A Location Dependent Connectivity Guarantee Key Management Scheme for Heterogeneous Wireless Sensor Networks

Kamal Kumar,
M.M. Engineering College Mullana, Ambala, Haryana, India.
kkmishra76@yahoo.co.in

A. K. Verma
Thapar Institute of Engineering and Technology
Patiala, Punjab, India
akverma@thapar.edu

R.B. Patel
M.M. Engineering College Mullana,
Ambala, Haryana, India.,
patel_r_b@yahoo.co.in

***Abstract* – Wireless sensor networks pose new security and privacy challenges. One of the important challenges is how to bootstrap secure communications among nodes. Several key management schemes have been proposed. However, they either cannot offer strong resilience against node capture attacks, or requires too much memory for achieving the desired connectivity. In this paper, we propose a LOcation dependent Connectivity guarantee Key management scheme for heterogeneous wireless sensor networks (LOCK) without using deployment knowledge. In our scheme, a target field is divided into hexagon clusters using a new clustering scheme crafted out of nodes's heterogeneity. Even without using deployment knowledge, we drastically reduce the number of keys to be stored at each node. A pair-wise, group wise and cluster key can be generated efficiently for among nodes. LOCK provides dynamicity by two ways; one by not completely depending upon pre deployed information and other by not completely depending upon location. Compared with existing schemes, our scheme achieves a higher connectivity with a much lower memory requirement. It also outperforms other schemes in terms of resilience against node capture and node replication attacks. Scheme is proved to support largest possible network using smallest storage overhead as compared to existing key management schemes.**

***Index Terms* – Deployment, Heterogeneous, Connectivity, Geographical Group**

I. INTRODUCTION

Wireless sensor networks (WSNs) are commonly used in ubiquitous and pervasive applications such as military, homeland security, health-care, and industry automation. WSNs consist of numerous small, low-cost, independent sensor nodes, which have limited computing and energy resources. Secure and scalable WSN applications require efficient key distribution and key management mechanisms.

These systems have traditionally been composed of a large number of homogeneous nodes with extreme resource constraints. This combination of austere capabilities and physical exposure make security in sensor networks an extremely difficult problem. Because traditional asymmetric encryption is not practical in this environment, a number of clever symmetric-key management schemes have been introduced. One well received solution that has been extended by several researchers is to pre-distribute a certain number of randomly selected keys in each of the nodes throughout the network [9], [4], [7], [16]. Using this approach, one can achieve a known probability of connectivity within a network. These previous efforts have assumed a deployment of homogeneous nodes and have therefore suggested a balanced distribution of random keys to each of the nodes to achieve security. Likewise, the analysis of those solutions relies on assumptions specific to a homogeneous environment. A deviation from the homogeneous system model has been increasingly discussed in the research community. Instead of assuming that sensor networks are comprised entirely of low-ability nodes, a number of authors have started exploring the idea of deploying a heterogeneous mix of platforms and harnessing the available "microservers" for a variety of needs. For example, Mhatre et al. [1] automatically designate nodes with greater inherent capabilities and energy as cluster heads in order to maximize network lifetime. Traynor et al. [32] extend this idea to mobile groups by having a more powerful node perform group handoffs for neighboring sensors.

In this paper, we propose LOCK without using deployment knowledge. In our scheme, a target field is divided into hexagon clusters using a new clustering scheme crafted out of nodes's heterogeneity. Even without using deployment knowledge, we drastically reduce the number of keys to be stored at each node. A pair-wise, group wise and cluster key can be generated efficiently for among nodes. LOCK provides dynamicity by two ways; one by not completely depending upon pre deployed information and other by not completely depending upon location. The rest of the paper is organized as follows. Section II and Section III discusses clustering approach. In section IV LOCK has been the proposed network's model, network deployment and discussed with section V discussing performance related issues. Finally concluded in section VI.

doi:10.4304/jait.1.3.105-115

II. NETWORK ELEMENTS

Basically, two architectures are available for wireless networks, distributed flat architecture and hierarchical architecture. The former has better survivability since it does not have a single point of failure, and the latter provides simpler network management, and can help further reduce transmissions. As we know, WSNs are distributed event-driven systems that differ from traditional wireless networks in several ways such as extremely large network size, severe energy constraints, redundant low-rate data, and many-to-one flows. It is clear that in many sensing applications, connectivity between all Sensor Nodes (*SNs*) is not required but some applications require explicit connectivity between every pair of nodes. Mostly wireless SNs merely observe and transmit data to those nodes with better routing and processing capabilities, and do not share data among themselves. Data centric mechanisms should be performed to aggregate redundant data in order to reduce the energy consumption and traffic load in WSNs (out of scope of our proposal). Therefore, the hierarchical heterogeneous network model has more operational advantages than the flat homogeneous model for WSNs with their inherent limitations on power and processing capabilities [11][12][13][8][12]. Moreover recent trend is towards secure connectivity between geographical neighboring nodes. This phenomenon requires of Group Key which is shared symmetric key among a group of neighboring nodes.

In this paper, we focus on large-scale WSNs with the same three-tier hierarchical architecture as in [2] [3]. *SNs* are divided into two categories namely H-Sensors and L-Sensors. H-Sensors are small number of *SNs* possessing higher memory, transmission range, multiple transmission ranges, processing power and battery life. Our network model has four different kinds of wireless devices on the basis of functionality; sink node/base station (*BS*), cluster head node (*CH*), Anchor Nodes (*AN*) and sensor node (*SNs*).

