OPTIMAL RESOURCE KEY MANAGEMENT PROTOCOL FOR CLUSTERED HETEROGENEOUS WIRELESS SENSOR NETWORKS

T.Kavitha[1] ***and Dr.D.Sridharan***[2]

[1, 2]Department of Electronics and Communication Engineering, Anna University-Chennai 25.
Email: [1]*haikavi18@yahoo.co.in*, [2]*Sridhar@annauniv.edu*

ABSTRACT

Secure communication in a Wireless Sensor Network (WSN) can be achieved through efficient key management techniques. Achieving security is not so simple due to the limited capability of radio module in wireless sensor network device. The probabilistic model provides a flexible and efficient solution when comparing various key distribution schemes developed so far. Here, we propose an Optimal Resource Key Management Protocol (ORKMP) considering the limited capabilities of WSN for a Clustered Heterogeneous Wireless Sensor Network (CHWSN) which follows a hierarchy. Better performance and scalable security solutions can be achieved by heterogeneous sensor nodes. The sensor nodes are deployed in groups, each controlled by a cluster head, which leads to less communication overhead. The keys distributed to nodes are based on the probabilistic model which combines the deployment knowledge with the sensor node's attributes of signal range and state. This hybrid effect reduces the memory requirement, by sustaining high connectivity and resilience. On the whole, the proposed key management protocol provides an efficient security solution with optimum resource requirement.

Keywords: Wireless sensor network, security, clustering, probabilistic key distribution, and sensor attributes.

1.0 INTRODUCTION

A wireless sensor network (WSN) is a network consisting of a collection of resource constrained sensor nodes which are capable of accomplishing various functions such as sensing, processing, transmitting and receiving to meet the application objectives. The sensor nodes deployed in a hostile environment can be used to detect, monitor and collect the data, and to perform decision making and evaluation. The sensor nodes in a hostile environment can be eavesdropped, captured and compromised. Hence, WSNs demand security which serves to conserve the confidentiality, integrity and availability of the transmitted information. But providing security is complicated compared to a traditional wired network and a mobile ad hoc network, because of the node's resource constraints, the wireless communication employed, the deployment methods, the location of the field of interest and huge number of nodes in the network.

The core design intention in the research on WSN is the efficiency in terms of energy, communication, computation, hardware complexity and memory. The clustering provides a scalable Medium Access Control (MAC), routing and also reduces the communication overhead [7]. Since the cluster head does the aggregation of data, the amount of data transmitted to the base station is reduced. There are homogeneous and heterogeneous types of cluster sensor networks. In a homogeneous network, the cluster head is overloaded with long distance transmission, high processing power and protocol synchronization, which lead to running out of energy before the other nodes. To overcome this difficulty, the cluster head is chosen randomly in such a way that all the nodes get their turn as said in Low Energy Adaptive Clustering Hierarchy (LEACH) [15]. But, the obstacle is that all the nodes need to have the required hardware capabilities leading to increase the cost of the network. In contrast, in a heterogeneous network, a few cluster heads are fixed by having extra hardware and energy. When the nodes use single hopping, those nodes which are away from the cluster head spend more energy to reach the cluster head. In multi hopping, nodes closer to the cluster head attains high communication overhead due to relaying other node's data to reach the cluster head. The obstacle is the non-uniform energy drainage pattern within the nodes of the network. In order to overcome this problem, the nodes could have the ability to move around the cluster region so that all the nodes have a chance to be a one hop neighbor of the cluster head.

To achieve security in WSNs, it is important to be able to perform various cryptographic operations including encryption, decryption, authentication, and so on. Applying any encryption scheme requires transmission of extra bits, hence extra processing, memory and battery power, which are very important resources for the sensors' longevity. Applying the security mechanisms could also increase delay, jitter and packet loss in WSNs. So, selecting the appropriate cryptography method for sensor nodes is fundamental to providing security services in WSNs. The process by which asymmetric and symmetric key cryptography schemes to be selected is based on the criteria given in [18]. The main issue in designing any protocol for WSN is the efficiency with respect to energy, communication, computation, hardware complexity and memory. Selecting asymmetric key cryptography is too expensive for many applications because of their integral complex mathematical calculations and increased code storage requirements. Keeping all these parameters in mind, symmetric key cryptography could be the best option to provide security for a WSN. However, symmetric key cryptography is not as versatile as asymmetric key cryptography scheme in case of key management is concerned. And hence, designing secure applications in symmetric key cryptography increases key management overhead. In order to reduce the key management overhead, the cluster based topology can be incorporated into the WSN [4, 19].

Secure communication in a wireless sensor network can be achieved through efficient key management techniques. The main task of a key management technique is the safe distribution of the secret keys to the communicating nodes before deployment (key pre- distribution), or safe agreement of keys between the communicating nodes after deployment. Secret key distribution can be a random or a deterministic approach. Even though the random approach gives the probabilistic model for connection establishment, it provides flexibility in key distribution to consider the attributes of the network and node, compared to the deterministic approach.

The efficiency of Key Pre-Distribution (KPD) can be improved by taking into account the network or sensor node attributes or both. The network attributes shall be the network model (distributed / hierarchical), the node distribution (linear/ non-linear) and the deployment model (single/ group and random/ grid). The node attributes shall be the identifier (node/group), the transmission range, the node state, the signal range, the attack probability, the deployment point (node/ group), the location (node/ group), and the distance of the node from the cluster head.

The rest of the paper is organized as follows. Section 2 provides some of the related work and section 3 describes fact, intuition, assumption and the proposed ORKMP protocol. Section 4 discusses the results and analysis of the algorithm with respect to the evaluation parameters like scalability, memory required, connectivity, resilience and communication overhead and the performance comparison with the existing schemes. Finally, section 5 provides the conclusion from the results.

2.0 RELATED WORK

In this section, we discuss some of the key pre-distribution schemes proposed earlier, based on the probabilistic approach using a key pool.

Eschenauer and Gligor (EG) [6] proposed the first probabilistic approach based on a balanced key pre-distribution scheme, for a distributed wireless sensor network with homogeneous nodes deployed randomly. Key chains are formed by randomly drawing K keys from the key pool of size P without replacement, and not including any attributes. A secure link between the two neighbor nodes shall be established if they share at least one key. The key *ID* or puzzle based shared key discovery (SKD) process is used. If the SKD fails, the path key is established using the key sharing graph (KSG). Chan et al. [3] extended the Basic Scheme [6] by having q common keys to establish a secure link between two neighbor nodes, which increases the resiliency. Rajgopal Kannan et al. [11], contributed to the distributed homogeneous network through the probabilistic balanced non-uniform key distribution scheme. Sensor nodes are preloaded with a combination of randomly derived and inherited keys. Here, the number of keys shared between any two nodes is more, compared to the Basic scheme [6].

Park.J et al. (PKK) [10] proposed a scheme for a distributed sensor work with random deployment that follows the probabilistic balanced non-uniform distribution of keys, using the state of the sensors. It avoids unnecessary key assignments, with reduced memory requirement and high connectivity. It also lowers the fraction of compromised communication. S.P. Chan et al. [12] suggested a random key pre-distribution scheme, using the probability of node compromise and the deployment knowledge for a hierarchical heterogeneous network. A network of N nodes is divided into G subgroups, each having a different compromising probability. It improves the resilience of that particular subgroup. A scheme was modeled by Takashi Ito et al. [13] based on random key distribution and it uses the deployment point of the sensor for a homogeneous distributed sensor network. It uses a key-position map and the probability density function of node deployment, where the key-position map shows which key is assigned to which position. For the different deployment model, a connectivity analysis is done. W. Du et al. (DDHV-D) [14] introduced a scheme for the distributed homogeneous network with grid-based group deployment. The probabilistic approach through an unbalanced key pre-distribution scheme uses a deployment model, was proposed. Here, the key pool is divided in such a way that the neighbor key pool shares more keys, and each corresponds to one deployment group. This method increases the local connectivity of the nodes.

