CHAPTER 5: MULTI TIER & MULTI CLUSTERED KMS FOR WSN USING $LL^T$

5.1 INTRODUCTION

WSN is a collection of autonomous sensor nodes placed spatially [3]. Unlike wired networks the nodes here are subject to resource constraints such as memory and power. To ensure security [21] high level encryption is necessary. The strength of encryption depends on the type of key used. Existing KMS for cluster-based hierarchical architecture [22] either supports group communication or node-to-node communication. The proposed KMS supports both group and node-to-node communication using $LL^T$ matrix. The main objective of the scheme is to strengthen the data transferring security mechanisms [20] and also to ensure efficient key generation and management along with authentication. The main feature of this proposed scheme is 100% local connectivity; efficient node revocation methodology, perfect resilience; three-level authentication cum key generation and the most importantly reduced storage and computation. The scheme and its detailed performance analysis are discussed in this chapter.

5.2 PROPOSED ARCHITECTURE OF THE KMS

5.2.1 Architecture

The four layered clustered architecture [23] [51] is proposed which is shown in Figure 5.1 and comprises of

- Base station at the top level
- Cluster heads at the second level
- High end sensors as the group head in the third level
• Low end sensor nodes at the bottom level

![Layered Clustered Architecture - WSN](image)

Figure 5.1 Layered Clustered Architecture - WSN

The multi-layered clustered architecture gives a clear idea of load, labour, number and types of nodes at each level. All the nodes in the system need not perform the same functions. The load can be optimally divided among them depending on what level they reside. Thus it provides division of labour system and also tells what types of node to be used at what level. The main objective [53] in WSN is to achieve 100% connectivity at low power consumption. More transmission power is required for long range communication, so usually transmission power is traded off with communication range. Typically, WSN, nodes are expected to work efficiently at low power consumption. Thus at low power supply the communication range of the nodes is cut down. The order of memory, processing speed and power decreases from base station to ordinary node. This entire process can be seen as an inverted triangle with
the base station at the base of triangle and the processing node at the apex as shown in the Figure 5.2.

![Inverted Triangle Schematic of WSN](image)

**Figure 5.2 Inverted Triangle Schematic of WSN**

In the proposed scheme low end sensor nodes are grouped under high end group [54] heads (nodes). If one node in a group wants to communicate with the node [76] in the other group multi-hopping is done via their respective group heads. By doing this data is more secured as the receiving node knows the source of the message, also there always exists exactly two nodes (the respective group heads) in between the sender and the receiver.

Thus the proposed scheme uses the “optimized connectivity with minimal power consumption” as the criterion for grouping nodes [55]. That is groups are formed based on the transmission power required for communication between them, i.e if two nodes requires more power for communicating then they both are placed under two different groups. And then they communicate by hopping through their group heads, Group heads being more powerful is supposed to consume less power for long range communication. The same criterion is followed for clustering group heads under powerful cluster heads. The cluster heads communicates directly with the base station.
A key is required for secure communication between two nodes in a network. A positive definite Hermitian matrix is decomposed into its lower and upper triangular matrices using CHOLESKY factorisation [56] [57]. Rows of the lower triangular matrices and columns of the upper triangular matrices are stored in each node, so that when two nodes communicate they transmit their column values and perform a simple matrix multiplication. This is yields two identical values because $L \times U = A$ (a positive definite Hermitian [56] of symmetric matrix). Multiplying one row with one column the key can be obtained. Since upper triangular matrix is the transpose of the lower triangular matrix it is necessary to store only the rows of the lower triangular matrix thus reducing the storage to half that stored in other matrix based KMS.

**5.2.2 Scheme Outline**

All the sensor nodes [36] are loaded with the KMS programs before deployment. Based on the locality of deployment, the sensor nodes are grouped under high end sensor nodes. Further the groups are clustered (depending upon the structure of deployment) under cluster head. Thus the knowledge about locality of sensor nodes is known in advance.

The base station is fed with the details of all sensor nodes, group as well as cluster heads such as number and IDs of all nodes belonging to a group; number and IDs of all group heads belonging to a cluster; number of clusters and ID of each cluster head.

Lower Triangular matrices decomposed from a symmetric matrices [36] form the basis of key generation. The symmetric decomposition is done using CHOLESKY Factorization.
CHOLESKY Factorization:

A positive definite symmetric matrix is decomposed into lower triangular matrix and its transpose using CHOLESKY factorization [56]. Let A be the symmetric matrix and by using CHOLESKY factorization $L$ & $L^T$ is obtained using the algorithm given below.

If order of the A matrix is N then,

1. Set $k = 1$

2. Repeat the following until $k \leq N$

3. For any $k^{th}$ value in generating L which is an $N \times N$ matrix the following 5 steps are executed:

   a) $a_{k,k} = \sqrt{a_{k,k}}$

   b) $a_{k+1:N,k} = a_{k+1:N,k} / a_{k,k}$

   c) $a_{k+1:N,k+1} = a_{k+1:N,k+1} - a_{k+1:N,k} * a_{k+1,k}$

   d) $a_{k+2:N,k+2} = a_{k+2:N,k+2} - a_{k+2:N,k} * a_{k+2,k}$ and so on

   e) Increment k by 1.