- *Sensor node* (*SNs*): Sensor nodes are L-Sensors which are inexpensive, limited-capability, generic wireless devices. Each *SNs* has limited battery power, memory size, data processing capability and short radio transmission range. *SNs* communicates with its, *CH* , *SNs* and *SINK* .
- *Cluster head node* (*CH*): Cluster head nodes are a kind of H-Sensors, have considerably more resources than the *SNs* . Equipped with high power batteries, large memory storages, powerful antenna and data processing capacities, *CH* can execute relatively complicated numerical operations and have much longer radio transmission range than *SNs* . *CHs* can communicate with each other directly and relay data between its cluster members and the sink node (base station). *SNs* which need to communicate with neighbors in neighboring cluster will relay its data through *CHs* .
- *Anchor Nodes (ANs*): Anchor Nodes are a kind of H-Sensors which have multiple power level for transmission. Thus *ANs* have capability to transmit in multiple ranges which can be changed at requirement. *ANs* are placed at triangular/Hexagonal points to realize a new clustering approach. We introduce a new clustering approach which divides the nodes into clusters of hexagonal shapes. This approach will classify our scheme into location dependent scheme but without using deployment knowledge.
- *Sink node/Base station* (*SINK* / *BS*): Sink node is the most powerful node in a WSN, it has virtually unlimited computational and communication power, unlimited memory storage capacity, and very large radio transmission range which can reach all the *SNs* in a WSN. Sink node can be located either in the center or at a corner of the network based on the application.

In our network model, a large number of *SNs* are randomly distributed in an area. A sink node/base station (*BS*) is located in a well-protected place and takes charge of the whole network's operation. After the deployment, *CHs* partition a WSN into several distinct clusters by using a clustering algorithm discussed ahead. Each cluster is composed of a *CH* and a set of *SNs* (distinct from other sets). *SNs* monitor the surrounding environment and transmit the sensed readings to their respective *CH* for relay. *SNs* may use multihop or single communication pattern for communication with *CHs* .

III. NETWORK DEPLOYMENT AND CLUSTERING APPROACH

SNs are large in number and have limited capabilities. *SNs* are deployed randomly in the field for deployment like can be dropped from an aircraft. *ANs* are placed uniformly and in controlled manner using a manned or unmanned deployment vehicle which is equipped with GPS system to connect with satellite to retrieve exact location for *ANs* . Using hexagonal/triangular deployment of *ANs* in the deployment field the network deployment field is roughly divided into hexagonal/triangular field using multiple transmission power levels of *ANs* . As shown in the Fig. 1 the lines in dark are transmission radius of *ANs* placed at triangular points. The higher is the transmission level larger is the transmission radius. For sake of convenience we approximated and drawn arc shaped lines by straight lines and thus resulting each field is subdivided into approximately triangular cells. Depending upon the number of *ANs* whose transmission ranges are aligned/covering the triangle completely, *SNs* in that triangular cell will receive the equivalent number of nonce, considering that each transmission level of a *AN* transmits an entirely different nonce. For e.g. Nodes in Blue Cluster receives Selected Nonce but from all from $AN5$ and N_{65} N_{66}, from $AN6$. *SNs* in other cells of same cluster receives nonce depending upon their location in the field. Further adjoining neighboring triangular Cells will form a Cluster and each cluster will be administered by *CH* . This process or step is followed

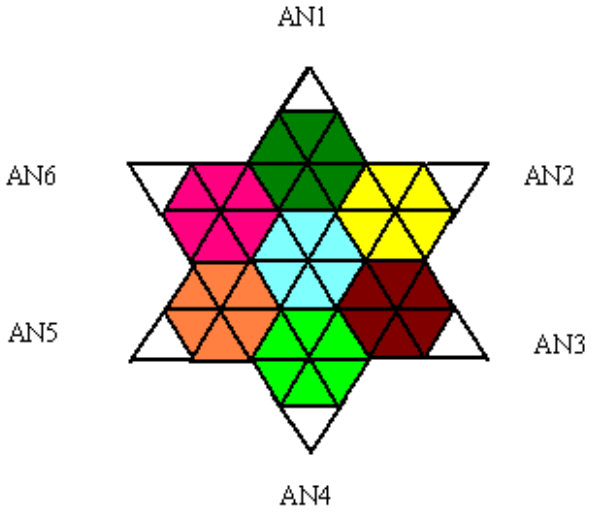

Figure 1: Hexagonal deployments of ANs and Resultant Hexagonal Clusters. For convenience the circular arcs are approximated as straight lines. Transmission ranges from closely placed Anchor Nodes at six corners intersect with each other and resulting into triangular shaped cells. Adjoining cells may be joined to give a hexagonal shaped clusters which are supposed to managed by cluster Head

by another controlled deployment using same GPS equipped vehicle, corresponding to H-Sensors which will work as *CHs*. Considering the placement of Nodes as shown in the Fig. 1, $AN1$, $AN2$, and $AN3$, $AN4$, $AN5$ and $AN6$ are able to transmit at different power level and thus can transmit in multiple ranges. We here assume that the Anchor Nodes are able to transmit at six power levels in Fig. 1.

IV. LOCK

A. Underlying Approach

In existing key pre-distribution schemes, two communicating sensors either use one or some of their shared pre-loaded keys directly as their communication key [15][9], or compose a pairwise key by their pre-loaded secret shares. Although this kind of mechanism has low computational overhead, it could lead to a serious security threat in practice. If some *SNs* are captured after the deployment, an adversary may crack some or even all the communication keys in the network by those compromised keys or secret shares. This node capture attack is the main threat to a key pre-distribution scheme. To address the limitations of existing key pre-distribution schemes, we propose to incorporate the location dependence with pre-distribution. Our proposal allows each pair of neighboring *SNs* has a unique pairwise key between them, which cannot be derived from the pre-oladed setup keys by other nodes. An adversary cannot crack the pairwise keys among non-captured *SNs*, even if some *SNs* are captured and their stored key information is compromised. Therefore, any *SNs* compromise can not affect the communication between non-compromised *SNs*.

B. Procedures in LOCK

Our proposed LOCK scheme has two phases,

(a) Setup keys assignment phase,
(b) Location dependent Keys Generation Phase

Key generation phase includes the generation of group, cluster and pair wise key between nodes. An off-line authority center called *SINK* is in charge of the initialization of the *SNs* in LOCK. Before deployment, each sensor node is assigned a unique *ID*; generated by *SINK*. Besides this each sensor node is also assigned the *IDs* of two *CH* which is assigned the part of the information required to generate pair wise key with its post deployment *CH*. *SINK* also generates a large size key pool P composed of more than 2^{20} distinct symmetric keys. For each sensor node SN_i, *SINK* randomly selects a secret key from P and stores it into SN_i s memory, this pre-loaded key is denoted as k_{SN_i-SINK}. k_{SN_i-SINK} is the shared pairwise key between node SN_i and the Sink node, and is be used to encrypt the exchanged data between the node SN_i and *SINK*.

