Security Model for Hierarchical Clustered Wireless Sensor Networks

Kalpana Sharma kalpanaiitkgp@yahoo.com
Department of CSE, SMIT,
Sikkim, India

M.K. Ghose mkghose2000@gmail.com
Department of CSE, SMIT,
Sikkim, India

Abstract

The proposed security system for the Wireless Sensor Network (WSN) is based on the WSN security design goal that 'to design a completely secure WSN, security must be integrated into every node of the system'. This paper discusses on two main components of the security framework viz. the secure key management module and the secure routing scheme. The incorporation of security mechanism during the routing protocol design phase is the main focus of this paper. The proposed security framework viz. 'Secure and Hierarchical, a Routing Protocol' (SHARP) is designed for the wireless sensor network applications which is deployed particularly for data collection purpose in a battlefield where the security aspect of the network cannot be compromised at any cost. SHARP consists of three basic integrated modules and each module performs a well defined task to make the whole security framework a complete system on its own.

Keywords: WSN, Security, Hierarchical Clustering, Cluster Heads, Routing, Key Management

1. INTRODUCTION

Wireless Sensor Networks (WSN) are a class of mobile ad-hoc network(MANET) which consists of a hundreds/ thousands of sensor nodes deployed in the area of interest to accomplish a particular mission like habitat monitoring, agricultural farming , battlefield surveillances etc. The sensor nodes are resource constraint devices in the sense that they've limited memory, computational capacity, limited transmission range and operate on a battery thus having limited energy also. Despite these inherent limitations these tiny nodes are used in a variety of applications as they have the ability to sense the environment, process the captured data, aggregate the data and send it wirelessly to the destination which is a powerful entity called the Base Station (BS). Complex cryptographic solutions which are meant for traditional networks cannot be used in the WSN because such algorithms demand a huge computation capability, a large storage space, large bandwidth and unlimited energy supply which a tiny node cannot provide. So instead of concentrating on such complex security algorithms, a lightweight security solutions sound more realistic for the resource starved sensor nodes. Further a security solution concentrating in only one layer, for example physical layer or link layer or network layer, proves to be insufficient for the WSN because such security solution doesn't provide sufficient security to all the layers. So a more practical security solution for WSN would be the development of a security framework consisting of many security service components to provide multi-level security services [6, 8]. Such a framework should interact with all the modules of various protocol layers to provide robust security to the WSN. As per Boyle et al. 'to achieve a completely secure WSN, security must be implemented into every node of the system' [1]. This paper attempts to integrate services provided to the link layer and the network layer to come up with an integrated security solution. The integrated framework of this paper consists of three basic modules forming the core of the proposed security platform. Nevertheless these modules can be used as stand alone modules as well.

There are basically three types of communication in a WSN environment. Theses are one to one communication, many to one and one to many. To secure these communications, the key management module establishes various kinds of keys. To secure the whole WSN, all communication types needs to be secured. Secure keying techniques presented in this paper provides a combination of different kind of keys for secure communication .The secure routing scheme presented in this paper ensures that the messages, using these keys,are securely routed by the nodes and the Cluster Heads (CH) to their final destination i.e. the BS.

The topology that has been considered in this paper is a hierarchical clustering approach. Using this topology the coupling of the security mechanisms like the key usage as well as the routing has been proposed. The proposed security framework is composed of three different modules. These modules are 1) Hierarchical Cluster formation module in which the issues like the formation of tracks, sectors, cluster heads, and neighbor selection are addressed. 2) Key Management module which is responsible for the distribution and maintenance of the keys used in the network. 3) Secure Routing module is responsible for the communication between the BS and other nodes of the network.

The rest of the paper is organized as follows. Section 2 deals with the overview on the related work followed by section 3 which deals with the details pertaining to the module which is responsible for cluster formation. Key management techniques of the proposed framework is presented in section 4 followed by section 5 in which proposed ‘Routing Scheme’ is discussed. Finally in section 6I results and discussions are presented followed by conclusions in section 7.

