KEY MANAGEMENT IN STATIC AND MOBILE SENSOR NETWORKS

By

ANVESH REDDY AILENI

Bachelor of Technology in Computer Science and Engineering

Jawaharlal Nehru Technological University

Hyderabad, India

2009

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2011

KEY MANAGEMENT IN STATIC AND MOBILE

SENSOR NETWORK

Thesis Approved:

Dr. Subhash Kak

Thesis Adviser

Dr. Johnson Thomas

Dr. David Cline

Dr. Mark E. Payton

Dean of the Graduate College

ACKNOWLEDGMENTS

It gives me immense pleasure in acknowledging all those people without whom, this thesis would have been impossible. First and foremost I would like to offer my deepest gratitude to my Thesis advisor Dr. Subhash Kak, who has supported me throughout my thesis with his patience and knowledge while encouraging me to work in my own way. It is his encouragement and effort without which this thesis would not have been completed. I could not wish for a better advisor.

I also like to acknowledge my gratitude and thanks to Dr. Priyank Jaiswal who has been a support to me throughout my study. I thank him for giving me an opportunity of setting up and working on a Linux lab which helped me understand all the administrative concepts. Above all his unflinching encouragement and support in various ways has helped me in achieving my goal, not just in completion of thesis but also in every major step I took. I would like to thank Dr. Eliot Attekwana, who supported my study at OSU and encouraged me to work towards my goal for which I would be ever grateful to him.

I would also like to thank Dr. Johnson Thomas, whose thought provoking question have made to consider all those points which I have missed and helped me to a conclusive research. I also like to thank Dr. David Cline for his encouragement and valuable comments which helped my thesis end up in a good shape.

Most important of all, I would like to thank my parents, Raghotham Reddy Aileni, Jyotsna Aileni and grandparents Mahendhar Reddy, Late Savitri for their unprecedented encouragement throughout my life. I would like to dedicate this Thesis to them for believing in me and teaching me to be a better human being. My brother Agnivesh Aileni has helped me building positive attitude towards everything I do. I also thank my Uncle Ajay Tadisina, Aunty Renuka and Cousins Sahiti, Linitha, Suraj, Varun, Rahul, Sandesh and extended family in giving their support and encouragement throughout my life.

Finally, I would like to thank my friends, Srujana Machiraju, Praveen Chandrahas, Medhamsh Vutpula, Tejaswi Battula, Singi Reddy Rohith Reddy, Yashwanth Raju, Venkat Ravinder, Pradeep Dantala, Vijay Singh, Hima Bindu, Shashank Sadalia, Parmeshwar Reddy, Chakradhar Reddy, Nitesh Reddy, Siddharth Echampati, Hari Kishan Kotha, Sagar Kodukula, Rohith Vaidya, Ansih Koppula, Sanath Chilakala, Shiva Busireddy, Raviteja Gunda, Praveen Kuruvada, Bipul Chandra, Sujith Bheemireddy for always being with me and giving support.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

A sensor network consists of spatially distributed sensors with limited computational and communication capabilities. These are used in varied applications such as monitoring environmental conditions, military and industrial purposes. Various kinds of sensor nodes are used in the network. They can be broadly classified based on their size, communication range and power capability. The sensors have limited computational capability and hardware complexity. One such example of a sensor is the SmartDust sensors [17] which has specifications shown in Table 1, giving us an idea of its limited resources. In this thesis, sensors and nodes are terms which are used interchangeably.

CPU	8b, 4MHz
Storage	8Kbytes program, 512B Memory, 512B EEPROM
Communication	916 MHz Radio
Bandwidth	10 Kbps
Operating System	Tiny OS

Table 1: Typical configuration of a sensor node

Each sensor runs on battery power and is equipped with certain kind of radio communication device which can communicate only in small range. There are sensors which use dual radio communication device, one for short range and the other for long range. These kind of dual radio communication devices are manufactured by Sensoria Corporation [14]. The software which is

used in the sensors must consume little computational power and thus the operating system used should be less complex than conventional operating systems. Examples of operating systems which are used are Tiny OS [15] and Lite OS [16]. Algorithms used for computational purpose should consume less energy and therefore use of certain cryptographic methods is precluded.

Typical sensor networks have a base station or a controller to monitor the activities in the network. The networks may be static, dynamic or both i.e., the network consisting of both static and mobile sensors. In this thesis we have dealt with two kinds of networks, one is static sensor network and other is mobile sensor network. If the network deals with static sensors, we call it Wireless Sensor Network (WSN) and if it deals with mobile sensors, we call it Mobile Sensor Network (MSN). WSN's are used particularly in areas which are very remote and direct human intervention is least possible. One such example is the area along the international borders where instead of human, if sensors are deployed; the situation could be monitored easily. Also, WSN's are used to monitor environmental conditions provoked by natural disasters such as volcanic eruptions, landslides, Tsunamis. Apart from these, WSN's can also be used for industrial and medical purposes and also to monitor traffic situations. Few areas where MSN's can be used widely include army at war. Sensors are deployed on moving objects such as soldiers and army vehicles, which helps them communicate information about the situation around them. Also, MSN can be used to monitor cattle, emergency vehicular information and many other purposes.

One of issues with sensors when distributed spatially should communicate with each other in a secured manner since the data associated with these sensors might be confidential. Efficient cryptography techniques are to be used for communication among them. Use of public key cryptography is precluded since it had been reported by Carman, Kruus, and Matt [18] that when compared to the energy consumption of 1024 bit AES and 1024 bit RSA, the latter takes higher computations than the earlier. Also, using master key for the entire network is infinitely scalable but might be a compromise on the security of the network. Thus, efficient key management

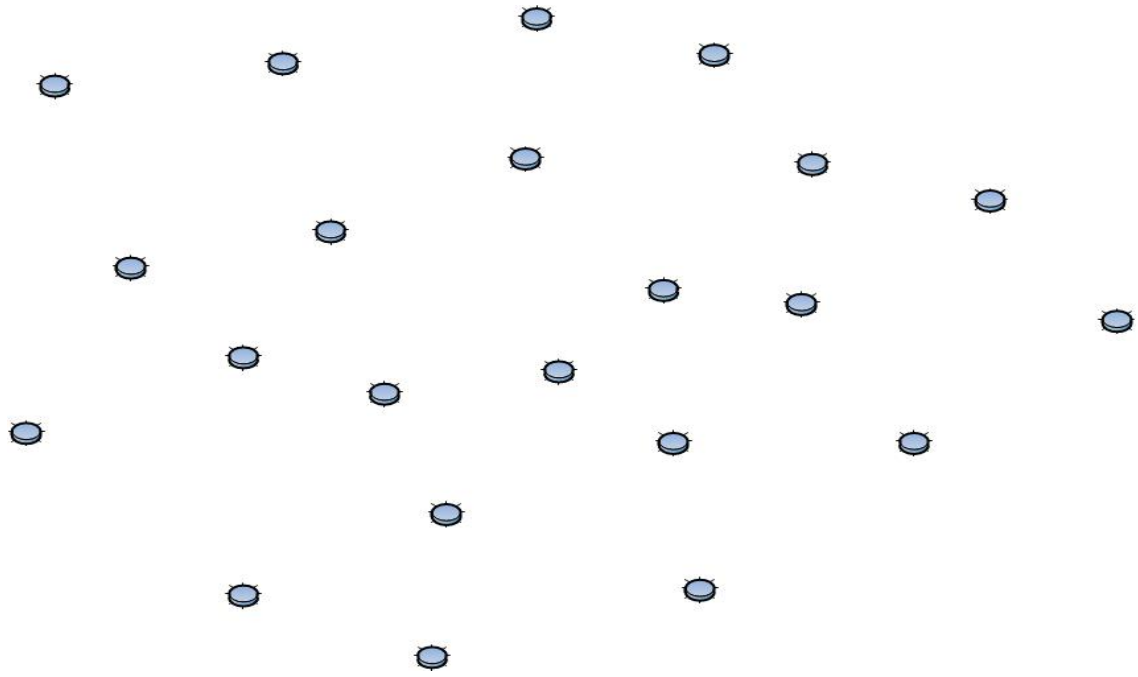

Figure 1 Randomly distributed Sensors

among sensors is of prime importance while designing a network. Key management among both MSN and WSN's would be considered here in this thesis.

