A Top-down Hierarchical Multi-hop Secure Routing Protocol for Wireless Sensor Networks

M. P. Singh[1] and Md. Zair Hussain[2]

[1]Department of Computer Science and Engineering, NIT Patna
writetomps@gmail.com
[2]Department of Computer Science and Engineering, MACET, Patna
mdzairhussain@gmail.com

ABSTRACT

This paper proposes a new top-down hierarchical, multi-hop, secure routing protocol for the wireless sensor network, which is resilient to report fabrication attack. The report fabrication attack tries to generate bogus reports by compromising the sensor nodes to mislead the environment monitoring application executed by randomly deployed wireless sensor nodes. The proposed protocol relies on symmetric key mechanism which is appropriate for random deployment of wireless sensor nodes. In the proposed protocol, base station initiates the synthesis of secure hierarchical topology using top down approach. The enquiry phase of the protocol provides assurance for the participation of all the cluster heads in secure hierarchical topology formation. Further, this methodology takes care of failure of head node or member node of a cluster. This protocol ensures confidentiality, integrity, and authenticity of the final report of the monitoring application. The simulation results demonstrate the scalability of the proposed protocol.

KEYWORDS

Algorithms, Design, Experimentation, Security, Cluster based wireless sensor network, Report fabrication attack, Secure routing protocol

1. INTRODUCTION

The advancement in hardware technology (micro-electro-mechanical systems (MEMS) technology & nano-electro-mechanical systems (NEMS) technology), and wireless communications, has enabled the development of new category of computing devices which are known as sensor nodes [4]. These devices integrate computation, communication and sensing components. Aggregation of these sensor nodes into communication infrastructure leverages the idea of wireless sensor network (WSN). WSNs, a specific kind of ad hoc network of resource constrained sensor nodes, have enabled wide range of applications. These applications which make use of several sensor nodes, are not suitably applicable to computing nodes on traditional wired networks, wireless networks, and ad-hoc networks. The examples of such applications are military surveillance, disaster detection & relief, space exploration, environmental monitoring, habitat monitoring, acoustic detection, seismic detection, inventory tracking, medical monitoring, smart spaces, process monitoring, structural health monitoring etc. [10, 13, 15, 23, 24, 26, 31, 33, 44, 45, 47, 51]. Some of these applications like monitoring of nuclear plants, sea beds, remote locations in mountain ranges or deserts, volcanic eruptions etc. involve physically challenging environments which are not conducive for human life. These monitoring applications can be used to provide better services to humanity in terms of improved climate control, security, and safety.

Sensor nodes which are being used in above mentioned applications are constrained in computational speed, communication range, and storage capacity. This handicap or limitation of sensor nodes makes the design of wireless sensor network a challenging task. The

10.5121/ijasuc.2010.1204

aforementioned constraints of sensor nodes make the implementation of security mechanisms in the wireless sensor network a challenging research issue.

The main concern of all possible applications of WSN is to fetch the data sensed/read by the each node and transfer to the end user without losing data integrity, freshness etc. The application with security requires more computation and/or communication in comparison to other applications. Hence, it is important to design application which may provide the accurate data from WSN against all odds, happens in WSN. The resource constrained nature of sensor nodes poses the unique challenges to the design of WSNs for their applications [30, 37]. The limited power of sensor nodes mandates the design of energy-efficient communication protocol with or without security. According to [37], 3000 instructions could be executed at the same energy cost as that of sending a bit for a distance of 100 meters using a radio. In another observation [30], the power consumed by a Berkeley mote to transmit 1 bit data is equivalent to the computation of 800 instructions. From the above two observations, it is noted that communication cost is an important factor. This communication cost can be reduced significantly by using cluster based communication as compared to one without clustering. These are being taken care of in the proposed protocol in section 3.

”Some of the applications of wireless sensor networks like military surveillance, disaster detection and relief, space exploration etc. require a secure communication channel for data transfer from the sensor nodes to the base station. Without security mechanisms, such applications may result in undesirable consequences.” The following examples are influenced from [25]. In the military surveillance application, there are several attacks like denial of service attack, eavesdropping attack (leaking of information to enemy), supply of misleading information attack (there is no enemy movement) etc. In the disaster detection and relief application, an adversary may send false information like bogus disaster warning causing huge financial loss as a result of larger scale evacuation, and deployment of disaster equipments. In the space exploration application, it is required that all the commands executed on space are authenticated, and all the collected data are encrypted, and authenticated. Otherwise this may cause a huge monitory loss as well as human life. The requirements of such applications necessitate the security as one of the major concerns of wireless sensor network design. The security attacks may be defended using strong cryptographic mechanisms [4, 14, 41]. But, the cryptographic mechanisms developed for traditional wired network, traditional wireless network and ad-hoc network may not be directly employable for the wireless sensor network due to various reasons enumerated next:

1. The sensor nodes have scarce resources [5]. The implementation of cryptography algorithm within a sensor node consumes extra memory, & processing time, which is certainly overload condition. This overload consumes extra battery power. Thus security algorithm for the wireless sensor network has to be developed in such a way that it consumes minimal amount of battery power, and occupies minimal additional memory space of sensor node as to allow seamless operation of the applications augmented with security.
2. Public key algorithms are not suitable for WSN because of computational complexity and larger code size associated with such algorithms.
3. In most of the application, it is assumed that base station is the trusted entity that is base station will not be compromised. It is also assumed that the base station is resource full scalability is not a bottleneck for the WSN [41].
4. In-network aggregation in WSN reduces the communication overhead, and hence minimizes the battery consumption. In-network aggregation is an important requirement for secured the wireless sensor networks too [25].
5. Sensor nodes are often deployed in a hostile environment where the sensor nodes are open to the physical attack. It is also one of the factors that require being paid attention in designing the security mechanisms.

Because the above mentioned limitations/requirements, a new feasible security mechanism needs to be developed to be used by the sensor nodes in a WSN.

The intruder may mount an attack by eavesdropping of wireless channel. So it is required to secure the environment. Otherwise, intruder may mount various attacks like insider attack, outsider attack. In addition, it might also be necessary to provide security mechanisms in order to provide data confidentiality, data integrity, data authenticity, data freshness, and availability. Each of these is explained in brief next.

Data Confidentiality*:* In the wireless sensor networks, the data confidentiality is related to the following observations [9, 36].

—Sensed data should not be leaked from the sensor node of a wireless sensor network to the adversary.

—Before sending the highly sensitive data like cryptography keys, establishment of a secure channel is necessary.

Data Integrity*:* Data integrity ensures the originality of data. Adversary and/or harsh communication environment hamper the data integrity.

Data Authenticity: In two party communications, a receiver has to ensure the sender identity using data authenticity approach.

Data Freshness: Data freshness ensures that data is recent. This is required to defend against the replayed attack. That is it ensures that no old data have been replayed. The data confidentiality, and the data integrity do not ensure the data freshness.

Availability: Denial of service (DoS) attack does not allow using the network service. Hence, it is required to make sure for availability of network service despite of denial of service attack.

Two broad categories of security attacks, namely outsider attack and insider attack [41] in the wireless sensor networks are discussed in the following.

Outsider Attack [4, 48]: The outsider attack is the one in which the adversaries try to attack the wireless sensor network without having the knowledge about the network. The security attacks such as eavesdropping the communication channel (where the adversary obtains the sensed values by hearing the communication channel), injecting malicious messages into the wireless sensor network, and denial of service attack (where the adversary jams the communication channel using signals of high strength) are the few examples of outsider attack.

Insider Attack [4, 19, 28, 32]: When the adversary compromises a sensor node, it gets hold of the information stored in the sensor node such as the cryptographic keys, location of the sensor node etc. Such information helps the adversary to launch the insider attacks in the wireless sensor network. The security attacks such as report fabrication attack (RFA), node replication, generation of bogus reports, Sybil attack (impersonate multiple network entities by obtaining their identities) are the few examples of insider attack.

So, there is need to develop a security protocol considering the resource constraints, security challenges, and security requirements. The same has been done in the section 3.

The rest of the paper is organized in following sections. Section 2 summarizes the related work of security protocols. This section also presents their limitations. Section 3 describes the assumptions, notations, and the detail explanation of all the phases of proposed protocol. Section 4 explains the handling of failures of either cluster members or cluster heads. Section 5 presents the security analysis of proposed protocol, like secure cluster setup, secure hierarchical topology setup, event detection, and secure reporting. Section 6 presents the analysis of memory overhead, computation overhead, communication overhead, and scalability. Section 7 presents implementation details and simulation results of with and without security. This section also shows the time consumption for the security implementation and finally, section 8 concludes the work and also presents the future works.