2. RELATED Work

The various efforts to design optimal security architectures for the WSNs that have been specified/implemented to-date have been described in [1]. The authors have reported that the symmetric key cryptography based architectures have been the main source of security in the WSN to date. Key management is an important activity for ensuring sensor data integrity and securing the WSN communications through cryptographic technique. Design efforts to achieve optimal security architectures of key management for the WSNs are discussed in [1].Key management techniques that have been reported so far can be categorized as follows [24, 25]:

a. Random key Pre-distribution scheme: An example of Random-key pre-distribution schemes is Peer Intermediaries for Key Establishment (PIKE) [26]. PIKE uses probabilistic techniques to establish pair wise keys between neighboring nodes in the network. However, in this approach, each node has to store a large number of keys.

b. Master-key-based scheme: In this scheme, the nodes share unique symmetric keys with the Base station. These keys are assigned before the network is deployed. This involves a significant pre-deployment overhead which is not scalable. Examples of this scheme are Security Protocols for Sensor Networks (SPINS) [2] and Localized Encryption and Authentication Protocol (LEAP) [3], which is discussed in detail in subsequent sections.

c. BS based scheme: Hierarchical Key Establishment Scheme (HIKES) proposed by Ibriq et al. [27] is an example of BS based scheme. In this scheme, the Base station, acts as the central trust authority and empowers randomly selected sensors to act as local trust authorities. These nodes authenticate the cluster members and issue all secret keys on behalf of the Base station. HIKES uses a partial key scheme that enables any sensor node selected as a CH to generate all the cryptographic keys needed to authenticate other sensors within its cluster. The main drawback of this scheme is the storage overhead of the partial key table in every node.

Undercoffer et al. [28] have proposed a security model considering the BS as a trustworthy authority of the entire framework of the sensor network. In this protocol the BS disallows a sensor node from participating in the network, if it detects a node behaving anomalously or becomes compromised. The main drawback of this scheme is the usage of shared keys among the nodes as Zhu et al. [10] have pointed out that a single keying scheme is not sufficient to secure the entire WSN.

Four different keying mechanisms have been provided in Localized Encryption and Authentication Protocol (LEAP) [3], keeping in mind the need for different security requirements for different types of messages. These include the Individual Keys, Group Keys, Cluster Keys and Pairwise Shared Keys. The usage of four different keys supports in-network processing and at the same time restricts the security impact of a node- compromise to only the group members of that node. LEAP + [4] is an improved version of LEAP. Unlike LEAP, a new node sends a message to establish both the pairwise key and the cluster key in LEAP +.

The concept of ‘Secure Triple-Key Management Scheme’ is proposed in [6]. The main drawback of the scheme is that it supports the pre-development key management scheme. TinySec [14] assumes to have a global common secret key among the nodes that is assigned prior to the deployment of the network in order to provide encryption and authentication in the link layer. A standard 8 byte key length is specified for use in the protocol [4], thus making it possible to address smaller sized messages. In [7] the authors have described a scheme for pre-key distribution based on the prior deployment knowledge of the sensor nodes. New schemes for key management for confidential communication between node and its cluster head in hierarchical sensor networks is discussed in [5] wherein the performance analysis was done which shows that Tree- Based Scheme exhibits better performance with some additional storage. SHARP is motivated by the security framework presented by Zia and A.Y. Zomaya, in their research paper “A Secure Triple-Key Management Scheme for wireless sensor networks” [6]. In their paper the authors have used the concept of secure triple-key management scheme. In [8] the authors have presented a security framework discussing the cluster formation and leader election process, secure key management scheme, secure routing and their algorithms, which is the main motivation of this research work presented in this paper. TinySec [14] has been the de facto security solution at Berkeley. The performance of the proposed framework SHARP has been compared with TinySec and is discussed in section 6. Raman et. al. in their paper [16] has bought out that WSN protocols are very deeply dependent on application scenarios, but most of protocols does not use any specific application in its design to achieve this interaction. In [17] the authors have in general described the relationship between the development of various security algorithms and the resources constraint nature of the sensor nodes. A Path Redundancy Based Security Algorithm for Wireless Sensor Networks is discussed in [18] but this security solution is also not integrated security solution and concentrates only in one aspect of security service for the WSN. In [19] Khalil et al. have described an interactive solution only for the resource allocation for the WSN but is not intended for security solution .So it is to be noted that very less research work is being reported in the integration of the security solution.

3. CLUSTER FORMATION MODULE

The main aim of hierarchical routing is to efficiently maintain the energy consumption of sensor nodes by involving them in multi-hop communication within a particular cluster and by performing data aggregation and fusion in order to decrease the number of transmitted messages to the sink. Cluster formation is typically based on the energy reserve of sensors and sensor’s proximity to the cluster head [20]. The hierarchical clustering technique proposed is described in this section.