Patrick Traynor et al. [9] considered a heterogeneous network which is deployed randomly. An unbalanced uniform key distribution through the probabilistic approach is proposed. It supports large savings of key space and reduces the communication overhead. For a heterogeneous hierarchical network, an unbalanced random key pre-distribution scheme is proposed by B. Maala et al. [1]. Here, the shared key discovery process is done through the tree construction process which reduces the key storage overhead. Nguyen et al. [8] proposed a scheme for a distributed homogeneous network, where nodes are deployed in a predefined location, which uses the square grid. Key pre-distribution uses the probabilistic approach, by considering the signal range of the nodes, where the neighboring nodes share more keys. Here, the connectivity of the scheme is always one with better resilience. K.Panyim and P.Krishnamurthy [5] have proposed a hybrid key pre-distribution scheme for the distributed sensor network that uses the spatial retreat to cope with jamming attacks. It combines the advantages of [6, 14] by using the group-based deployment model.

Mittal and Novales (MN) [16] introduced a key pre-distribution scheme that uses deployment knowledge to divide deployment regions into overlapping clusters. This approach improves the resilience without any compromise, improves the local connectivity of the network. The scheme also provides improved global connectivity and flooding overhead, when compared to a perfectly resilient scheme, which uses group-based deployment knowledge. Additionally, it does not place upper or lower bounds on the key ring size.

Albert Levi et al., [20] introduced re-usable key pool-based probabilistic key distribution scheme that increase the scalability. Bechkit et al., [21] proposed a tree-based probabilistic key distribution scheme and hash function that increases the resiliency. Khan et al., [22] used a symmetric matrix and generator matrix of Maximum Rank Distance (MRD) codes for symmetric key generation and pre-distribution scheme, which provides low communication overhead, high connectivity and scalability. Bechkit et al., [23] proposed a key management scheme using enhanced unital design theory which provides high network scalability and good key sharing probability. Kur et al., [24] proposed two novel improvements to random key pre-distribution. The first one uses limited length collisions in secure hash functions which provide high connectivity and the second one introduces hash chains into the key pool construction which provides high resilience. Use of network security concepts has even been found in some natural language processing applications, [25].

3.0 PROPOSED WORK

From the discussion on introduction and related work, the facts observed are: i) when the attributes of the node or network is combined with the key pre-distribution that leads to effectiveness in various dimensions of performance. ii) Designing a protocol for a CHWSN is effortless and also provides efficiency in various aspects like energy, computation, communication and scalability. iii) Adopting the deployment knowledge and signal range of nodes is straightforward in case of group deployment is concerned. iv) Design of key distribution algorithm using hybrid attributes of the nodes have not addressed so far. From these facts, it is the intuition that design of a key management protocol for the Clustered Heterogeneous Wireless Sensor Network by adopting the hybrid attributes of the nodes like state [10] of the node along with the signal range [8] and deployment knowledge [14] of the group. We are the first to take the effort to combine the state, and signal range of the node along with deployment knowledge for key pre-distribution. Here cluster head key ring is derived using the inheritance property whereas for the nodes probabilistic method is used. This combined effect will improve the performance of the network in terms of connectivity, resilience, scalability and reduces the communication overhead and memory requirement. To design this protocol, some of the assumptions are made which are given in network model and attacker model.

3.1 Network model

A wireless sensor network, consisting of a large number of resource limited sensor nodes, is considered with a cluster head and base station following a three-tier architecture, where each tier has different capabilities. The base station maintains all information about the clusters. The cluster heads are assumed to be equipped with high resources compared to the sensor nodes, in terms of memory, communication range, computation speed and energy.

They are also assumed to have tamper resistant hardware. Such an assumption is feasible, because the number of cluster heads is very less compared to the number of nodes. A set of nodes with a cluster head will form a cluster, based on the location and communication range, which are identified by the cluster *ID*. The clusters are deployed from the top of the deployment point associated with each cell by the deployment vehicle, for example air craft, which provides the assistance for sensor node deployment. The nodes within a cluster are assumed to be reachable by the cluster head, and they can reach the cluster head in single or multi hops. The cluster head and the nodes are capable of having limited mobility within the cell. The nodes perform sensing and relaying, whereas the cluster head performs key updating, revocation and aggregation of data that are received from its nodes, and then forwards it to the base station directly or through another cluster head.

3.2 Thread model

Assume that the adversary can eaves-drops all data transfer between any two nodes due to their radio nature. Assume that capturing and compromising of some nodes in a cluster of the network will make it possible for the adversary to take control of that cluster. Assume that to compromise the node, the intruder needs at-least some time, which is greater than the bootstrapping time. Once the nodes are captured, their memory content can be read and modified. Cluster heads are assumed to be tamper proof, and hence, cannot be compromised. Assume that the adversary does not have any knowledge of the content of the nodes prior to a node compromise, and can capture in any part of the network. The target of the adversary is to reveal partial or entire keys by capturing the minimum number of nodes to compromise a fraction of the secure links or the whole network itself. The models considered here for node capture are (1) Random node capture attack and (2) Selective node capture attack, which are considered here.

3.3 Terms

The list of definitions for the technical terms that will be used throughout this paper is given in Table 1.

Table 1: Terms

Cell	-	Smallest partition of the field E.g.: hexagon cell.
Unit	-	Hexagon cell enclosed by square.
Deployment point	-	Position from where the nodes are deployed, i.e., center of the cell.
Deployment group	-	Smallest set of nodes which are deployed from a deployment point at a time.
Cluster head	-	It controls the set of nodes in a cell.
Cluster	-	The smallest set of nodes in the network grouped together.
State group	-	Set of nodes which are in the same state, at the same time-interval.
Physical neighbors	-	Set of nodes which come within the communication range of the node.
State neighbors	-	Set of nodes which are active at the same time.
Secure neighbor	-	Set of nodes that shares at-least one key.

Communication neighbor	-	Set of nodes which are physical, state and secure neighbors
Network key pool	-	Set of keys that are used to the network.
Cluster key pool	-	Set of keys that are used to the cluster.
State key pool	-	Set of keys that are used to the state group.
Key Ring	-	Set of keys that are assigned to the node or cluster head.

3.4 System Architecture

The system architecture has seven stages as given in Fig 1:

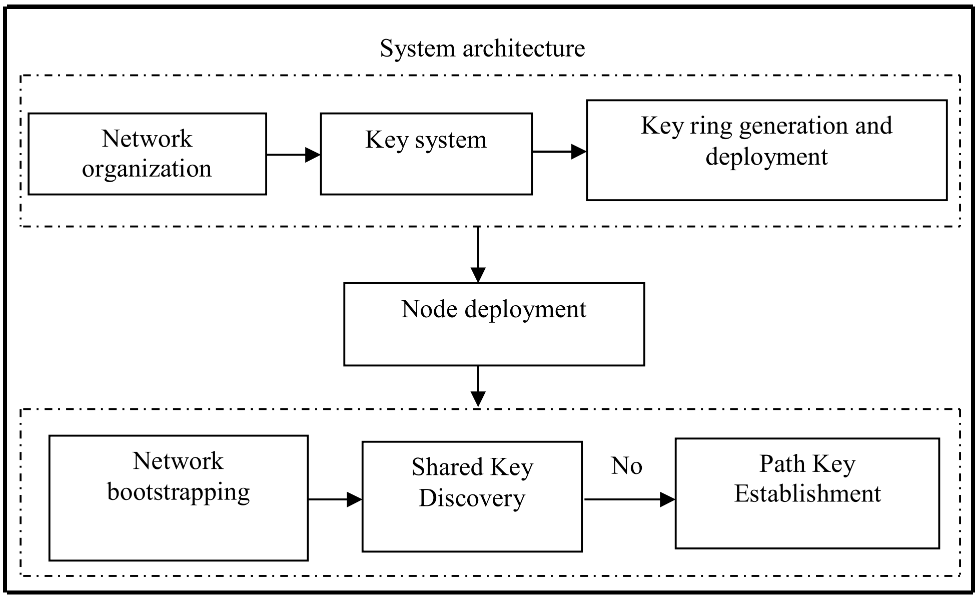


Fig. 1: System architecture

1. The Network organization is the first step in which the nodes are arranged as clusters, each with a cluster head. 2. In Key system, the network key pool is divided in to cluster key pool which in turn divided into state group key pool for each cluster and state group. 3. In key chain selection, the key chains are formed for each cluster head and cluster members; and they are deployed into the cluster head and its cluster members. 4. Once the nodes are deployed with keys, then each cluster is deployed randomly from top of the center of its cell. 5. In bootstrapping stage, Cluster head registration with BS and clustering are done. 6. In the shared key discovery process, communication neighbor set discovery process is initiated by each node. From this communication neighbor set, shared key for any two nodes is found. 7. If SKD process fails, the path key establishment process will be executed, which finds the secure path by

using other secure neighbors. In this section, all the seven stages are discussed in detail. Explanations of symbol used in this proposed work are given in Table 2.

