Chapter 8

A Hierarchical and Role Based Secure Routing Protocol for Large and Mobile Wireless Sensor Networks

8.1 Introduction

8.2 Background and Related Work

8.2.1 Key Management Approaches
   8.2.1.1 Single Master Key Scheme
   8.2.1.2 Full Pair Wise Key
   8.2.1.3 Random Key Pre-distribution
   8.2.1.4 Group Based Key Method

8.2.2 Some Security Protocols

8.3 Problem Statement

8.3.1 Statement
8.3.2 Security Goals in WSN
8.3.3 System Model and Assumptions
8.3.4 Difficulties for Implementing Security Schemes

8.4 Proposed Technique

8.4.1 Secure Hierarchical Routing Protocol for Mobile WSN (SHaRP) Framework
   8.4.1.1 SHaRP Components
   8.4.1.2 Various Role Assignments and Routing Protocol Overview
   8.4.1.3 Functioning of SHaRP Framework
      8.4.1.3.1 Initial System Establishment Module
      8.4.1.3.2 System Operation Module
      8.4.1.3.3 Key Management Module

8.4.2 Security Analysis of SHaRP Framework

8.5 Efficiency Analysis

8.5.1 Security Properties
8.5.2 Implementation Analysis
8.5.3 Defense against Network Layer DoS attack Analysis
8.5.4 Spreading of Attack
8.5.5 Traffic Analysis (Message Complexity)

8.6 Conclusion
Nodes in Wireless Sensor Networks are highly resource constrained and topology of such networks remains highly dynamic due to sudden death of sensor nodes and break of links. When mobility to such nodes is introduced it gives rise to a highly complex network system since the topology becomes even more dynamic. Providing security in the routing layer of wireless sensor network is a highly critical issue mainly due to the stringent resource constraints and dynamic topology. In this chapter a Secure Hierarchical Routing Protocol for Mobile Wireless Sensor Network (SHaRP) is proposed which is a combination of symmetric and asymmetric cryptographic approach. In the SHaRP framework the sensor field is divided into some logical clusters and each cluster contains nodes with different roles such as Ordinary Sensor Node (OSN), Cluster Head (CH) and Gateway Node (GN). Various keys are used by the nodes with different roles for message encryption and decryption. The overall security has been entrusted on different roles in a distributed manner instead of relying on one authority and that is why this approach is safe from single point failure problem. This also enhances the security in the network system. In this chapter focus is more on various security aspects in the routing process rather than issues like cluster formation and role distribution algorithms. Most of the computing overhead is shifted to the Base Station as it is assumed to be a resourceful and reliable node. This obviously improves the energy efficiency of the protocol. Author also reports a security analysis along with implementation issues of the proposed protocol.

A part of the work done in this chapter is published in the form of a paper in IEEE ICIMT 2010.

8.1 Introduction

Wireless sensor network systems (WSNs) are composed of large number of sensor nodes which are low powered and also resource constrained sensing devices [16]. These sensor nodes communicate wirelessly mainly through radio. The number of sensor nodes deployed to study a phenomenon may be in the order of hundreds or thousands depending on the application. Therefore, a WSN may be of large scale as well as of high density [97]. Wireless sensor networks are typically deployed in large geographic areas to collect data from the environment (some times harsh or hostile) of interest through the sensor nodes and finally present the data to users located far away from the geographic area under consideration [74][93][98]. It is a kind of data collection through sensor nodes and their wireless links from remote areas or environments which may be even hostile at times depending on the type of application of the WSN. The data collected by the sensor nodes may be some environmental parameters such as temperature, pressure, humidity etc or it may be even image or video depending on the kind of application and the kind of sensory device installed in the node [98][99][100][101]. There is wide range of applications of WSN starting from environmental monitoring, smart home applications, healthcare, wildlife monitoring, forest fire detection and alarming system etc even up to battlefield awareness and target tracking including some more defense applications [98][102][103]. Depending on the kind of application the level of security of information in the WSN varies. Some applications may not need security at all but in the defense applications and in some mission critical applications the security of information in the network is of highest priority.

Within a wireless sensor network, the sensor nodes communicate among themselves to exchange sensory data and network control information. Since the sensor nodes communicate through the wireless medium such communications are vulnerable to various active and passive attacks. Overall, a wireless sensor network may have to suffer from various active and passive attacks [95] on its communication protocols and communication devices. Therefore, secure communication among sensor nodes is of paramount importance for successful deployment and operation of a WSN.

Encryption and subsequent decryption through some keys are the major techniques by which any communication can be made secure. Looking at the resource constraints of the sensor nodes and WSN as a whole it is infeasible to deploy Public Key Infrastructure (PKI) in WSN. Therefore, Symmetric Key Cryptography (SKC) is one alternate solution to this issue. But the level of security that can be offered by SKC is not that high, therefore, there has to be a compromise between the computational complexity of the security algorithm and the level of security offered by the security algorithm. An ideal approach would be an optimized design of security algorithms in a given condition (i.e., various resources in the system) without significant degradation of the level of security. Whatever the approach followed for ensuring security in WSN the key management in WSN is a very tricky issue. It is so due to the resource constraints in the WSN system. Because of the limited memory in the sensor nodes, it is infeasible for a sensor device (node) to store a shared key value for every other sensor nodes in the WSN system. The memory limitation of the sensor nodes restricts the size of the keys and the
number of such keys that can be preinstalled in each sensor nodes into short and a small number respectively. For example, the capabilities of sensor nodes so far Smart Dust Sensors [80][104] are concerned, are very limited i.e., each node has only 8 kb of program and 512 bytes for data memory. At the same time small key pool size increases security vulnerabilities [105][106]. Thus key management itself in WSN is an issue of higher complexity.

Routing is a network layer issue [16] and routing in WSN is different from routing in classical TCP/IP based networks [53][2]. In-network aggregation is one aspect of WSN which makes the design of secure routing protocol complicated and also makes it different from classical TCP/IP based network [2]. In conventional networks a secure routing protocol generally takes care of guaranteed message availability only. However other related security issues such as message integrity, authenticity and confidentiality are handled at a higher layer by end-to-end security mechanism such as SSH or SSL [95][107]. Such types of end-to-end security mechanisms are not possible for WSN because of in-network processing requirement which is again an important aspect of WSN. Due to this the intermediate nodes in a route in WSN need direct access to the contents of the messages and this complicates the entire process of secure routing.

Secure routing in WSN is a highly important issue specifically for any mission critical application or defense application. This can be achieved by integrating key management schemes with regular data gathering schemes. In this chapter, a secure routing algorithm for large scale WSN is proposed. The secure routing algorithm has been designed keeping security as a major goal in the routing process. It is a hierarchical one and the major portion the entire key management process is based on Group Based [108][109] method. While designing the protocol, various resource constraints of WSN are considered and it has been designed in an energy efficient manner by putting much emphasis on reduction in communication overhead and computational overhead. The entire sensor field is logically divided into some clusters and communication takes place in a hierarchical manner. This approach gives energy efficiency as well as scalability. Provisions are kept to make the protocol fault tolerant. Moreover, security to the information is ensured through various levels and also in distributed manner. It is difficult to divulge secure information in the WSN by single point of compromise and this needs at least a threshold number of nodes to be compromised at the same time. This approach improves the security level significantly.

8.2 Background and Related Work

In WSN, the communication media is always vulnerable to different attacks, moreover limited node capabilities and possible insider threats put big challenge in front of the designer in order to ensure security to such network systems. The adversaries can use powerful laptop computers with high energy and also with long communication range with an intention to attack the WSN system and again this makes secure routing protocol design task a non trivial one [85].
8.2.1 Key Management Approaches

There are several security related proposals on the basis of key management for the overall security of WSN. These proposals can be classified into following categories:

i) Single Master Key, ii) Full Pair wise Key, iii) Random Key Pre-distribution Scheme, & iv) Group Based Key.