2. Related Work

This section presents the related work on security protocols for wireless sensor networks.

Localized encryption and authentication protocol (LEAP) [55] is a key management protocol for the wireless sensor network. It supports in networking processing, while at the same time providing security properties. In the LEAP, every node creates a cluster key, and distributes this key to its immediate neighboring nodes using pairwise keys that it shares with each of its neighboring nodes. This scheme has some drawbacks as follows. (i) The bootstrapping phase is quite expensive. (Bootstrapping phase establishes a secure communication channel among sensor nodes. These nodes may have some preloaded keys but does not have any information regarding the neighboring nodes.) (ii) The storage requirements on each sensor node are also nontrivial since every node has to store a number of pairwise, and cluster keys. This number is proportional to the number of actual neighboring nodes. (iii) This scheme performs well against the outsider attack in comparison to the insider attack.

The many other key management schemes for wireless sensor networks have been proposed in [8, 12, 20, 21, 27, 36, 39]. Security mechanism for denial of service attack in the wireless sensor networks has been discussed in [48].

A study on the effects of hello flood attack, and Sybil attack on different routing protocols such as directed diffusion, GEAR, LEACH, TEEN, and PEGASIS is presented in [29].

SAC: Secure Adaptive Clustering protocol in wireless sensor network is successful in preventing attacks caused by adversary like *hello flooding* and provides resilience to sensor nodes captured by adversary [38].

A security mechanism for the S*ybil attack* in the wireless sensor network is presented in [19, 32]. In this attack, a malicious node assumes the identity of a large number of the sensor nodes of the network. It does this by impersonating other sensor nodes or by claiming the false identities. An approach to defend against the Sybil attack is resource testing [19]. Resource testing verifies that whether or not each identity has as much of tested resource as is expected from the identity. This approach is not suitable in case of the wireless sensor network because of resource limitations. For example, an attacker can deploy a physical device with capabilities that are several orders magnitude larger than those of the normal sensor nodes. Other defenses such as radio resource testing, verification of set for random key pre-distribution, registration, and position verification have been proposed in [32].

A security protocol, *An intrusion-tolerate routing protocol for wireless sensor networks* attempts to design a routing protocol that can tolerate intrusions rather than detecting the intrusion [17]. This is done by choosing the redundant paths from any sensor node to the sink. But this allows only the base station to broadcast information to protect the routing protocol from attack.

In *trust routing for location aware sensor networks* (TRANS) [46], each sensor node calculates trust value of its neighboring nodes. Based these calculated trust value; a secure path is being setup by bypassing the insecure locations.

Security protocols for sensor networks (SPINS) [35] uses two low level secure building blocks, namely SNEP, and μTESLA. SNEP provides data confidentiality, two-party data authentication, data integrity, and data freshness between the sensor nodes & the sink. μTESLA provides authentication for data broadcast. SPINS uses Rivest Cipher 5 (RC5), is a kind of block cipher, [1] because of its small code size. SPINS assumes that only base station is the only point of trust. It does not allow node to node communication directly. Node to node communication necessitates authentication via the base station. Hence, SPINS may not be suitable for relatively larger size of the wireless sensor network, especially hierarchical wireless sensor networks.

The three levels of security for three different types of data are considered in [43]. Each node has a set of master keys initially, but one of them is active at a given time. All three security levels derive keys from the master key. For the first level of security, the master key is used for the messages, which are infrequent. For the second level of security the network is divided into hexagonal cells. All the member nodes in a cell share a unique location based key. The nodes at the border of the cell store the keys of the cell to which they belong to allow traffic to pass through. This model requires that the sensor nodes be able to discover their exact location, which allows them to organize into cells, and produce a location based key. This is expensive. The third level of security uses a weaker encryption with a focus on computational overhead. The authors assume that the sensor nodes are tamper proof. Thus, the set of the master keys, and other preloaded information cannot be revealed by compromising the node.

An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks tries to make the cluster based wireless sensor network which will be resilient to *report fabrication attack* [55]. Report fabrication attack is the insider attack. In this attack, a compromised node or a set of compromised nodes are used to generate forged report for a non existing event. This forged report can either misguide the system or deplete the resources of the sensor nodes in the wireless sensor network. The *report fabrication attack* is a severe security attack in reactive sensor networks. This attack is one of the most recent security attacks which have been paid less attention in the wireless sensor network security literature [4, 25]. This protocol tries to detect, and filter out the false data packets either at or en route to the base station. There are some drawbacks with this approach which are as follows:

1. It assumes that all the clusters contain at least n+1 nodes (including itself). But when the sensor nodes are randomly deployed, the number of nodes varies from cluster to cluster [18, 42]. So, this approach is not appropriate for the random deployment.
2. It does not handle the unavailability of the nodes either due to the physical damage of the sensor nodes or failure of the sensor nodes [35].
3. If a single node fails to report, then the report generated by the cluster becomes invalid.
4. It uses *localized encryption and authentication protocol* (LEAP) [54] for neighborhood discovery, and for establishing the shared key among neighboring nodes which itself is prone to security attacks [18].
5. It is energy expensive as it requires fixed path to be maintained from the cluster head to the base station.

The proposed approaches in [41, 55] focus mainly on the enroute filtering framework. In this, the forged reports will be dropped enroute from the source to the base station. Further, the node that has dropped the forged report, generates alarm message about the cluster that has generated the forged report. In this approach, the base station trusts the node which has generated the alarm message. But there is a possibility of generation of bogus alarm messages about the lower associated clusters by a compromised node.

The strength of the approaches to defend against the report fabrication attacks mainly depends on the followings:

1. The ability of a wireless sensor network to suppress the generation of the false reports.
2. The ability of the base station to differentiate between the forged report & legitimate report, and take an action against the neighborhood that has generated the forged report.
3. The ability of the wireless sensor network to restrict the scope of attacks of the compromised nodes to a particular cluster in which they reside.

Due to the above reasons, the enroute filtering does not perform well in the cluster based wireless sensor networks.

The security, and privacy of data centric wireless sensor network have been proposed in [40]. In [50], an attempt was made to provide security solution for *false data injection attack* in the wireless sensor network.

In [49], an attempt was made to provide effective security solution for the flat wireless sensor networks. It uses the location dependent approach, where the sensor nodes need to be deployed at predefined location depends on the requirement of the application. However, the solution is not optimal for the cluster based networks [34].

The studies on these protocols helped in comprehensive analysis of the problem of development of security mechanism as presented in section 3. Authors also note that it is better to incorporate the security mechanism during the design phase of the system development rather than implementing the security mechanism on top of a developed system. The latter approach fails in many cases as detailed in [48]. In this proposed paper security issues are taken up along the development of clustering mechanism and routing protocol.

3. The Proposed Secure Hierarchical Multi-hop Routing Protocol

This section describes the working of the proposed top-down, hierarchical, multihop, secure routing protocol in detail. First, the threat model under consideration is presented. Second, this section notes the assumptions about WSN and its components. Third, it enumerates the notations used. At last, it presents the algorithm of each phase with detail explanation.

3.1 Threat Model

If a sensor node is compromised then: All the information stored in sensor node will be revealed to the adversary. The adversary can create the clones of the sensor node and deploy them across the sensor network. It may illegitimately take multiple identities.
The adversary may either eavesdrop the communication channel in the sensor network or replay the old messages or inject malicious messages into the sensor network or physically destroy the sensor node.