3.4.1 Network organization

A new way of grouping the nodes as a disjoint cluster process is done before deployment. It has clustering and state grouping process which is done as given below.

Clustering: Initially the nodes are arranged based on node identifier starting with $N_1, N_2 \dots N_N$. Then, the given set of N nodes is grouped using the node identifier into C clusters based on the size n_i of each cluster. Each cluster is controlled by one cluster head (CH_i) and it is identified by a unique identifier (C_i). Clustering allows scalable key management and distribution of activities (sensing, data aggregation and secure data transmission) among the cluster head and the nodes. The deployment point of each cluster is determined by mapping the cluster with the cell of the target field. Here, the nodes in the clusters have a high probability to be in the communication range of one another. All the nodes in the cluster come within the signal range of the cluster head, but not vice versa.

State Grouping: Within the cluster, the state of the sensor attribute is exploited. For simplification, the state of the sensors can be idle, sleep or active. But, in general, there are several sleep states available. Based on the number of states of the nodes, each cluster C_i is again divided in to S_i number of state groups based on the size n_{ij}' of the state group of the cluster; which is identified by the unique identifierS_{ij}. Now the nodes in a j^{th} state group belonging to i^{th} cluster head are arranged as follows:

$$S_{ij} = SN_{i.j.1}, SN_{i.j.2}, \dots, SN_{i.j.k} \qquad (1)$$

The nodes in the state group are in the same state at the same time with high probability. Each state group will be in the active state at different points of time. But at a time, more than one group will not be in the same state. State groups which come under one cluster are controlled by its cluster head. The following conditions are satisfied when grouping the nodes:

$$N = n_1 + n_2 + \cdots + n_C \qquad (2)$$

$$n_i = n_{i1}' + n_{i2}' + \cdots + n_{iS_i}' \qquad (3)$$

$$C_i = S_{i1} \cup S_{i2} \cup \dots \cup S_{iS_i} \qquad (4)$$

$$Network = C_1 \cup C_2 \cup \dots \cup C_{n_i} \qquad (5)$$

Table 2: Symbol

Symbol	Significance	Symbol	Significance
N	Network size or number of nodes	KRN_{ijk}	k^{th} Key ring of the j^{th} State group key pool that belongs to the i^{th} cluster key pool
C	Number of clusters	KRC_i	i^{th} cluster head key ring
C_i	Cluster ID	m'	Node Key ring size
CH_i	i^{th} cluster head	m	Keys taken from each KS_{ij} for KRC_i
n_i	Number of nodes in a cluster	L , W	Length and Width of the target field
S_i	Number of state group in a cluster	$U_{i,j}$	Unit identifier
n_{ij}'	Number of nodes in a state group of cluster	$R_{i,j}$ or $C_{i,j}$	Cell identifier of *H/V* Hexagon
S_{ij}	j^{th} state group *ID* of the cluster i.	r	Radius of the circum circle of hexagon
N_i	Node identifier	a	half width of hexagon
$SN_{i.j.k}$	k^{th} sensor node of j^{th} state group of i^{th} cluster	s	Side of hexagon
K_{C_iB}	i^{th} Cluster-base station pair wise key	N_r	number of rows
KN	Network key pool	$N_{cr_{even}}$	number of cells in even row
K	Size of network key pool	$N_{cr_{odd}}$	Number of cells in odd row
KC_i	i^{th} Cluster key pool	P and P'	Point (x , y)
k_i	Size of cluster key pool	(x_{min}, y_{min})	Coordinates of the unit which takes minimum value
KS_{ij}	j^{th} State group key pool of i^{th} cluster	(x_{max}, y_{max})	Coordinates of the unit which takes maximum value
k_{ij}'	Size of state group key pool	G_i	Deployment group identifier
K_i	Key identifier	x	Number of nodes captured

3.4.2 Key system

The key system explains how the keys are arranged in order to do the key distribution. It has four steps, as follows. Generate the C pair-wise Cluster-Base station keys (K_{C_iB}) for each cluster C_i as given in Equation (6), which is shared between the base station and the i^{th} cluster head (CH_i).

$$K_{C_iB} = f(v||ID_{Base}||C_i) \qquad (6)$$

Here, v is a random number and f is one way function. Generate a large network pool KN of K symmetric keys, with a required length of 64 or 128 bits, depending upon whether the AES or DES algorithm is used. Initially arrange the keys based on key identifier starting with$K_1, K_2 \dots K_N$. The nodes in different clusters will not communicate with each other. So it is unnecessary for the nodes in different clusters to share the keys. This unnecessary key assignment should be avoided. So, the network key pool is divided in to C distinct cluster key pool KC_i of size k_i. With each cluster key pool, include a Cluster-Base station pair-wise key K_{C_iB}.The nodes within the cluster will transfer the data to the cluster head through a single or multi hops. But all the nodes in the cluster will not be in the same state simultaneously. So, an active node can transfer the data only through a node which is in the active state. Therefore, it is not necessary to share the keys between all the nodes in the cluster. Here, the keys are assigned in such a way that the nodes should have high probability for sharing the keys with in the same state groups (time neighbors), but not with the different state groups. In order to pull off this situation, each cluster key pool is again divided in to S_i distinct state group key pool $KS_{ij}$of size $\mathrm{k_{ij}}'$. The following conditions are satisfied when grouping the keys:

$$K = k_1 + k_2 + \cdots + k_C \qquad (7)$$

$$k_i = k_{i1}' + k_{i2}' + \cdots + k_{iS_i}' \qquad (8)$$

$$KC_i = KS_{i1} \cup KS_{i2} \cup \ldots \cup KS_{iS_i} \qquad (9)$$

$$KN = KC_1 \cup KC_2 \cup \ldots \cup KC_{n_i} \qquad (10)$$

3.4.3 Key ring generation and deployment

This section explains the method of assigning keys to the nodes, cluster heads and base station.

Table 3: key ring generation and deployment

```
for ($i$=1 to $C$)
•  begin
•  Assign (BS , $CH_i$ ) ←$K_{C_iB}$
   for ($j$=1 to $S_i$)
   •  begin
   •  Randomly select m keys from $KS_{ij}$
   •  $KRC_i$←$m$ keys
      for ($k$=1 to$n_{ij}'$)
      •  begin
      •  Randomly select $m'$keys from $KS_{ij}$to form $KRN_{ijk}$
      •  Assign $SN_{i.j.k}$←$KRN_{ijk}$
      •  End
   •  End
•  Assign $CH_i$←$KRC_i$
•  End
```

It is performed before deployment by the setup server. A probabilistic unbalanced non-uniform clustered key sharing algorithm is proposed. This is done by mapping the key system with the network organization. Select m^' keys randomly from the state group key pool KSij without replacement to form the key ring of node KRNijk, which is installed in the sensor node SNijk in the state group Sij. The cluster head CHi of the state groups Sij is assigned with the key ring of cluster KRCi of Si × m keys where m keys taken randomly from each state group key pool KSij, where〚 m〛^'≫m. And also each cluster head CHi and the base station is loaded with KCiB pair-wise key, which provide the authentication between the cluster head and the base station. The pseudo code for key ring generation and deployment is given in Table 3.