One solution to the above discussed problem is to use unique key for each pair of nodes. This is not just robust but also attains maximum resilient during node captures and adversaries attack. But, due to the storage limitations of the sensors, all the unique keys cannot be stored. Consider a total of *m* nodes of which two nodes share a unique key. This will result in a total of $\left(\frac{m(m-1)}{2}\right)$ keys in the network with each node storing *(m-1)* keys. This is very high considering the storage limitations of a node if *m* is large. Also considering the ad-hoc nature of the sensor networks, nodes add to the network over time. Thus, efficient key management has to be performed addressing solutions to all the problems that occur in designing the network.

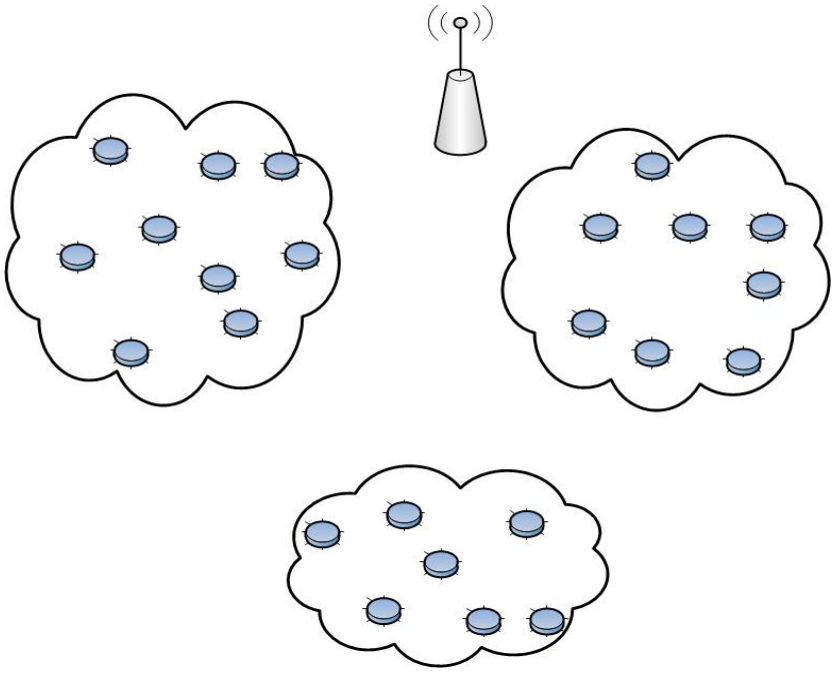

Figure 2 A typical networks with clusters divided and a base station

One of the issues these sensors face is broadcasting of messages. They cannot broadcast longer ranges, so to send messages to sensors which are at longer distance than the range, it should send via multi hops and these hops should be finite. Thus, while designing the network, prime criteria which should be met is that the number of hops required for a message to be delivered at destination should be less, relatively.

Rest of the document is divided into chapters with introduction being chapter 1. Chapter 2 deals with literature review followed by methodology in chapter 3, findings in chapter 4 and conclusion in chapter 5.

CHAPTER II

LITERATURE REVIEW

There has been wide range of research going on in key management of wireless and mobile sensor networks. Symmetric key pre distribution and pair wise key distribution are the approaches used earlier but this scheme uses Birthday Paradox [8],[9],[10] in identifying the number of keys that are to be stored in each sensor. In addition, for clustering of the network, *K-means++* algorithm [7] has been employed. Also, Dijkstra's shortest path algorithm [11],[12] is used which would be discussed in detail along with the usage in chapter 3.

CLUSTERING

K-means++ is an improved version of *k*-means algorithm which reduces the complexity in finding the *k* means [32]. This algorithm can be used to identify *k* means out of *n* nodes in the network and assigning rest of the nodes to the means thus forming clusters. In *k-means++* clustering algorithm, the number of clusters is fixed *a priori*. We assume the number of clusters chosen to be *k* and this choice is based on the network size and geographical position of each node.

According to *k-means++* clustering algorithm, *k* means are identified initially, one for each cluster. For this, following steps are to be followed,

1. Choose an initial centre *x*, from the set of all node points say μ, uniformly at random.
2. Choose next mean from the set and name it *y* with probability $\frac{D(y)^2}{\sum_{x\in\mu} D(x)^2}$ proportional to $D(x)^2$
3. Repeat step 2 until all the *k* centers are identified.

Binding is done in order to form clusters by assigning nodes to the means. Hence, new mean positions are calculated and binding is done. This process is repeated until there is a change in the position of the means.

BIRTHDAY PROBLEM

In the proposed scheme, as mentioned earlier, birthday problem is used in distributing keys among nodes in the network [8],[9],[10]. Birthday paradox or birthday problem is a probabilistic approach, which selects pairs of people from a set of randomly chosen people who share same birthday. This is also a problem for identifying number of people who share same birthday with certain probability. Considering an year with 365 days, for the probability to be 1, the number of people to be considered are 366.

Let the probability with which we have to find number of people *n* in a room to be *p(n)* and number of days to be *d*. So, according to the birthday problem

$$p(n) = 1 - \frac{n!\binom{d}{n}}{d^n}$$

The equation can be approximated to

$$p(n) = 1 - \left(1 - \frac{1}{d}\right)^{C(n,2)}$$

RELATED WORKS

Eschenauer and Gligor [2] proposed a “key ring” scheme using random graphs wherein, from a large pool of keys *S*, few keys were selected and stored in the node such that two neighbors share at least a key with certain probability *p*. Each node carries a subset of keys called the key ring of size *m* from *S* and finds common set of keys among subset of any two randomly chosen key rings. This scheme requires three main steps namely, key pre-distribution, key discovery and path key identification. In the key discovery phase every node sends a message to all other nodes. The neighbors upon receiving the message determine which keys are shared with which node. Also a path is established among nodes if they do not share a key and this is done in path key identification. This populates the entire network and this process may have to be repeated every time a node is added or a node is replaced.

When the “key ring” approach by Eschenauer and Gligor [2] requires at least one node to be shared among the neighbors, the *q*-composite scheme proposed by Chen et. al. [19] requires *q* keys of neighbors to be shared which increases the resilience in the network. This scheme reduces the size of the key pool but increases the key ring size at each node by repeating the keys to be stored. Also, the network has more chance of being compromised when the number of nodes captured is more than 1% of the total network size but it is considered advantageous when it is less than 1%. Zhou et. al. [20] based on symmetric polynomials, proposed a key management scheme with *t* – degree *(K+1)*-variate symmetric polynomial with *K* credentials selected for each sensor node. Using these credentials, polynomials share is computed and then both the credentials and polynomials shares are stored in the nodes. Two sensors can produce same key only when the credentials differ in one dimension.

There are also schemes which assume the deployment knowledge so as to improve key management performance in WSN’s [20,21]. A proposed key management protocol by Du et.al.

[21] which is based on [2] uses Guassian probability distribution function (pdf). In this scheme, the network area is divided into square cells with each cell being associated to one group of the sensors. There is division of key pool into sub key pools and there are as many sub key pools as the squared cells. Also, sub key pools of each cell have certain keys related to those of their neighboring cells. Each node randomly selects *m* nodes from the sub key pool and stores, thus improving the protocol performance when the network size is huge. But there are still few problems associated with this protocol, main being static nodes. Another scheme which is similar to [21] is Zhou et. al. [23]. They proposed a protocol called LAKE which is based on *t*-degree tri-variant symmetric polynomial. Each node has two credentials *(n1, n2)* where *n1* is cell identity and *n2* is node identity. Based on these credentials, nodes calculate polynomial share and if nodes have one mismatch in their credentials, they can generate a common key. So, all the nodes in a cell can establish a connection among themselves but there are only two specific nodes from different cells sharing a key directly with each other.

Du et. al. [24] proposed a pair wise distribution scheme which is based on blom's key distribution scheme. In blom's scheme [28], a $(\lambda + 1) \times N$ matrix G is defined which is public, where N being the size of the network. Also, it generates several symmetric D matrices of $(\lambda + 1) \times (\lambda + 1)$ which are private. The scheme is λ resilient; it means that the network still remains secure even if λ nodes are compromised. But if more than $\lambda+1$ nodes are compromised, the remaining nodes cannot communicate and entire network is compromised. The blom's scheme is modified by Du et. al. [24] in order to increase the resilience in the network but the scheme even then requires $O(\lambda)$ memory to generate matrix and secret numbers which will be used in generating keys among neighboring nodes. In this scheme, large λ increases the resilience at the expense of large memory. Moreover, there is network over head in computing the keys which ultimately degrades the overall performance of the network. Another scheme has been proposed by Yu et al. [29] which also uses blom's scheme for the key management. They have divided the network area into

hexagonal cells and associated D and G matrices are stored in sensors based on the prior deployment knowledge. Cells are of two types, one is base cell and other normal cell. Normal cells are neighbor with two base cells but the base cells are not neighbors. A confidential matrix B_i is allocated to each base cell and also to its six neighbor cells so as to produce required information to the sensors. Though the connectivity of the scheme is close to one, the memory used by sensors is very high.