3.2 Assumptions

A1: Each sensor node has a unique ID.
A2: Broadcast message sent by the cluster head is received correctly within finite time by all of its one hop neighboring nodes.
A3: Network topology is static during algorithm execution that is sensor nodes are not mobile.
A4: Packet broadcast by the base station is correctly received by some of the cluster heads.
Discussion: As the hierarchy formation is initiated by the base station, the deployment of the base station is assumed flexible to justify this assumption otherwise it would not be possible to synthesize the hierarchical topology in the wireless sensor network.
A5: The base station is static, resourceful, and trusted entity.
Discussion: The base station may be a laptop or a PC class device. Mobility of the base station is not allowed. The base station need to keep the information about all the sensor nodes in order to avoid the report fabrication attack. Hence, the base station assumed to be resourceful. The base station is an interface between wireless sensor network and other networks. The base station is physically protected. Hence, the base station is assumed to be trust worthy. If the base station is not static, the hierarchy of clusters would change. And hence the cluster head has to establishment a path to the base station each time before sending the data. Further, if the base station does not have enough resource, then memory overhead, computation overhead, and communication overhead will make it unsuitable for larger size of wireless sensor network. If the base station is not trust worthy, then this can result in a single point failure of the proposed security protocol. In such a case, securing the wireless sensor network is of no interest.
A6: All the sensor nodes in the wireless sensor network have same capabilities in computation, and have equal communication range initially.
Discussion: The sensor nodes are homogeneous in hardware capability. Design of clustering protocol would change, and hence routing protocol based on the clustering protocol will also change.
A7: The wireless sensor network will not be attacked for the first '*t*' second [18, 34, 54] that is during the secure cluster setup phase, and secure hierarchical topology discovery phase.
Discussion: Assumption of tamper resistance to design a security protocol for wireless sensor network is not correct [6, 43]. The proposed protocol assumes that the adversary node requires minimum time, $T_{compromise}$ to compromise a sensor node whereas $T_{topology}$ time is required to setup

clusters, and hierarchy, which is smaller than $T_{compromise}$. That is the wireless sensor network will not be attacked for the first $T_{topology}$ time same as [18, 34, 54]. This protocol believes that this is a reasonable assumption for most of the wireless sensor networks, and adversaries same as [18, 34, 54]. Otherwise, it could be easy for adversary to compromise the deployed sensor nodes and then capture the wireless sensor network. These assumptions are used along with threat model in the secure routing protocol presented next.

3.3 Notations

cid: Cluster ID. *Connected Cluster Head (CCH):* Cluster head connected to partially or fully formed hierarchical topology. E_{K_m} : Encryption using key K_m. *holdback:* Holds randomly generated value which is used for cluster head making decision. $K_{i,j}$: Shared key between nodes i and j. *levelstatus:* Whether or not the cluster head is the part of hierarchical topology. *MAC*(*k; s*): *M*essage *A*uthentication *C*ode computation on message *'s'* using key *'k'*. *MN:* Member nodes of any cluster. *RFT*: Report Fabrication Threshold. *status:* Whether the node is cluster member or cluster head.

3.4 Working of the Proposed Hierarchical Multi-hop Secure Routing Protocol

The proposed protocol can be divided into two phases namely, *(1)* Pre Deployment Phase, and (*2)* Post Deployment Phase. Pre deployment phase takes place before the deployment of the sensor nodes in the given target area. This phase initializes the parameters. These parameters are useful for post deployment phase. In the pre deployment phase, each sensor node is preloaded with keys, which are used for security mechanisms like encryption, and decryption. Post deployment phase takes a secure hierarchical topology based on clusters. This phase also detects events that occur in the given target area, and reports to the base station in secure. In nutshell, the proposed protocol follows the steps as described below:

1. Before deployment of sensor nodes, each node initializes some keys in order to provide security in WSN.
2. After deployment of sensor nodes, the proposed protocol divides WSN into non-overlapping clusters (groups). Each cluster has one cluster head.
3. After that, the proposed protocol synthesizes the hierarchy of cluster heads.
4. After that, the proposed protocol send the report of any event, arises in any cluster, to end user in hop-by-hop strategy that is cluster head to another cluster head towards the end user. The report is being authenticated at each hop and integrity of the report is also maintained during transmission.
5. All the communications done in all of the above steps, if any are secure communications.

In next, the each phase is explained in detail.

Algorithm 1: Pre deployment Phase

```
1: Each sensor node initializes the following its own parameters
2: cluster ID, cid = null, status=null, levelstatus=false,
3: holdback value to some randomly generated number,
4: Nodeid ($N_i$), MasterKey ($K_m$) NodeKey ($K_i$), ClusterKey ($K_i^c$), RFT
```

Pre Deployment Phase: The pre deployment phase takes place before the sensor nodes are deployed in the given target area. During this phase each sensor node *'i'* is assigned with unique identifier N_i ranging from 0 to *n-1* (where *n* is the number of sensor nodes in the network). Each node initializes its own *cid* (Cluster ID) to *null* value, randomly generated value into *holdback*, *status* to null, and *levelstatus* to false. The parameter *status* is used for checking the status of the sensor node. Sensor node plays its role either as a cluster head or as a cluster member. The parameter *levelstatus* is used to check whether or not node is connected with hierarchical topology. Each sensor node is preloaded with the following keys:

Master Key (K_m): A key of 64 bits is shared among all the sensor nodes, and the base station. This key is used during the secure cluster setup phase, and secure hierarchical topology setup phase. In order to ensure the security, this key will be erased after hierarchical topology setup phase.

Node Key (K_i): This is a secret key of 64 bits. Each sensor node shares this key with base station. This key will be used for authentication purpose while reporting to the base station.

Cluster Key (K_i^c): Each node *'i'* is loaded with unique cluster key K_i^c **,** and will be used by the sensor nodes that will become cluster heads.

Report Fabrication Threshold (RFT) RFT is the parameter that determines the number of nodes that have to endorse the occurrence of the event. It is determined based on the number of nodes in the cluster as follows

$$m = \lceil RFT * \text{no. of nodes in the cluster} \rceil ; \quad \textit{if (no. of nodes)>2}$$
$$m = 2; \textit{ if (no. of nodes)} \leq 2 \text{..} (1)$$

The RFT value varies from 0.5 to 1.0 (which ensures that at least 50 percent of the cluster members have to report the occurrence of event). If RFT is 1 then all the member nodes in the cluster have to endorse the occurrence of the event. Depending on the security requirements of the applications the RFT value can be set. Once all the sensor nodes are preloaded with the required keys, the base station is informed about the sensor nodes, and key mappings. The sensor nodes are deployed redundantly.

Post Deployment Phase: The post deployment phase is further divided into four phases namely, *(1) Secure cluster setup phase, (2) Secure hierarchical topology setup phase, (3) Secure enquiry Phase, and (4) Event detection and secures reporting to base station*. These phases are presented in detail in Algorithm 2, 3, 4, and 5 given below.

Algorithm 2: Secure cluster setup phase

```
after every $t_c$ seconds, each sensor node decrements the holdback value by one
if (holdback == 0 && status == NULL) then
    set status = cluster_head and cid = node id,
    Initializes the packet.type=cluster_head_hello
    broadcasts cluster_head_hello end if
On receiving the cluster_head_hello broadcast
if (status == NULL && holdback , 0 && packet.type== cluster_head_hello) then
    cid = SendID, /*ID of node that has broadcast the cluster head hello*/
    set status = cluster_member,
    compute the shared key using hash function `h'
    reply with cluster_hello_reply message to cluster head end if
cluster head on receiving cluster_hello_reply message
computes the shared key with cluster members using hash function `h'
```

Secure Cluster Setup Phase: This phase partitions the wireless sensor network into non-overlapping groups, known as clusters. During this phase, only those sensor nodes with *holdback* = 0, declare cluster heads themselves, and start broadcasting the message *cluster_head_hello* to form the cluster.

$$cluster\_head\_hello : \{cluster\_head\_hello, C_{id}, K_i^c\}E_{K_m} \text{ ....(2)}$$

where C_{id} is node id of the node broadcasting *cluster_head_hello.* Upon receiving the message, the sensor node decrypts the message, and decides to become cluster member, if it is not a part of any other cluster, and inform to the cluster head about its membership by sending the packet of type *cluster_hello_reply*. They respond by

- Computing the shared key with the cluster head using secure hash function *'h '*

$$K_{C_{id,j}} = h(K_i^c, C_{id} \mid K_m \mid j) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\text{(3)}$$

- Acknowledging the *cluster_head_hello* by sending *cluster_hello_reply*, so that cluster head knows the members of its cluster, and computes the shared key as in Equation 3.

$$cluster\_hello\_reply : \{cluster\_hello\_reply, C_{id}, MAC(K_{C_{id,j}}, j \mid C_{id})\}E_{K_m} \text{ (4)}$$

After fixed time t_c, every node decreases its *holdback* value by one. By the end of this phase, the wireless sensor network is divided into clusters. The member nodes are at one hop distance from their cluster heads. Each cluster member computes a shared key with the cluster head. Cluster head also does the same, and computes the *'m'* value (which determines the number of nodes within the cluster that have to endorse the occurrence of a particular event).