3.4.4 Node deployment

Deployment of nodes follows the uniform distribution of group deployment for the network which uses the location information. The field of interest is divided into cells in order to decide the possible location of the cluster. The cell may be a square, circle or hexagon. The nodes are deployed randomly inside the hexagonal cell. So, the topology formed by the sensor nodes is irregular. But, the cluster heads are placed in the center of the cell. Hence, all the cluster heads are forming the regular hexagon topology where cluster heads have six neighbors. This hexagon topology provides highly scalable, efficient, and easily optimizable networking protocols [17], which are eye-catching for the resource constrained large-scale wireless sensor networks. The Hexagonal wireless sensor networks are well suited for multi-hop real-time communications [17] which provides timely delivery of data.

The target field has a set of cells, by dividing it into N_r rows or N_c columns which depends on whether hexagon structure is a Horizontal (H-Hex) or Vertical (V-Hex) which is shown in Fig 2. A cell of the target field is denoted by $R_{i,\,j}$ or $C_{i,\,j}$ which depends on whether the field is divided into rows (H-Hex) or columns (V-Hex). Arrange the deployment group G_i in order before the deployment, based on the cluster identifier C_i.

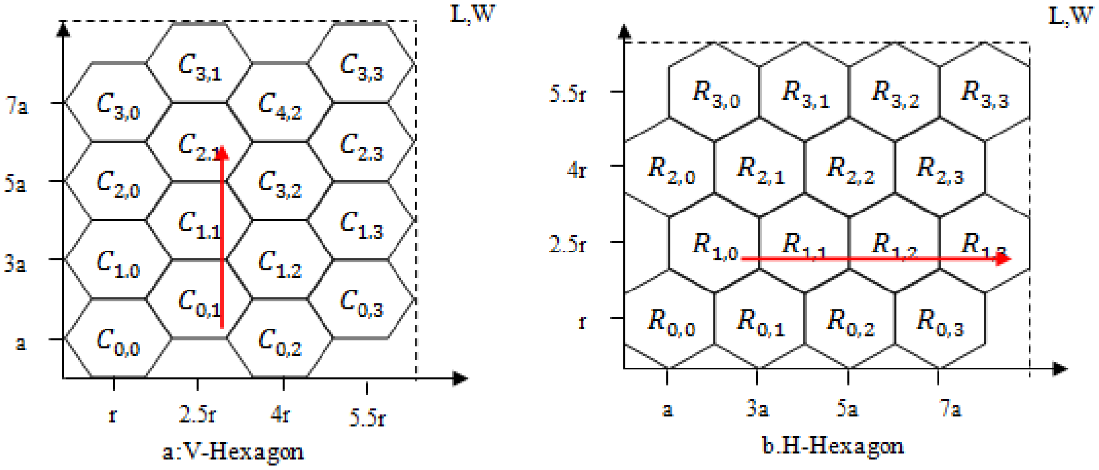
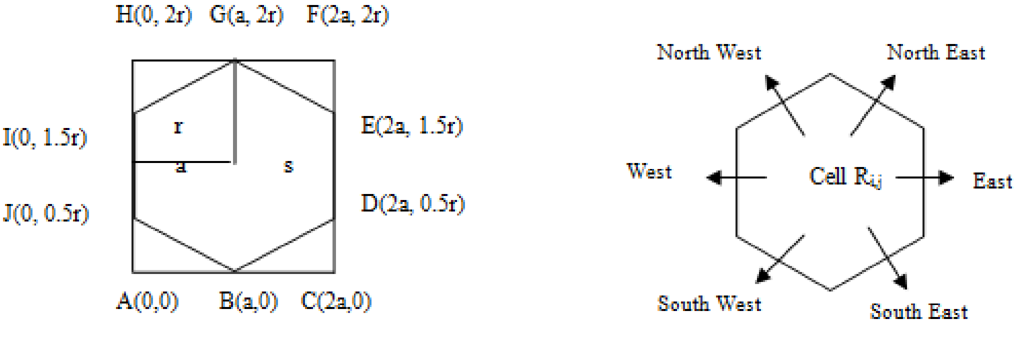


Fig. 2: Structure of Hexagon

The deployment group G_i corresponds to cluster C_i which consists of its own state group nodes and its cluster head. For each deployment group G_i, center of the cell is chosen as the deployment point, so that all the deployment points fall with equal distance. This process leads to the deployment of the sensor nodes uniformly over the entire region whereas the nodes within the cluster are distributed non-uniformly. The deployment probability density function for the nodes in group G_i is following the Gaussian distribution. The resident point of the cluster head CH_i is assumed to be at the center of the cell in order to get entire coverage of the nodes which are deployed randomly within the cell. Table 4 explains the algorithm for the deployment in case of H-Hexagon.



a. Circumcircle(r), side(s),half width (a) of H-hexagon cell & Coordinates of origin unit $U_{0,0}$

b. Directions

Fig. 3: Origin unit coordinates and directions of cell in H-Hex structure

Table 4: Node deployment in H-Hex structure

```
1. Get the value of circum circle, ‘$r$’ of hexagon.
2. Calculate the half width of hexagon $a = (\sqrt{3}/2)r$.
3. Get the length $L$ of target area
4. Get the width $W$ of target area
5. Calculate number of rows $N_r = ((W/r - 2)/1.5) + 1$
6. Calculate number of cells in each row.  Number of cells in even rows: $N_{cr_{even}} = L/2\,a$.
                                           Number of cells in odd rows: $N_{cr_{odd}} = N_{cr_{even}} + 1$
7. Select deployment point for sensor nodes and cluster head as follows.
   for($i$=0 to $N_r - 1$)
    {
     If ‘$i$’ is even, $end = N_{cr_{even}} - 1$
     Else $end = N_{cr_{odd}} - 1$
     for (j=0 to $end$)
       {
          7.1.  Calculate the coordinates of unit $U_{i,j}$ based on the directions as given in Fig 3b.
                {
                 If ($i$= =0 and $j$= =0)      // Origin unit
                  Calculate the coordinates of origin unit$U_{0,0}$ as given in Fig 3a.

                 If ($i$ is odd and $j$ = = 0)      //north west unit.
                  Calculation of coordinates of unit $U_{i,0}$ using coordinates of unit$U_{i-1,0.}$
                  $P = (X,Y)$ then $P^{'} = (X-a\ ,\ Y+1.5r)$

                If ($i$ is even and $j$ = = 0)      //north east unit
```

```
                  Calculation of coordinates of unit$U_{i,0}$ using coordinates of unit$U_{i-1,0}$.
                  $P = (X,Y)$ then $P' = (X+a , Y+1.5r)$

                 Else     // east Unit
                  Calculation of coordinates of unit $U_{i,j}$ using coordinates  of unit$U_{i,j+1}$
                  $P = (X,Y)$ then $P' = (X+2a , Y)$.
                }
        7.2.    Validate the coordinates of unit $U_{i,j}$
                  a.   Find $(x_{min}, y_{min})$and $(x_{max}, y_{max})$ of the unit.
                  b.   Check $x_{min} \leq X \leq x_{max}$ and $y_{min} \leq Y \leq y_{max}$.
        7.3.    Generate set of points, lying inside the unit $U_{i,j}$
        7.4.    Group the generated points of Unit $U_{i,j}$ which are lying inside the hexagon cell $R_{i,j}$.
                  a.   Form the three side equations for all four corner triangle's of cell $R_{i,j}$.
                  b.   Calculate the centroid for all four triangles.
                  c.   Substitute the given point and centroid of the triangle in the LHS of the equation,
                       if both are same sign for all the three equations of the triangle, then given point $P$
                       is within the triangle, else outside.
                  d.   If the Point lying outside the all four triangles, then the point is lying inside the
                       hexagon $(i,j)$.
        7.5.    Choose deployment points randomly, from the grouped points, for each sensor node in a
                 cluster $(i, j)$.
        7.6.    Choose centre of hexagon $(i, j)$ as deployment point for cluster head in cluster $(i, j)$.
    }
}
```