8.2.1.1 Single Master Key Scheme

In this scheme, all nodes share a common key known as master key. This technique can secure data transmissions between sensor nodes. Simplicity of this technique may encourage adapting it specifically in a storage constrained environment like WSN system in which the sensor nodes have very limited memory to store the keys. This scheme is also very easy to implement. But the security level it can provide is very low, since if an adversary becomes successful in capturing the master key, the entire network becomes compromised. [110]

8.2.1.2 Full Pair wise Key

In this scheme, each pair of nodes in the WSN system adopts a unique pair key to encrypt and decrypt messages. Thus, each node needs to store n-1 keys where n is the total number of nodes in the WSN system. Moreover, these keys to the nodes are to be distributed prior to the operation of the network. Therefore, in the entire WSN there shall be n(n-1)/2 keys. The memory requirement in the nodes is very high. Again, along with the increase or decrease in the number of nodes in the WSN system there is a need to update pair wise keys for all nodes which also increase the computational overhead in the system and hence more power consumption. This computational overhead in re-keying and huge memory requirement for storing the keys i.e., storage overhead make this scheme inappropriate for large scale WSN. The security level this scheme can provide is moderate. A compromised node can effect the secure link between itself and other nodes directly connected with it but does not effect the security among the uncompromised nodes. [111][112]

8.2.1.3 Random Key Pre-distribution

This scheme relies on a probability value p of a common shared key k between two nodes i and j. Each node i in the WSN system is associated with a key ring of K randomly chosen keys from a system key pool P. If any two nodes i and j have a common key then it is said that there exists a secure link (i → j) between i and j. Therefore, a larger key ring size increases the probability of sharing common keys by different nodes and therefore, connectivity in terms of secure link also increases in the network system. But, it also increases the storage overhead at each sensor node. The security level of this scheme is moderate. The bigger key ring size also raises the compromised effect on a path shared between uncompromised nodes. In this scheme, a node i may share a key with several nodes and then just based on the shared key, it becomes impossible for i to
recognize uniquely with which node it is currently connected. Though this may be solved adding some additional information tag with each message transmitted, this shall obviously raise the computational overhead and also the packet size. This increases the energy requirement in the system. It can provide moderate security level at the expense of moderate storage. This scheme provided flexibility for the designers of sensor networks to tailor the network deployment to the available storage and the security requirements [113][114].

8.2.1.4 Group Based Key Method

In this scheme, the entire sensor field is divided into some disjoint groups. These groups sometimes are termed as clusters or zones. Each sensor node shares a common intra-group key for secure data communication with its neighbors within the same group. Similarly, some special sensor nodes within a group may share inter-group key with some other special nodes from other groups for having secure communication with other neighbor groups or so on. This mechanism outperforms other methods as mentioned above in terms of performance, scalability, storage overhead etc [108][109].

8.2.2 Some Security Protocols

This part is in addition to the state of art regarding secure routing in wireless sensor networks reported in chapter 2 of this thesis. There are few security protocols proposed so far and all of them consider wireless sensor networks with static sensor node. Perrig et al. (2002) proposed SPINS [88] which is a security protocol suite optimized for wireless sensor networks. It has two security building blocks namely Secure Network Encryption Protocol (SNEP) and µTESLA which run on top of TinyOS. µTESLA is the micro version of TESLA (Timed Efficient Stream Loss-tolerant Authentication). Jing Deng et al. (2002) designed INSENS (INtrusion-tolerant routing protocol for wireless SEnsor NetworkS) [90]. INSENS does not rely on detecting intrusions but it can tolerate intrusion by bypassing the malicious nodes. INSENS protects the network from spreading of the attack throughout the network. Zhu et al (2003) proposed LEAP(LoCalize Encryption and Authentication Protocol)[115] which is a key management protocol for wireless sensor networks motivated by the observation that different types of messages with different security requirements get propagated through the network. Karlof et al. (2004) proposed TinySec [116]. It provides similar services like SPINS including access control, message integrity and confidentiality. ZigBee (2005) [117] specification outlines the design of network layer that operates just above the PHY and MAC layer specified by the IEEE 802.15.4 standard. The concept of “trust center” is introduced in the specification. There are three types of keys specified for use in ZigBee security services: master key, link key and network key [117]. A D Wood et al. (2006) designed SIGF (Secure Implicit Geographic Forwarding) [91] which is a family of configurable secure routing protocols for wireless sensor networks which can provide resource bound security solution that is good enough and has high performance. Zhen Cao et al.(2007) proposed FBSR (Feedback based Secure Routing Protocol for wireless sensor networks) [96] which makes use of the feedback of the neighbor nodes for making the forwarding decision in secure and energy efficient manner. It uses Keyed One Way Hash Chain
(Keyed-OWHC) to avoid feedback fabrication. Nasser et al. (2007) proposed SEEM (Secure and Energy-Efficient Multipath routing protocol) [78] for wireless sensor networks and it uses multi-path alternately as the path for communication between two nodes. SEEM is resistant to some specific routing layer attacks that have the character of pulling all traffic through the malicious nodes (i.e., wormhole, sinkhole, selective forwarding attack). Junqi Zhang et al. (2008) [118] proposed a new security scheme for wireless sensor networks based on ID-based secure group key management approach. This approach minimizes the key storage requirement and also the number of the communication messages for re-keying. Zhong Su et al. (2009) [119] proposed an efficient technique for secure communication in large-scale wireless sensor networks which is based on random key pre-distribution scheme. In this approach (Random Pairwise Keys Establishment Scheme) the nodes are categorized either as auxiliary nodes or ordinary nodes prior to network deployment and make use of these nodes for establishing pairwise keys. All these already proposed secure routing protocol consider wireless sensor network consisting of static node. None of these addresses the issue of mobility of the nodes in the network.

8.3 Problem Statement

This section defines the problem undertaken in this chapter. The entire problem is stated through different phases as mentioned under.

8.3.1 Statement

The main objective is to design a secure routing protocol for a wireless sensor network in which the nodes as well as the Base Station are mobile. The protocol should be energy efficient, scalable and robust against different attacks in the network system in a balanced manner.

8.3.2 Security Goals in WSN

In a broader sense the major threats to a wireless sensor network are eavesdropping and spurious data injection. Therefore, putting defense against these threats is the major security goal. It demands various security services as mentioned below in order to make the data secure in the network.

a) Authentication: This service ensures that the message has originated from a legitimate source node and the message has also reached only the intended destination node. Authentication proves the identity of the source.
b) Data Integrity: This service ensures that the message has reached the destination in its correct state i.e., the message has not been changed or forged during the transmission.
c) Access Control: This service implies that only authorized entities or nodes can access the necessary messages.
d) Data Confidentiality: This service ensures that no third party other than the sender and the intended recipient can divulge the content of the message.
e) **Non-Repudiation**: It proves the source of a packet. Non-Repudiation prevents the source (i.e., sender) from denying that it sent the packet.

f) **Data Freshness**: It ensures that a malicious node does not resent an old packet which has been captured previously by the node.

This indicates key agreement is an essential requirement in order to achieve the security goals through the security services mentioned above.

### 8.3.3 System Model and Assumptions

The nodes in the network system are moderately mobile. The nodes are resource constrained in terms of various resources such as available onboard memory, computing capability, communication bandwidth, and most importantly available battery power. It has been assumed that the source of energy of the sensor nodes which battery is not renewable. The sensor nodes are not reliable and they need to be authenticated. The Base Station is resourceful and also reliable. The Base Station is also moderately mobile but it always resides far away from the Sensor Field in spite of its mobility. The BS never comes into the area of deployment of the sensor nodes. It has been assumed that there exists some GPS free solution to know the geographic location of each sensor node by the Base Station [121][122]. The sensor network uses radio-link as communication medium which is assumed to be insecure and vulnerable to attack. Attackers eavesdrop the radio transmissions, they can inject spurious data and also they may replay some already heard data packets. Attackers can also deploy some identical (hardware) nodes in the field to provide vague information and mislead legitimate nodes.

### 8.3.4 Difficulties in Implementing Security Schemes

Resource requirements (i.e., processing capability, storage space, memory, and communication bandwidth) are generally high for a node to participate in the execution of security protocols. There are various cryptographic schemes which demand different volume of such resources in the network system. Asymmetric Key Cryptography is not feasible for a wireless sensor network system if low cost sensor nodes are used in the system. Again security level provided by Symmetric Key Cryptographic techniques may not be very high. Applicability of Elliptic Curve Cryptography (ECC) is shown in [123]. ECC is an approach to Public Key Cryptography which is based on the algebraic structure of elliptic curves over finite fields [95][124]. These are the real life constraints due to which it is extremely difficult to put security solutions in practical applications of WSN. Mica Mote [125] is a commercially available product regarding sensor node. It has an 8-bit microprocessor running with 4 MHz clock speed, 4 KB RAM and 512 KB flash memory. The energy source for these nodes is alkaline battery that can work between 3.2 and 2 V. Therefore, battery life is another major concern. And therefore, considering all these constraints one has to design best possible balanced security solution for a given situation.
8.4 Proposed Technique

The proposed secure routing algorithm is hierarchical one. An important feature of the proposed protocol is that most of the computing burden has been shifted to the Base Station and the number of communications between Base Station and ordinary sensor nodes are minimized. After deployment of the sensor nodes in the field the Base Station divides the sensor field into some clusters. Each cluster is nothing but a group of closely located sensor nodes. It is assumed that the Base Station uses some efficient clustering algorithms and in this work focus is on routing part of the system in which the decision regarding which node to whom and when the data are to be forwarded is of prime concern. Each cluster has one Gateway Node (GN), two Cluster Head Nodes (CH), and remaining as Ordinary Sensor Nodes (OSN). The various roles such as GN, CH or OSN to different sensor nodes are assigned by the Base Station. A typical clustered sensor field is depicted in the Figure 8.1 below:

![Figure 8.1 A typical Clustered Sensor Network under SHaRP framework.](image)

A particular setup of cluster and various nodes normally works for a predefined time interval called cycle. But, still there exists some exception conditions like severe partitioning of the sensor network due to the mobility of the nodes or sudden death of some significant number of sensor nodes in the network etc, when the clustering algorithm and role assignment process are to be called before the current cycle ends.