Setup Key Assignment Phase: Before *SNs* are deployed, setup keys need to be pre-loaded into them in a certain way to ensure any two nodes can find some common keys after the deployment. Besides this each sensor node is also assigned *IDs* of two *CH* which is assigned the part of the information required to generate pair wise key with its post deployment *CH*. For each SN_i, *SINK* randomly selects some keys from P and pre-loads them into the intended $SN's$ memory. In our scheme, this pre-loaded information is named as network setup keys. Besides these a common key K is preloaded as a common setup key into the memory of each *SN*. To ensure any two *SNs* share some keys after deployment, depending upon its location beside common key K, we use a simple but efficient setup key assignment method for Heterogeneous Wireless Sensor Networks(*HWSN*).

Suppose there are n *SNs* in the network. First, *SINK* randomly selects n distinct keys from key pool P and constructs a two-dimensional ($m \times m$) matrix M, where "$m = \sqrt{n}$". Fig. 2 illustrates an example of the constructed key matrix M, in which each entry is a symmetric key with a unique two-dimensional id denoted by "$k_{i,j}(i, j = 1,2,...,m)$." For convenience, we use R_i and "$C_j\ (i, j = 1,2,...,m)$" to represent the i^{th} row and the j^{th} column in M, respectively. An equivalent representation of the Matrix M is given in Fig. 4, where nodes in black represent the diagonal entries of the matrix and also root of the Dual skewed Hash Binary Tree (DHBT) a modification of Hash Binary Tree in Fig. 3. Root can be used to derive node's left skewed branch and right skewed branch. For e.g. $k_{3,3}$ can be used to derive the row 3 completely. Similarly all the diagonal elements of the matrix M. Besides these each *SN* is informed a number N, such that $N = 2t$ with "$t(1 \le t \le m)$" values of which t values represents row numbers and remaining t values represents column numbers, assigned to *SNs* by their post deployment *CH* (Deployment of *CHs* is discussed in previous section). Before we can generate the complete rows of the matrix we need to customize the key matrix with respect to *SN's* Location. To customize and to make the scheme location dependent, the diagonal elements of a *SN* in a cluster in conformance to its administered cluster and corresponding geographical location; *CHs* computes the common content of the broadcast received by all constituent cells and sends the common broadcast vector to each node in its cluster using a plain broadcast message or by encrypting using cluster key.

Equation (1) is used to customize the diagonal elements in M, where $K_{i,i}^{j}$ is the customized diagonal element of ith row and *ith* column with respect to location of *jth* cluster. $COMM_j$ is defined as common content received by each *SNs* node in the jth cluster and is defined as "$COMN_j = K_1 \oplus K_2 \oplus K_3 \oplus ...$" where K_1, K_2 etc are nonce/keys shared by all the nodes in the jth cluster (CH_j). $COMM_j$ is a vector and is informed by the cluster head to each node in its broadcast.

$$K_{i,i}^{j} = H_{COMM_j}(K_{i,i}) \tag{1}$$

Now each node is provided with localized keys which represent the diagonal elements of the Key Matrix K. Next we propose to use Dual Skewed Hash Binary Tree (DHBT)where left or right branch can be generated using hash of left shifted value or right shifted value. The diagonal elements are considered as roots of these DHBT. Applying the procedure repeatedly results in generation of complete Key Matrix M where hash function is considered to be hardwired in *SN*.

Now consider the network setup keys pre-loaded in a *SN* when $t = 2$. In our case, each SN stores only m instead of m^2 or keys in its memory. This is alternative to use t rows and t columns thus $2 \times t \times m$ values in the storage [10]. This is where our scheme performs better in terms of memory requirements as it requires only m keys in the memory. For higher values of t this saving in memory requirements are even higher. For higher values of t this memory requirement shoots up exponentially [10] and thus our scheme offers a memory efficient approach for establishing pair wise keys in HWSNs. Any two *SNs* share at least $2t^2$ common keys in their memories, therefore, our setup key assignment which is deployment knowledge independent but location dependent in manner compared to procedure in [10] but still guarantee the connectivity between any two nodes in the network. Compared to scheme proposed in [6] our scheme ensures 100 percent connectivity among nodes of WSN. So, compared with existing key pre-distribution schemes, our approach is the first one to support full network connectivity without any prior deployment information and no matter how the *SN* are deployed and offers higher memory efficiency and computational efficiency, which are the main contribution of our proposed scheme.

Key Generation Phase: This phase includes the procedures for generation of Inter Cluster, Administrative keys, Cluster key and pair wise symmetric keys.

Inter Cluster Key Establishment $(K_{CH_a - CH_b})$: Each node is assigned a node *ID* by *SINK*. Provided CH_a and CH_b are the participating cluster heads, *CHs* can generate the pair wise key between them using (2) where sh_1 and sh_2 are shares of the symmetric keys exchanged between participating *CHs*

$$\left(K_{CH_a - CH_b} = H_K(sh_1 \oplus sh_2)\right) \tag{2}$$

Administrative Key ($K_{CH_i - SN_i}$) Generation: The nodes are preloaded with a symmetric key i.e. $K_{SN_i - CH_i}$ which can be used directly. *CHs* has to construct this pair wise symmetric key using the information stored in *SN*. Each *SN* is provided with *IDs* of two *CHs*. These *IDs* are sent to *CH* of the parent cluster. *CH* will receives the shares k_1, k_2 from two *CHs* whose *IDs* is sent to *CH* by *SN*. Equation (3) is used to set up $K_{CH_i - SN_i}$, where K is preloaded common setup key.