3.4.5 Network bootstrapping

Once the nodes are deployed, the cluster heads begin the registration process with the base station and the clustering process is done. This process is explained as follows.

a. **Cluster head registration with BS**:

After deployment, each cluster head establishes a connection with the base station. The cluster head forwards an initialization packet, having its identifier CH_i and the identifier of the cell it belongs $R_{i,j}$, to the base station in a single or multiple hops. The initialization packet is encrypted by a key K_{CiB}. After receiving the initialization packets from the cluster heads, the base station checks with its database for authentication and it replies that the registration was done successfully.

b. **Node discovery and clustering**:

The cluster head starts its registration process which follows the node discovery process. The cluster head broadcasts the announcement of its identifier announcing its presence. Upon receiving the cluster head announcement, the node checks whether it belongs to that cluster. If yes, the node sends its identifier to the cluster head to join the cluster. If not, the node moves with applicable mobility and checks whether it can receive any more announcements from its own cluster head. Once the node receives the announcement from its parent cluster head, then it will join with that cluster. Otherwise, it will become an orphan node. Since the clusters are deployed from the center of the cell, all the nodes in the cluster will reside within the cell itself. So, the probability of getting an orphan node is very less. After the node discovery process, the cluster head continuously monitors the nodes under its control.

3.4.6 Shared key discovery

Once the bootstrapping process is over, each node has the need to find communication neighbor set with its neighbors. Each node A in the cluster broadcast its identifier which includes state group identifier and cluster identifier to its each physical neighbor B. The receiving physical neighbor B checks whether it belongs to the same cluster and state group. If so, node B responds with acknowledgement which has its set of key ring identifier. Now node A compares its key ring identifier with each receiving node *B's* key ring identifier. If any key identifier match is found, then node B is included in the communication neighbor set and the corresponding key is considered as a shared key for the node A and B. Once this communication neighbor discovery process is over, then the entire sensor network develops a key space sharing graph which is defined as follows:

Definition 1: Let *KSG (V, E)* be a key space sharing graph, where V represents the nodes and E represents the link between the nodes in the sensor network. For any two nodes $\{a, b\} \in V$, there exists an edge between them, if and only if they are *i*) physical neighbors, *ii*) Time neighbors and *iii*) secure neighbors.

3.4.7 Path key establishment

There is a possibility that, two nodes which are physical and time neighbors that may not be secure neighbors. In such a case, they need to find a secure path, by using one or more of the other communication neighbor nodes. It can be done easily by using the key space sharing graph, which was developed in the earlier step.

3.4.8. Key updating technique and revocation

A key updating technique is proposed using one way function to dynamically update and remove the old keys. This technique mainly enhances the resilience of the key distribution schemes. Table 5 gives the steps involved in the technique.

4.0 PERFORMANCE EVALUATION

In this section, the evaluation of the ORKMP's performance is shown through simulation using Castalia simulator. The performance metrics considered are memory, connectivity, resilience and communication overhead. The results shown are the average value of ten simulation experiments with different random seeds. Here, the results are compared with the existing key pre-distribution schemes of EG [6] basic random scheme, PKK [10] state based scheme which uses mutually exclusive key pool, EG-D scheme which is the basic random scheme with deployment knowledge, and W. Du et al.[14] DDHV-D scheme for group deployment which uses overlapped key pool. When C=1 & S=1, the scheme converts to basic scheme[6]; when C=1 & S=3, the scheme converts to state based scheme [10]; when C>1 & S=1, the scheme behaves as EG-D scheme with hierarchical clustered network which uses mutually exclusive key pool; when C>1 & S=3, it leads to the ORKMP, it behaves as hierarchical clustered network which uses deployment knowledge, state and mutually exclusive key pool.

Table 5: Key updating technique

1) Initially, the base station selects a new random number v and a new pair-wise Cluster-Base station keys (K_{C_iB}) for each cluster C_i.
2) The base station then constructs the new pair-wise Cluster-Base station keys for each cluster head.

$$K'_{C_iB} = f\left(v'\middle|\middle|ID_{Base}\middle|\middle|C_i\right), 1 \leq i \leq C$$

where v' is the new random number and *f* is the one way function

3) Then the base station forwards this new Cluster-Base station key by encrypting with old Cluster-Base station keys to concern CHs.

$$Basestation \rightarrow CH_i: E_{K_{C_iB}}\left[Q\middle|\middle|v'\middle|\middle|K'_{C_iB}\right]$$

Where Q is the indication of updating mesasge

4) Cluster head $\mathrm{CH_i}$ calculates updated keys for each key in the key ring of its own.

$$K'_j = f\left(v'||K_j\right) where\ 1 \leq j \leq mandK_j \in KRC_i$$

5) Each CH_i uses its shared key with its cluster members to encrypt v', and transmit it to its cluster members.

$$CH_i \rightarrow CM_i: E_{K_{shi}}[v'||Q]$$

6) CM_i then utilizes v' to re-compute the new keys in the key ring of its own.

$$K'_j = f\left(v'||K_j\right) where\ 1 \leq j \leq m'andK_j \in KRN_{ijk}$$

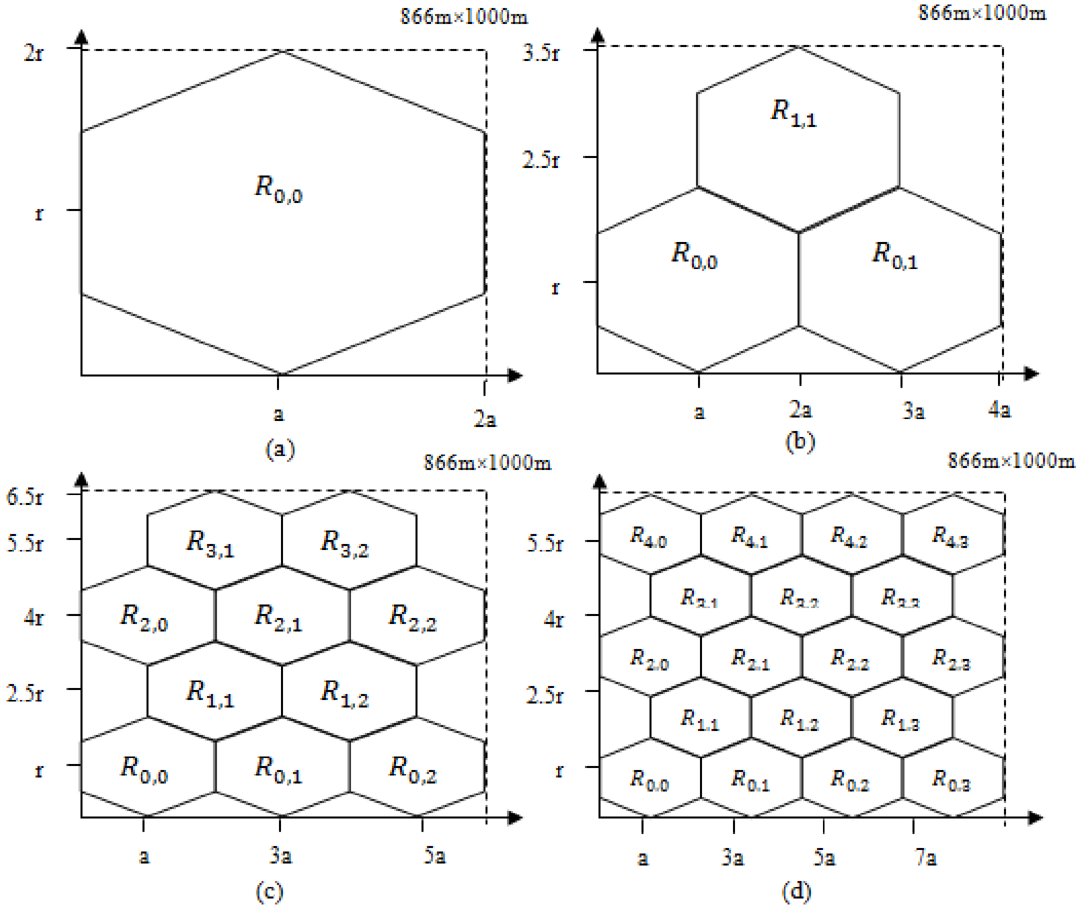
7) Following the above estimation, all the nodes and cluster heads delete initial key information.