Figure 8.2 shows the communication hierarchy present in the sensor field. The Ordinary Sensor Nodes communicate with the Cluster Head Node and Cluster Head Nodes communicate with the Gateway Node; finally the Gateway Nodes are responsible for transmitting data to the Base Station.
In this approach, most of the message encryption-decryption processes are symmetric in the sense that same keys are used for encryption as well as decryption process. The key generation and encryption-decryption process for CH ↔ GN pair is based on Elliptic Curve Cryptography and this is an instance of Asymmetric Key Cryptography. A combination of various key management schemes such as Single Master Key, Pair wise Key and Group Based Key is used. The common key ($K_{COM}$) is a kind of Single Master Key scheme, $K_{HGN}$ is an instance of Pair wise Key scheme and $K_{ICK}$ & $K_{ECK}$ are instances of Group Based Key. These significances of these various keys are described in the following section.

In the proposed security framework the entire sensor field is organized into some virtual clusters and each cluster contains one Gateway Node (GN), two Cluster Head Nodes (CH) and the remaining as Ordinary Sensor Nodes (OSN) as shown in the Figure 8.1 above. The OSN forwards the sensory data towards the respective CH and again CH forwards the data gathered towards the GN. Finally the GN forwards the data towards the BS either directly or in a multi-hop fashion. Thus the communication in the WSN system takes place in a hierarchical fashion and more precisely it is an instance of three tier communication architecture as shown in the Figure 8.2 above. All the GNs present in the sensor field along with the BS may form a network (based on link availability) as shown in the Figure 8.3 below.
8.4.1 Secure Hierarchical Routing Protocol for Mobile WSN (SHaRP) Framework

This section gives the details of the proposed SHaRP security Framework. It is reported through various sections as mentioned below. A schematic diagram depicting the SHaRP framework is given in Annexure II.

8.4.1.1 SHaRP Components

Several secret keys and protocol messages constitute the SHaRP framework. This section gives an overview of various secret keys and different protocol messages used in the protocol.

Various Security Keys and Protocol Messages

There are several keys used at various levels in the network. This sub-section provides overview of all types of keys used with SHaRP framework.

There are five major types of secret keys used in the system. Those are Initial Key (K(INI)), Preliminary Key (K(PRE)), Intra-Cluster Key (K(ICK)), Inter-Cluster Key (K(ECK)), and Common Key (K(COM)). These keys either individually or in combined manner, are used for encrypting different messages at various levels depending on the sender-receiver pair in the network system and subsequently also for decrypting various messages. Some keys are generated though one way hash function and some keys are generated using the principles of Elliptic Curve Cryptography (ECC) [124].

Initial Key (K(INI)) & Preliminary Key (K(PRE)): Every sensor node is embedded with an Initial Key before deployment along with its Identification Number, ID. This key is generated prior to deployment and loaded into the memory of the sensor node. Each node i uses this pair < (K(INI)i), (ID)i > for initial authentication with Base Station. This key is valid till the initial registration cum authentication process just after the deployment is over. Each node i applies to the BS for registration cum authentication process encrypting
the message containing the pair \<(K_{INI})_i, (ID)_i\> and this encryption is carried out using the initial key \(K_{INI}\). The Base Station replies to the node \(i\) and sends an authentication message containing the pair \<(ID)_i, (K_{PRE})_i\>. \(K_{PRE}\) is the secret key for the node \(i\) for any further secret communication to be carried out with the Base Station. The Initial Key \(K_{INI}\) is valid only for one round of communication with the Base Station as mentioned above and thereafter the node \(i\) as well as the BS deletes the Initial Key from its memory. Thus the Preliminary Key \(K_{PRE}\) replaces the Initial Key and this new key is again a pair wise key generated by the BS using one way hash function and taking ID of the node \(i\) as well as an input to the function. The details of this key generation process has been given in the section 8.4.1.3.3 below. The Initial Key is deleted mainly to ensure that a failed node can not be reinserted to the network by the attacker. Apart from the Initial Key all other Keys are volatile in the sense that if a node fails due to the battery power decay the node also loses all the Key related information except the Initial Key. The Initial Key of a node \(i\) is lost only when it is automatically deleted by the node \(i\) after the initial registration and authentication process is over. In some situations a failed node may be captured by attacker and after recharging the battery the attacker may reinsert the same node in the network as its own agent. The concept of Preliminary Key will immediately detect such an attack and the Base Station can reject such a node.

**Intra-Cluster Key \((K_{ICK})\):** This kind of key is used for the communication inside a cluster. There are two different types of communication inside each cluster based on the sender-receiver pair. Ordinary Sensor Nodes communicate with the Cluster Head Node and vice-versa. Similarly Cluster Head Node communicates with the Gateway Node and vice-versa. Therefore, there are two different sets of keys that are used inside the same cluster e.g., one set of Key a) for OSN→CH & CH→OSN Communication and another set of Key b) for CH→GN & GN→CH Communication. The Base Station generates an Intra-Cluster Key \((K_{ICK})\) for each cluster \(k\) and distributes this key along with cluster ID to the respective nodes. This key distribution is done by using the Preliminary Key \((K_{PRE})\). Now the key set used for the communication of type a) is \(\{K_{ICK}, K_{COM}\}\); here \(K_{ICK}\) is the Intra-Cluster Key and \(K_{COM}\) is the common key (discussed in the section below). The key set used for the communication of type b) is \(\{K_{ICK}, K_{HGN}\}\) where \(K_{HGN}\) is a pair wise key between the (CH) and GN inside cluster \(k\) and this key is generated through the principles of Elliptic Curve Cryptography (ECC) as mentioned in the section 8.4.1.3.3 below. This key information titled In-Cluster Keys has been summarized in the Table 8.1.

<table>
<thead>
<tr>
<th>Communication Type</th>
<th>Key Set Used</th>
<th>How to Generate</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSN→CH &amp; CH→OSN</td>
<td>({K_{ICK}, K_{COM}})</td>
<td>BS generates and distributes both (K_{ICK}) &amp; (K_{COM})</td>
</tr>
<tr>
<td>CH→GN &amp; GN→CH</td>
<td>({K_{ICK}, K_{HGN}})</td>
<td>GN &amp; CH generates (K_{HGN}) through ECC [40]</td>
</tr>
</tbody>
</table>

Table 8.1: Different Keys used inside a Cluster in SHaRP framework, In-Cluster Keys
Inter-Cluster Key ($K_{ECK}$): This key is used by the GN for external communication in order to forward data towards the BS. Here two types of communication are possible, for example, c) communication between two GNs (multi-hop communication towards the BS) and d) direct communication between a GN and the BS. Inter-Cluster Key is generated by the BS and also distributed by BS encrypting it through the Preliminary Key. For both the communication types c) and d) same Inter-Cluster Key is used along with the Common Key ($K_{COM}$). This key information titled Ex-Keys by GN has been summarized in the Table 8.2.

<table>
<thead>
<tr>
<th>Communication Type</th>
<th>Key Set Used</th>
<th>How to Generate</th>
</tr>
</thead>
<tbody>
<tr>
<td>GN$_i$ → GN$_j$ &amp; GN$_j$ → GN$_i$</td>
<td>${K_{ECK}, K_{COM}}$</td>
<td>BS generates and distributes both $K_{ECK}$ &amp; $K_{COM}$</td>
</tr>
<tr>
<td>GN$_i$ → BS &amp; BS → GN$_i$</td>
<td>${K_{ECK}, K_{COM}}$</td>
<td>do</td>
</tr>
</tbody>
</table>

Table 8.2: Different Keys used for External Communication by GN in SHaRP framework, Ex-Keys by GN

There has been a possibility of using a pair wise key for GN$_i$→GN$_j$ Communication and another pair wise key for GN$_i$→BS & reverse Communication. Looking at the overhead of storing so many pair wise keys in the GN$_i$ this idea has been overlooked though it could have provided higher security level.

Common Key ($K_{COM}$): It is a common key shared by each node for secret communication with the Base Station. As this key is shared by all the authenticated nodes in the network system, this key can always be used for any secret communication between any two authenticated entities irrespective of their roles in the system. This key is generated and distributed by the Base Station. The key is also refreshed and redistributed by the Base Station after a regular interval of time. The key is generated by using one way hash function. The common key is distributed by encrypting it through the Preliminary Key ($K_{PRE}$). Refreshed Common Key (after a regular time interval $t_{key}$ defined by the BS) is distributed by encrypting it through suitable keys as discussed in detail in the section 8.4.1.3.2 below.
The role and lifetime of various keys (Table 8.3) and various keys to be stored at different nodes (Table 8.4) are summarized below.