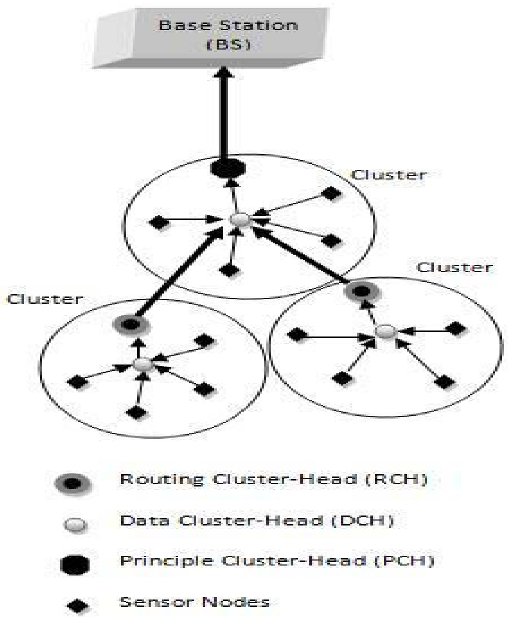

FIGURE1: DCH/RCH of a cluster

Division of the whole network is done in terms of track and sector in order to provide energy efficient and storage efficient key establishment. A detail pertaining to track- sector is available in [9]. The Base station (BS) would divide the network into tracks and sectors. Tracks are required in order to reduce the communication with the BS as most of the nodes belonging to track 1 would serve as “Routing CH” (RCH). In this context the BS would be assumed to be in track 0. Tracks are further divided into sectors. The nodes belonging to the same sector communicate with each other as buddy and in order to be buddy ,each node after getting its sector information and track information from the BS, starts “Neighbor finding phase’ or ‘buddy discovery phase’. Once the neighbors are identified each node stores their neighbor information in ‘buddy info table’. This work is an extension of [11] in which a naïve idea about the selection of the Data Cluster Head (DCH) and Routing Cluster Head (RCH) as well as ‘ Neighbor finding” algorithms are already presented. The reason for dividing the Cluster Heads (CHs) into RCH and DCH is as follows: Rather than a single CH performing both data aggregation and data transmission to the BS this load is divided among two sensor nodes which act as the DCH and RCH in a sector. The BS selects two sensor nodes in each sector as the DCH and the RCH as shown in figure 1. The node preferably in the centre of each sector is selected as the DCH and the node with the minimum distance to the BS in that sector is selected as the RCH. Figure 1 has been reproduced from [13].

DCH is responsible for data aggregation and sending the aggregated data to the RCH. RCH in turn transmits these data packets to the BS if it is in track1 or else it sends to the nearest DCH/RCH belonging to a track of higher level in the track-sector hierarchy. When the number of the nodes in a particular sector is less, then a single CH would act as both RCH/DCH in order to conserve energy.

Division of the network into track-sector and selection of RCH/DCH is the responsibility of the BS. The following assumptions are made in the context of the role of BS in the proposed security framework.

1) Location of each node is known to the BS.
2) Time Division multiplexing (TDM) is used for communication in a group.
3) The BS is very powerful with a huge computational capability.

4) Before any transmission takes place, all the nodes have to register themselves with BS.
5) The BS does the clustering on the basis of the location of the nodes thus dividing the whole network into say 'n' tracks and 'm' sectors where n>>m. The base station also selects cluster head for each sector.
6) The responsibility of the Cluster Head is as follows:
   i) Data aggregation
   ii) Encoding of the message sent by the nodes belonging to its track/sector.
   iii) Communication with the base station.
   iv) The "key table" maintenance.
7) The role of CH is rotational as CH has additional duties in comparison with other nodes. This is done in order to conserve energy or to increase the longitivity of the network.

4. KEY MANAGEMENT MODULE

It is a proven fact that a single keying technique is not appropriate to secure all the communication types of the WSN [10]. So a good keying mechanism should consist of a combination of variety of keying techniques like 'in-network generated keys', 'pre-deployed keys' and 'broadcast keys' [5]. The key management module of this work does not rely on a single key type but makes use of all the above mentioned keys.

4.1. Types of Keys of the Proposed Secure Keying Scheme

Once the 'neighbor discovery 'phase is over and all the nodes have updated their respective 'buddy info table', the BS initiates the key distribution process. In general pair-wise key distribution scheme is set up between the neighbors [12].