Table 6: Properties of cell

S. No	Cluster arrangement	Scheme	No of Cluster and State group	Area	cell radius	Half Width
				L×W	r	a
1	As given in Fig 4a	EG scheme	*C=1 & S=1*	866×1000 2a×2r	500	433
2		PKK scheme	*C=1 & S=3*		500	433
3	As given in Fig 4b	EG-D (3 clusters)	*C=3 & S=1*	866×1000 4a×3.5r	250	216.5
4		ORKMP(3 clusters)	*C=3 & S=3*		250	216.5
5	As given in Fig 4c	EG-D(10 clusters)	*C=10 & S=1*	866×1000 6a×6.5r	166.66	144.3
6		ORKMP(10 clusters)	*C=10 & S=3*		166.66	144.3
7	As given in Fig 4d	EG-D(18 clusters)	*C=18 & S=1*	866×1000 8a×8r	125	108.3
8		ORKMP(18 clusters)	*C=18 & S=3*		125	108.3

4.1 System configuration

Simulation is performed with the following setup. The number of sensor nodes in the network is 10,000. For the deployment area 866m×1000m, the number of cluster considered are 1, 3, 10 and 18 as shown in Fig 4. And the corresponding cell radii are 500m, 250m, 170m and 125m which is the communication range of the cluster head deployed in the center of the cell and for each sensor nodes communication range is 40m. The cluster head is always in active state and the sensor nodes are changing their states to be active, idle and sleep periodically. The network key pool is fixed as 100,000. The node key ring size is fixed as 200 and the cluster head key ring size is fixed as 1000. Table 6 gives the properties of cell for each cluster arrangement as given in Fig 4.



(a)

(b)

(c)

(d)

Fig. 4: Clusters (a) *C*=1. (b) *C*=3. (c) *C*=10. (d) *C*=18

4.2 Connectivity

The connectivity is the metric which says about the probability of two nodes being connected securely. It can be of two types: local connectivity and global connectivity. Here, the local connectivity of a node is the probability of having communication neighbor set. As the local connectivity increases, the links established between the nodes also increases in the KSG. In the simulation, local connectivity for each node *A* in the cluster is calculated. Then local connectivity for the cluster is calculated using local connectivity of the time group within the cluster. Using the local connectivity of the cluster, local connectivity for the network is calculated using the formulas as below:

$$\alpha_{ij} = \left(\sum_{k=1}^{n_{ij}'} \lambda_k/\delta_k\right)\Big/ n_{ij}' \qquad (11)$$

$$\beta_i = \sum_{j=1}^{S_i}\left(TGC_{ij} \times n_{ij}'\right)\Big/ n_i \qquad (12)$$

$$\gamma = \sum_{i=1}^{C}(CC_i \times n_i)\Big/ N \qquad (13)$$

Where α_{ij} - j^{th} time group in i^{th} cluster local connectivity; β_i - i^{th} cluster local connectivity; γ - network local connectivity; λ_k - number of nodes in a communication neighbor set; δ_k - number of nodes that are physical and time neighbors of node k.

In this simulation, we can evaluate how the number of clusters where deployment knowledge is used and the state of the sensor nodes improve the local connectivity. By using (13) local connectivity γ is calculated for the schemes EG, PKK, EG-D, DDHV-D and ORKMP for various memory requirement scenarios. The outcomes are plotted in Fig 5. In EG scheme the local connectivity is very low and connectivity increases with increase in key ring size. When deployment knowledge is combined with the EG i.e., EG-D scheme, the local connectivity increases to an extent compared to EG scheme. The PKK scheme is superior to the EG-D scheme, since the state of the knowledge of the node is used. As in the case of DDHV-D scheme, it provides low connectivity for the small key ring sizes compared to EG-D and PKK; and outperforms the PKK and ORKMP (C=10, S=3) and finally reaches 1. This is because of the number of key spaces carried by each sensor. There is a discrete step for a given number of key spaces when key ring size increases [14]. For the proposed scheme, the exposure of more number of clusters and the states drastically improve the local connectivity of the network.

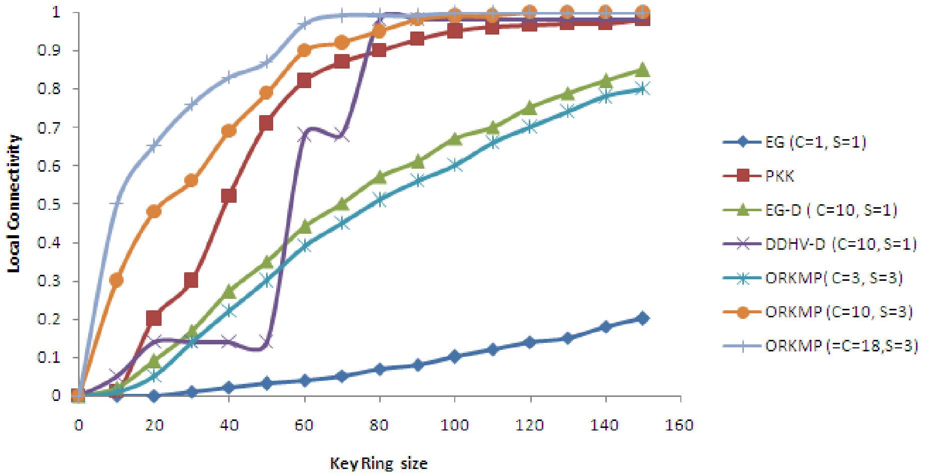


Fig. 5: Local connectivity performance comparisons K=100,000.

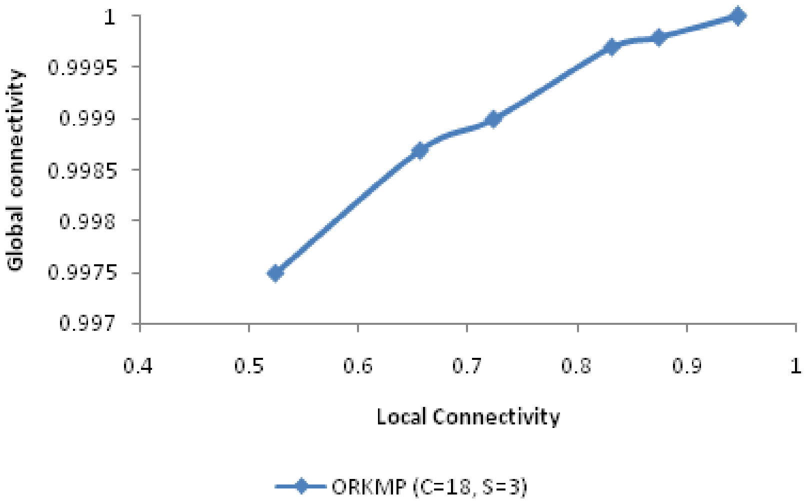


Fig. 6: Local connectivity *Vs* global connectivity

Global connectivity is the ratio of number of nodes in the largest connected component to the network size. When the local connectivity increases, links established between the nodes also increases in the KSG. But it is not necessary that the graph is fully connected, it can have isolated components. So it is essential to measure the global connectivity also. Fig.6. shows the relation between the local connectivity and global connectivity. From the Fig 6 it is seen that, when the local connectivity is 52 percent, only 0.25 percent of the node becomes orphan because of deficient in secure links; when the local connectivity is 95 percent, all the nodes are able to establish secure links.