$$K_{CH_i - SN_i} = H_K(k_1 \oplus k_2) \tag{3}$$

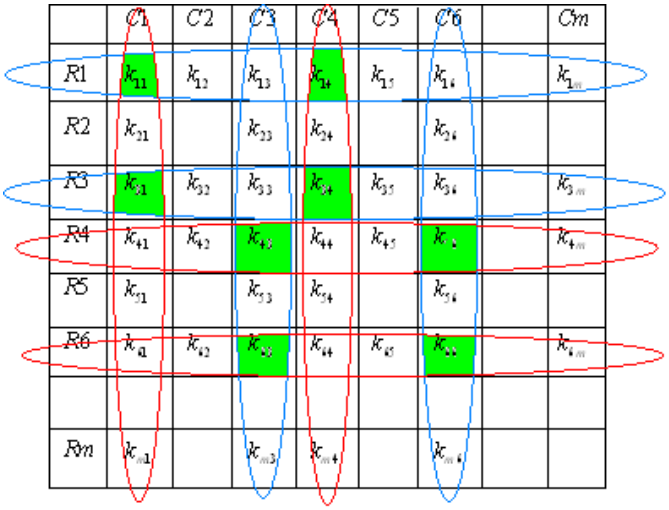

Figure 2: Setup key Matrix and Keys Assignment

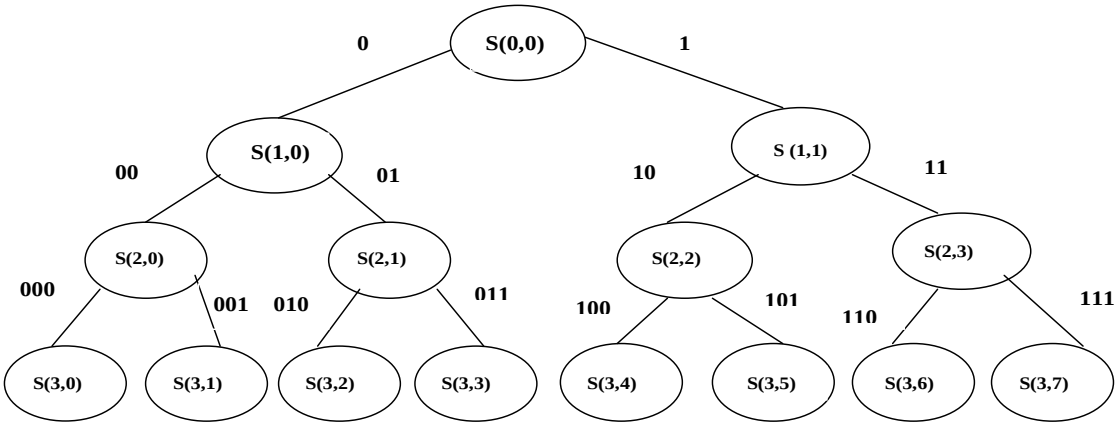

Figure 3: Hash Binary Tree. S(0,1) is obtained as Hash(leftShift(S(0,0))). Similarly S(1,1) is obtained as Hash(RightShift(S(0,0))). The complete HBT can be obtained in this manner and upto required height.

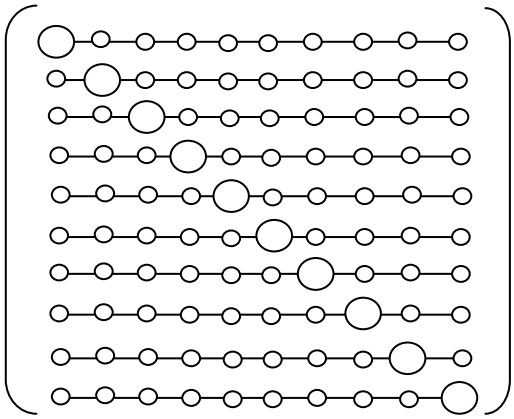

Figure 4: Dual Skewed Hash Binary Tree Representation of Key Matrix K

Pairwise Key Generation Phase: To secure the communication between two neighboring nodes, any SN needs to generate a pairwise key with each of its one-hop neighbors after the deployment.

In our proposed scheme, the pairwise key generation phase has three steps. First, node SN_i randomly selects "$l(1 < l < t)$" rows and l columns from its stored setup keys and SN_i generates a random nonce rn_i. Then, node SN_i broadcasts a handshaking message including its node ID_i, the random nonce rn_i, and indices of it selected rows and columns to its one-hop neighbors. After two neighboring nodes exchanged the handshaking message, they can generate a pairwise key using their shared setup keys and the random nonce. To explain the procedure clearly, we use an example to illustrate how two communicating nodes generate a pairwise key between them. Suppose nodes SN_a and SN_b are two neighboring SNs after the deployment. As shown in Fig. 2, SN_a has been pre-loaded the $3rd$ and $6th$ columns, and the $1st$ and $4th$ rows indices of key matrix K in its memory, SN_b has the $1st$ and $4th$ columns, and the $3rd$ and $6th$ rows indices of key matrix K pre-loaded in its memory.

To establish a pairwise key between nodes under consideration, first SN_a generates a random nonce rn_a. Then, SN_a broadcasts a handshaking message "$\{SN_a, R_1, R_4, C_3, C_6, rn_a\}$" to node "$SN_b$". Similarly, SN_b generates a random nonce rn_b, and broadcasts "$\{SN_b, R_3, R_6, C_1, C_4, rn_b\}$"to node. After exchanging their handshaking messages, node SN_a obtains rn_b as well as its shared setup keys indices with "$SN_b\langle k_{1,1}, k_{1,4}, k_{3,3}, k_{6,3}, k_{4,1}, k_{4,4}, k_{3,6}, k_{6,6}\rangle$" which are the intersections of the corresponding key rows and columns. Node SN_b also can get rn_a and the shared setup keys with SN_a at the same time. Now, nodes SN_a and SN_b can calculate a pairwise key between them by Equation (4):

$$pk_{N_a-N_b} = rn_a \oplus k_{1,1} \oplus k_{1,4} \oplus k_{3,3} \oplus k_{6,3} \oplus k_{4,1} \oplus k_{4,4} \oplus k_{3,6} \oplus k_{6,6} \oplus rn_b \cdots (4)$$

In (4), "$\oplus$" is the exclusive-or operator, $pk_{N_a-N_b}$ denotes the pair wise key between nodes SN_a and SN_b, rn_a and rn_b are two random nonce generated by SN_a and SN_b respectively.