Algorithm3: Secure hierarchical topology setup phase

```
Base station initiates this phase /*Top-Down Approach*/
Base station initializes the following parameters
Packet type= base_hello;  levelstatus=TRUE; Level = 0,
Now, base station initiates the hierarchical topology setup phase by broadcasting the base_hello message
cluster head(s) on receiving first base_hello message
if (packet.type== base_hello && status == cluster head && levelstatus==FALSE) then
    Set its own levelstatus=TRUE,
    Increase the level count by one,
    Remember the its parent address,
    Initialize the packet.type=base_forward_hello
    compute the shared key using hash function `h' and broadcast the base_forward_hello packet to form the next level
    send the base_reply end if
on receiving the base_reply,  call Algorithm 6
cluster head(s) on receiving first base_forward_hello message
if (packet.type==base_forward_hello && status == cluster_head && levelstatus==FALSE) then
    Enqueue the packets /*May receive packets from more than one cluster heads*/
    Set its own levelstatus=TRUE,
    Increase the level count by one & maintain the list of parents,
    compute the shared key using hash function `h' and broadcast the base_forward_hello packet to form the next level
    send the base_reply end if
on receiving the base base_reply Algorithm 6
```

Secure hierarchical topology setup phase: Base station starts this phase by broadcasting the packet of type *base_hello*. This finds out the cluster heads for the level 1 in hierarchical topology formation. Base station is at level 0. Upon receiving the first *base_hello* message each cluster head responds to it by

- Computing the shared key with the base station or the cluster head from which it has received the *base_hello* as in Equation 3.

$$base\_hello : \{base\_hello, C_{id}, hop\_count\}E_{K_m} \ldots\ldots\ldots\ldots\ldots\ldots\text{(5)}$$

- Sending the *base_reply* to the sender of the *base_hello*. *Base_reply* acts as an acknowledgment to the *base_hello*. It also contains the payload holding details of the cluster head, and the cluster members destined to the base station.

$$\begin{aligned} &base\_reply : y_0 \leftarrow \{id's\, of \quad the \quad cluster \quad members\} \\ &y_1 \leftarrow \{y_0, MAC(K_{C_{id}}, y_0)\}E_{K_{C_{id}}} \\ &y_2 \leftarrow \{base\_forward\_reply, C_{id}, y_{1\}} \\ &y_3 \leftarrow \{base\_reply, C_{id}, MAC(K_{C_{id,j}}, j \mid C_{id}), y_2\}E_{K_m} \end{aligned} \qquad \ldots\ldots\ldots(6)$$

base_reply consists of all the values calculated in y_0, y_1, and y_2 that is y_3 is nothing but the *base_reply*.

- Forwarding the *base_hello* to the downstream cluster heads by incrementing the *hop_count*.

On receiving the *base_reply* the cluster head responds by computing the shared key as in Equation 3, and forwards payload *base_forward_reply* to the base station. On receiving the *base_reply* the receiving cluster head responds by computing the shared key as in Equation 3. In turn, the cluster heads at level 1, broadcast the packet of type *base_forward_hello* to find out the cluster heads for the level 2, and so on.

By end of this phase, wireless sensor network is organized as a connected graph and base station learns about the secure topology of the wireless sensor network.

Algorithm 4: Secure enquiry phase

```
if (levelstatus=FALSE && status == cluster_head) then
the cluster head is not the part of hierarchical topology
Cluster head initiates this by sending scan_hello packet to its members, end if
on receive scan_hello
Members scans cluster heads which are already the part of hierarchical topology by broadcasting the packet.type = scan,
on receiving scan
if (packet:type==scan && status==cluster_head && levelstatus==TRUE) then
compute the shared key using hash function ('h')
cluster heads reply by sending the packet of type=scan_reply, end if
Member nodes receive the packet and decide to join the hierarchical topology
if (packet.type== scan_reply && status==cluster_member && ReceiveID == NodeID) then
Enqueue the packets,
Set its own levelstatus=TRUE,
Increase the level count by one,
Maintain the list of parents,
compute the shared key using hash function ('h')
Forward the packet.type==scan_reply forward to its cluster head,
send scan_reply_ack towards base station end if
on receiving scan_reply_ack
if (receiving node of scan_reply_ack == cluster_head && levelstatus ==TRUE) then
forward the scan_reply_ack towards to base station end if
After receiving the scan_reply forward packet
    Enqueue the packets, Set its own levelstatus=TRUE, Increase the level count by one, Maintain the list of parents,
    send scan_reply_ack
on receiving scan_reply_ack
if (receiving node of scan_reply_ack == cluster_member && levelstatus ==
TRUE) **then**
34: forward the *scan_reply_ack* to its parents towards the base station **end if**
```

Secure Enquiry Phase: After the *secure hierarchical topology setup phase*, there may be some cluster heads which are not the part of hierarchical topology because such cluster heads are not at the directly reachable from any other cluster heads. But member nodes of such clusters connect the one or more CCHs. This phase handles such case of the wireless sensor networks. Such cluster heads start the *enquiry phase* by sending a packet *scan_hello* to its member nodes. Cluster heads know about its cluster members during *cluster setup phase*, and calculated the share key during the secure cluster setup phase.

$$sacn\_hello : \{scan\_hello, C_{id}\}E_{K_m} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(7)$$

where C_{id} is node id of the node broadcasting *scan_hello.*
Now, the members nodes broadcast the packet of type *scan* to find out the reachable CCHs.

$$sacn : \{scan, C_{id}\}E_{K_m} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(8)$$

Only those CCHs, who receive $scan$ packet, compute the shared key with member node from which it has received the $scan$ packet as in Equation 3, and reply by sending the packet of type *scan_reply*.

$$sacn\_reply : \{scan\_reply, C_{id}, hop\_count\}E_{K_m} \ldots\ldots\ldots\ldots\ldots(9)$$

After checking the *scan_reply* packet, the member nodes accept this packet, and become the member of the next level of the hierarchical topology by computing the shared key with the node from which it has received the *scan_reply* as in Equation 3. Further, it sends the *scan_reply_ack* toward base station. Further, the same member node sends the packet of type *scan_reply_forward* to its cluster heads. The cluster heads accept this packet only from its members. Now, these cluster heads connect themselves to the next level of their member nodes, and send the packet *scan_reply_ack* towards base station.

$$scan\_reply\_ack : y_0 \leftarrow \{x\}$$
$$y_1 \leftarrow \{y_0, MAC(K_{C_{id}}, y_0)\}E_{K_{C_{id}}}$$
$$y_2 \leftarrow \{scan\_reply\_ack, C_{id}, MAC(K_{C_{id,j}}, j \mid C_{id}), y_1\}E_{K_m}$$

where x is node id of member nodes (which connect their cluster head to the partially formed topology) in case of line number 19 of Algorithm 4, and id's of the cluster members in case of line number 24 of Algorithm 4.

After the phases, namely cluster setup phase, hierarchical topology setup phase, and enquiry phase (optional phase), none of the deployed sensor nodes of wireless sensor network possesses master key, K_m that is all the sensor nodes erase the master key, K_m. An adversary may have eavesdropped on all the communications during the above phases. But without K_m , the adversary cannot encrypt the false report to send it towards base station or decrypt any of the messages. Hence, eavesdropping does not help adversary. However, after $T_{compromise}$ the adversary may compromise sensor nodes and obtains the keys available with the compromised nodes. This results in localizing the security impact. The false information injected by the compromised node will be detected during reporting to base station.

Algorithm 5: Event Detection and Secure Reporting to Base Station

```
1: **if** the cluster member detects an event **then**
2: it sends *report* to its own cluster head **end if**
3: **if** the cluster head receives at least `m' reports from its members **then**
4: it prepares the *final_report* and routes it to base station in hop by hop authenticated
fashion **end if**
```

Event Detection and Reporting to Base Station: This phase reports the any event, that occurs in any cluster, to the base station in multi hop. If any event occurs, each sensor node *'i'* senses the event, and informs to its cluster head as in Equation 10.

$$y_0 \leftarrow \{event\_\inf o\}$$
$$y_1 \leftarrow \{y_0, MAC(K_i, y_{o)}, MAC(K_{C_{id,i}}, y_0)\}E_{K_{C_{id,i}}} \quad \text{.......................(10)}$$
$$y_2 \leftarrow \{report, C_{id}, y_1\}$$

The cluster head receives at least *'m'* data from its members and takes MAC's (MAC's have been computed for the base station). Further, cluster head computes XOR of all the MAC's in order to reduce the communication overhead that is XORing all the MAC's rather than sending individually. Now, Cluster head prepares the report, and includes its own identity in the report as in Equation 11. Finally, cluster head sends this report to base station in hop by hop authenticated fashion.