Location based protocol proposed by Kihyun et al. [26] uses three phases in establishing efficient routing in mobile sensor networks. Although this scheme does not mention about key management, it has an approach in establishing routing protocol among mobile sensors. In first phase node sends advertisement messages to all the neighbor nodes and then receives reply from them in second phase if only when they want to send the reply. In third phase, data is forwarded upon receiving the reply message from sensors. This way communication is made and messages are exchanged among nodes. This uses global positioning system through which each sensor knows about the location of itself and also the location of the sink node. Kifayat et al. [27] proposed group based key management for mobile sensor networks in which they have both static and mobile sensor nodes and the network is divided into groups. Key ring k, from pool of key S are loaded into static and mobile sensors in pre-deployment phase in a similar fashion as employed in [2]. With these keys they can communicate with sensors in an out of each group and moreover each group has a group leader which receives data in encrypted format from nodes in the group and sends to sink node. Mobile nodes roam in the network and establish connection with at least one sensor in each group with a probability P within its communication range. This scheme does not talk about connectivity of mobile sensor if they are not in the communication range of sensors in any group and also has one major drawback as when group leader compromise entire group of sensor nodes get compromised.

Dongg et al. [25] proposed group based key pre-distribution scheme. This scheme has two parts, one is group based EG in which key pool size at any instance of time is 500 and each sensor node randomly selects 50 keys each from in-group and cross-group instances. Whereas the second one, group based PB scheme at any instance of time include 49 degree bivariate polynomial. Polynomial share from in-group and cross-group is assigned to each sensor node. The scheme which comes close to the proposed scheme on WSN is a cluster based group key management scheme proposed by Lin et al. [30]. This scheme divides the entire network in clusters and each cluster has a cluster head and sensors and a base station for the network. This scheme use LEACH [31] protocol in electing new cluster heads. Cluster heads are ordinary sensor nodes which could communicate to base stations and all the cluster heads in single hops. This seems to be practically impossible as sensors with such limited communication range cannot interact in single hop.

As mentioned, the proposed scheme deals with two kinds of networks, one being Wireless Sensor Networks (WSN) and other Mobile Sensor Networks (MSN). Both the networks cover most of the problems mentioned above and also deal with different kind of key management approach which has not been used earlier. In addition, both the networks will be divided into clusters based on standard clustering algorithm whose usage will be discussed in detail. The main criteria are to reduce the total number of keys to be stored at each sensor and to minimize the over head at each sensor. The proposed approach requires that data be routed over many nodes before it reaches its destination. This indicates that there is an underlying connection between routing and security and that implicit security technique [3],[4],[5] may be used to route data via multiple channels, thus spreading the vulnerability over several nodes and communication channels making an adversary's task harder. There is also the question of node compromise which may be countered by using tamper-resistance hardware in each node [6]. Further, overlay architecture has been used to achieve balance between number of keys per node and routing complexity [13].

CHAPTER III

METHODOLOGY

As mentioned earlier, we would be dealing with two kinds of networks here; one is Static Wireless Sensor Network (WSN) and other is Mobile Sensor Network (MSN). In both, the network will be divided into clusters and would use similar kind of key management technique.

STATIC WIRELESS SENSOR NETWORKS

In this network, none of the sensors move. All the nodes are deployed randomly and the network will be divided into clusters, not necessarily of same size. Each divided cluster will have a sub controller in addition to a main controller which belongs to all the clusters. The next phase after clustering is to load the keys into the memory of the nodes and these keys stored in the memory is called a key set. Keys are also to be distributed among the sub controllers for their communication, as all the sub controllers may not be in the communication range. After key distribution is done, the deployment phase begins and the communication among the nodes is established. Figure 3 gives an idea of how the network is divided into clusters.

In a cluster, say of *n* nodes, there are $\left(\frac{n(n-1)}{2}\right)$pairs of keys. Each sub controller has to find a way of populating the key set of each sensor node. Also, main controller will do the work similar to that of the sub controller in distributing the keys among the sub controllers. Once the network is divided into clusters, it is assumed that each node interacts with its peers within the cluster and

there is no direct interaction with nodes outside that cluster.

Other assumptions are, each sub controller is within the communication range of its own cluster and main controller can communicate with all the sub controllers. Also, sub controllers have more communication and computation capabilities than other nodes in the network; likewise the main controller has more computational capabilities than sub controllers and nodes. Moreover sub controllers in the network may not communicate with main controller in single hop. For that reason it may require multiple hops for message to be broadcasted to the main controller from sub controllers.

CLUSTERING OF WSN

Using *K-means++* algorithm, clusters and *k* means are identified. Once this is done, each node in the network is associated to one of the other means and with this association entire network with divided clusters are formed. The main goal is to minimize the sum of the square distances within each cluster and is given by (1)

$$\sum_{i=1}^{k} \sum_{x_j \in S_i} ||x_j - m_j||^2 \longrightarrow (1)$$

where $m_1, m_2, m_3 \ldots .\ m_k$ are the means of the clusters initially defined.

$$S_i^t = \{ x_j : || x_j - m_i^t || \leq || x_j - m_{i^*}^t || \text{ for all } i^* = 1, 2, \ldots k\}$$

xj is the relative distance of *(x,y)* in the network from origin.

mi is the relative distance of mean *(x,y)* from origin.

Updating the means:

$$m_i^{t+1} = \frac{1}{|S_i^t|} \sum_{x_j \in S_i^t} x_j$$

t – Ranges from 1, 2… till the means does not change

j – 1, 2 ….. n.

i – 1, 2 ….. n.

Thus, the k means locations will be identified and these will remain as locations for controllers of those clusters.

KEY DISTRIBUTION

During key distribution phase, we can draw an analogy of birthday problem to relate to the present scenario as follows,

1. Total number of nodes in a cluster can be associated to number of days.
2. Number of people in a room can be associated with number of nodes sharing keys with each node in the cluster.

By this analogy, we can identify the number of keys to be stored in the sensors. In WSN, as the network is divided into clusters and is based on the assumption that sensors in the clusters cannot directly communicate with sensors in other clusters; two types of key managements are required.

1. Key Distribution among nodes.

2. Key Distribution among sub controllers.

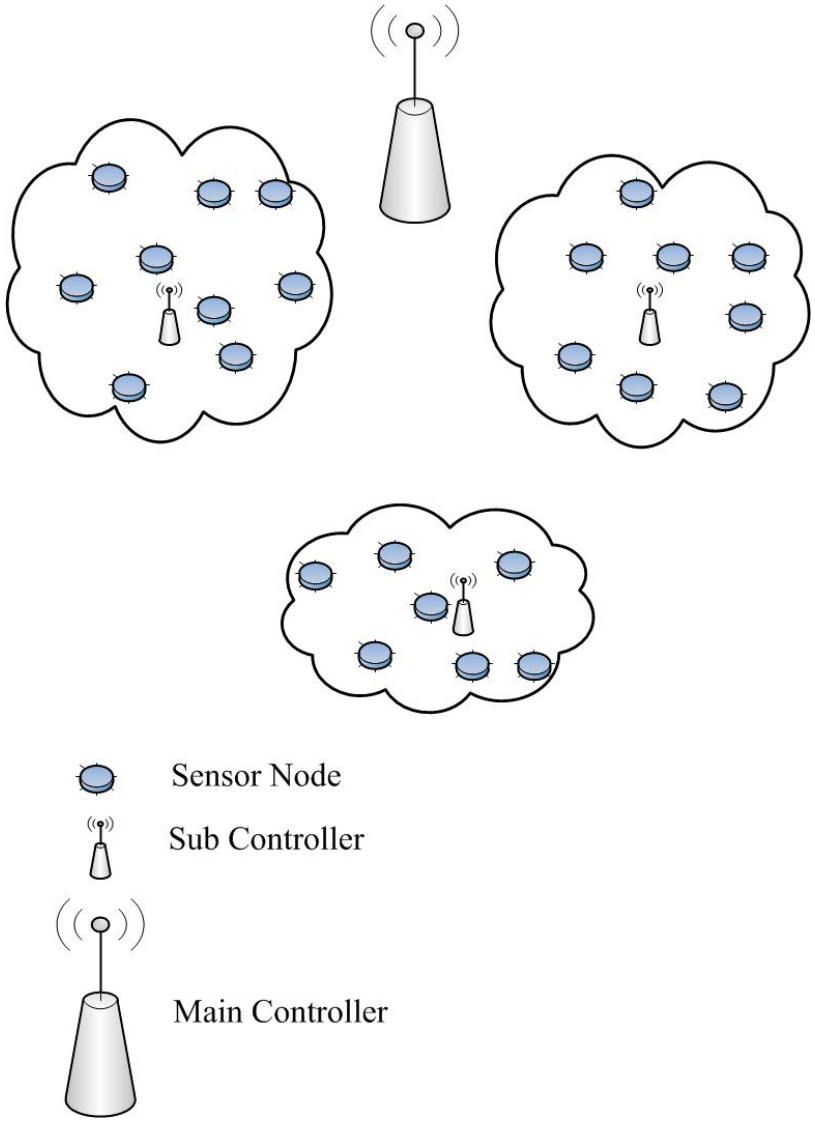

Figure 3 Network divided into clusters with sub controllers and a main controller