In LOCK, each SN stores m diagonal keys from the constructed matrix M and t rows and t column indexes. Since each pair of row and column has an intersection entry between them, any two SNs can find $2t^2$ common keys after they exchange the handshaking messages, which means, any two SNs which are members of same cluster, within their radio transmission range, can directly setup a secure link without the third node's participation. In other words, the path-key establishment phase of existing key pre-distribution schemes is eliminated in our approach, which not only reduces the communication overhead, but also increases the security level of the generated pairwise keys. On the other hand, since each generated pairwise key is distinct to others, LOCK improves the network resilience against node capture attack. Further customizing the diagonal elements to a cluster results in strengthening the resilience against node capture attack as same node may never be used outside the cluster heads transmission range.

*Geographical Group Key Generation (k_{GoG}) :*Sensor nodes in the same geographical group i.e. triangular cell, can construct a group key k_{GoG} using the broadcast received from ANs and membership information obtained from CHs as follows:

$$K_{GoG} = H_{K_{CH_i}}(k_{11}, k_{12,}, ..., k_{ij}, ..., list\_of\_IDs) \ldots (5)$$

where k_{ij}'s are key broadcast from AN_i and transmitting at jth power or transmission level, $list\_of\_IDs$ is unique value obtained as a result of X-OR operation on the IDs of the nodes residing in a cell as defined in (6):

$$list\_of\_IDs = ID_{i,1} \oplus ID_{i,2} \oplus ... \oplus ID_{i,m} \quad (6)$$

where $ID_{i,j}$ is the jth Node's ID in ith cell assigned at pre-deployment stage by the $SINK$. $list\_of\_IDs$ is securely sent to the SNs using pair wise symmetric key i.e. $K_{CH_i-SN_i}$.

*Cluster Key Generation (K_{CH_i}):*Equation (7) can be used to generate K_{CH_i}:

$$K_{CH_i} = H_K(COMN_i) \quad (7)$$

where "$COMN_i = K_1 \oplus K_2 \oplus K_3 \oplus ...$" where K_1, K_2 etc are keys shared by all the nodes in the ith cluster (CH_i), and K is pre deployed in SNs as described earlier. Successive uses of Common keys is replaced by K_{CH_i} as K will be deleted after bootstrapping is over.

V. Security Analysis And Performance Evaluation

We analyze the security property and evaluate the performance of our proposed LOCK scheme in this section.

A.. Security Analysis

Node Replication Attack: Because of the unattended mode operation, some SNs could be physically captured

by an adversary during the operating period. Thus Node replication attack is a severe threat for WSNs due to its infrastructure less architecture. In [9], the pair-wise keys are directly used from the pre-loaded keys. After the network bootstrapping phase, if *SN* is captured and all its stored keys are compromised, the adversary can duplicate some malicious node and deploy them into the network to execute some attacks such as eavesdropping, Denial-of-Service (DoS), etc.

In LOCK the keys are not same throughout the operational life of the *SN* . Cluster key is updated as and when needed using most recent broadcast from the ANs. Geographical group key is updated using new and remaining list of nodes from cluster head and new broadcast from the ANs . Diagonal entries of the matrix got customized with respect to the corresponding cluster using the common part of the broadcast received by the nodes in the cluster head's coverage range. Moreover any pair of SNs has a unique pairwise key between them after network initialization phase, which can be used to authenticate the communicating parties mutually. Without the proper authentication, any stranger's packets will just be ignored. Consequently node replication attack can be totally prevented by our proposed scheme.

Resiliency against Node Capture Attack: Adversary can physically capture some SNs to compromise the secret information. Node capture attack is the most serious threat in WSNs.

The communication between non-captured nodes could be cracked even they are not physically captured. In [15], if each SN stores 200 keys in its memory and the probability that any two nodes share at least one common key is 0.33 and 50 nodes' capture could compromise 10% of the communication among the non-captured nodes. Although [9] claims that the network resilience can be improved if two nodes share at least $q(q > l)$ common keys to establish a secure link, it only works when the number of captured nodes is less than a critical value. When the number of captured nodes exceeds the critical value, the fraction of compromised communication among non-captured nodes increases even at a much faster rate than [15]. In LOCK, after the pairwise key generation phase, each pair of neighboring nodes have a unique pairwise key between them, hence any node's capture can not affect the secure communication between non-captured nodes. In other words, our approach can guarantee the communication security among non-captured nodes no matter how many *SNs* are captured by the adversary, which is one of the main contributions of our work. Fig.5 shows that above 30% of the communication between non-captured nodes are compromised in [9] when 200 nodes are captured; if the number of captured nodes increases to 500, more than 60% of the communication of the rest network will be compromised. On the contrary, no communication between non-captured nodes could be compromised in LOCK no matter how many SNs are captured by the adversary. Fig. 6 shows the LOCK cluster size supported. assorted. As a result of location dependence the network size supported is much larger than any existing key management scheme.

If we assume the number of clusters is 7 and the size of the network is almost 7 times of cluster, the network supported is drawn in Fig. 6. In LOCK, the maximum supported cluster size exponentially increases when the key ring size increases linearly, which means our proposed scheme has better scalability than any of the existing key pre-distribution schemes till date. Think of the network size that can be supported in our deployment. Random key pre-distribution key management scheme can support a network of size 200 nodes using 50 keys per node. Q-composite[4] key distribution is not much better than Random key pre-distribution key management scheme.

In LOCK the same matrix got localized and thus same equation which earlier denoted the size of equation which earlier denoted the size of the network, exploits the size of cluster. Moreover due to our proposal of storing only diagonal entries the memory requirements are even lesser. Network Connectivity: Random key pre-distribution schemes cannot guarantee any two SNs establish a pairwise key directly. To increase the network connectivity, intermediate nodes need to be involved in a path-key establishment procedure. Even so, based on probability theory, some SNs or some portions of a network are still possibly isolated from the network if no path-keys can be established.

LOCK can guarantee a completed network connectivity since any two SNs can find common setup keys between them, which is the second contribution of our work. Fig. 7 shows that LOCK can generate a connected network with only one-hop neighbors' information exchange. For random key pre-distribution schemes, two or three more hops neighbors need to be involved to setup an almost connected network, which not only reduce the security of the established pairwise key, but also produce more communication overhead in the network.