The base station on receiving the report checks for the source (that is the cluster head from which the report has originated), and validates the report by computing the MAC. If the report fails the validation check, the base station discards the report, otherwise the base station initiates appropriate action. In next, handling of sensor nodes failure is presented.

$$y_0 \leftarrow \{event\_\inf o, MAC_1 \oplus MAC_2 \oplus ....\oplus MAC_{m\}}$$
$$y_1 \leftarrow \{y_0, MAC(K_{C_{id}}, y_0)\}E_{K_{C_{id}}} \quad \text{..............(11)}$$
$$y_2 \leftarrow \{final\_report, C_{id}, y_1\}$$

Algorithm 6: Algorithm for base reply

```
if receiving node of base_reply == base_station then
    compute the shared key using `h'
    learn about the cluster using base_forward_reply
else if receiving node of base_reply == cluster_head && levelstatus == TRUE then
    compute the shared key using the information in base reply
    forward the base_forward_reply part of it to next hop towards the base station end if
```

4. Handling Node Failures

The sensor nodes are often deployed in hostile area, and left unattended. There is great probability of physical damage to the sensor nodes. There is a need to handle the sensor node failures effectively as the failure of single sensor node may lead to disastrous consequences.

Cluster Member Failure: In order to handle the cluster member failure, this protocol assumes that the cluster heads frequently ping their cluster members, and keeps a track of them. If a cluster member fails to respond to its cluster head, then the cluster head notifies it to the base station so that the base station recomputes the *`m'* value, and send it to the cluster head.

Cluster Head Failure: As the cluster members contain the information about the cluster heads within the communication range (the cluster member gathers this information during the secure cluster setup phase). If cluster head becomes unavailable, the cluster members may execute orphan adoption protocol as in [34].

5. Security Analysis

Secure Cluster Setup phase, Secure Hierarchical Topology setup phase, and Secure Enquiry Phase: The security of these three phases, namely secure cluster setup phase, secure hierarchical topology setup phase, and secure enquiry phase, is based on the assumptions that the sensor nodes will not be tampered for the first *'t'* seconds after their deployment in the given target area. The sensor nodes will not behave non-deterministically until and unless they are tampered. So, the above assumptions [18, 54] ensure that only legitimate senor nodes become part of the wireless sensor network. In order to ensure confidentiality all the messages are encrypted using master key (K_m), which is shared by all the nodes of the wireless sensor network. The MAC (Messages Authentication Code) which ensures authenticity, and integrity of the message, is used where ever required (for multi-hop communication). This work also assumes that all the sensor nodes share a random counter value with base station as well as single hop neighboring nodes, as it offers semantic security [52], and as well as protects against replaying of messages.

Event Detection and Secure Reporting: During the normal of the network, the effect of the sensor node compromise depends upon the type of the sensor node compromised. The compromised node either may be cluster head or may be cluster member. If the cluster members are compromised, then the adversary tries to generate the bogus data, and send it to the cluster head. But as the valid event must be endorsed by at least *m* sensor nodes, it is violated in such case. So if the cluster head receives less than *m* data then it detects the sensor node compromise within the cluster, and takes an action either by re-computing and distributing the new keys or by issuing a refresh command so that all the sensor nodes hash the keys [18]. From the above approach, the adversary will only be able to generate a forged data once it has compromised at least *m* sensor nodes in the cluster.

If the cluster head has been compromised then the adversary can generate bogus reports, and route to the base station. As the base station verifies the authenticity of each report, the bogus report can be easily identified by the base station as the base station is aware of each cluster heads and their member during the secure hierarchical setup phase. The base station has a unique shared key for each cluster heads along with their member nodes. The master key, plays in important role in calculating shared key, is erased after the secure hierarchical setup phase. In the proposed security protocol, member nodes of any cluster compute MAC based on node key (K_i). The individual keys of the each deployed sensor node are shared with the base station. At least *m* number member nodes send their MAC to their cluster head. Further, the cluster heads compress the entire MAC into one MAC by using XOR operation. This compression scheme is secure as XOR-MAC scheme is secure [7]. The cluster heads send the compressed MAC along with the report and its cluster key toward base station. Now, base station again computes individual MACs and applies XOR operation on these MACs to get one compressed MAC. If this calculated compressed MAC is same as the received MAC then base station accepts the compressed MAC otherwise rejects. The base station may respond to the bogus report by revoking the cluster from the wireless sensor network as in [16]. The malicious cluster head may also report about the unavailability of the cluster members unnecessarily. So the base station does not isolate the sensor node as soon as it receives such report. It handles such situation by declaring the sensor nodes as ORPHAN by unicasting the orphan status to it. So, if the cluster member is available it may join other cluster using the orphan adoption protocol [34].

6. Overhead Analysis and Performance Analysis

This section analyzes the memory overhead/ storage requirement, computational and communication overheads of the security scheme used in the proposed security protocol. It also analyzes the scalable nature of the proposed security protocol.

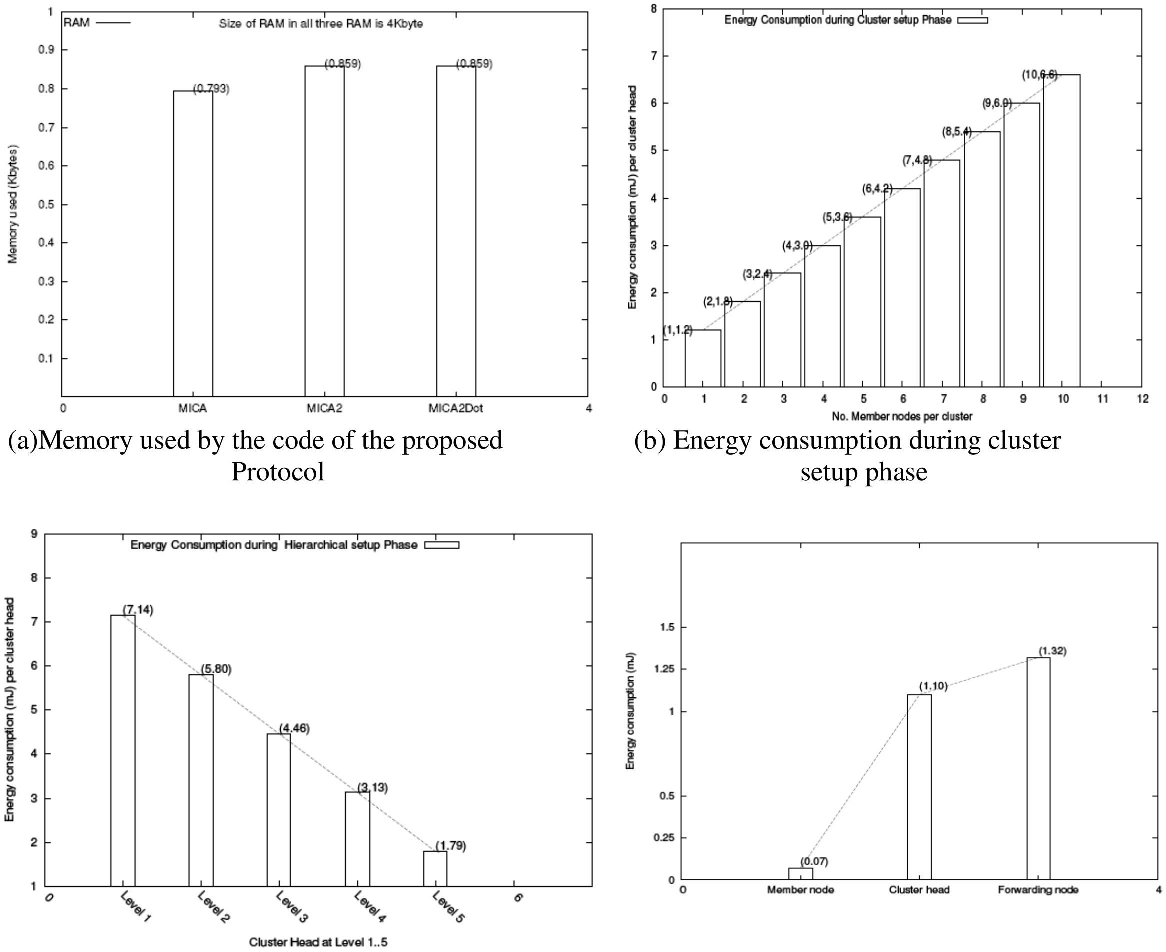
***Memory Overhead/Storage Requirement*:** Here, memory overhead represent the memory space required to store the number of keys preloaded, and the key generated during execution of the proposed security protocol. However, the memory required to store master key (deleted after hierarchical topology setup), and temporary memory used during execution of the proposed protocol are not taken into consideration. In the proposed protocol, all the nodes are homogeneous whereas plays two different roles, namely as a member node and as a cluster head. The member nodes need to keep three different keys, namely one node key (K_i), one cluster key (K_i^c), and one pairwise key with their cluster heads ($K_{c_{id,i}}$). The cluster heads need to keep one node key (K_i), one cluster key (K_i^c), pairwise keys with their member nodes

($Kc_{id,j}$), and one pairwise key with either base station or another cluster head ($Kc_{id,k}$). Here, i = 1...n (n is total number of deployed sensor nodes); j = 1...m (m is total number of member nodes of the clusters; k is the id of cluster head which is parent in hierarchical topology.