In the key management module of this paper there are three broad categories of key 1. Pre-deployed keys 2) In network generated keys and 3) BS broadcasted keys. Before introducing the keys used in this paper, the type of communication of WSN are to be noted. These are as follows:

1. Type 1 communication :Node to Node communication i.e. N:N
2. Type 2 communication :Node to Cluster Head Communication i.e. N : CH
3. Type 3 communication :Cluster Head to BS Communication CH:BS
4. Type 4 communication :Node to BS i.e. N: BS
5. Type 5 communication :BS to Nodes i.e. BS:N
6. Type 6 communication :DCH to RCH Communication i.e. D:R

The following keys are used in the proposed protocol;

1) Buddy key (K_b) is calculated by all the nodes once the neighbor finding work is over. It is used to communicate by each node within its own sector/cluster i.e. for N: N communication.
2) My-Own-Key (K_o) is used by each and every node initially. All the nodes are preloaded with its id and this key is a function of node id, sector-id, track-id, and its residual energy. Ko is used for N: BS communication.
3) Network key (K_n) is issued by the BS after authentication to all the nodes. If a node joins the network it has to send a request to BS for acquiring network key. This request is sent by all the nodes to the BS encrypting it by K_o. BS sends the 'K_n' to the requesting nodes encrypting it with 'K_o' of that particular node. Only the node which is authentic can decrypt this 'K_n'. K_n is used for BS: N as well as for N: CH communication.
4) Cluster Key (K_c) is calculated by CHs for D:R communication.
5) Broadcast Key (K_{bro}) is issued by the BS after authentication as CHs. K_{bro} is used for CH: BS communication.

Node to node communication is nothing but the intra cluster communication which is done using buddy- Key. For simplicity all the keys used in this proposed security framework is listed in the table 1 given below.

Type of key	Composition	Origin	User/communication type
My-Own-Key (K_o)	f(node-id, sector-id, remaining energy level)	Node id preloaded, Nodes calculate it and BS knows it all	All the nodes of the Network i.e. N: BS and BS: N communication.
Buddy-Key (K_b)	*Idsender + f(Idreceiver ,track-id)	All the nodes belonging to the same sector calculate it. Here sectorid (i)=sectorid(j)	All the neighboring nodes belonging to the same sector. For N: N Communications
Network key (K_n)	Calculated by BS	BS to all the nodes	All the nodes.
Broadcast key (K_{bro})	BS Generated	Distributed by BS to RCH & DCH	RCH to BS.
Cluster key (K_c)	f(node id, sector-id, track-id)	Calculated by DCH & RCH.	DCH and RCH.

TABLE 1: Keys usage in various WSN Communication types

* + indicates concatenation operation

4.2 Achieving Security Through the Usage of the Keys

Four rules have been devised for the usage of the keys and Routing.

1. Key distribution: Not all the keys are distributed by the BS. Keys like K_b, K_c and K_o are calculated by each node in the network.
2. Key usage : The key usage rules are discussed in section 4.2.1
3. Key refreshment
   - Broadcast key (K_{bro}) as well as Network key (K_n) are refreshed by the BS at regular interval of time. This refreshment ensures that the nodes belonging to the network is well authenticated from time to time.
4. Key Maintenance: Each node in the network maintains the databases of the following
   - Its own key (K_o).
   - Network key (K_n).
   - Buddy key (K_b).
   - Broadcast key (K_{bro}) and Cluster Key (K_c) (In case of the nodes playing the role of DCH or RCH).

4.2.1 Key Usage

It is essential for all the nodes to know/ calculate their own key K_o. Also all the nodes should possess K_n to take part in the communication as K_n is used for encrypting/ decrypting all the messages broadcasted by the BS from time to time.

Let the initial message to be sent to the DCH by the ordinary node be IM. This first message format would look like this:

$$\boxed{\text{Message(M) = \{ } K_b\text{, TS,MAC, IM\}}}$$

Here TS is the timestamp used to avoid replaying of the messages, MAC is used for authenticating the message. This first message is encrypted using K_b because RCH/DCH is nothing but a buddy to the node in the same sector.

DCH and RCH communicate with each other using cluster key. DCH and RCH compute their cluster-key which is simply a function of their own id, sector-id and track-id. Here sector id may be dropped but for generalization sector id is also considered though all the DCH and RCH should belong to the same sector. But to address the situations where RCH may not be directly communicating with the BS in hierarchical clustering nature of the network, a DCH may have to communicate with DCH or RCH of the cluster above it. So sector id is retained. DCH, RCH communication is encrypted using cluster key. Since DCH is responsible for data aggregation

DCH would send the aggregated message to RCH encrypting it using the K_c and the message format looks like this:

DCH Message(DM) = { K_c,{ K_b, TS,MAC, aggregated (M)}}

Since the number nodes in a particular sector varies and if the number of nodes are less then RCH will function as both DCH and RCH to save energy.
Now for type 3 communication, apart from K_c, another key called K_{bro} is needed. To get hold of Kbro DCH/RCH sends the request message to BS encrypting it using its own-key. BS knows which node has dual function as RCH, RCH and which all have separate DCH, RCH. Accordingly it sends K_{bro} to the requesting CH encrypting them with my-own-key of the requesting nodes. So once DCH, RCH acquire the K_{bro}, communication to RCH to BS takes place encrypting the message by K_{bro} and the message format looks like this

RCH Message(RM) = { K_{bro},{ K_c,{ K_b, TS,MAC, (DM)}}}

The real challenge here is the maintenance of the buddy key. It is already stated that all the nodes set up their keys by communicating with their neighboring node. If a node has to communicate with a large number of neighbors then the WSN has to compromise on its space requirement vs security.