KEY DISTRIBUTION AMONG NODES

In the network, assume total number of nodes to be *m* and is divided into *k* clusters. Choose a cluster randomly with *n* nodes and probability *p(s)*, which is given by

$$p(s) = 1 - \left(1 - \frac{1}{n}\right)^{C(s,2)}$$

Here, *s* is the number of nodes with which the node shares a key. So the key size of each node would be at least *s*. As the value of *p* changes, *s* value changes and this gives us the total number of nodes where two nodes share a common unique key. Consider a random node *i* in the cluster and randomly choose *s* nodes which share a unique key with node *i* in the cluster. Out of $\left(\frac{n(n-1)}{2}\right)$ total keys, those nodes with which node *i* shares a key are stored in the memory. This process is repeated till all the nodes of the clusters stores keys.

KEY DISTRIBUTION AMONG SUB CONTROLLERS

Not every sub controller can interact with all other sub controllers due to its limited communication and storage limitations. So, a key distribution should be present among the sub controllers to have an effective communication. To accomplish this, all those sub controllers which are within the communication range are to be identified first. Then, the same method as that of the nodes can be employed here.

Let *l* be the number of clusters identified within a communication range of distance *d*. From the birthday problem, we identify the number of sub controllers sharing keys among themselves. Say a sub controller *si* has to share keys with *ss* sub controllers within its range out of the total *l* number of sub controllers, the probability *p(ss)* can be determined as,

$$p(ss) = 1 - \left(1 - \frac{1}{l}\right)^{C(ss,2)}$$

Here we get the value of *ss* i.e., total number of sub controllers with which sub controller *si*, shares a key. Those *ss* sub controllers are chosen randomly from *l* which is in the range of that sub controller under consideration.

PATH IDENTIFICATION

After key distribution phase, path has to be established among different nodes which may either belong to same cluster or to a different cluster. There are two ways in which this can be done.

1. Path among sub controllers
2. Path among nodes.

PATH AMONG SUB CONTROLLERS

When there is a inter cluster communication, there has to be a path that exists between sub controllers. For this purpose Dijkstra's shortest path algorithm [11],[12] is used. All the weights are assumed to be equal to1.

According to the shortest path algorithm the following steps are to be followed in order to identify the path from one sub controller to other.

1. Initially assign a sub controller and identify all those sub controllers which can interact with it.
2. Assign all the sub controllers status as not visited.
3. When all the nodes neighboring the current nodes are visited, mark the current node visited.

If by the 3rd step all the nodes are visited at least once, search end there, if not, mark the next node as the current node and repeat the process from step 2.

Thus, by this procedure all the paths between different sub controllers are identified and stored in their respective memory.

PATH AMONG NODES

Each path has to be identified from a source node to a destination node. Instead of populating the network at the beginning, the path is established between two nodes when required. Hence, when a message has to be transferred from one node to another for the first time, a path will be identified using the key sets and this path is stored for the next time use. This way, path is identified between different nodes and stored in sub controller. The two cases that are to be dealt are:

1. The source node and the destination node belonging to the same cluster.
2. The source node and the destination node belonging to the different clusters.

CASE 1

Assume a path to be identified between the nodes *i* and *j* which are in the same cluster. Node *i* sends message, if *i* and *j* share a key, the message is broadcasted directly, otherwise, *i* checks for node *n*, where

$$\{n \mid \text{if } n \in \{\text{key set of } i\} \text{ and } n \text{ where } dkj < dxj, x = \{\text{key set of } i\}\}$$

dnj defines a node *n* which is the closest node to *j*.

When the message is broadcasted to *n*, the next nearest node of *j* is searched and this process is repeated till there is a path directly from *n* to *j*.

As an example, assume a path to be identified from node *c* to *r*. From the procedure described above, we get *n*'s to be nodes *o, m, p, u, t, e,* and thus the path from *c* to *p* is *c-o-m-p-u-t-e-r*. This path identified will be stored and it will be used for the next time when a communication is required from *c* to *p.*

CASE 2

When a node has to communicate with a node in another cluster, sub controller communication is also needed along with node communication. Thus, when a node in one cluster has to communicate with a node in another cluster, that node sends a message to its sub controller. This sub controller based on other sub controller's information sends a message to the destination cluster's sub controller through the path which has already been defined. Thus, after reaching the sub controller of destination cluster, it will send a message to the first node of its cluster and then the process as discussed in case 1 is repeated. The path thus established here again will be stored.

MOBILE SENSOR NETWORK

This network has been designed considering an army situation at war. We considered sensors having two kinds of radio communications, one being lone range and other being short range. In this network, all the nodes are mobile and they move in groups most of the time. But few sensors may move out of the group either to join other groups or to form a new group. Hence we divide the network into clusters and elect a cluster head based on the *K-means++* algorithm. Alongside, there is a base station to monitor the activity of the sensors in the network.

We calculate the mean position and the sensor at that position is the cluster head. Cluster heads switch on their long range radio in order to communicate with other cluster heads. In doing so, they will add extra load on the battery and this load has to be distributed among all the sensors in their respective clusters. This is possible since all the sensors are move and hence there would be change in the mean position of the clusters. Also, the nodes are added and deleted from the clusters frequently. Therefore, cluster heads change frequently and thus distributing the load.

CLUSTERING OF MSN:

Clustering is done using *K-means++* algorithm as used earlier for WSN. Initially, positions of *K-means* are identified and these positions are assigned to the cluster heads. Since there is frequent addition and deletion and also change in the relative positions of the nodes, cluster head elections are done periodically by the base station. Thus, instead of load on individual node, the load is distributed on almost all the nodes by the change of cluster heads.

In identifying the mean position, the main goal is to minimize the sum of square of distance within each cluster given by equation (1)

$$\sum_{i=1}^{k} \sum_{x_j \in S_i} ||x_j - m_j||^2 \longrightarrow (1)$$

Updating the means:

$$m_i^{t+1} = \frac{1}{|S_i^t|} \sum_{x_j \in S_i^t} x_j$$

t – Ranges from 1, 2… till the means does not change

j – 1, 2 ….. n.

i – 1, 2 ….. n.