Communication Overhead: In random key pre-distribution schemes, each *SN* exchanges all of its stored key information with its neighbors. For a large-scale communication and memory storage overheads are produced in this procedure. LOCK guarantees any two nodes to establish a pairwise key directly; therefore, its communication overhead is much lower than the previous schemes.

C. Performance reviewed in the light of multiple Transmission levels of Anchor Nodes

Scheme proposed in previous sections has support for all three types of keys namely cluster key, Pair wise key and group key. Key refresh mechanism of cluster and pair wise keys is achieved with the help of periodic or event based broadcast from ANs. The information broadcast used by nodes in a cluster to generate cluster and pairwise keys. To measure the effect of number of power levels and radius of broadcast we reinvestigate the performance

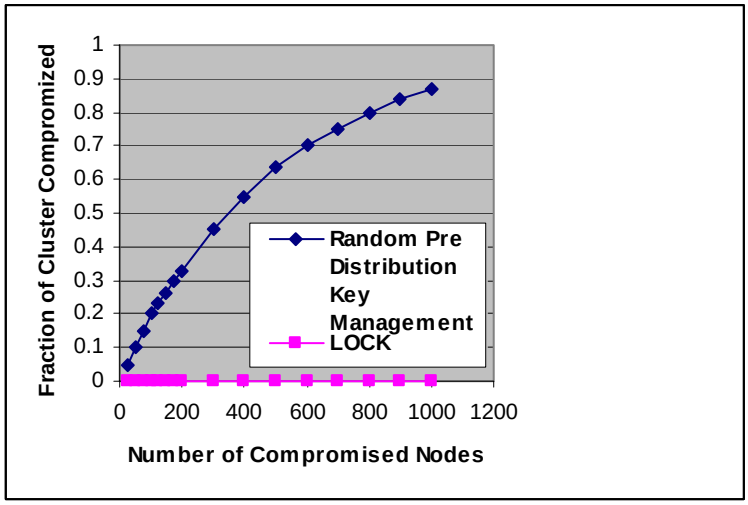

Figure 5: Fraction of Compromised Nodes

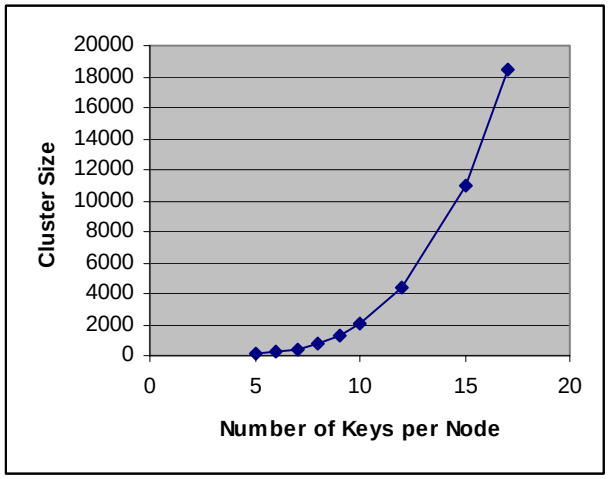

Figure 6: Cluster Size Supported

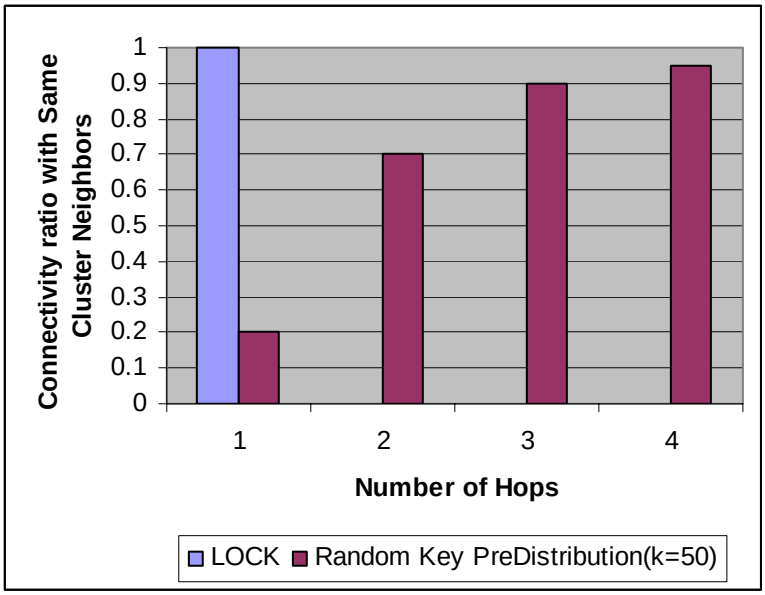

Figure 7: Network Connectivity vs. Number of hops needed for pair wise keys

and subject to various configuration and analyze the effect on memory and connectivity performance. We start by investigating the expected number of keys stored on each sensor node when using LOCK. This gives a measure of memory capacity of every sensor that needs to be devoted for LOCK while refreshing.

Location Dependence measurements: The number of keys stored on a sensor node is only momentarily contributed by the number of messages that a node receives from various *ANs* and almost completely by number of generation keys stored on each sensor node. Starting with memory required by the former factor. Each message contains a nonce which is then used to form or customize the pre-distributed keys with respect to its location in the deployment field and to derive a key used for geographical group and to derive a cluster key. After these uses the nonce obtained by the nodes are deleted.