5. SECURE ROUTING MODULE

BS initiates the routing process once the hierarchical clustering topology is fixed and also key distribution is done. Each node is authenticated by their unique id BS has in depth knowledge about a) Id of each node b) Sector no of each node. c) Track no of each node d) Energy level of each node. The BS keeps track of this information in a table which is updated from time to time. When the sensor nodes send a request to BS to join the network encrypting this request with K_o, the BS will be able to decrypt the request message of only the genuine nodes of the network. Once a node is authenticated BS sends the requesting node the network key. When any CH node makes a request for K_{bro} the BS sends the authentic node the broadcast key. Message pertaining to k_{bro} is encrypted using my-own-id key of the requesting CH. If BS needs to broadcast any message to all the nodes then it encrypts the message using K_n. The 'data collection table' which is maintained by the BS is updated after processing the data and obtaining the original message sent by the nodes.

The algorithm for 'Secure Routing' is presented in 5.1.

5.1 Algorithm BS

```
Algorithm: Secure Routing
Begin
Step1.  if BS receives request for authentication from a node then
                goto step2
            else
                goto step1
            end if
Step2.   Check the authenticity of the node
        Step2.1.   Request node sends its request message to BS
                Step2.1.1. $M'=E (M, K_0)$
                Step2.1.2. SEND M'
        Step2.2.  BS decrypts the request message
                Step2.2.1. RECEIVE M'
                Step2.2.2. $M = D (M', K_0)$
Step3. Send network key after node is authenticated by BS
        Step3.1. $M'=E (K_n, K_0)$
        Step3.2. Decrypts message to obtain network key
```

```
    Step3.3. $Kn=D(M',K_0)$
Step4. if CH node makes a request then
         BS sends broadcast key
        $M'=E(K_{bro},K_0)$
    end if
Step5. if BS needs to broadcast message then
        $M' = E (M, K_n)$
    else
        listen
        goto step1
    end if
Step6. if decryption successful at node then
        goto step7
    else
        discard M'
        goto step10
    end if
Step7. Compute $M = D (M', K_n)$
Step8. Update 'data collection table'
Step9. goto step11
Step10. if data retransmission necessary then
        Broadcast message
     else
        goto step1
     end if
Step11. End
```

This algorithm is implemented in C++ and the results have been compared with that of TinySec [14] and Triple-Key security algorithm [6].

It is also to be noted that the encryption/ decryption technique used in this algorithm is RC5 block cipher. A detail on feasibility on the RC5 usage in the WSN is available in [21, 22].

6. RESULT ANALYSIS

To analytically evaluate the cost of the security of the proposed security platform computation overhead, communication overhead and storage overhead for packet processing is considered.

To extend the battery life of resource constraint sensor nodes it is necessary to limit the energy consumption. But to provide security to the WSN a price have to be paid in terms of significant amount of energy consumption for bare minimum and unavoidable requirement of security services like encryption, decryption, and key management. Key management also demands extra storage space to store the required keys for the secure communication. The performance comparison provided in this section presents the space and computation overhead.

The packet format for SHARP is shown below. The performance comparison is done with TinySec [14] and Triple Key [6, 8] the packet format. All the values are in bytes.

6.1 Packet Formats

Dest	AM	GRP	Length	Data	CRC
2	1	1	1	29	2

Tiny OS

Dest	track-id	keys	L	sector id	Src	TS	Data	MAC
1	1	5	1	1	1	1	29	4

Proposed packet format

Dest	AM	Length	Data	MAC
2	1	1	29	4

TinySec –Auth

Dest	AM	Length	Src	Ctr	Data	MAC
2	1	1	2	2	29	4

TinySec-AE

ID	Keys	TS	Data	MAC
5	4	1	29	4

Triple Key

For SHARP since all the keys are the function of node-id, sector-id, track-id, at most (5+1+1+1+1) bytes i.e. 9 bytes are required for key and ID storage. MAC is used for message authentication and integrity. All the values are in bytes. In SHARP the data packet is not more than 44 bytes long and these packets can be transmitted easily in sensor nodes available in the market today.