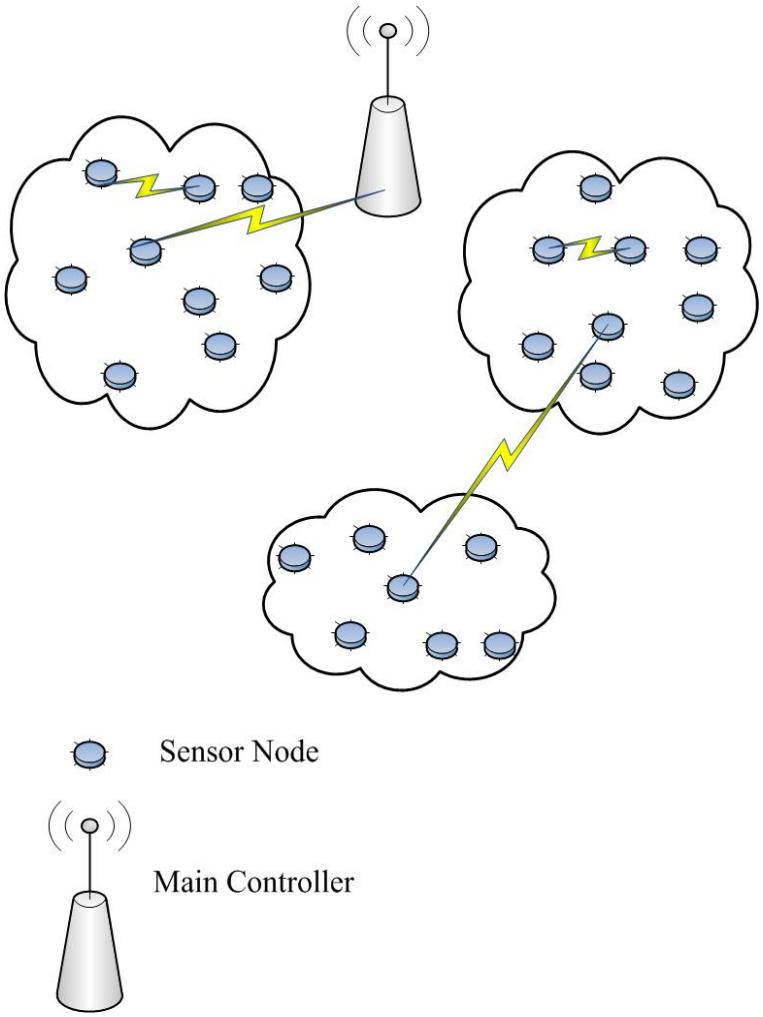

Figure 4 MSN showing long range and short range radio communication

KEY DISTRIBUTION

As the nodes move continuously, the key distribution would be dynamic and therefore the keys are not pre distributed except for the key that is shared with the base station. Using this common key, nodes decrypt the encrypted key set received from the base station. Similar to what we have discussed earlier in WSN, there will be two kinds of communications; one is among the nodes within a cluster and the other among the nodes belonging to different clusters.

Henceforth, there will be two kinds of key distributions,

1. Key distribution among the nodes within the cluster
2. Key distribution among the cluster heads.

KEY DISTRIBUTION AMONG NODES

Keys are distributed among nodes within a cluster using birthday problem. We can similarly draw an analogy with slight modifications and can be related as follows.

1. The number of nodes in a cluster at a moment of time with total number of days.
2. The number of people in a room is associated to the number of keys shared.

If we consider the number of nodes in a cluster to be *n* and number of keys with which each node in a cluster shares keys to be *s*, then probability *p(s)* can be calculated as

$$p(s) = 1 - \left(1 - \frac{1}{n}\right)^{C(s,2)}$$

At different values of *p(s)* you get different values of *s* and by these we say that each node at a particular moment of time store *s* keys. Consider a node *j*, it shares keys with *s* other nodes randomly chosen among *n* nodes in the cluster. Each time the node is added and deleted from the cluster, the new key set is given to the sensors.

KEY DISTRIBUTION AMONG CLUSTER HEADS

Base station will randomly assign ID's to all the clusters. We can similarly draw an analogy using these ID's as follows,

1. The number of clusters at a moment of time with the total number of days.
2. The number of people in a room is associated with the number of keys shared.

Let *l* be the number of clusters identified within the communication range of distance *d*. Each cluster head *ch* shares keys with *sh* number of other cluster heads, then

$$p(ch) = 1 - \left(1 - \frac{1}{l}\right)^{C(sh,2)}$$

From the value of *sh*, we identify the number of cluster heads with which *ch* shares keys. So each time a sensor is elected as a cluster head, it stores additional keys in order to communicate with other cluster heads. When the cluster head change, the base station will send the packet of information containing encrypted key set to it.

PATH IDENTIFICATION

The path identification among sensors in MSN is bit different from that of WSN. Each time the path might not be same from the source to the destination as the nodes are mobile. There are two kinds of paths which have to be defined.

1. Path among nodes within a cluster.
2. Path among nodes of different clusters.

PATH AMONG NODES WITHIN A CLUSTER

Path among nodes *i* and *j* within a cluster is identified using following procedure.

{n | if n ∈{key set of i} and n where dnj < dxj, x = {key set of i}}

dnj defines a node *n* closest possible node to *j*.

Assume a path has to be identified from node *i* to *j* not sharing a common key. Using the procedure described, nearest node to *j i.e., n,* sharing common key with *i,* is added to the path from *i* to *j*. This process is repeated till a node is found which shares a common key with *j*.

PATH AMONG NODES OF DIFFERENT CLUSTERS

If both the nodes belong to different clusters, the head nodes of the clusters will also have to involve in the communication along with the normal nodes within the cluster. Thus the source node initially has to send message to the head node of that cluster. Head node gets the cluster ID to which the destination node belongs to from the base station based on its ID. Further, it sends the message to that cluster head following the same procedure as discussed earlier in the path among nodes within a cluster. For the head nodes to communicate with each other, they switch on their long range radio communication. Upon receiving the message the head node of that cluster sends message to the destination node.

ADDITION/DELETION OF SENSOR NODES

When the nodes are added, the information is first updated to the base station and then based on its location it joins in any of the clusters. As per the algorithm, if the new node is unable to join in any of the clusters, that node will form a new cluster with zero other nodes in it. This node will switch on its long range radio communication for communicating with other nodes. As these nodes are mobile, this node at any moment of time will join any one of the clusters and then it turns off its long range radio in turn reducing the over head on the battery.

When the nodes, either lose their battery power or is compromised or is completely dead by any reason, we say that the node has been deleted. The deleted node's information is updated to all the nodes within the cluster to which it belonged. Then new set of keys are generated and the normal operation are resumed. If the deleted node is the head node, then the new head node is selected using clustering algorithm and this information is sent to all other nodes by the base station.

NODE MOVEMENT FROM ONE CLUSTER TO ANOTHER

These nodes will always in movement and thus might have a chance of moving away from the clusters. Therefore, the message will be notified to the base station about its present status. When the node joins any of the other clusters, it receives new set of keys from the base station. If it does not join in any cluster, then as discussed earlier in the addition section above, it would form an another cluster with zero other nodes in it.

CHAPTER IV

FINDINGS

RESULTS AND ANALYSIS OF WSN

Simulations are done to investigate the results using network of 4000 nodes which is divided into 4, 8 and 12 clusters in each case.

Number of hops on an average in 4 clusters is presented in graph of Figure 5. In a cluster of 1000 nodes with 499500 keys, we store key set of size which varies from 15 to 204, 15 when the probability is 0.1 and 204 when the probability is 0.999999999. The number of hops on an average is 28.7 at probability 0.1 and it is 2 at probability 0.999999999.

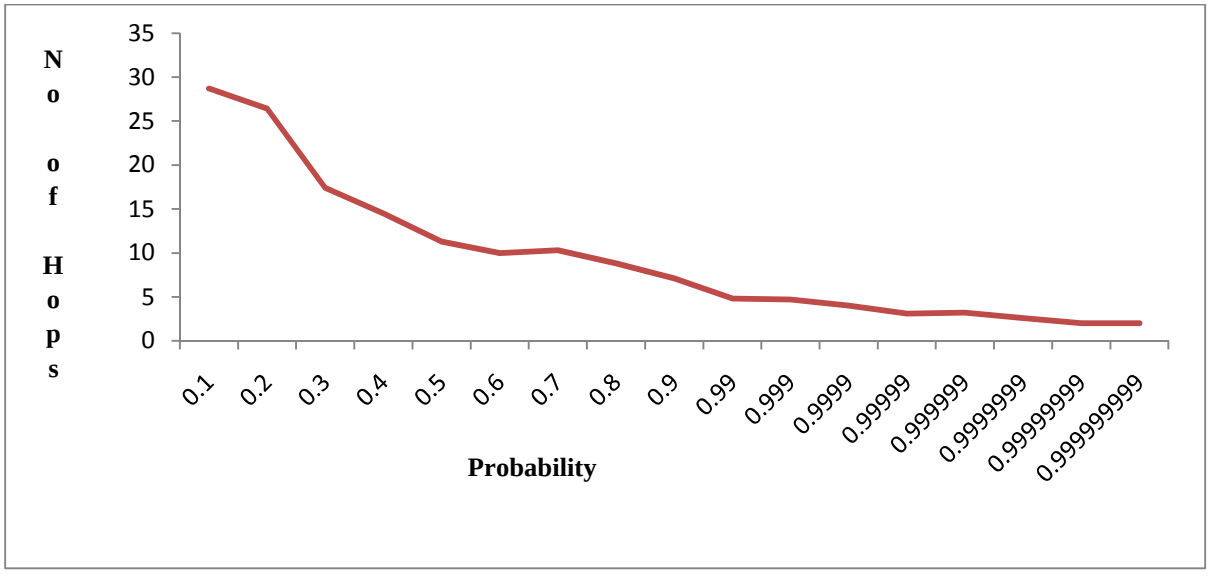

Figure 5. No of hops on an avg vs probability for 4 clusters

Figure 6 presents the case of 8 clusters for a network of size 4000 nodes. In a cluster of 500 nodes with 124750 keys, we take key set of size which varies from 10 to 144, 10 when the probability is 0.1 and 144 when the probability is 0.999999999. The number of hops on an average is 19.3 at probability 0.1 and it is 1.9 at probability 0.999999999.