<table>
<thead>
<tr>
<th>Key</th>
<th>Utility</th>
<th>Lifetime</th>
<th>Refresh Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Key (KINI)</td>
<td>For Initial Authentication along with the (ID)_i</td>
<td>Till the initial authentication is over</td>
<td>Not refreshed; One time use only.</td>
</tr>
<tr>
<td>Preliminary Key (KPRE)</td>
<td>For other key (KICK, KECK, KCOM) distributions</td>
<td>Till other key (KICK, KECK, KCOM) distribution is over</td>
<td>Not refreshed</td>
</tr>
<tr>
<td>Intra-Cluster Key (KICK)</td>
<td>For communication within the Cluster</td>
<td>Duration of the cluster setup, i.e., a cycle</td>
<td>Not refreshed within a cycle</td>
</tr>
<tr>
<td>(KHGN)</td>
<td>For communication between CH↔GN</td>
<td>Duration of the cluster setup, i.e., a cycle</td>
<td>Not refreshed within a cycle</td>
</tr>
<tr>
<td>Inter-Cluster Key (KECK)</td>
<td>For communication outside the Cluster</td>
<td>Duration of the cluster setup, i.e., a cycle</td>
<td>Not refreshed within a cycle</td>
</tr>
<tr>
<td>Common Key (KCOM)</td>
<td>For secret commn with BS; and also as a joint Key</td>
<td>A regular time interval t_key defined by the BS</td>
<td>t_key</td>
</tr>
</tbody>
</table>

Table 8.3: Role, Lifetime and Refresh Period of different Keys in SHaRP framework

Since the sensor nodes are limited in their available memory space, effort has been made to minimize the number of secret keys to be stored at various nodes. Still, based on the role of a sensor node in the sensor field it needs to store few secret keys (2 to 4) and those are summarized in the Table 8.4 below (Node’s Stored Keys).

<table>
<thead>
<tr>
<th>Node</th>
<th>Key to be stored</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSN</td>
<td>Intra-Cluster Key (KICK)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Common Key (KCOM)</td>
<td></td>
</tr>
<tr>
<td>CH</td>
<td>Intra-Cluster Key (KICK)</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Common Key (KCOM) (KHGN)_i</td>
<td></td>
</tr>
<tr>
<td>GN</td>
<td>Inter-Cluster Key (KECK)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Common Key (KCOM) (KHGN)_1 &amp; (KHGN)_2</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.4: Various Keys to be stored at different Nodes, Node’s Stored Keys
It is worthy to mention that in order to enhance security, various keys as mentioned above are stored in different nodes in the encrypted form only.

**Protocol Messages:** In the proposed SHaRP framework four different types of messages are used: Control Message (CM< >), Data Message (DM< >), Routing Message (RM< >) and Exception Message (EM< >). Control Messages are used for establishing the network system. Various tasks of this type include adding node to the network system, choosing Cluster Head Nodes, Gateway Node, cluster organization related messages, authentication, key refreshing etc. For example, after a regular interval of time keys are refreshed and the updated communication regarding key refreshing happens through Control Message. Data Messages are used for data transmission and reception along with data queries. The sensor nodes sense various environmental data and finally these data are to be sent to the Base Station through CH and GN. Routing messages are used for establishing routes from a source to the destination. Generally Base Station distributes routes to the concerned nodes encrypting it with proper keys. A node may request for a route by sending route request message RREQ to the Base Station and the BS replies by sending route reply message RREP. Exception messages are used under exception conditions such as node removal, any circumstance (eg., node moving far away) due to which topology changes, call for cluster reorganization etc. For example some Cluster Head nodes may go far away from their respective subordinate nodes due which links may break down between CH and some subordinate nodes which significantly influence the throughput level at the Base Station. Then BS may detect such event (a kind of network partition) and depending on the overall scenario in the entire network system BS may call for cluster reorganization and such communication happens through exception message. The security keys for encryption of all these four messages depend on the sender-receiver entity. Depending on the role (i.e., BS, GN, CH, OSN) and sometimes identity of the peer, suitable keys are used for encryption.

**8.4.1.2 Various Role Assignments and Routing Protocol Overview**

After random deployment of the sensor nodes in the field the Base Station forms some logical clusters. Each cluster is a group of geographically closely located sensor nodes. Each cluster contains a Gateway Node (GN), two Cluster Head Nodes (CH), and the remaining Ordinary Sensor Nodes (OSN). The Base Station selects these various nodes for the different roles i.e., GN, CH and OSN as mentioned just above. The OSNs are responsible for sensing the environment and then forward the sensed data towards the Base Station through the CH nodes. Each cluster contains two CH nodes. This is because the sensor nodes under consideration are moderately mobile. Due to the mobility there may be link break between the CH and some of the OSNs and in order to address this issue there is a provision to keep two CH nodes inside the same cluster and some OSNs are instructed to send reports to one Cluster Head i.e., CH1 and other OSNs are instructed to report to the other Cluster Head i.e., CH2. These two sets i.e., {CH1, OSN for CH1} and {CH2, OSN for CH2} are selected in such a way that the OSNs do not miss the link with the respective CH and link failure probability due to the movement of the nodes remains at the minimum. The two Cluster Head nodes inside each cluster again forward the reports to the single Gateway Node inside the same cluster. The CH nodes do local
data aggregation before forwarding the data reports to the GN. Finally the GN forwards the aggregated reports to the Base Station either directly or again in multi-hop fashion through some other GNs. The communication pattern for each GN is distributed by the Base Station. This communication pattern for a GN may be direct or multi-hop depending on the current position of the other GN and the BS. This is dynamically decided by the BS and sent to each of the GNs. Since sometimes the multi-hop communications are more energy efficient and at the same time GN→BS is a long distance communication and involves massive data transmission, this decision regarding the route for the data communication between GN and BS is intelligently taken by the BS and then passed on to the concerned GNs.

In this routing protocol, the communication from the OSN to the BS happens in three-tier fashion and those tiers are OSN→CH, CH→GN, GN→BS. Different sensed data are grouped into some reports and finally forwarded to the next level. Therefore, the terms data and report are used interchangeably.

Various algorithms such as cluster formation algorithm, role assignment algorithm or dynamic communication pattern finding algorithm between GN and BS are not discussed here. In this work focus is on various security issues that may surface during the data communication process. In this work algorithms are devised to make the routing process a secure one.

**Role of the Base Station:** The Base Station is a highly resourceful and reliable node located far away from the sensor field. The Base Station plays the role of control centre also, at which sensory data from the field are gathered. In the proposed security framework SHaRP, the major computing load is shifted towards the BS and therefore, the BS plays a very vital role in the entire scheme. For example, Clustering, selection of Gateway Node and Cluster Head Nodes and similarly Ordinary Sensor Node Role Assignment (all inside a cluster) are the major computing tasks carried out by the BS. BS also decides the communication pattern for the Gateway Nodes. Moreover the selection of OSNs for the respective CH nodes is also carried out by the BS. Again various security keys as mentioned in Table 8.1 and Table 8.2 are generated at and distributed from the BS. BS is also responsible for periodically refreshing the common key and distributing it. Under exception situation BS calls for re-clustering even before the current cycle ends. BS does initial authentication of all the nodes just after the deployment and also it does authentication for newly added nodes after initial setup.

**Role of the Gateway Node:** Each cluster contains one Gateway Node which is responsible for forwarding the aggregated data (GN does data aggregation) towards the BS either directly or in multi-hop fashion. The GN inside a cluster receives data from the two Cluster Head nodes. GN is also responsible for generating the secret key (as mentioned in Table 8.1) for secure communication between the CH and itself.

**Role of the Cluster Head Node:** There are two Cluster Head Nodes for each cluster and both are active at the same time. But the set of Ordinary Sensor Nodes which sends report to one of the CH nodes inside each cluster is unique and is determined by the BS. This
arrangement is adapted to maintain connectivity between the OSNs and the CH throughout the cycle (since the nodes are mobile). CH node also takes part in generating secret key for secure communication between GN and itself (as mentioned in Table 8.1). CH nodes do data aggregation after collecting data from the Ordinary Sensor Nodes.

**Utility of Clusters:** The entire sensor field is logically divided into some clusters after the deployment of the sensor nodes in the field. Each cluster is a group of closely located sensor nodes and it covers certain geographic area. The cluster based approach makes the network management tasks simple and efficient and also it gives rise to a scalable solution. At the end, each cluster is responsible for covering a certain geographic area of the environment and sense and forward the sensory data towards the control center i.e., Base Station. The formation clusters instigates the hierarchical communication in the WSN system.

### 8.4.1.3 Functioning of SHaRP Framework

The functioning of the entire SHaRP framework may be described in terms of three major modules- Initial System Establishment module, System Operation module and Key Management module. This section provides overview of these modules.

#### 8.4.1.3.1 Initial System Establishment Module

This module consists of four different phases namely Authentication Phase, Cluster Formation Phase, Role Assignment Phase and Communication Pattern Establishment Phase.