6.2 Performance Comparison –Computation Overhead

The comparison of packet size overhead for TinySec, TripleKey and SHARP is shown in Table 5.2. The values pertaining to TinySec is obtained from [14] whereas TripleKey values are obtained from [21].

	Application data (bytes)	**Packet Overhead**	**Total Size**	**Time to transmit (ms)**	**Increase over TinyOS Stack (%)**	**Latency overhead (%)**	**Energy overhead (%)**
TinySec-Auth	29	8	37	26.6	1.5	1.7	3
TinySec-AE	29	12	41	28.8	8	7.3	10
TripleKeys	29	11	40	28.3	6.3	5.9	2.8
SHARP	29	15	44	30.7	12.8	11.5	12.2

TABLE 2: Comparison of SHARP with TinySec & Triple-Keys

There is an increase in the packet size for the proposed security solution. It results in an increase in the usage of bandwidth and energy needed to send the packets, as evident from Figure 2 and Table 2. However, this framework provides multiple layers of security. It is also observed that there is an increase in time to transmit parameter. Accordingly, there is 9% increase over current TinyOS stack, which can be compensated by multiple-layer of security provided by SHARP.

The energy overhead of TinySec and Triple keys is comparatively lesser than that of SHARP. However, SHARP overcomes the shortcomings of all these protocol with acceptable energy overhead. The graphical representation based on the packet sizes are provided in Figure 2.

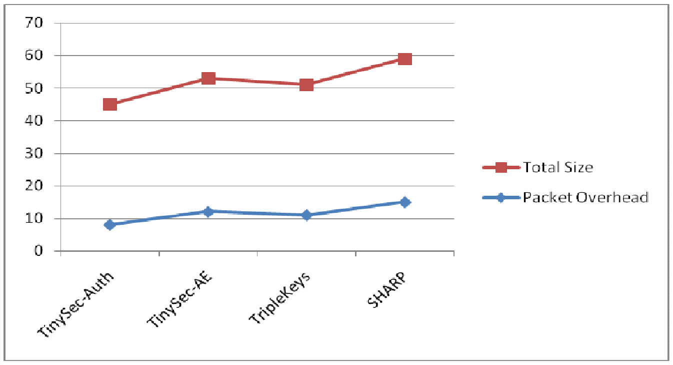

FIGURE 2: Packet size comparison.

6.2.1: Advantages of SHARP over TinySec:

TinySec is not a complete security framework. It is based on a number of assumptions as defined below:

- TinySec concentrates only on Link Layer Security.
- There is no particular keying mechanism specified for TinySec.

SHARP addresses both these shortcomings but it is at the cost of increased packet size. This increase in packet size is an acceptable to be used for the sensor nodes available in the market.

6.2.2: Advantages of ISF over TripleKeys

Triple-Keys have shown considerable savings in bandwidth and battery life, thus have energy advantage over both TinySec and SHARP. However, it has following deficiencies, which have been addressed in the proposed ISF scheme.

- To reduce the packet size, the AM field of the of the triple-key packet format is dropped. Hamed et al [26] have observed that the removal of AM type field introduces several major problems for upper layer services as it directly affects the Active Message Model of TinyOs.
- The id and key fields are combined. Moreover, the number of bytes reserved for the source / destination fields of the message are not mentioned.
- The scheme uses three keys, two of which are pre-deployed keys. These important pre-deployed keys when compromised, the whole network is compromised. ISF being a multilayered security scheme, the compromise of a key of a particular sector effect only that cluster but not the entire network.

6.3 Storage Overhead

As it is known that all the nodes in this WSN need to store keys for communication. A simple node stores at least three keys for encryption process throughout and the minimal keys being my-own-id key, network key and buddy key (which is with respect to its neighbors).Apart from this if the node is DCH / RCH it needs to store all the three above mentioned keys as well as cluster key and Broadcast key. It obviously increases the storage overhead. But not all the keys are pre-loaded. Two keys i.e. network key, and broadcast keys are always broadcasted by the BS and my-own-id keys are only pre-loaded. Yet the storage overhead is expressed as (for any node)

$$\boxed{\text{Storage overhead= [Size of}(K_o)+K_b\ (X-Y)+K_n]}$$

where X is the total number of the nodes in the WSN and Y is the subset of X which is nothing but the population of only those nodes which belong to the same sector i.e. selected neighbors of that particular node belonging to the same sector. The storage overhead for DCH/RCH would be

$$\boxed{\text{Storage overhead= [Size of}(K_o)+K_b\ (X\text{-}Y)+K_n+K_c+K_{bro}]}$$

Even if the default key size is considered to be 10 bytes each and the maximum number of neighbors is taken as 50, the storage overhead for an ordinary node would be 60 bytes and for a CH it would be at most 90 bytes which is less than 1 KB. This overhead is acceptable as almost all the sensor nodes available in the market including Berkeley's motes have a memory size of more than 4 KB.