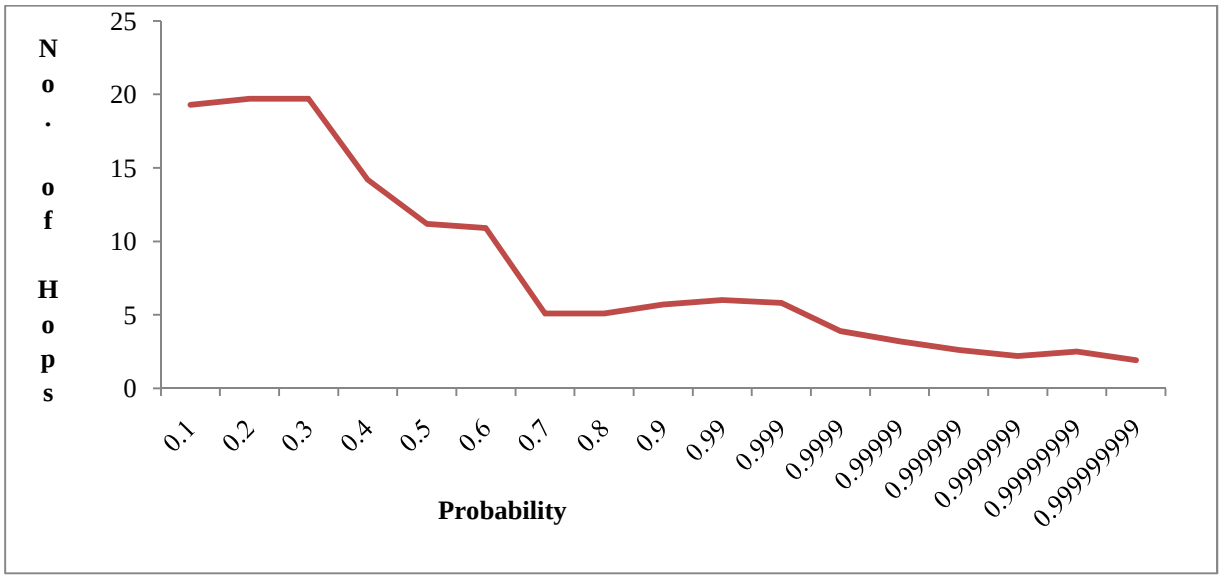

Figure 6. No of hops on an avg vs probability for 8 clusters

COMPARISON OF HOPS AND KEY SET SIZE FOR CLUSTERS

Figure 7 represents a graph which shows the key set size being reduced when the number of clusters is increased at different probabilities even when the network size is same. At probability 0.5, the key set size is 37, 35 and 21 when the network is divided into clusters of 4, 8 and 12 respectively.

Similar observations are done for the number of hops on an average for 4, 8 and 12 clusters. Figure 8 shows at different probabilities that the number of hops reduce when the number of clusters increase even when the network size remains the same. At probability 0.5, the number of hops is 11.3, 11.2 and 7.8 for the network of 4, 8 and 12 clusters respectively.

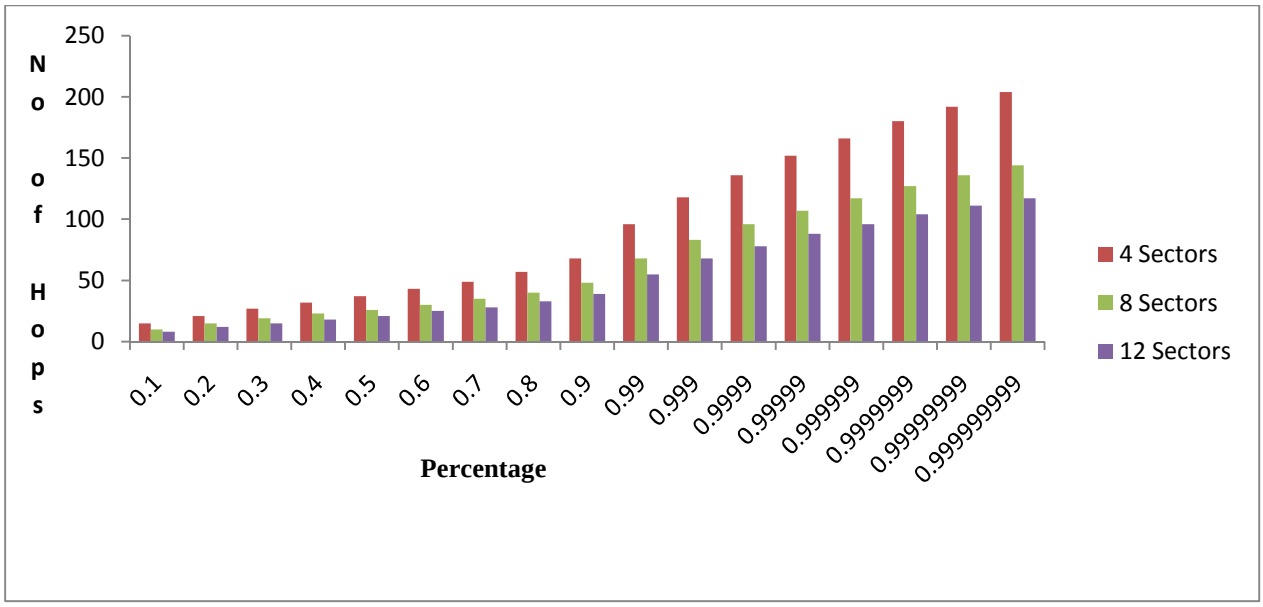

Figure 7. Number of Hops versus Percentage

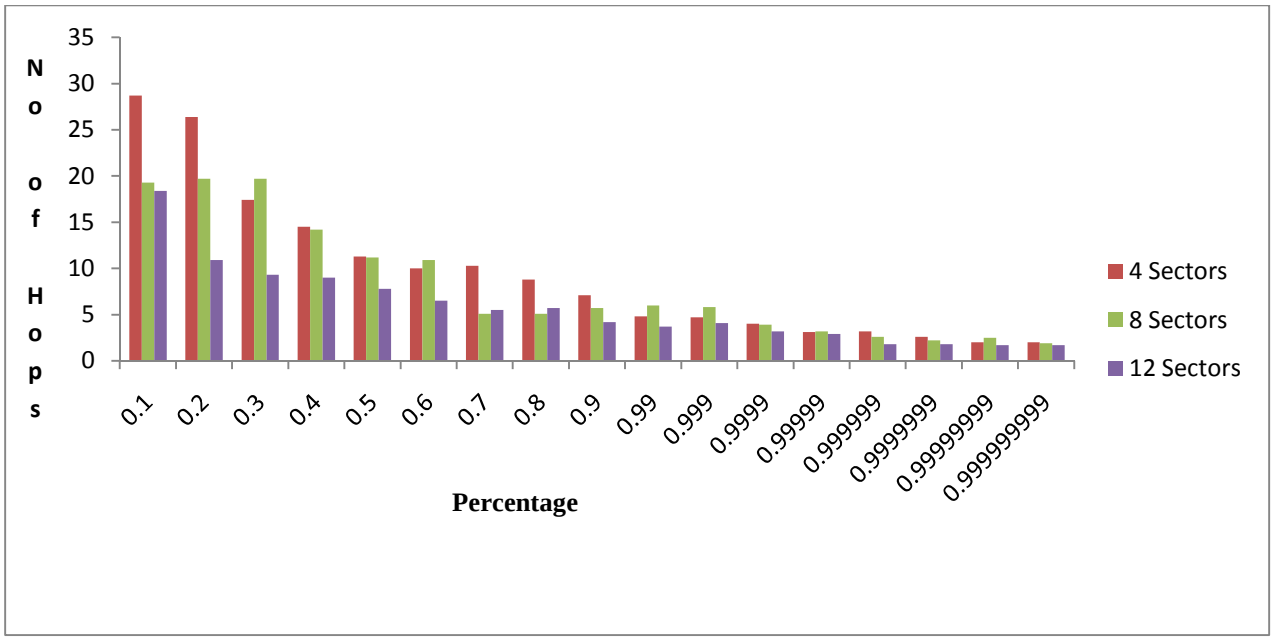

Figure 8. Number of Hops versus Percentage

RESULTS AND ANALYSIS OF MSN

Simulations were carried out for a network of 12000 nodes with a maximum of 300 and minimum of 200 nodes in each cluster. At different percentages, each simulation was run for 30 times. According to birthday problem, the number of keys each sensor store at 20 percent is 8 and 37 at 90 percent assuming 300 to be the node size in the cluster. Further, if a sensor is the cluster head it has to store an additional of 3 to 10 keys to be able to communicate with other cluster heads. Figure 9 represents graph showing the number of hops at different percentages. It is observed clearly that upon increase in the number of keys stored, the number of hops decrease.