Hence we need to store these keys momentarily and delete thereafter. Thus there is not major consumption on the memory of the individual nodes as a result of receiving broadcast from *ANs* . If we assume the memory consumption is contributed equally by former factor, then we need to determine the expected number of messages received by a sensor node. In order to this we divide the messages transmitted by each *AN* into N_P different categories, where N_P is the number of power levels on each *AN* . The messages transmitted at the ith power level are called $type-i$ messages. $Type-1$ correspond to the lowest power level while $type-N_P$ messages correspond to the highest power level. Therefore, when sensor node receives $type-i$ messages then it also receives messages of $type-j$ where "$j>=i$".

$$E_N = \sum_{j=1}^{N_P}\sum_{i=1}^{N_a} i(N_P - j + 1)\left(\frac{\rho_L \pi\left(R_j^2 - R_{j-1}^2\right)}{i!} e^{-\rho_L \pi\left(R_j^2 - R_{j-1}^2\right)}\right) \quad (8)$$

The equation (8) as derived in [5] represents the number of nonce which a node receives corresponding to the ith power level. Where radius R_i is the outer radius for annulus centered on SN under consideration with R_{i-1} inner radius with of the annulus, ρ_L indicates the density of AN deployment, N_P is the number of power levels N_a is the total number of anchor nodes in the network, area of annulus may be calculated by "$A_a = \prod\left(R_i^2 - R_{i-1}^2\right)$". All the nonce broadcast from *ANs* at higher transmission level is also received. Dependence on the average number of keys on each node. We further assumed the maximum transmission radius R_{N_P} of an *AN* . We want to determine the number of sub-keys would be needed to ensure high degree of location dependence Given this equation we analyze the effect of number of power levels and thus measure of location thereby reduce the memory requirement at pre-distribution stage.

For Lower values of R_{N_P} the degree of location dependence is very high and approaches to Lower levels for higher values of R_{N_P} . This phenomenon is attributed to the fact that with increased R_{N_P} we are able to cover more nodes by the same *AN* and thus the probability of having same diagonal contents shared among neighbors increases and thus lowers Location dependence. We can achieve desired connectivity but compromise ratio will have a boost as a result. To achieve connectivity ratio of 1 and low compromise ratio we are required to increase the size of matrix to achieve desired uniqueness in row and column assignment.

The memory requirement is dependent upon number of power levels. The reason to this issue is attributed to the behavior is that with even single power level the node in coverage of *AN* will receive one nonce/subkey. The pre-distributed contents and thereafter customized contents are same throughout and thus will require a large sized matrix and thus higher memory requirement at each *SN* to achieve desired connectivity at low compromise ratio.

With only one power level impact of compromised nodes is very severe. To avoid the effect on the compromise ratio we need to increase the size of matrix and thus memory requirement at each *SN* .

To achieve the uniqueness in pre distributed information we need extremely large sized key matrix. Thus memory requirement is heavenly dependent upon the number of power levels. This is because when using a single power level any node in the transmission range of an *AN* knows the of all secrets transmitted by the *AN* . When the number of power levels increases for the same value of R_{N_C} , the number of secrets of *ANs* known by the sensor node depends upon the distance of the sensor from *ANs* of interest.

Consider the case where all the intermediate power levels are eliminated. Then sensor in any region will receive all the secrets from all the transmitting anchor nodes. Thus compromise of any node in the region will jeopardize the communication of any other sensor node in the network unless we increase the matrix size. On the other hand by having three power levels for each *AN* , the nodes in any region will not receive all the secrets from *AN* . In such a case compromise of a node leads to a lesser number of secure links between non-compromised nodes being jeopardized. This effect is attributed to the fact that with the increase in the number of power levels the degree of location dependence increases thus causing a reduction in number of *SN* in each cell and thus in a cluster. This will reduce the size of matrix by many folds equivalent to the number of clusters obtained as a result of sectoring of network deployment field. This factor will continue to affect the memory requirement; in other words lower down the memory requirement unless each node in a separate cell. We can increase the number levels to a degree such that there should be at least two

nodes in each cell. Even at high degree of location dependence the connectivity ratio is 1 for every node in the same cluster. Beyond a threshold i.e each *SN* in a separate cell, this factor will not affect / have impact on the compromise ratio but reduces the connectivity ratio to 0. Thus the compromise ratio as well as connectivity ratio is sensitive only to very low and very high values of N_P .

To study the effects of density, number of power levels and transmission levels of *AN* on the compromise ratio. For a sensor node we consider only the density and maximum transmission range.

For a sensor node connectivity remains same and compromise ratio increase as the density of sensor is increased. This is because with the increase in sensor density there are more nodes that share the same customized diagonal entries. As more nodes are close by and thus able to connect with their neighbors, a node is able to set up secure links with more of its neighbors. In addition compromise of a node is still unaffected as long as we able to have uniqueness in row and column assignment. With the increase in the density the size of matrix may required be increased to bring uniqueness in row and column assignment and thus increasing the memory requirement at each node.

Further as the transmission radius of the sensor nodes is increased, the nodes have more neighbors and a node is able to communicate with a node only if they share commonly customized matrix. Moreover if node belongs to the neighboring cluster node will not be able to communicate. We have not considered such scenario, but of course will reduce linked node compared to potential neighboring nodes. The connectivity ratio on the other hand could be reduced. Increasing radius results in increasing the neighbors; some of which might not be sharing the same secrets; as new neighbors might not be covered by the same *ANs* as node concerned. Thus reducing the capacity of connecting to all the neighbors of a sensor node thus reducing connectivity ratio.

The compromise ratio on the other hand should not be affected. Changing the transmission range will not affect the number of non compromised nodes impacted due to compromise of any node. It is because a non-compromised node is impacted only when it shares keys with the compromised nodes. And sharing of keys is not governed by the transmission range of a sensor node. Increasing the transmission range might allow more number of non-compromised nodes to set up secure links and the fraction of these new links that are impacted cannot be predicted.

Increasing the number of power levels N_p on an *AN* while keeping the density of *ANs* as well as the maximum transmission range R_{N_P} the same also does not affect either the connectivity ratio or the compromise ratio.

Increasing the density of *ANs* without changing either N_p or R_{N_P} has positive impact on both the connectivity ratio or compromise ratio. This is because by increasing the number of *ANs* more number of Sensor Nodes can receive the beacons/nonces which allow them to derive their own customized diagonals. This also has a positive impact on the compromise ratio by reducing the compromise ratio since location dependence increases with increase in the density of *ANs* . Increasing the maximum transmission radius of a *ANs* has negative impact on Location dependence. This is because by increasing the R_{N_P} more number of sensor nodes will receive beacons from the same *AN* . This makes it easier for neighboring nodes to share common diagonal. This will also result in increasing compromise ratio. Thus from above we can conclude that the *AN* density has to increased while ensuring that both N_p as well as R_{N_P} are not large in order to reduce the impact of compromised nodes. But this could increase the cost associated with the deployment. If compromise of nodes can be tolerated then the system can deploy a low density of *ANs* with large transmission range and fewer power levels.