Though there is an overhead in storage but on the flip side because of the hierarchical clustering techniques there're two CHs in a single cluster which results in a multi-hop communication which obviously results in energy efficiency in comparisons with those clustering techniques in which there is only one cluster head, as in [13]. The simulations results reported in [13] are comparable with that of LEACH [15] in certain cases.

To summarize the overall performance of SHARP it is to be noted that when the algorithm presented in section 5 is to be implemented in the same environment where TinySec is implemented and the same sized application data is considered then it can be predicted that there would be an increase in 'Time to transmit parameter. It is because of the increase in the packet size. Thus there would be at most 9% increase over current TinyOS stack. But this increase is compensated by 'multiple-layer' of security provided by SHARP.

Again it is to be noted that the security framework presented in [6] have single point of failure as there's only one cluster head. Moreover when the cluster keys are to be stored then it requires more storage space than SHARP as the clustering in this work is not only hierarchical but also energy efficient because of the division of the network into tracks and sector .As already stated the number of neighbors is restricted because of the division of the network into tracks and sectors. Tracking and sectoring not only restricts the no. of neighbors with whom key establishment needs to be done but also aids in energy conservation. Energy efficient key management technique is thus made possible by adapting the concept of sectoring and tracking which is a new concept in this proposed protocol. In case of a node compromise, the nodes belonging only to the' buddy set' are compromised but not the entire network. To manage the buddy set, computation is preferred over communication as communication is much more expensive than computation in WSN.

7. CONCLUSION

This work is motivated by the research work presented in [6, 8]. In this paper an effort has been made to couple Routing with Secure key management. Track /Sector and selection of two CHs has been unique features of this paper. The security scheme presented in this paper is energy efficient and at the same time it ensures that the whole network is never compromised even if there has been an attack in the network. This is possible because of clustering in terms of tracking and sectoring. However, there's a scope of improvement in this framework also. First of all if the nodes calculate most of the in its own level rather than depending on the BS to communicate the keys then there would be a lot of energy saving. If the packet size is reduced by dropping some of the redundant fields by decreasing the size of the keys stored then there would be less bandwidth usage. Further the whole security framework can be simulated/ implemented using the 'real sensor network scenario' using real sensor nodes. Further, the security framework can be implemented in Berkeley's motes for accurate results and for performance comparison with TinySec.

8. REFERENCES

1. David Boyle and Thomas Newe," Securing Wireless Sensor Networks: security Architectures" (2008), Journal of Networks, VOL. 3, NO. 1,pp 65-77.

2. Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and J.D.Tygar. Spins: Security protocols for sensor networks. Wireless Networks, 8:521 – 534, 2002.

3. Zhu, S., Setia, S., Jajodia, S. (2003) ‘LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks’, CCS ’03, Washington D.C., USA, 27 – 31 October 2003, New York, USA: ACM Press, 62-72.

4. Zhu, S., Setia, S., Jajodia, S. (2006) ‘LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks’, ACM Transactions on Sensor Networks TOSN,2(4), 500-528.

5. A.S.Poornima and B.B.Amberker,” Key Management Schemes for Secure Communication in Heterogeneous Sensor Networks”, International Journal of Recent Trends in Engineering, Vol 1, No. 1, May 2009.

6. T.A Zia and A.Y. Zomaya, ‘A Secure Triple-Key Management Scheme for wireless sensor networks’, in the proceedings of INFOCOM 2006,25th IEEE International Conference on Computer Communications, Barcelona, pp1-2 ,23-29 April 2006 .

7. W. Du, J. Deng, Y.S. Han, P.K. Varshney, A key predistribution scheme for sensor networks using deployment knowledge, IEEE Transactions on Dependable and Secure Computing, Vol. 3, issue 1, Jan-March 2006 pp.62-77.

8. Tanveer Zia and Albert Zomaya,” A Security Framework for Wireless Sensor Networks “, SAS 2006 – IEEE Sensors Applications Symposium Houston, Texas USA, 7-9 February 2006.