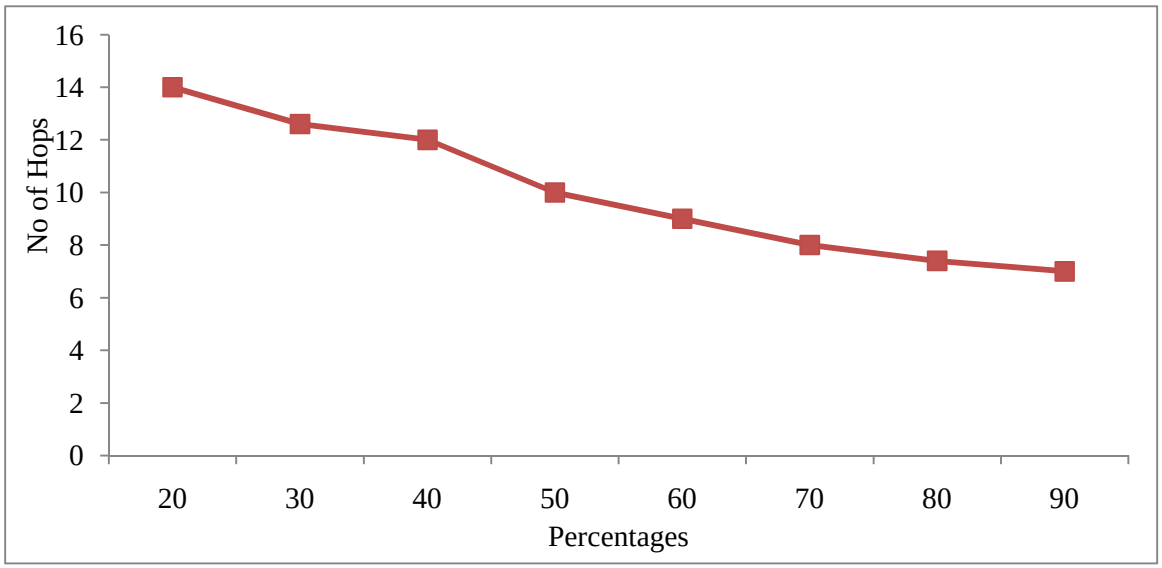

Figure 9. No of Hops vs Percentage

Simulations were also carried on to check the resilient nature of the network. This was done by removing few sensors from the cluster and then carrying on the simulation. Figure 10 is a graph representing the number of hops to the percentage of nodes deleted. It is observed comparing figures 9 and 10 that the number of hops overall has increased. But there is fall in the number of hops upon increase in the number of deleted nodes. This is because of decrease in total number of sensors.

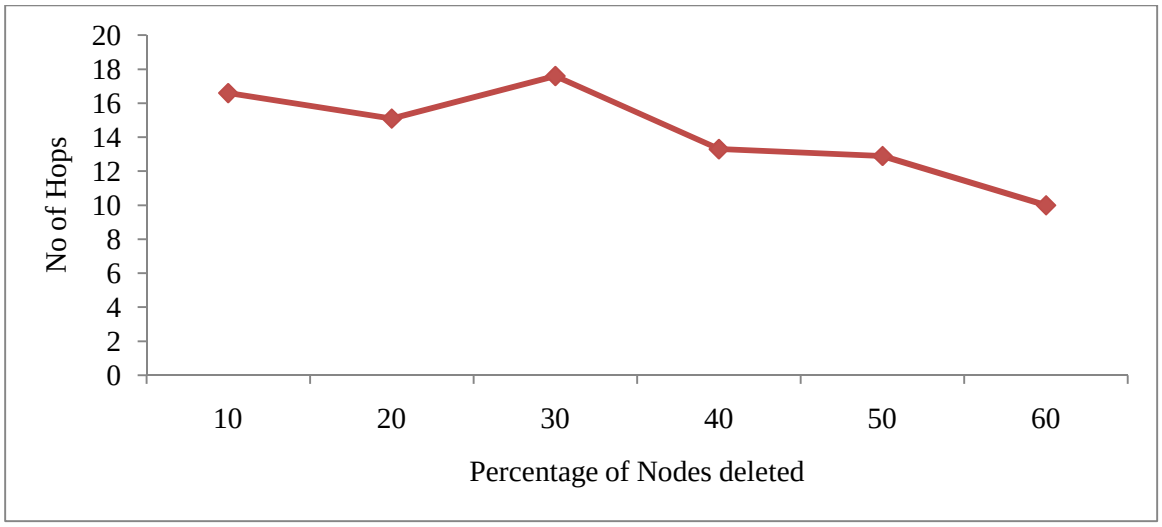

Figure 10. No of Hops vs Percentage of nodes deleted

Figure 11 shows us the probability of link failures at different percentages of deleted nodes. If 20 percent of the overall nodes in a cluster are deleted, 2 percent of nodes are unreachable. Similarly when 70 percent of nodes are deleted, 5.6 percent of nodes are unreachable. If more than 70% of the nodes in the network are deleted, the network is assumed to be compromised since there is steep increase in the link failure.

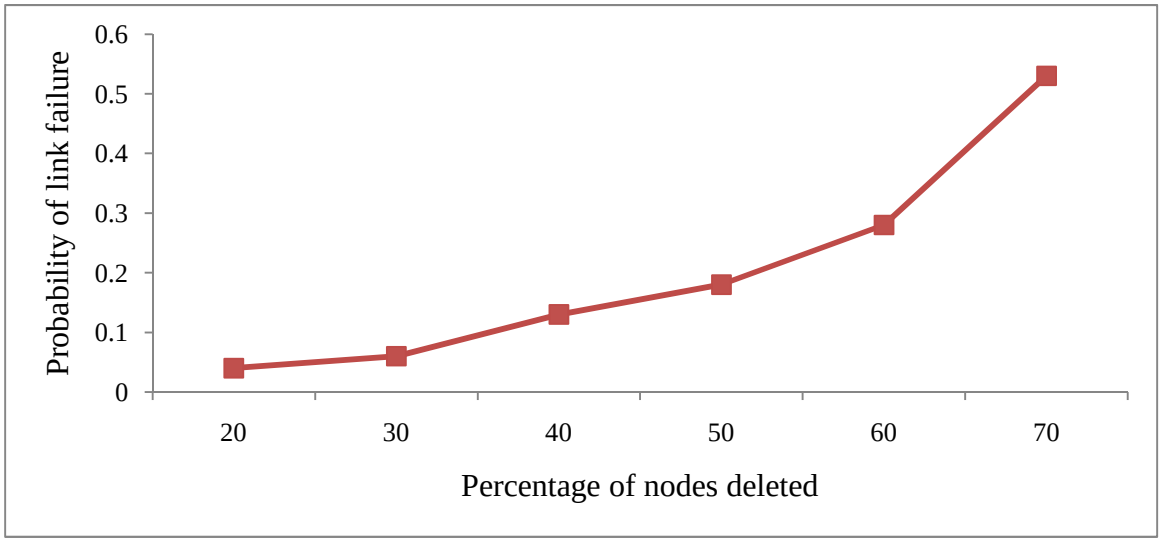

Figure 11 Probability of link failure vs Percentage of nodes deleted

Most of the time sensors operate only with their short range radio and therefore overall power dissipation should not be very high. But when the sensor is a cluster head, it operates on its long range radio thus making it to consume more power and if a particular sensor continues its operation on its long range radio, there is a chance that it loses its battery power soon. Hence, there should be a change in the role of the cluster head very often in order to reduce over head on individual sensors. Since sensors in the network are mobile they tend to change their relative positions very often and so is the change of the role of the cluster head.

CHAPTER V

CONCLUSION

This thesis presents key management scheme for static and mobile sensor networks. The scheme presented for static sensor network is an effective protocol for dividing the network into clusters and for distributing keys among them. This method is efficient when the nodes in the network are divided randomly and can be clustered easily as compared to the case where the nodes were distributed in a uniform fashion. Simulations were carried on for the proposed design and the results are presented in graphs. These results show that the performance in terms of number of hops and number of keys stored in a node improves as the number of clusters increases.