The proposed protocol uses a PC of processor of Intel(R) Pentium4 with speed of 3.2 GHz, and 1 GB RAM for execution. TOSSIM (TinyOS simulator) executes the code of the proposed protocol for MICA mote, MICA2 mote, and MICA2Dot mote. Fig. 1(a) shows the consumption of RAM of these mote. The memory size of RAM in all three mote is 4Kbyte [2]. Fig 1(a). shows that the code of the proposed protocol consumes less size of RAM for all three types of motes in comparison to 4Kbyte.

Computational Cost: The computation cost arises mainly due to establishing pairwise key and report authentication. Both require a number of encryption, and decryption.

Assume the total number of cluster head is *C* in the wireless sensor network that is there are *C* clusters in the wireless sensor network. These cluster heads establish pairwise keys with their member nodes. All the member nodes also establish a pairwise key with their cluster heads. The establishment of pairwise keys uses encryption and decryption. The total number of encryptions and decryptions depends on total number of cluster heads and their member nodes. Assume, N_j



(a)Memory used by the code of the proposed Protocol

(b) Energy consumption during cluster setup phase

(c) Energy consumption during Hierarchical setup phase

(d) Energy consumption during reporting to base station.

Fig. 1. Overhead analysis and performance analysis

is the total number of member nodes, where j = 1...m. The total number of encryptions during cluster setup phase is N_e = C + $N_e = C + C * \sum_{j=1}^{m} N_j$. The total number of decrypts during cluster setup phase is $N_d = 2 * C * \sum_{j=1}^{m} N_j$ .The total number of encryptions during hierarchical topology setup phase is $H_e = 1 + 2 * \sum_{i=1}^{l-1} \sum_{j=1}^{n} L_j$, where l is the total number of level of hierarchical, and L is the total number of cluster heads at particular level. The total number of decryptions during hierarchical topology setup phase is $H_d = L_1 + 2 * \sum_{i=1}^{l-1} \sum_{j=1}^{n} L_j$. The total number of encryptions, and decryptions in the WSN are $C_c + H_e$ and $C_d + H_d$ respectively.

Report Authentication: In the proposed security protocol, each member nodes, cluster heads, and forwarding nodes (the node forwards the message received from cluster heads at higher level in the hierarchical) computes two MACs for an event occurred in the given target area. These two MACs use node key and pairwise key as a MAC key.

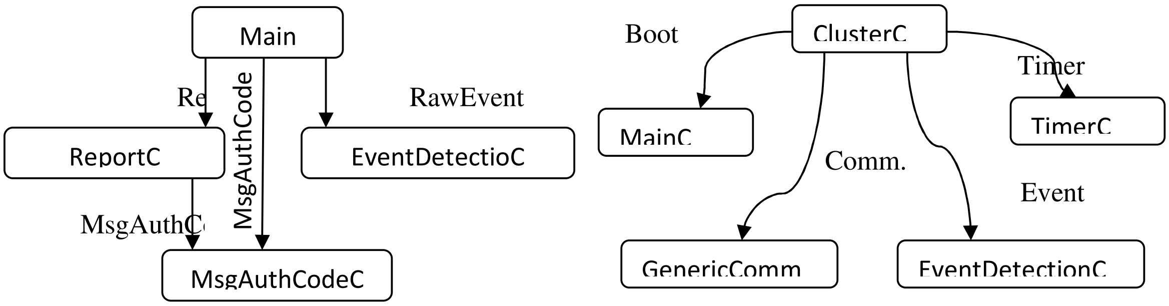
Communication overhead: The communication overhead is much higher than that of computation overhead. This protocol uses the following consumption rates similar as [53]: Mica2 mote consumes 10mA current when the node is in idle, and receiving state whereas it consumes 13mA for transmitting. Based on the battery voltage, 3V and data rate, 19.2Kbps the Mica mote consumes 16.25μJ/byte for transmission, 12.25μJ/byte for reception. This protocol uses RC5 block cipher [22] for MAC and hash computation. Both computation take about 0.5ms and consumes about 15 μJ. The default size of packet in TinyOS is 36 byte. The size of packet with authentication is 37 byte, and the size of packet with encryption, and authentication is 41 byte [22]. Fig. 1(b) shows that the consumption of battery power of the cluster head during cluster setup phase, which is the summation of consumption on encryption of cluster heads' hello packet & then transmit the same of size 36 byte, on receiving the acknowledgment from member nodes, and on hash function to compute pairwise key with their member nodes. The total consumption of battery power of the cluster head gets change because of second, and third terms of the summation, depends on total number of its member nodes. Fig. 1(b) show that the consumption of the battery power of the cluster head increases by an approximately same amount, 0.6mJ for each member node. Assume that the cluster of this cluster head has total number of member nodes varying from 1 to 10. Each member node of the cluster consumes battery power of an amount of 1.28mJ, which is summation of consumption on receiving packet from cluster head (decrypt the message of size 36 byte), on hash function to compute pairwise key (15μJ), and on encryption the acknowledge packet & then transmit the same of size 41 byte. Assume that there is maximum six levels on any path from base station to last level. Fig. 1(c) shows that the extra consumption of battery power of cluster heads on level 1 to level 5 during hierarchical topology setup phase. The cluster heads at level 1 consumes more battery power than level 2 to 5 and so on. The consumption of battery power is lowest at the level 5 in comparison to higher level (level 4 to level 1).

Fig. 1(d) depicts the battery power consumption of a member node & its cluster head, and forwarding node during report preparation and forwarding the same towards base station. The each member nodes of the cluster consumes battery power of a amount of 688.75μJ (.07mJ) to prepare a information of the event occurs in the cluster, and transmit the same to the its cluster head. The cluster head consumes battery power of an amount of 1.10mJ (in case of 5 member nodes) to prepare a final report and transmit to its parent (parent is the cluster head at lower level). The forwarding nodes consumes battery power of an amount of 1.32mJ per report, which is summation of battery power consumption on receiving & decryption of a report of size 41 byte, and on encryption and transmitting the same of size 41 byte. The forwarding node is the cluster head which receives a report from lower level. The cluster heads at the highest level, and base station of the wireless sensor network are not a forwarding node.

Scalability: Each member nodes receive one cluster message to setup a pairwise key. Member nodes need to keep only one node key, cluster key, and one pairwise key with their cluster heads. Hence, memory overhead, computation overhead, and computation overhead of cluster member nodes remains same as the number of sensor nodes increases. The memory overhead of cluster heads depends on size of cluster, and total number of their member nodes. The total number of member nodes in a cluster will be approximately same as number of sensor nodes increases but the density of sensor nodes per unit area remains same. Hence, memory overhead, computation overhead, and computation overhead of cluster heads remains approximately same as the number of sensor nodes increases. In the proposed security protocol, the base station need to keep node key of all the deployed sensor nodes, information of cluster heads, and their member nodes. Hence, memory overhead, computation overhead, and computation overhead of base station increases as the number of sensor nodes increases. This is not a bottleneck in the proposed security protocol as the base station is supposed to be resourceful.

7. Implementation Details and Simulation Results

The proposed secure hierarchical multi-hop routing protocol is implemented in NesC programming language with the underlying operating system TinyOS and simulation is done by using TOSSIM simulator. TOSSIM is a discrete event simulator for TinyOS. TinyOS is operating system for sensor nodes. Instead of compiling a TinyOS application for a sensor node (mote), users can compile it into TOSSIM, which runs on a personal computer. TOSSIM builds directly from TinyOS code. This allows users to debug, test, and analyze algorithms in a controlled environment. Fig. 2(a) shows the relation between the components of the proposed protocol.