4.3 Resilience

The application of sensor network requires deployment of the sensor nodes in adversarial region which enables the intruder to capture the nodes. Then, the intruder gets access to the keys from the captured nodes. Theses captured keys may be in use to establish a secure link between some other nodes in the cluster. Hence, the capturing of the nodes does not only affect the captured nodes, but also the other nodes which share common keys to the captured nodes. A fraction of additional communication is compromised based on the knowledge gathered from the captured nodes. So, it is important to know how an attack on x number of sensor nodes by an intruder influence the rest of the nodes in the network. Fig.7. shows the resilience performance comparison based on the fraction of communication compromised among un-captured nodes by using captured nodes. For this experiment, an attack area is assumed to have a radius of 250m in the simulation. The keys of few clusters inside the attack area are affected because the adversary focuses only inside the attack area. So, the clusters away from the attack area are not probable to be affected at all. Hence, the average resilience performance of clusters is satisfactory for the given x captured nodes. The key factor for this improved resilience is the clustering where each cluster uses mutually exclusive key pool.

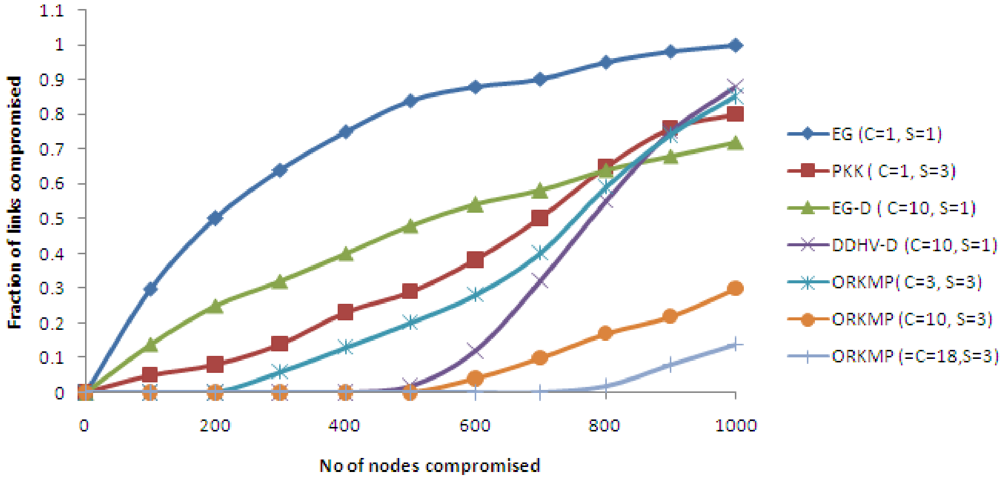


Fig. 7: Resilience performance comparisons, key ring size 200, Local connectivity 0.5 and attack circle radius 250

Here, the comparison of the fraction of links compromised for the EG, PKK, EG-D, DDHV-D, and ORKMP where *C* is 3, 10, 18 & *S*=3 *vs* different number of captured nodes is shown in Fig 7. From the Fig 7, it is seen that ORKMP outperforms the EG, PKK, EG-D but less than the DDHV-D when *C*=3 & *S*=3. But, when *C* increases, ORKMP outperforms the DDHV-D scheme too. This is because when *C* increases, n_i decreases which leads to decrease in n_{ij}. So the number of nodes sharing KS_{ij} is very small and the resilience increases with increase in *C*.

Fig 8 shows the relation of the resilience with memory usage and local connectivity. As the memory usage increases, connectivity also increases. To get the relation between memory usage and resilience, connectivity needs to be maintained constant. In order to maintain the constant connectivity, key pool size is increased which also provides better resilience. Fig 8a depicts that the resilience improves linearly and reaches 100 percent as the memory usage increases with *C*. Fig 8b represents that the resilience decreases linearly with increase in local connectivity. When *C* is 18, the decrease in resilience with local connectivity is very small compared to when *C* is 10 and 3. So it is obvious that the connectivity and resilience increases as the number of clusters increases.

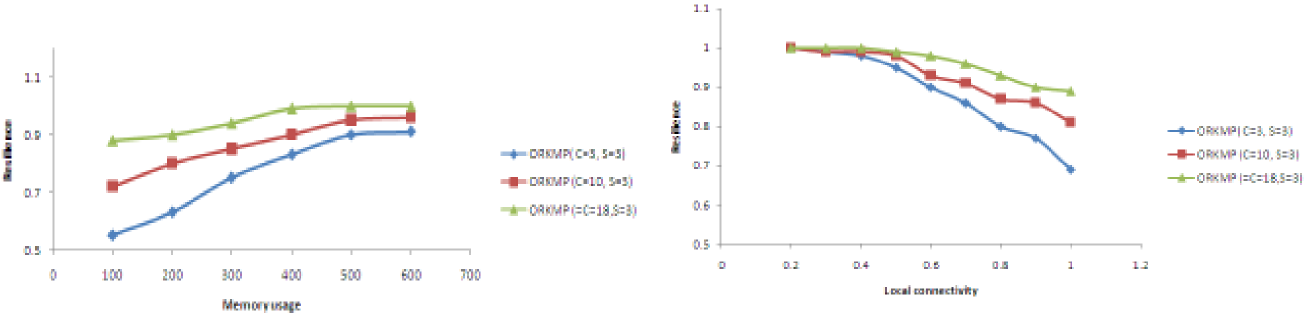


Fig. 8 (a)

Fig. 8 (b)

(a) Resilience *Vs* memory usage, No. of nodes compromised= 200, Local connectivity = 0.5, Attack circle radius = 250.
(b) Resilience *Vs* local connectivity, No. of nodes compromised= 200, key chain size = 200, Attack circle radius = 250.

4.4 Communication overhead

In practical, achieving 100 percent network local connectivity is impossible. So there are some nodes which are not able to establish a secure direct communication path between source and destination. These nodes need to undergo the path key establishment phase in which a path is established using more than one hop between source and destination. As the number of hop increases for path establishment, the communication overhead increases which leads to more energy consumption. Hence, the communication overhead depends on the local connectivity. When the local connectivity increases, i.e. secure connection established using single hop increases, the communication overhead decreases. The performance comparison of communication overhead for the different key ring sizes of EG, PKK, EG-D and ORKMP ((*C,S*) = (3,3), (10,3) and (18,3)) schemes is shown in Fig 9. From the Fig 9, it is observed that ORKMP outperforms when C increases, compared to the other schemes even for small key ring size.

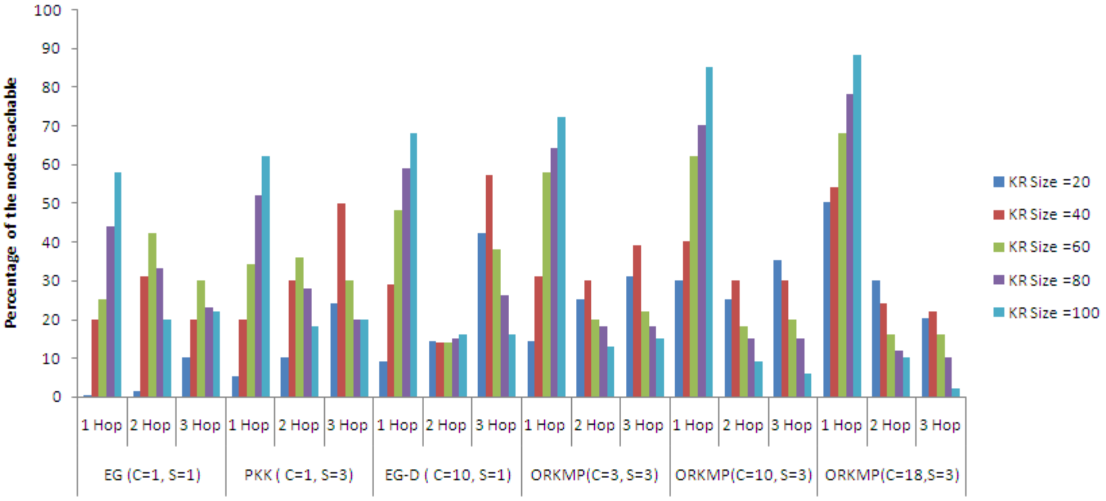


Fig. 9: Performance comparison of Communication overhead

5.0 CONCLUSIONS

In this paper, we have described the hybrid attribute based key management protocol design. The proposed protocol considered the clustered heterogeneous wireless senor network with certain degree of mobility. The proposed key management protocol made use of the probabilistic approach that exploits the deployment knowledge and the sensor node attributes. The kind of network considered, nature of the key pre-distribution used, knowledge of the deployment, and the sensor node attributes, all together lead to an excellent performance. This ORKMP provides high connectivity without compromising resilience and memory requirement which in-turn decreases the communication overhead and resource requirement. In future, we plan to adopt the dynamic mobile cluster head in ORKMP and study how the performance of ORKMP is influenced.