**Authentication Phase:** Each node joining the network system must get authenticated by the Base Station. Each node comes with an embedded key known as Initial Key (K_{INI}) and Identification Number (ID) which are already recorded with the Base Station before deployment (i.e., pre deployment registration). After random deployment of the nodes in the sensor field, each node requests the Base Station for authentication and receives authentication as per the following procedure (Procedure 1).
**Procedure 1**

Step 1: Each node i sends *request for authentication* message to the BS as 

\(< (K_{INI})_i, (ID)_i> \) encrypting through the initial key \((K_{INI})_i\)

Step 2: BS decrypts the message at Step 1 through the initial key, \((K_{INI})_i\), of the sender node i and matches the content with the records of BS

Step 3: *If* match occurs at Step 2

- BS sends the *authentication message* to i as \(<(ID)_i, (K_{PRE})_i> \) encrypting through \((K_{INI})_i\) and \((ID)_i\)
- *Else*
  - reject node i

Step 4: BS and Node i deletes \((K_{INI})_i\) and stores \((K_{PRE})_i\)

Note: \((K_{PRE})_i\) is the secret key for the node i for any further secret communication to be carried out with the Base Station.

BS generates a common key \(K_{COM}\) for all the authenticated nodes and distributes to each authenticated node encrypting it by \((K_{PRE})_i\). Whenever a node i receives the common key \(K_{COM}\) it understands that it is the completion of the authentication process. At the end of the post deployment authentication process each node i is left with two secret keys: \((K_{PRE})_i\) and \(K_{COM}\)

**Cluster Formation Phase:** The Base Station forms the virtual clusters in the sensor field. A cluster is a group of sensor nodes in the field which are geographically closely located. These clusters are valid for a predefined interval of time called cycle. But due to the movement of the nodes as well as sudden death of the nodes the re-clustering process may have to be called before the current cycle expires. And this is decided by the Base Station and necessary actions are taken. In the SHaRP framework it has been assumed that the clusters are formed first and then only various roles such as GN, CH, OSN are assigned to suitable nodes. Here one of the objectives behind clustering is to keep the geographically closely located sensor nodes together under the control of same GN and CH set so the GN and CH nodes are geographically uniformly distributed. The BS forms the clusters (approximately 5% of the total number of nodes deployed as per [22]) and assigns cluster identification number, CID. BS also informs each node about its CID. It is assumed that the BS applies some energy efficient clustering algorithm and computes the cluster [6][58] and focus is on the security aspect of the protocol.
Procedure 2

Step 1: BS computes n clusters in the field and assigns CID to each cluster.

Step 2: BS sends to each sensor node i a message as <i, CID> encrypting it through (K_{PRE})i and (K_{COM})

Note: n is 5% of the total number of sensor nodes deployed in the field.

Role Assignment and Role based Key Acquisition Phase: Each cluster contains three different types of roles e.g., Ordinary Sensor Node (OSN), Cluster Head Node (CH) and Gateway Node (GN). Ordinary Sensor Nodes collect data through their sensing unit and sends to the respective Cluster Head Node and again Cluster Head Node forwards the data to the Gateway Node. And finally the Gateway Node forwards the data towards the Base Station either directly or via some other GNs. Each cluster contains one Gateway Node, two Cluster Head Nodes and rests are Ordinary Sensor Nodes. Inside each cluster each sensor node has exactly one CH node to forward to. The whole set of the OSNs inside each cluster are segregated into two sub sets and each subset is assigned to each of the CH node inside the same cluster. This approach helps in maintaining connectivity between OSN and CH throughout the epoch, since the nodes are mobile. Moreover the work load on the Gateway Node gets reduced due to the responsibilities taken by the Cluster Head nodes. The approach for selecting the suitable nodes for the roles GN, CH and OSN has not been considered in this work rather focus is on the security aspects. Here, it is assumed that the nodes for the roles of GN and two CHs are selected by the BS in such a way that the connectivity (between OSN & CH and CH & GN) inside the cluster remains stable throughout the cycle.
Chapter 8

The Base Station distributes various keys to different roles as per the following procedure (Procedure 3).

Procedure 3

do for each cluster (CID)_i

Step 1: BS identifies GN, two CHs and OSNs

Step 2: BS distributes Inter-Cluster Key (KECK) to GN i as <(GN)_i, KECK> encrypting through (KPRE)_i and (KCOM)

Step 3: BS distributes Intra-Cluster Key (KICK) to two Cluster Head nodes CH1 and CH2 < (CH1), (KICK) > and < (CH2), (KICK) > encrypting through respective (KPRE)_{CH1} / (KPRE)_{CH2} and (KCOM)

Step 4: BS distributes Intra-Cluster Key (KICK) to each OSN i encrypting it by respective (KPRE)_i and (KCOM)

Step 5: (GN,CH1) and (GN, CH2) generates pair wise key (KHGN)1 and (KHGN)2 respectively through ECC [as mentioned in Section 8.4.1.3.3] and stores subsequently

Step 6: Each node i deletes (KPRE)_i

Step 7: End

At the end of the Role Assignment and Role Based Key Acquisition Phase every node based on its role is equipped with necessary security keys i.e., Node’s Stored Keys.

Communication Pattern Establishment: A communication pattern is equivalent to a route. In this phase routes for intra cluster communication and inter cluster communication are established. BS distributes proactively all the routes for all concerned nodes. If a node does not receive the route information the node sends RREQ message to the BS and BS sends the RREP message containing the route information. And this message for OSN is encrypted through the Common Key. Similarly, the messages containing route information for CH and GN are encrypted through different keys as mentioned in the later part of this section. Route information distributed by the BS contains sequence number generated by the BS. The sequence number indicates freshness of the route information. The route computation task is totally shifted to the BS and this approach reduces the massive route computing burden on the sensor nodes and also contributes in energy saving in the resource constrained sensor nodes.
For an Ordinary Sensor Node the next hop to which data are to be forwarded is the respective Cluster Head. Similarly, a CH forwards data to its Gateway Node. And finally each GN forwards data towards the BS either directly or in multi-hop fashion. The BS distributes this route information containing the exact ID of the node to which a particular node shall forward the data as the next hop. Thus, a route under SHaRP framework may look like (OSN→CH→GN$_m$→GN$_n$→---→BS) or (OSN→CH→GN→BS). At the time of computing multi-hop route for GN the BS selects only those intermediate GN which offer better energy efficiency in comparison to direct transmission and also better energy efficiency than other candidate GN available at that point in time. Let us consider three nodes A, B and C as shown in Figure 8.4. A can transmit to C either directly or through B. A will transmit to C through B subjected to the constraint:

$$E_T(k,a) + E_T(k,b) < E_T(k,c)$$  \hspace{1cm} (8.1)

where, $E_T(k,a)$ indicates transmission energy required for transmitting k bits over a distance a. Same logic is also applicable when two or more candidate GN are available for considering those as intermediate hop.

$$E_T(k,a) + E_T(k,b) < E_T(k,c)$$  \hspace{1cm} (8.1)

Once various routes are computed by the BS it distributes those encrypting through different keys as mentioned below:

**Route Message for OSN, i:**

< $i$, CID, (CH)$_i$, seq no> encrypted through $K_{COM}$ & $K_{ICK}$

**Route Message for CH, j:**

< $j$, CID, (GN)$_j$, seq no> encrypted through $K_{COM}$, $K_{ICK}$ & ID (i.e, $j$ in this case)

**Route Message for GN, k:**

< $k$, CID, ((GN)$_m$, (GN)$_n$, ----, (BS)), seq no> encrypted through $K_{COM}$, $K_{ECK}$ & ID (i.e, $k$ in this case)

where CID indicates Cluster ID and seq no indicates sequence number of the Route Message.
8.4.1.3.2 System Operation Module

In this phase, the operation of the network system starts during which main concerns are data transmission from the ordinary sensor nodes to the Base Station and continuous security check in the system.

**Secure Data Routing:** The Ordinary Sensor Nodes sense data from the environment and forward to the respective Cluster Head Node as the communication pattern established during the Initial System Establishment module. The Cluster Head node carries out local data aggregation and forwards the summarized data to the respective Gateway Node. The Gateway Node again does the second level of data aggregation inside the same cluster in order to reduce data duplication. Finally the Gateway Node transmits the summarized data to the Base Station as per the communication pattern established before. The key sets used for data packet encryption are dependent on the roles of the sender-receiver pair. **In-Cluster Keys** and **Ex-Keys by GN** are the various key sets used for intra-cluster (internal) and inter-cluster (external) communication. For local data aggregation at the CH Node and Gateway Node some standard data aggregation algorithm such as CAG algorithm [46] may be integrated with the SHaRP framework.