(a) Components of the proposed protocol (b) Configuration file in TOSSIM

Fig. 2 Components and Configuration file of the proposed protocol

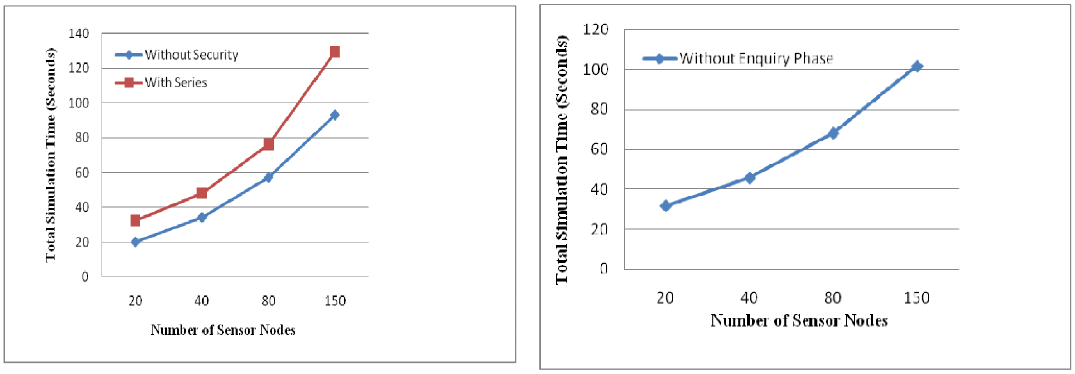
CluserC: ClusterC is the main component. It Contains the logic for initialization phase, cluster setup phase, hierarchical topology setup phase, enquiry phase, and maintenance phase.

EventDetetionC: This component is responsible for monitoring the environment in which the sensor nodes are deployed, for occurrence of events of interest.

MsgAuthC: This component computes MAC, and uses RC5 encryption algorithm to encrypt the messages.

ReportC: This component prepares the report of the occurred event, and reports to the base station.

Due to the resource constrained nature of sensor nodes, the cryptographic algorithms are not only selected based on their strength, but also based on the energy conserving nature. As evaluation of security mechanisms in wireless sensor networks identifies the RC5 encryption algorithm as the right candidate for sensor nodes [22]. So, the proposed protocol uses RC5 for implementation. In order to save the memory the same algorithm is used for performing all cryptographic primitives as in [36] that is the same algorithm has been used for all purposes such as encryption, key computation, and MAC computation. Fig. 2(b) depicts the configuration files of the application in TinyOS. For the simulation purpose, the value of t_c is 700 milliseconds that is after every 700 milliseconds holdback value of every sensor node decreases by one. The simulation graph in Fig.3(a) shows the relationship between the number of sensor nodes, and simulation time for with and without security in the top-down, hierarchical, multi-hop, secure routing protocol for the wireless sensor network.



(a) No. of Nodes vs. Simulation Time With and without security

(b) No. of Nodes vs. simulation time without enquiry setup phase

Fig. 3 Simulation Results

It also shows that the implementation of security methodology consumes the significant amount of simulation time. Fig. 3(a), and Table 1 analyze that hierarchical topology setup, enquiry setup, and event detection & reporting to base station phases consume the significant amount time. Fig. 3(b) shows the relationship between the number of sensor nodes and simulation time for secure sensor network setup in the absence of the *enquiry setup phase*.

Table1: Number of nodes vs. simulation time for cluster setup phase

Number of Nodes	Simulation time for cluster setup phase (in Sec.)		Difference Col2 – Clo3
	without security	with security	
20	11.804	14.735	2.931
40	11.854	14.810	2.956
80	14.105	16.356	2.251
150	16.735	20.917	4.082f

Analytical observations: The simulation results, shown in Fig 3(a), depend on the deployment/distribution of sensor nodes in the target area. Variation in simulation time is mainly due to time required for *cluster setup phase*, and *enquiry phase* that is initial holdback value of each sensor node and how many cluster heads are not connected to secure hierarchical topology after the completion of *secure hierarchical topology setup phase*.

8. Conclusion and Future Work

This paper has narrated the new top-down hierarchical, multi-hop, secure routing protocol for the reactive hierarchical wireless sensor networks. This protocol is resilient to Report Fabrication attack that is the presented protocol suppresses the bogus report generated either by adversary node or by captured member node or by cluster head. The proposed solution relies on symmetric key mechanism which is appropriate for random deployment of sensor nodes in the given target area. This protocol divides the wireless sensor network into non-overlapping clusters in which each cluster member is at most one hop distance from their cluster heads. In the proposed protocol, all the sensor nodes participate in cluster formation either as a cluster member or as a cluster head. Further, the proposed protocol also ensures the participation of all the cluster heads of wireless sensor network in hierarchical topology formation. The proposed protocol is scalable. The memory overhead, computation overhead, and communication overhead of the proposed protocol has been evaluated. The battery power consumption of member nodes, and cluster heads during cluster setup phase, hierarchical topology setup phase, and report preparation and forwards towards base station have been also calculated. It also ensures the confidentiality, integrity, and authenticity of sensed data, and reports.

The proposed routing protocol has been designed with the assumption of static sink node, and sensor nodes. Hence, future work can be done to enhance the protocol considering the mobility of sink node, and sensor nodes. In such a case, the proposed security protocol also needs to be modified.

References

1. http://en.wikipedia.org/wiki/RC5
2. 2004. Crossbow - wireless sensor networks product page, http://www.xbow.com/products/wireless_sensor_networks.htm.
3. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. 2002. Wireless sensor networks security: a survey. *International Journal of Computer and TelecommunicationsNetworkin,38,* 4, 393–422.
4. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. 2004. A survey on sensor network. *IEEE Communication Magazine 40,* 8 (August), 102–114.
5. Anastasi, G., Falchi, A., Passarella, A., Conti, M., and Gregori, E. 2004. Performance measurements of motes sensor networks. In *Proceedings of the Seventh ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '04).* ACM Press, New York, NY, USA, 174–181.
6. Anderson, R. and Kuhn, M. 1996. Tamper resistance - a cautionary note. In *proceeding of the Second Usenix Workshop on Electronic Commerce*. 1–11.
7. Ballare, M., Guerin, R., and Rogaway, P. 1995. XOR MACs: New methods fro message authentication using finite pseudo random function. In *Proceedings of Crypto'95*.
8. Basagni, S., Herrin, K., Bruschi, D., and Rosti, E. 2001. Secure pebblenets. In *Proceedings of the Second ACM international symposium on Mobile ad hoc networking and computing(MobiHoc'01)*. ACM, New York, NY, USA, 156–163.
9. Carman, D. W., Krus, P. S., and Matt, B. J. 2000. Constraints and approaches for distributed sensor network security. Technical report 00-010, NAI Labs, Network Associates, Glenwood, MD. September.
10. Cerpa, A., Elson, J., Hamilton, M., Zhao, J., Estrin, D., and Girod, L. 2001. Habitat monitoring: Application driver for wireless communications technology. In *SIGCOMM LA '01: Workshop on Data communication in Latin America and the Caribbean*. New York, NY, USA, 20–41.
11. Chaithanya, L., Singh, M. P., and Gore, M. M. 2006. Secure data management in reactive sensor networks. In *Proceedings (Lecture Notes in Computer Science Series numbered 4332 published by Springer Verlag) of the Second International Conference on Information SystemSecurity (ICISS-06)*. Springer Verlag, Kolkata, 235–248.
12. Chan, H., Perrig, A., and Song, D. 2003. Random key predistribution schemes for sensor networks. In *Proceedings of the IEEE symposium Security and Privacy*. 197–213.
13. Charny, B. Wireless research senses the future. ZDNet News.
14. Chen, K. Y., Trappe, W., and Martin, R. 2007. Attack detection in wireless localization. In *Proceedings of the Twenty Sixth Annual IEEE Conference on Computer Communications (IEEE INFOCOM'07)*. 1964–1972.
15. Chong, C.-Y. and Kumar, S. P. 2003. Sensor networks: evolution, opportunities, and challenges. In *Proceedings of the IEEE*. Vol. 91(8). 1247–1256.
16. Deng, J., Han, R., and Mishra, S. 2003. A performance evaluation of intrusion tolerate routing in wireless sensor networks. In *Proceedings of the IEEE International Workshop on Information Processing in Sensor Networks (IPSN'03)*. 349–364.