9. Navin Gautam, Won-Il Lee, Jae-Young Pyun, "Track-Sector Clustering for Energy Efficient Routing in Wireless Sensor Networks," cit, vol. 2, pp.116-121, 2009 Ninth IEEE International Conference on Computer and Information Technology, 2009.

10. Du X, Xiao Y, M Guizani, Chen H.H, (2007) “Effective key management for sensor networks, an effective key management scheme for heterogeneous sensor networks”, Ad Hoc Networks, Volume 5, Issue 1, 1 January 2007, Pages 24-34.

11. Kalpana Sharma, S.K. Ghosh, and M.K. Ghose ‘Establishing an Integrated Secure Wireless Sensor Network System: A New Approach’, International Journal of Next Generation Networks (IJNGN), Sept.2010

12. D. Liu and P. Ning, “Establishing Pairwise Keys in Distributed Sensor Networks”, ACM CCS 2003.

13. Kalpana Sharma, Anurag S. Rathor, S. R. Biradar, M.K Ghose ,‘Power-efficient Routing & Increased Yield Approach for WSNs ‘,International Journal on Computer Science and Engineering (IJCSE),Vol. 02, No. 03, 2010, pp 586-592.

14. C. karlof, N. Shastry and D. Wagner, TinySec: A link layer security architecture for wireless sensor networks, SenSys’04, November 3-5, 2004, Baltimore, Maryland, USA

15. Heinzelman, A. Chandrakasan and A. Balakrishnan,’ Energy-Efficient Communication Protocol for Wireless Microsensor Networks’, proceedings of the 33rd Hawaii International Conference on System Science, Jan 2000.

16. Bhaskaran Raman et. al, ‘Censor Networks: A Critique of “Sensor Networks” from a Systems Perspective’, ACM SIGCOMM Computer Communication Review, Volume 38, Number 3, July 2008.

17. Al-Sakib Khan Pathan et al. “Security in Wireless Sensor Networks: issues and Challenges”, ICACT2006 in Feb. 20-22, 2006, ISBN 89-5519-129-4 pp(1043-1048).

18. Sami S., Wakeel and Eng. Saad A. AL-Swailem,"PRSA: A Path Redundancy Based Security Algorithm for Wireless Sensor Networks", WCNC 2007 Proceedings, pp (4159-4163).

19. Ayman Khalil, Matthieu Crussière and Jean-François Hélard,'Cross Layer Resource Allocation Scheme under Heterogeneous constraints for Next Generation High Rate WPAN (2010),International Journal of Computer Networks and Communications(IJCNC) vol 2, No. 3.

20. A.R. Masoum, A.H. Jahangir,Z. Taghikhani, and R. Azarderakhsh, (2008) "A new multi level clustering model to increase lifetime in wireless sensor networks", Proceedings of the Second International Conference on Sensor Technologies and Applications, pp 185-190.

21. Germano Guimaraes, Eduardo Souto, Djamel Sadok, Judith Kelner (2005), "Evaluation of Security Mechanisms in Wireless Sensor Networks", proceedings of the 2005 Systems Communication (ICW '05) ,P 428-438, 0-7695-2422-2/05.

22. Kyung Jun Choi and Jong-In Song (2006), " Investigation of Feasible Cryptographic Algorithms for Wireless Sensor Network", in the proceedings of International Conference of Advanced Communication Techniques (ICACT 2006), 89-5519-129-4.

23. Soroush Hamed, Salajegheh Mastooreh and Dimitriou Tassos (2007) "Providing Transparent Security Services to Sensor Networks" Proceedings of IEEE International Conference on Communication (ICC'07).24-28 June 2007.

24. Zhou, Y., and Fang, Y. (2007), 'A two-layer key establishment scheme for WSN'. IEEE trans. Mobile Computing, Volume 6, No. 9, pp: 1009-1020.

25. Zhou, Y., Fang, Y., and Zhang, Y. (2008), 'A survey of Securing Wireless Sensor network'. IEEE communication surveys, Volume 10, No. 3, pp: 6-28.

26. Chan, H.,and Perrig, A., (2005), 'PIKE: Peer Intermediaries for Key Establishment in Sensor Networks'. In Proceedings of IEEE Infocom, Miami, Florida, pp: 524-535.

27. Ibriq, J., and Mahgoub, I. (2007), 'A Hierarchical Key Establishment Scheme for Wireless Sensor Networks'. 21st International Conference on Advanced Networking and Applications, pp: 210-219.

28. Undercoffer, J., Ayancha, S., Joshi, A., Pinkston, J. (2004), 'Security for Wireless Sensor Networks', Wireless Sensor Networks, Kluwer Academic Publishers Norwell, USA, pp: 253-275.