Though secret keys are used for encryption and decryption process, the SHaRP framework also relies on message code for verifying the integrity of the transmitted data. Each node exploits a rapid and one way hash function to compute a Message Authentication Code (MAC) and then appends the same with the encrypted message. For this purpose Common Key, $K_{COM}$ is used as the secret key.

In order to ensure data freshness, each secure data packet is also appended with a globally unique sequence number assigned by the sender. The sequence number is designed in such a way that it also indicates the node identity. For example, <cluster_id, node_id, i> where i=1,2,-------n, n is an integer is a globally unique sequence number.

Data Packets are made secure by using some secure information inside each packet. Each secure data packet will contain the following security components as shown in Figure 8.5 a) and Figure 8.5 b):

<table>
<thead>
<tr>
<th>Encrypted Message</th>
<th>MAC</th>
<th>Unique Sequence Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 8.5 a) Security Component of a Secure Packet Format (<em>ensuring confidentiality, integrity &amp; freshness</em>)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Encrypted Message</th>
<th>MAC</th>
<th>Unique Seq Number</th>
<th>Unique Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 8.5 b) Security Component of a Secure Packet Format (<em>also ensuring non-repudiation</em>)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Again to ensure non-repudiation each secure packet contains a unique signature generated by each sender node. This unique signature is generated by each sensor node using an one way hash function which takes three input e.g., node ID (nID_i), cluster ID (cID_i) & current geographic location of i (x,y)_i. Thus the hash function appears to be $F_{\text{sig}}(nID_i, cID_i, (x,y)_i) \rightarrow (\text{sig})_i$. This unique signature of i, $(\text{sig})_i$ is valid through a span of time $t_{\text{sig}}$ decided by the BS based on the mobility level (high, moderate, low) of the sensor nodes. This exact signature can not be generated by another node j for the node i since it is impossible for the node j to know the current geographic location of i (as per the assumption). This is an optional part in the SHaRP framework regarding data routing and this packet format (Figure 8.5 b) is used when non-repudiation is also necessary otherwise packet format in (Figure 8.5 a) may suffice.

**Continuous Security Check and Authentication:** The Base Station continuously authenticates the nodes by distributing a refreshed Common Key, $K_{\text{COM}}$. The current $K_{\text{COM}}$ is valid only for a specified period of time $t_{\text{key}}$. After each such period the BS generates a fresh Common Key by applying one way hash function and distributes to each of the authenticated nodes in the system encrypting it by proper key set i.e., In-Cluster Keys and Ex-Keys by GN. Each such key distribution comes with a sequence number (i.e., $< K_{\text{COM}}, \text{seq num} >$) as well in order to ensure that the nodes receives the latest key and also convince themselves about the latest key distributed by the BS. If a node does not encrypt messages with the latest Common Key the messages are ignored in the system and as a whole the network rejects those nodes which do not posses the latest Common Key.

**Various Procedures for Exception Conditions:** When the system is in operation some nodes may go far away from the range of the respective Cluster Head node and this may cause significant degradation in the throughput at the Gateway Node level and in turn at the Base Station. It may be assumed that at the time of selection of the GN and two CH nodes inside each cluster, the BS selects those intelligently so that GN and concerned CH nodes remain connected throughout the epoch with maximum probability. But still there are enough chances remaining that some nodes may die out and this again may cause link break and in turn throughput degradation. If such effect significantly influences the throughput and causes network partition re-clustering may be necessary. By looking at the throughput level, under exceptional condition BS can call for location information from the sensor nodes in the field. Subsequently re-clustering may be initiated by sending exception messages to various nodes. Through this exception message the BS also asks different GN and CH nodes to relinquish their responsibilities. Under normal condition a particular cluster setup continue for an epoch and then re-clustering is called. The following procedure (Procedure 4) describes how to handle exception conditions.
Chapter 8

Procedure 4

At Base Station

Step 1: Observe expected throughput $T_p$ i.e., message per unit time after a regular time interval $t$

Step 2: if $T_p < (threshold)_{throughput}$, call for location information from each node i else wait till next interval and go to Step 1

Step 3: analyze location information and radio range of sensor nodes

if number of broken link between OSN & CH > $(threshold)_{brink}$

and

number of effected cluster > $(threshold)_{effCluster}$
call for re-clustering

else

{ count dead node, ‘$n_d$’

if $n_d > (threshold)_{deadNode}$

wait till next interval and go to Step 1

else

suspect attack & call for re-clustering
}

Step 4: End

When re-clustering is called by then the nodes lose the Initial Key and the Preliminary Key but left with other keys. In next cycle the initial authentication happens through the Node ID, Common Key and Geographic Location of a node at that instant of time.

8.4.1.3.3 Key Management Module

This module describes how the various keys are generated and messages are encrypted and also decrypted in the SHaRP security framework.

Initial Phase: Each node comes with an ID$_i$ and an Initial Key (K$_{INI}$)$_i$ prior to deployment. After post deployment registration the BS generates the Preliminary Key (K$_{PRE}$)$_i$ for each node i as below:

$$(K_{PRE})_i = SH (ID_i, Loc(x,y)_i, (K_{INI})_i)$$

where, SH is an one way hash function such as HMAC or SHA [95], and $Loc(x,y)_i$ is the latest location information of i collected by the BS.
**Chapter 8**

**Secret Key Generation Phase between CH and GN:** The secret key between the Cluster Head Node and the Gateway Node is a pair wise key. This pair wise key is generated on the basis of the principles of Elliptic Curve Cryptography (ECC). Elliptic curve (EC) cryptosystems were first suggested by Miller and Koblitz [95]. ECC is an approach to public key cryptography. A main feature that makes EC attractive is the relatively short operand length relative to RSA and systems based on the discrete logarithm (DL) in finite fields. Cryptosystems which explore the discrete logarithm problem over elliptic curve can be built with an operand length of 150-200 bits. Much research has been conducted on fast algorithms and implementation techniques of elliptic curve arithmetic over various finite fields [95]. The following is a generic procedure on the basis of which two parties can generate public key and respective private keys.

A key exchange between users M and N can be accomplished as follows:

<table>
<thead>
<tr>
<th>Step 1:</th>
</tr>
</thead>
<tbody>
<tr>
<td>M selects an integer ( n_M ) less than ( n ). This is M’s private key, M then generates a public key ( P_M = n_M \times G ); the public key is a point in ( E_q(a,b) ).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2:</th>
</tr>
</thead>
<tbody>
<tr>
<td>N similarly selects a private key ( n_N ) and computes a public key ( P_N ).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3:</th>
</tr>
</thead>
<tbody>
<tr>
<td>M generates the secret key ( K = n_M \times P_N ).</td>
</tr>
<tr>
<td>N generates the secret key ( K = n_N \times P_M ).</td>
</tr>
</tbody>
</table>

A complete description of this procedure is available in [95]. To break this scheme an attacker would need to be able to compute \( k \) given \( G \) and \( kG \) which is assumed to be hard.

(The security of Elliptic Curve Cryptography depends on how difficult it is to determine \( k \) given \( kP \) and \( P \). This is referred to as the elliptic curve logarithmic problem. For detail description about this technique [95] may be referred.)

The key set generated this way and used for secret communication between CH and GN has been termed as \( (K_{CHGN}) \) as a cumulative key set.

Every unique pair of intended nodes i.e, (CH and GN) inside a cluster in the network generates keys according to the principle of Elliptic Curve Cryptography as mentioned above.

**Intra-Cluster & Inter-Cluster Key Generation:** The Base Station generates the Intra-Cluster and Inter-Cluster Keys for various clusters through one way hash functions. And designers can choose some efficient hash functions for this purpose.

**Data Encryption Process:** Most of the data encryption process in the SHaRP framework happens through multiple keys. A general purpose encryption process with two keys has been listed below:

\[
C = E_{N_{key1, key2}}(M)
\]
where, C is the cipher text, EN is the encryption algorithm, (key1, key2) are various keys used in the encryption process and the selection of these keys (i.e., In-Cluster Keys, Ex-Keys by GN) depends on the roles of the sender-receiver pair entities.

**Data Decryption Process:** On receiving an encrypted message the receiver having proper keys can only decrypt the message as below:

\[ M = D_{key1, key2}(C) \]

where, DE is the decryption algorithm. Same keys are used for decryption as used in the encryption process.

### 8.4.2 Security Analysis of SHaRP Framework

An attacker who tries to divulge secure information is said to be successful if and only if it can access correct information from at least \( \partial \) number of secret keys \( (1 \leq \partial \leq 3) \) and hence determine the exact information. This is because most of the encryptions happen through multiple keys and so decryption also requires same number of keys. The security of the proposed scheme relies on the properties of Hash Function and ECC.

**P1:** Let \( H \) be a one-way Hash Function

- It is infeasible to derive the value \( v \) from a given \( H(v) \) [126].
- It is again infeasible to determine values \( v \) and \( u \) such that \( H(v) = H(u) \) [126].