The scheme for mobile sensor network presents an effective ways of distributing keys among sensors which are mobile. The network is divided into clusters based on their physical location and maximum distance between any two points in the cluster is the communication range of a sensor. By dividing the network into clusters, we are actually limiting the key pool size to have keys of only nodes in the cluster. Usage of a sensor node which has two types of radio communications would help the sensors to communicate with sensors at longer distance via cluster heads cluster heads. Simulations are performed which show the improved overall performance of the system.

REFERENCES

[1] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler. Spins: security protocols for sensor networks. *Wireless Networks Journal (WINE)*, No. 8: pp. 521–534, 2002.

[2] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *CCS '02: Proceedings of the 9th ACM conference on Computer and Communications Security*, pages 41–47, New York, 2002.

[3] A. Parakh and S. Kak, Online data storage using implicit security. *Information Sciences, vol.* 179, pp. 3323-3331, 2009.

[4] A. Parakh and S. Kak, Internet voting protocol based on improved implicit security. *Cryptologia*, vol. 34, pp. 258-268, 2010.

[5] A. Parakh and S. Kak, Space efficient secret sharing for implicit data security. *Information Sciences*, vol. 181, Issue 2, pp. 335-341, 2011.

[6] S. Kak, On secret hardware, public-key cryptography. *Computers and Digital Technique (Proc. IEE - Part E)*, vol. 133, pp. 94-96, 1986.

[7] D. Arthur and S. Vassilvitskii, k-means++: the advantages of careful seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.* pp. 1027–1035, 2007.

[8] E. H. McKinney, Generalized birthday problem. *American Mathematical Monthly* 73, pp. 385–387,1966.

[9] M. Klamkin and D. Newman, Extensions of the birthday surprise. *Journal of Combinatorial Theory 3*, pp. 279–282, 1967.

[10] J.B. MacQueen, Some methods for classification and analysis of multivariate observations". *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press.* pp. 281–297, 1967.

[11] D. MacKay, Information Theory, Inference and Learning Algorithms. *Cambridge University Press.* pp. 284–292, 2003.

[12] E.W. Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik 1:* pp. 269–271, 1959.

[13] A. Parakh and S. Kak, Efficient Key Management in Sensor Networks, In *Preceedings IEEE Globecom:* pp. 1584-1589, Miami, 2010.

[14] Sensoria Corporation. www.sensoria.com

[15] Tiny OS. www.tinyos.net

[16] Little OS. www.liteos.net

[17] K.S.J. Pister, J.M. Kahn and B.E. Boser, Smart Dust: Wireless networks of millimeter-scale sensor nodes (1999).

[18] D. W. Carman, P. S. Kruus and B. J. Matt,"Constraints and approaches for distributed sensor network security," September 1, 2000. *NAI Labs Technical Report* pp 00-10.

[19] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy*, page 197, Washington, DC, USA 2003. IEEE Computer Society.

[20] Yun Zhou and Yuguang Fang. A Scalable Key Agreement Scheme for Large Scale Networks. In *Proceeding of the IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pages 631-636, Lauderdale, Florida, USA, April 2006. IEEE.

[21] Wenliang Du, Jing Deng, Yunghsiang S. Han, Shigang Chen, and Pramod Varshney. A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge. In *Proceeding of the 23th IEEE Conference on Computer Communications (INFOCOM)*, pages 586-597, Hong Kong, March 2004. IEEE.

[22] Mahalingam Ramkumar and Nasir D. Memon. An Efficient Random Key Pre-Distribution Scheme. In *Proceeding of IEEE Global Telecommunication Conference (GLOBECOM)*, pages 2218-2223. IEEE, December 2004.

[23] Yun Zhou and Yuguang Fang. A Two-Layer Key Establishment Scheme for Wireless Sensor Networks. *IEEE Transaction Mobile Computing*, 6(9):1009-1020, September 2007.

[24] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili. A pairwise key predistribution scheme for wireless sensor networks. *ACM Trans. Inf. Syst. Secur.*, 8(2):228–258, 2005.

[25] D. Liu, P. Ning, W. Du, "Group-Based Key Pre-Distribution in Wireless Sensor Networks," In *Proceedings of 2005 ACM Workshop on Wireless Security (WiSe 2005)*, pp. 11-20 September 2005.

[26] Kihyun Kim, Jeongbae Yun, Jangkyu Yun, Byeongjik Lee and Kijun Han, "A Location Based Routing Protocol in Mobile Sensor Networks," *11th International Conference on Advanced Communication Technology (ICACT)*, pp. 15-18 Feb. 2009, Phoenix Park, Korea.

[27] Kifayat K, Merabti M, Shi Q, Llewellyn-Jones D. Group-based Key Management for Mobile Sensor Networks. In *Proceedings of 2010 IEEE Sarnoff Symposium, Princeton, NJ, USA, 2010.* pp. 1–5.

[28] Rolf Blom. An Optimal Class of Symmetric Key Generation Systems. *In Proceedings EURO-CRYPT 84*, pages 335-338, Paris, France, April 1985. Springer.

[29] Zhen Yu and Yong Guan. A Key Management Scheme Using Deployment Knowledge for Wireless Sensor Networks. *IEEE Transaction on Parallel and Distributed Systems*, 19(10):1411-1425, 2008.

[30] L. Shen; H. Feng, Y. Qiu, and H. Ding, "A New Kind of Cluster-based Key Management Protocol in Wireless Sensor Network", *Proceedings of the IEEE Conference of Networking, Sensing and Control*, 2008, pp. 133-136.

[31] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient communication protocol for wireless microsensor networks", *Proceeding of the 33rd Annual Hawaii Int'l Conference on System Sciences, IEEE Computer Society*, 2000, pp. 3005-3014.

[32] MacQueen, J. B. (1967). "Some Methods for classification and Analysis of Multivariate Observations". In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press. pp. 281–297.

VITA

ANVESH REDDY AILENI

Candidate for the Degree of

Master of Science

Thesis: KEY MANAGEMENT IN STATIC AND MOBILE SENSOR NETWORKS

Major Field: COMPUTER SCIENCE

Biographical:

Education:

Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in May, 2011.

Completed the requirements for the Bachelor of Science in Computer Science at Jawaharlal Nehru Technological University, Hyderbad, India in 2009.

Experience:

Research Assistant — August 2009 – Present
Oklahoma State University — *Stillwater, OK*

- Design and administration of seismic lab running on RedHat Linux using NIS, NFS
- Design and maintain of website for Boone Pickens School of Geology

Research Internship: — January 2009 – April 2009
International Institute of Information Technology — *Hyderabad, India*

- Researched cryptographic method and employed in broadcasting of a message to users
- Developed hierarchical key structure for broadcasting encrypted messages to users having different key level

Intern Software Developer: — May 2007 – August 2007
Free Software Movement of India — *Hyderabad, India*

- Oversaw in building project PGOCTAVE, a Graphical User Interface for GNU Octave in a team of 3 people
- Recommended in embedding this into E-Swecha OS, which is an OS based on Debian based GNU/Linux

Professional Memberships: Computer Society of India (CSI), International Society of Technical Education (ISTE)

Name: ANVESH REDDY AILENI Date of Degree: July, 2011

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: KEY MANAGEMENT IN STATIC AND MOBILE SENSOR NETWORKS

Pages in Study: 34 Candidate for the Degree of Master of Science

Major Field: Computer Science

Scope and Method of Study:

Wireless sensor networks consist of sensor nodes with limited computational and communication capabilities. These networks are dynamic in nature that they allow addition of nodes to the network thus increasing the size of the network over time. The nodes communicate with each other in a secure manner and send processed data to the base station. This thesis concerns Key management in static and mobile sensor nodes. The network of sensor nodes is divided into clusters based on their physical locations in both cases. In addition, efficient ways of key distribution among the nodes within the cluster and among heads of clusters are discussed. The security of the network is considered through efficient key management by taking into consideration the network's power capabilities is discussed.

Findings and Conclusions:

The scheme presented in this paper is an effective ways of distributing keys among static and mobile sensors. The network is divided into clusters based on their physical location and the maximum distance between any two points in the cluster is the communication range of the sensor. By dividing the network into clusters we are limiting the key pool size to make it contain keys of only nodes in the cluster. The simulation performed confirmed the overall improved performance of the system.

ADVISER'S APPROVAL: Dr. Subhash Kak