REFERENCES

[1] Boushra Maala and Yacine Challal and Abdelmadjid Bouabdallah. "HERO: Hierarchical kEy management pRotocol for heterOgeneous wireless sensor networks". *in IFIP International Federation for Information Processing. Wireless Sensor and Actor Networks II*; Ali Miri; (Boston: Springer), Vol 264, 2008, pp. 125–136.

[2] D. W. Carman. "New directions in sensor network key management". *in International Journal of Distributed Sensor Network*. vol. 1, no. 1, Jan. 2005, pp. 3–15.

[3] H. Chan, A. Perrig, and D. Song. "Random Key Pre-distribution Schemes for Sensor Networks". *in IEEE Symposium on Security and Privacy.* May 2003, pp. 197–213.

[4] I.F. Akyildiz, W. Su, Y. Sankarsubramaniam and E. Cayirci. "Wireless Sensor Networks: a survey". *Computer Networks*, Vol. 38, March 2002, pp. 393-422.

[5] Korporn Panyim and Prashant Krishnamurthy. "A Hybrid Key Predistribution Scheme for Sensor Networks Employing Spatial Retreats to Cope with Jamming Attacks". *in Journal of Mobile Network and Application*, 2010.

[6] L. Eschenauer and V. D. Gligor. "A Key-Management Scheme for Distributed Sensor Networks". *in Proceedings of 9th ACM Conference on Computer and Communication Security,* Nov. 2002, pp. 41–47.

[7] V. Mhatre, C. Rosenberg. "Homogeneous vs heterogeneous clustered sensor networks: a comparative study". *in proceedings of IEEE International Conference on Communications*, Vol.6, 2004, PP: 3646 – 3651.

[8] H.T.T. Nguyen, M.Guizani, Jo Minho, Eui-Nam Huh. "An Efficient Signal-Range-Based Probabilistic Key Predistribution Scheme in a Wireless Sensor Network". *in IEEE Transactions on Vehicular Technology, I*, Vol. 58, Issue 5, Jun 2009, pp.2482 – 2497.

[9] Patrick Traynor, Raju Kumar, Heesook Choi, Guohong Cao, Sencun Zhu, and Thomas La Porta. "Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks". *in IEEE Transactions On Mobile Computing*, Vol. 6, No. 6 June 2007, pp. 633 -677.

[10] Park.J, Kim.Z, and Kim.K. "State-Based Key Management Scheme for Wireless Sensor Networks". *in proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems*. 2005.

[11] Rajgopal Kannan, Lydia Ray, Arjan Durresi and S. S. Iyengar. "Security-Performance Tradeoffs of Inheritance based Key Predistribution for Wireless Sensor Networks". *arxiv.cs*, June 2004.

[12] S.-P. Chan, R. Poovendran, and M. T. Sun. "A Key Management Scheme in Distributed Sensor Networks Using Attack Probabilities". *in Proceedings of IEEE GLOBECOM*, Nov 2005.

[13] Takashi Ito, Hidenori Ohta, Nori Matsuda, andTakeshi Yoneda. “A Key Pre-Distribution Scheme for Secure Sensor Networks Using Probability Density Function of Node Deployment”. *SASN05*. November 7, 2005.

[14] W. Du, J.Deng, Y.S.Han ,PK.Varshney “A key predistribution scheme for sensor networks using deployment knowledge”. *in IEEE Transaction on Dependable Secure Computing,* vol. 3, No.1, 2006, pp.62–77.

[15] W. Heinzelman, A. Chandrakasan and H. Balakrishnan. “An Application Specific Protocol Architecture for Wireless Micro sensor Networks”. *in IEEE Transactions on Wireless Communications*, Vol.1, No. 4, October 2002.

[16] N.Mittal, R.Novales, "Cluster-Based Key Predistribution Using Deployment Knowledge," *in IEEE Transactions on Dependable and Secure Computing,* vol.7, no.3, 2010, pp.329-335.

[17] K. S. Prabh. “Real-Time Wireless Sensor Networks”. *Ph.D. Thesis, Department of Computer Science, University of Virginia*, Charlottesville, VA, 2007.

[18] Mohammad AL-Rousan, A. Rjoub and Ahmad Baset, “A Low-Energy Security Algorithm for Exchanging Information in Wireless Sensor Networks”, *in Journal of Information Assurance and Security*, Vol 4, pp.48-59, 2009.

[19] I. F. Akyildiz and I. H. Kasimoglu, “Wireless Sensor and Actor Networks: Research challenges”, *in journal of Ad hoc Networks (Elsevier),* 2004.

[20] Albert Levi, Sinan EmreTaşç, Young Jae Lee, Yong Jae Lee and Ersoy Bayramoğlu, “Simple, extensible and flexible random key predistribution schemes for wireless sensor networks using reusable key pools”, *in Journal of Intelligent Manufacturing*, Vol. 21, no. 5, 2010, pp. 635-645.

[21] W. Bechkit, Y. Challal, A. Bouabdallah, and A. Bencheikh. “An efficient and highly resilient key management scheme for wireless sensor networks”, *in proceedings of IEEE 35th Conference on Local Computer Networks*, 2010, pp. 216 - 219.

[22] E. Khan, E. Gabidulin, B. Honary, and H. Ahmed, “Matrix-based memory efficient symmetric key generation and pre-distribution scheme for wireless sensor networks”, *in Wireless Sensor Systems, IET* , vol.2, no.2, 2012, pp.108,114.

[23] W. Bechkit, Y. Challal, A. Bouabdallah, and V. Tarokh, “A Highly Scalable Key Pre-distribution scheme for Wireless Sensor Networks”, *in IEEE Transactions on Wireless Communications*, vol. 12, no. 2, 2013, pp. 948 – 959.

[24] Jiri Kur, Vashek Matyas, and Petr Svenda. “Two Improvements of Random Key Predistribution for Wireless Sensor Network”, *in Secure Communication 2012, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Vol. 106, 2013, pp. 61-75.

[25] R.G. Raj, V. Balakrishnan. “A Model for Determining The Degree of Contradictions in Information”. Malaysian Journal of Computer Science, 2011, 24(3): 160 – 167.

Authors:

T.Kavitha pursuing her Ph.D in the Faculty of Information and Communication Engineering, Anna University Chennai, India. She received her M.E. degree in Systems Engineering and Operations Research from Anna University, Chennai India in the year 2006. B.E. in Electronics and Communication Engineering from Bharathidasan University, India in the year 2000. In 2000 she joined in the department of Electronics and Communication Engineering as a lecturer in Jerusalem College of Engineering. Presently she is working as an Assistant professor in Jerusalem College of Engineering. Her fields of interests are Wireless Networks, Wireless Sensor Network and information security, etc. She has published 12 papers in national/International conferences and 4 in International Journals. She is a life member of ISTE, 2011.

Dr.D.Sridharan received his Ph.D degree in the Faculty of Information and Communication Engineering, Anna University, Chennai, India in 2005. M.E. degree in Electronics Engineering from Madras Institute of Technology, Anna University, Chennai, India in the year1993. B.Tech. degree in Electronics Engineering from Madras Institute of Technology, Anna University in the year 1991.He is currently working as Associate Professor in the Department of Electronics and Communication Engineering, CEG Campus, Anna University, Chennai, India. He was awarded the Young Scientist Research Fellowship by SERC of Department of Science and Technology, Government of India. His present research interests include Internet Technology, Network Security, Distributed Computing, VLSI for wireless Communications and Wireless Sensor Networks. He has published more than 30 papers in National/International Conferences and Journals.