**P2:** It is computationally a hard problem to compute \( k \) given \( G \) and \( kG \) [95].

**P3:** If a cipher text \( C \) is produced through \( n \) encryption processes by \( n \) unique secret keys over a plain text \( M \) using symmetric cryptography it is not possible to decrypt the cipher text \( C \) and regenerate \( M \) without knowledge of exactly \( n \) unique secret keys.

**Conjecture 1:** It is infeasible for attacker \( A \) to compute the Preliminary Key of node \( i \), \( (K_{PRE})_i \), until and unless \( A \) acquires exact Initial Key of \( i \), \( (K_{INI})_i \), along with \( (ID)_i \) and also the Preliminary Key Encryption Algorithm \( E \).

**Conjecture 2:** It is not feasible to compute any other key in the SHaRP framework without the knowledge of at least Preliminary Key \( (K_{PRE})_i \).

The proposed security framework SHaRP is robust against following kinds of attack:

**Impersonation attack**

Let an attacker \( A \) captures \( n-1 \) nodes and tries to reveal information of each node \( < IN_i(1), IN_i(2), \ldots, IN_i(n-1) > \). But still it is not possible to recover the exact information each node \( i \) contains as it is infeasible to compute the value of the initial key of each node \( i \), i.e., \( (K_{PRE})_i \) due to Conjecture 1. Therefore, attacker \( A \) fails to acquire the Common Key \( (K_{COM}) \) and that is why \( A \) fails to prove its queries to be authentic against all other nodes.
and the Base Station \[ \text{since if a node i does not encrypt a message also by the Common Key then i is rejected} \]. And that is why A fails to retrieve the correct answer.

Alternatively the attacker A may try to poison the information and may try to make a legitimate node fool by forcing to accept some spurious information. But it is not feasible since any query response has to be sent in encrypted manner \[ \text{encrypted through the proper key sets} \] maintaining the data integrity intact that too implicitly \[ \text{by including MAC} \]. Therefore, masquerader cannot succeed in any way to break out protocol.

**Node Insertion attack**

Some of the nodes may die out due to the decay of the battery power level or due to some other environmental factors [16]. Let an attacker A gathers this information and tries to reinserst some other nodes of A’s interest against the failed nodes’ ID. These newly inserted nodes will fail to get authenticated by the base station due to unavailability of the proper Initial Key \((K_{\text{INI}})\) \(\text{as the initial key is deleted automatically just after the first round of authentication process}\) and therefore, the newly inserted nodes can not participate in the network function.

Alternatively the attacker A may refill the battery of the failed node i \(\text{though it is not possible in many applications}\) and reinserst the node i in A’s own interest. Still i can not take part in the network functions as i fails to get authenticated again due to the unavailability of the Initial Key \((K_{\text{INI}})\) as well as other secret keys \(\text{since all other secret keys are volatile}\). Moreover newly reinserted node i can not acquire any other keys due to Conjecture 1 and Conjecture 2.

**Denial of Service (DoS) attack**

This is a very common form of attack in a set up like wireless sensor network. Adversaries can disrupt the network services by draining the limited battery power in the sensor nodes. Adversaries can overburden the resource constrained sensor nodes by compelling more and more computing work to do which are actually unecessary and thus nodes lose its energy and finally die. In the proposed scheme the computational requirement at the sensor node level is negligible and therefore, attacker cannot make the nodes busy with computational intensive operations. Thus proposed SHaRP framework is free from this form of DoS attack. A variant of DoS attack is jamming [85], where the attacker \(\text{though the attacker fails to be authentic}\) sends undesired service request that results extra traffic in the network by all the nodes to BS and BS to user. To avoid this sort of attack in the proposed SHaRP framework the messages from unauthenticated nodes are rejected. The nodes which do not possess the current Common Key \((K_{\text{COM}})\) are considered to be unauthenticated. Thus messages not encrypted through the current Common Key \((K_{\text{COM}})\) as one of the secret keys are rejected. Thus SHaRP framework resists DoS attacks. Moreover the BS keeps monitoring and comparing expected throughput versus actual throughput in the system. If the actual throughput degrades significantly it may be due to either network partition \(\text{node death}\) or it may be indirect
result of some form of attack e.g., DoS. The BS then calls for re-clustering and a new setup starts building.

**Node capturing attack**

If an attacker captures a few nodes even then the attacker can not continue participating in the network operation after the *key refreshing interval* $t_{key}$ at the end of which the common key ($K_{COM}$) is refreshed. This is because of the following reason: just before the refreshing of the Common Key the BS collects the geographic location information $(x,y)_i$ of each node $i$ in the field. And then the BS refreshes the current Common Key ($K_{COM}$). The refreshed current Common Key is redistributed. The location information of each node $i$ is used in the encryption/decryption process during the refreshed Common Key distribution phase. It has been assumed that the attacker cannot know the current geographic location of any node as it is computed by itself. Now in one situation the attacker may give its own geographic location information $(x,y)$ but with the ID of some other nodes. Then BS will receive multiple (two or more) location information but with single ID. BS easily detects attack and rejects both the (or all the involved) nodes. In another situation the attacker may not transmit its location information to the BS but waits till the captured nodes receive the refreshed Common Key. Now the attacker cannot reveal the refreshed Common Key since it does not have the location information of $i$ as it was communicated to the BS by $i$ itself.

Thus by knowing the secret of few captured nodes, it is infeasible for an attacker to know the actual information in the network. Apart from the reasons mentioned above this is due to the lack of the correct information from other nodes in the network (*keys are different for different clusters and some pairs of nodes*). Moreover attacker can not know the secret of the BS.

**Replay attack**

The use of unique sequence number in the data packets as well as key distribution packets (*i.e., for the distribution of current Common Key*) insists the nodes to reject the replayed messages. Moreover, the Common Key is valid for a particular duration only ($t_{key}$) which may be a short span of time; any replayed messages encrypted with the past common key are automatically rejected as per the protocol (*once the interval $t_{key}$ is over*).

**Guessing attack**

The attacker may guess the various keys of legitimate nodes and execute the protocol. The attacker is assumed to be successful, if can get the exact information in the network. In the SHaRP framework the attacker has to make correct guess about 2/3/4 keys for different roles such as OSN/CH/GN in order to become successful. This is similar to dictionary attack and it is not possible for large size of the keys.
8.5 Efficiency Analysis

In this section, the author justifies the efficiency of the proposed scheme based on the security strengths and implementation complexity. The author considers a WSN system in which the nodes as well as the Base Station are mobile. Therefore, it is not suitable to compare the proposed scheme directly with already proposed schemes such as mentioned in the section 8.2.2, since those schemes do not consider mobility of nodes. Yet author compares the proposed framework with SPINS [88] and TINYSEC [116] for some compatible parameters as mentioned in the Table 8.5.

8.5.1 Security Properties

The author compares the security properties of the proposed scheme SHaRP against SPINS [88] and TINYSEC [116]. Results are summarized in the Table 8.5.

<table>
<thead>
<tr>
<th></th>
<th>SPINS [88]</th>
<th>TINYSEC [116]</th>
<th>SHaRP [proposed]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>(Session) Key Agreement</td>
<td>Yes(^1)</td>
<td>Yes(^2)</td>
<td>Yes(^3)</td>
</tr>
<tr>
<td>Data Integrity</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Data Freshness</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Access Control</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Non Repudiation</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 8.5 Comparison of different Security Properties

Note:
1. It is based on delayed key disclosure approach and the master key is maintained at the Base Station.
2. Any keying mechanism can be employed; no particular keying mechanism is specified. Suggested keying mechanisms include single network-wide keys, per-link keys between neighboring nodes and group keys.
3. It is based on combination of various key management schemes such as Single Master Key, Pair wise Key and Group Based Key.
Various design parameters of SPINS, TINYSEC and SHaRP are compared and results are summarized in Table 8.6.