17. Deng, J., Han, R., and Mishra, S. 2006. INSENS: intrusion-tolerate routing in wireless sensor networks. *Journal of Computer Communications 29*, 216–230.
18. Dimitriou, T. and Krontiris, I. 2005. A localized, distributed protocol for secure information exchange in sensor networks. In *Proceedings of the Nineteenth IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)*. IEEE Computer Society,Washington, DC, USA, 240.1.
19. Douceur, J. R. 2002. The sybil attack. In *Proceeding of the First International Workshop on Peer-to-Peer Systems (IPTPS'02)*. Springer-Verlag, London, UK, 251–260.
20. Du, W., Deng, J., Han, Y. S., and Varshney, P. K. 2003. A pairwise key pre-distribution scheme for wireless sensor networks. In *Proceedings of the 10th ACM conference on Computer and communications security (ACM CCS'03)*. ACM, New York, NY, USA, 42–51.
21. Eschenauer, L. and Gligor, V. D. 2002. A key management scheme for distributed sensor networks. In *Proceedings of the Ninth ACM conference on Computer and communications security (ACM CCS'02)*. ACM, New York, NY, USA, 41–47.
22. Guimar~aes, G., Souto, E., Sadok, D. F. H., and Kelner, J. 2005. Evaluation of security mechanisms in wireless sensor networks. In *Proceedings of the 2005 Systems Communications (ICW'05, ICHSN'05, ICMCS'05, SENET'05)*. IEEE Computer Society, 428–433.
23. Hill, J., Horton, M., Kling, R., and Krishnamurthy, L. 2004. The platforms enabling wireless sensor networks. *Communication of the ACM 47,* 6 (June), 41–46.
24. Hills, R. 2001. Sensing the danger. Karlof, C. and Wagner, D. 2003. Secure routing in wireless sensor networks: Attacks and countermeasures. In *Proceeding of the First International Workshop on Sensor Network Protocolsand Applications (SNPA'03*. 113–127.
25. Law, Y. W. and Havinga, P. J. M. 2005. How to secure a wireless sensor network. In *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. Melbourn, Australia, 89–95.
26. Li, M. and Liu, Y. 2007. Underground structure monitoring with wireless sensor networks. In *Proceedings of the Sixth ACM International Conference on Information Processing in Sensor Networks (IPSN'07)*. ACM, New York, NY, USA, 69–78.
27. Liu, D. and Ning, P. 2003. Establishing pairwise key in distributed sensor networks. In *Proceedings of the Tenth ACM conference on Computer and communications security (ACM CCS'03)*.
28. Liu, F., Cheng, X., , and Chen, D. 2007. Insider attacker detection in wireless sensor networks. In *Proceedings of the Twenty-Sixth Annual IEEE Conference on Computer Communications(IEEE INFOCOM'07)*. 1937–1945.
29. Karlof, C. and Wagner, D. 2003. Secure routing in wireless sensor networks: Attacks and countermeasures. In *Proceeding of the First International Workshop on Sensor Network Protocolsand Applications (SNPA'03*. 113–127.
30. Madden, S., Franklin, M. J., Hellerstein, J. M., and Hong, W. 2002. TAG: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Operating System Review 36,* SI (December), 131–146.
31. Martinez, K., Ong, R., and Hart, J. 2004. Glacsweb: a sensor network for hostile environments. In *Proceeding of the First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON'04)*. 81–87.
32. Newsome, J., Shi, E., Song, D., and Perrig, A. 2004. The sybil attack in sensor networks: analysis and defences. In *Proceeding of the International Workshop on Information Processing in Sensor Networks IPSN'04*.
33. Nishimura, C. E. and Conlon, D. M. 1994. Dual use: Monitoring whales and earthquakes using sosus. *Journal Marine Technology Society 27,* 4, 13–21.
34. Oliveira, L. B., Wang, H. C., and Loureiro, A. A. 2005. LHA-SP: secure protocols for hierarchical wireless sensor networks. In *Proceedings of the Ninth IFIP/IEEE International Symposium on Integrated Network Management*. Department of Computer Science, Federal University of Minas Gerais, Brazil, 31–44.
35. Perrig, A., Stankovic, J. A., and Wagner, D. 2004. Security in wireless sensor networks. *Communication ACM 47,* 6, 53–57.
36. Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V., and Culler, D. E. 2002. SPINS: security protocols for sensor networks. *Wireless Network 8,* 5, 521–534.
37. Pottie, G. J. and Kaiser, W. J. 2000. Embedding the internet: wireless integrated network sensors. *Communications of the ACM 43,* 5 (May), 51–58.

38. Rao, G. V. K., Singh, M. P., and Gore, M. M. 2005. Secure adaptive clustering protocol for wireless sensor networks. In *Proceedings of the Thirteenth International Conference on Advanced Computing and Communication (ADCOM 2005)*. 14–19.
39. Savino, I. and Dini, G. 2006. *S2RP*: A secure and scalable rekeying protocol for wireless sensor networks. In *Proceeding of the Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS'06)*. 457–466.
40. Shao, M., Zhu, S., Zhang, W., and Cao, G. 2007. pDCS: Security and privacy support for data-centric sensor networks. In *Proceedings of the Twenty-Sixth Annual IEEE Conference on Computer Communications (IEEE INFOCOM'07)*. 1298–1306.
41. Shi, E. and perrig, A. 2004. Designing secure sensor networks. *IEEE Wireless Communications 11,* 6 (dec), 38–43.
42. Singh, M. P. and Gore, M. M. 2005. A new energy-efficient clustering protocol for wireless sensor networks. In *Proceedings of the Second International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2005)*. Melbourn, Australia, 25–30.
43. Slijepcevic, S., Potkonjak, M., Tsiatsis, V., Zimbeck, S., and Srivastava, M. 2002. On communication security in wireless ad-hoc sensor networks. In *Proceedings of the Eleventh IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*. 139–144.
44. Steere, D. C., Baptista, A., McNamee, D., Pu, C., andWalpole, J. 2000. Research challenges in environmental observation and forecasting systems. In *Proceedings of the Sixth annual international conference on Mobile computing and networking*. 292–299.
45. Szewczyk, R., Osterwell, E., Polastre, J., Hamilton, M., Mainwaring, A., and Esterin, D. 2002. Wireless sensor networks for habitat monitoring. In *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*. ACM, New York, NY, USA, 88–97.
46. Tanachiwiwat, S., Dave, P., Bhindwale, R., and Helmy, A. 2003. Secure location: Routing on trust and isolating compromised sensors in location-aware sensor networks. In *Proceedings of the First international conference on Embedded networked sensor systems (SenSys'03)*. ACM, New York, NY, USA, 324–325.
47. Tsai, H. M., Viriyasitavat, W., Tonguz, O., Saraydar, C., Talty, T., and Macdonald, A. 2007. Feasibility of in-car wireless sensor networks: A statistical evaluation. In *Proceedings of the Fourth IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communicationsand Networks (IEEE SECON'07)*. 101–111.
48. Wood, A. and Stankovic, J. 2002. Denial of service in sensor networks. In *IEEE Computer*. 54–62.
49. Yang, H., Ye, F., Yuan, Y., Lu, S., and Arbaugh, W. A. 2005. Toward resilient security in wireless sensor networks. In *Proceedings of the Sixth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'05), Urbana-Champaign, IL, USA, May 25-27, 2005*. ACM, 34–45.
50. Yang, Y., Wang, X., Zhu, S., and Cao, G. 2006. SDAP: a secure hop-by-hop data aggregation protocol for sensor networks. In *Proceedings of the Seventh ACM international Symposium on Mobile Ad-hoc Networking and Computing (MobiHoc'06)*. ACM, New York, NY, USA, 356–367.
51. Yarvis, M. D., Conner, W., Krishnamurthy, L., Elliott, B., and Mainwaring, A. 2002. Real-world experiences with an interactive ad hoc sensor network. *Proceedings of International Conference on Parallel Processing Workshops*, 143–151.
52. Ye, F., Luo, H., Cheng, J., Lu, S., and Zhang, L. 2002. A two-tier data dissemination model for large-scale wireless sensor networks. In *Proceedings of the Eighth annual international conference on Mobile computing and networking (MobiCom'02)*. ACM Press, New York, NY, USA, 148–159.
53. Ye, F., Luo, H., Lu, S., and Zhang, L. 2004. Statistical en-routing filtering of injected false data in sensor networks. In *Proceeding of INFOCOMM 2004*.
54. Zhu, S., Setia, S., and Jajodia, S. 2003. LEAP: efficient security mechanisms for large-scale distributed sensor networks. In *Proceedings of the Tenth ACM Conference on Computer and communications security (CCS'03)*. ACM Press, New York, NY, USA, 62–72.
55. Zhu, S., Setia, S., Jajodia, S., and Ning, P. 2004. An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE Computer Society, 259–271.