VI. Conclusion And Future Works

With the proposal above we are able to highlight the effect of Heterogeneity on the performance of Key Management Scheme in Wireless Sensor Networks. We considered a special kind of heterogeneity i.e. Number of Power levels and were able to draw the effect on performance in terms of memory requirements and size of the network supported by LOCK. Average number of nuances/keys received depend not only the maximum transmission radius but also on number of power levels. Although some issues like simulation results etc still needs be addressed but hopefully in our next work we come out with better results on the issue. Future scope lies in making this scheme scalable with respect to new node addition , routing aware and thus achieve secure communication. We have not considered much of inter-cluster communication model among the Sensor nodes thus open challenge.

VII. Refernces

[1] Mhatre, V. P., Rosenberg, C., Kofman, D., Mazumdar, R., and Shroff, N., “A Minimum Cost Heterogeneous Sensor Network with a Lifetime Constraint”, In Proceedings of IEEE Transactions on Mobile Computing 4, 1 (Jan. 2005), 4-15.

[2] M. Younis, M. Youssef, and K. Arisha, “Energy-Aware Routing in Cluster-Based Sensor Networks,” In Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002), 2002.

[3] K. Arisha, M. Youssef, and M. Younis, “Energy-Aware TDMA-Based MAC for Sensor Networks,” In Proceedings of the IEEE Workshop on Integrated Management of Power Aware Communications, Computing and Networking (IMPACCT 2002), May, 2002.

[4] H. Chan , A. Perrig, D. Song, “Random Key Predistribution Schemes for Sensor Networks”, In the Proceedings of the 2003 IEEE Symposium on Security and Privacy, p.197, May 11-14, 2003

[5] Anjum, F., "Location dependent key management using random key-predistribution in sensor networks", In Proceedings of the 5th ACM Workshop on Wireless Security (Los Angeles, California, September 29 - 29, 2006). WiSe '06. ACM, New York, NY, pp 21-30.
[6] Kausar Firdous, Sajid Hussain, Laurence Tianruo Yang, Masood Ashraf, "Scalable and efficient key management for heterogeneous sensor networks", In Journal of Supercomputing 45(1): 44-65 (2008)
[7] W. Du , J. Deng , Y. S. Han , P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks", In the Proceedings of the 10th ACM conference on Computer and communications security, October 27-30, 2003, pp. 42 - 51 , Washington D.C., USA
[8] B. Liu, Z. Liu, and D. Towsley, "On the capacity of hybrid wireless networks", In Proceedings of IEEE INFOCOM, April 2003, volume 2, pages 1543--1552, San Francisco, CA.
[9] L. Eschenauer and V. Gligor, "A Key Management Scheme for Distributed Sensor Networks", In Proceedings of the 9th ACM Conference on Computer and Communication Security, pp. 41-47, November 2002.
[10] Cheng Y., Aggarwal D.P., " An Improved key mechanism for large scale hierarchical wireless sensor networks" ,Proc. of Security Issues in Sensor and Adhoc Networks, Elsevier, Vol. 5(1), p.p. 35-48.
[11] P. Gupta and P. Kumar, "Internets in the sky: The capacity of three dimensional wireless networks", In Proceedings of Communications in Information and Systems, 1(1), pp. 33--50, 2001.
[12] S. Zhao, K. Tepe, I. Seskar, and D. Raychaudhuri, "Routing protocols for self-organizing hierarchical ad-hoc wireless networks," In Proceedings of IEEE Sarnoff 2003 Symposium, 2003.
[13] P. Gupta and P. R. Kumar, "The capacity of wireless networks," IEEE Trans. Inform. Theory, vol. 46, no. 2, pp. 388–404, Mar. 2000.
[14] Gang Zhou, Chengdu Huang, Ting Yan, Tian He, John A. Stankovic and Tarek F. Abdelzaher, "MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks," In Proceedings of IEEE INFOCOM 2006, Barcelona, Spain, April 2006.
[15] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," Computer 35(10):54–62, 2002.
[16] D. Liu and P. Ning, "Location-based pairwise key establishments for static sensor networks," in Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Security of Ad Hoc and Sensor Networks in Association with 10th ACM Conference on Computer and Communications Security, Fairfax, Va, USA, October 2003, pp. 72–82.

Kamal Kumar received his M.Tech. as well as B.Tech degree from Kurukshetra University, Kurukshetra, India. Presently he is working as Associate Professor in Computer Engineering Department in M.M. Engineering College, Ambala, India. He is pursuing Ph. D from Thapar University, Patiala, India.

A. K. Verma is currently working as Assistant Professor in the department of Computer Science and Engineering at Thapar University, Patiala in Punjab (INDIA). He received his B.S. and M.S. in 1991 and 2001 respectively, majoring in Computer Science and Engineering. He has worked as Lecturer at M.M.M. Engg. College, Gorakhpur from 1991 to 1996. From 1996 he is associated with the same University. He has been a visiting faculty to many institutions. He has published over 80 papers in referred journals and conferences (India and Abroad). He is member of various program committees for different International/National Conferences and is on the review board of various journals. He is a senior member (ACM), LMCSI (Mumbai), GMAIMA (New Delhi). He is a certified software quality auditor by MoCIT, Govt. of India. His main areas of interests are: Programming Languages, Soft Computing, Bioinformatics and Computer Networks. His research interests include wireless networks, routing algorithms and securing ad hoc networks.

R. B. Patel received a PDF, Highest Institute of Education, Science & Technology (HIEST), Athens, Greece, 2005. He received a PhD in Computer Science and Technology from Indian Institute of Technology (IIT), Roorkee, India. He is member IEEE, ISTE. His current research interests are in Mobile and Distributed Computing, Security, Fault Tolerance Systems, Peer-to-Peer Computing, Cluster Computing and Sensor networks. He has published more than 100 papers in International Journals and Conferences and 17 papers in national journal/conferences. Two patents are also in the credits of Dr. Patel in the field of Mobile Agent Technology and Sensor Networks.