<table>
<thead>
<tr>
<th></th>
<th>SPINS</th>
<th>TINYSEC</th>
<th>SHaRP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Management Approach</td>
<td>Master Key</td>
<td>Any</td>
<td>Combination of Master, Pair-wise &amp; Group Based Key</td>
</tr>
<tr>
<td>Time Synchronization</td>
<td>Required</td>
<td>Not Required</td>
<td>Not Required</td>
</tr>
<tr>
<td>Computing overhead at Node</td>
<td>Moderate</td>
<td>Moderate</td>
<td>Very Less</td>
</tr>
<tr>
<td>Computing Load at BS</td>
<td>Moderate</td>
<td>Moderate</td>
<td>Heavy</td>
</tr>
<tr>
<td>Topology</td>
<td>Flat</td>
<td>Flat</td>
<td>Hierarchical</td>
</tr>
<tr>
<td>Scalability</td>
<td>Limited</td>
<td>Limited</td>
<td>High</td>
</tr>
</tbody>
</table>

Table 8.6: Comparison of different Design Parameters

8.5.2 Implementation Analysis

The proposed scheme does not impose much computational overhead at the sensor nodes rather computational tasks are shifted to the Base Station. Different keys are generated at the Base Station and finally distributed to various sensor nodes. The only key set which is generated by sensor node is \( K_{HGN} \) which is a pair wise key between the Cluster Head and Gateway Node. This key set is generated through the principles of ECC [95]. ECC is an approach to public-key cryptography and is based on the algebraic structure of elliptic curves over finite field [95]. ECC based algorithms offer reasonable computational loads and smaller key sizes for equivalent security level as compared to other techniques. Smaller key sizes reduce the size of message buffers and also the implementation cost of protocols. Apart from the communication between the CH and GN, all other communications are based on symmetric key cryptography. Asymmetric cryptographic algorithms consume more energy and therefore, not suitable to implement in the resource starved sensor nodes. Since the proposed security framework mostly relies on symmetric encryption/decryption, it does not involve burden intensive computations at the sensor nodes. If the application demands non-repudiation (as it is optional), each node needs to generate a unique signature executing one hash function only. Except this the ordinary sensor nodes do not need to compute any key.
8.5.3 Defense against Network Layer DoS attack Analysis

The network layer of wireless sensor network may suffer from several potential threats. There has been an exhaustive survey of these threats reported in [85]. In reality there are a large and increasing number of threats to the security of WSN. They can be broadly classified as attacks against confidentiality of the network data, denial of service (DoS) attack, impersonation or replication attack and physical attacks. DoS attack may take many forms and they are known to be any attack that can undermine a network’s capacity to perform its expected functions. For example, Neglect & Greed, Homing, Spoofing, Black Holes, Flooding etc. are instances of DoS attack [127][128]. The proposed security framework SHaRP can detect the DoS attack by a simple technique, i.e., the BS keeps on monitoring the throughput level in terms of successful packet delivery at the BS. If this parameter drops significantly and below a threshold level set by the user, the BS suspects either DoS attack or network partition (which again may be a result of DoS attack). Then the BS triggers re-clustering through exception messages and therefore, it can stop further deterioration in the network system.

SPINS consists of two secure building blocks namely Secure Network Encryption Protocol (SNEP) and µTESLA. SNEP is used to provide confidentiality through encryption and authentication. It also provides integrity and freshness. µTESLA is used to provide authentication for broadcasted data. µTESLA requires that the Base Station and the sensor nodes be loosely time synchronized. Schemes like µTESLA is based on delayed key disclosure and it can suffer from denial of service attacks [129]. Similarly, TINYSEC [116] also provides security services like access control and message integrity through authentication and confidentiality through encryption. Semantic security is achieved through the use of a unique initialization vector for each invocation of the encryption algorithm. But TINYSEC may also suffer from denial of service attack as reported in [129]. Here SHaRP framework has an edge of having capability to detect the probable DoS attack quickly and take preventive measure for it. SPINS, TINYSEC and SHaRP are compared for their strength regarding network layer DoS attacks and result is summarized in Table 8.7.

<table>
<thead>
<tr>
<th></th>
<th>SPINS</th>
<th>TINYSEC</th>
<th>SHaRP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defense against DoS Attack</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 8.7: Comparison regarding DoS Attack

8.5.4 Spreading of Attack

The entire network is organized into some logical clusters after the deployment of the sensor nodes. Each cluster has unique set of keys for different types of communication such as intra-cluster or inter-cluster. Therefore, even if an adversary can get hold of some
keys from some nodes, the attack can never spread beyond the boundary of the cluster to which the nodes belong to, with that much of disclosed secret information. Therefore, any attack in the SHaRP framework is always localized and confined to a relatively small zone in the sensor field.

### 8.5.5 Traffic Analysis (message complexity)

The major source of energy consumption in the sensor node is the communication [22]. Again transmission of messages by the transceiver unit of the sensor nodes consumes more energy than the reception of messages by the same. Therefore, essentially the number of transmissions should be minimized at the sensor node level in order achieve an energy efficient scheme. In this section, author reports a traffic analysis of the SHaRP framework regarding the traffic related to the overall management of the routing security in the WSN system. Since the sensor nodes are resource constrained the author analyzes the number of transmission and subsequent reception in the sensor node level. The Base Station is a resourceful entity, therefore, traffic sent and arrived at the Base Station level may be overlooked for the time being.

**a. During Initial Registration and Authentication:** Each sensor node in the field transmits one request to the Base Station for initial registration and authentication. In reply each sensor node receives one message with the respective preliminary key from the Base Station. Table 8.8 depicts the message complexity of SHaRP framework during the initial registration and authentication process.

<table>
<thead>
<tr>
<th>Node, N</th>
<th>Transmission, T(N)</th>
<th>Reception, R(N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each Sensor Node</td>
<td>(1)</td>
<td>(1)</td>
</tr>
</tbody>
</table>

Table 8.8: Message Complexity (Initial)

**b. During Key Distribution:** Based on the role assigned to different sensor nodes each node receives different secret keys from the Base Station. In this phase the sensor nodes do reception only as the Base Station performs a proactive function of distributing the secret keys. Ordinary Sensor Nodes (OSN) and each Cluster Head node (CH) receive the Intra Cluster Key and Common Key from the Base Station; each Gateway Node (GN) receives Inter Cluster Key, Common Key and Intra Cluster Key from the Base Station. Again each CH node and the respective GN do one round of communication (based on ECC) with each other regarding the generation of the secret key for communication between them (i.e., $K_{CHG}$).

Table 8.9 depicts the message complexity of SHaRP framework regarding communication with the BS by each role during the key distribution process. Again Table 8.10 summarizes the message complexity regarding the communication between CH and GN during the key distribution process.
### Table 8.9: Communication with the BS (per Role)

<table>
<thead>
<tr>
<th>Node, N</th>
<th>Transmission, T(N)</th>
<th>Reception, R(N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordinary Sensor Node</td>
<td>(nil)</td>
<td>(2)</td>
</tr>
<tr>
<td>Cluster Head</td>
<td>(nil)</td>
<td>(2)</td>
</tr>
<tr>
<td>Gateway Node</td>
<td>(nil)</td>
<td>(3)</td>
</tr>
</tbody>
</table>

### Table 8.10: Communication between CH and GN (per Role)

<table>
<thead>
<tr>
<th>Node, N</th>
<th>Transmission, T(N)</th>
<th>Reception, R(N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Head</td>
<td>(1)</td>
<td>(1)</td>
</tr>
<tr>
<td>Gateway Node</td>
<td>(2)</td>
<td>(2)</td>
</tr>
</tbody>
</table>

### c. During Cluster Formation and Role Distribution:

This is one of the major parts of the entire secure routing process. And numbers of rounds of communication between the sensor nodes and the Base Station happen as mentioned below. For creating the logical clusters BS collects location information from each node and then distributes cluster information (i.e., **Clustering**) to each node. The BS proactively distributes the role (i.e., **Role Distribution**) and communication pattern (i.e., **Commn. Pattern Distribution**) to each sensor node. But, due to the loss of messages in the network, if a node does not receive the communication pattern it sends a RREQ message to the BS; and BS then sends the RREP message again. Under **Exception Condition** the BS collects location information from each sensor node. Again the BS does **Continuous Authentication** by transmitting the refreshed Common Key to each node in the field irrespective of the roles. Table 8.11 summarizes all these different communications and associated message complexities:

### Table 8.11: Communication with the BS (per Role)

<table>
<thead>
<tr>
<th>Activity</th>
<th>by Node, N_i</th>
<th>T(N)</th>
<th>R(N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clustering</td>
<td>each node</td>
<td>(1)</td>
<td>(1)</td>
</tr>
<tr>
<td>Role Distributions</td>
<td>each node</td>
<td>(nil)</td>
<td>(1)</td>
</tr>
<tr>
<td>Commn. Pattern Distribution</td>
<td>each node</td>
<td>(nil)-(1)</td>
<td>(1)</td>
</tr>
<tr>
<td>Exception Condition</td>
<td>each node</td>
<td>(1)</td>
<td>(1)</td>
</tr>
<tr>
<td>Continuous Authentication</td>
<td>each node</td>
<td>(nil)</td>
<td>(1)</td>
</tr>
</tbody>
</table>

Table 8.11: Communication with the BS (per Role)
8.6 Conclusion

In this chapter, author presents SHaRP, a novel hierarchical and role based secure routing protocol for mobile wireless sensor networks. The proposed protocol design is motivated by the complex nature of a wireless sensor network in which both sensor nodes as well as the Base Station are mobile. Since the sensor nodes are resource constrained some communication and processing intensive functions are shifted to the Base Station. The storage requirements in the sensor nodes are minimized. Author presents a detail analysis on how the proposed protocol can defend different routing layer attacks. The security strengths of the SHaRP framework are compared with that of SPINS and TinySec. Author introduces a novel cluster organization and overall key management scheme which is suitable for a large wireless sensor network system. As a future scope of this work, implementation of the proposed security framework in real physical sensor network can be planned.