For illustration if a $4 \times 4$ symmetric matrix is assumed as shown below

$$
A = \begin{bmatrix}
1 & 1 & 4 & -1 \\
1 & 5 & 0 & -1 \\
4 & 0 & 21 & -4 \\
-1 & -1 & -4 & 10
\end{bmatrix}
$$

Then the L and $L^T$ matrix is obtained using the above described algorithm and the generated matrix is shown below
\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
1 & 2 & 0 & 0 \\
4 & -2 & 1 & 0 \\
-1 & 0 & 0 & 3
\end{bmatrix}
\begin{bmatrix}
1 & 1 & 4 & -1 \\
0 & 2 & -2 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 3
\end{bmatrix}
= 
\begin{bmatrix}
1 & 1 & 4 & -1 \\
1 & 5 & 0 & -1 \\
4 & 0 & 21 & -4 \\
-1 & -1 & -4 & 10
\end{bmatrix}
\]

Assuming there are a maximum of \( c \) clusters, \( g \) groups and \( n \) nodes. Then \( C \times C \) symmetric matrix is allotted for inter-cluster communication along with the base; \( c \) \( G \times G \) symmetric matrices for inter group (intra cluster) communication within a cluster; \( g \) \( N \times N \) symmetric matrices for inter node (intra group) communication. The trick of the trade is that values of order of the symmetric matrices are kept as large as possible. This is done to achieve better scalability. Here \( C \), \( G \) and \( N \) are slightly greater than \( c \), \( g \) and \( n \) respectively. This is to allow addition of new nodes in the future and also some of the rows of the matrices are allotted for group communication and node revocation.

Using separate sets of matrices for different layers of architecture, different sets of keys are generated for each layer. Each layer is a completed graph with \( m \) nodes (\( m \) is appreciably less than the order of the symmetric matrix allotted to it). For commanding purpose say from base station to cluster heads or from cluster heads to its group heads or from group heads to its nodes a unique key is generated for each group.

A key [75] array consisting of possible keys with which a node can communicate is stored in its memory. This ensures authenticated communication between nodes. A common substitution array for generating indices for the randomly generated pseudo random numbers is used for encrypting the message. Periodical checks are made by respective heads to test whether a node is alive or dead. Also the nodes have great resistance against tampering.
5.2.3 System Components and Functionalities

Base station is the master node of the network. It is at the topmost level of the architecture [47] [49]. It commands and controls all its co-ordinate nodes. It receives the aggregated data from various cluster heads and processes it. It stores cluster IDs, group IDs, number of clusters, number of groups in a cluster, number of nodes in a group along with their IDs. Further it stores one row of $C \times C$ matrix for establishing pair wise key between cluster heads for inter cluster communication; one common row from $C \times C$ matrix for broadcasting. It also stores key pool substitution array.

Cluster head [32] is a node and serves two purposes: one is that it reduces the burden of the base station by performing data aggregation and distribution of messages from/to various group heads; the other is that it aids inter-group communication by acting as a mediator. Moreover it initiates group head revocation. It stores cluster IDs, group IDs and number of groups under control. One row of $C \times C$ matrix for inter cluster communication; one row from the allotted $G \times G$ matrix for intra-cluster(inter-group) communication; the common row stored in base station (to receive message broadcasted by the base station) and one common row of $G \times G$ matrix for broadcasting (to group heads) purpose. Further it also stores the substitution array and key pool list.

High end sensor nodes / group heads is the node [22] that plays the role of cluster heads at this level, i.e it performs data aggregation and distribution of messages from/to its nodes. This also takes the role of initiating node revocation. It stores IDs of node belonging to it and IDs of group heads belonging to same cluster. One row of $G \times G$ matrix for inter group communications; one row of $N \times N$ matrix for intra-group (inter-node) communication; the common row of $G \times G$ matrix stored in it's cluster
head (to receive the message broadcasted by cluster head) and one common row of $N \times N$ matrix for broadcasting (to nodes) purpose. Further it also stores the substitution array and key pool list.

Lower end sensor nodes which is also called as a node is the working node of the system that senses and transmits sensed data to its group heads. The group ID, one row of $N \times N$ matrix for communicating with group head and the common row is stored. (to receive the message broadcasted by it). Further it also holds the substitution array and key list with two elements one the key value for communicating to group head and the other for receiving the broadcasted message.

Note: Different sets of matrices are dedicated to different cluster heads. Though the substitution array stored in the nodes is same for all, the key pool list varies in accordance with the matrices allotted to it.

### 5.3 CLUSTERING METHODOLOGY

![Figure 5.3 Four-Layered Multi-Clustered Structures](image)

Figure 5.3 Four-Layered Multi-Clustered Structures
The entire network is divided into many clusters under cluster heads as shown in Figure 5.3. This cuts down the communication range of nodes. The node that belongs to one particular cluster must use multi-hopping techniques [25] to communicate with a node in other cluster. This is a drawback as frequency of hopping is high. Thus in order to achieve inter-node communication and to reduce the burden of the cluster heads in performing hopping in addition to data aggregation, nodes in a cluster are further grouped under group heads. This further limits the communication range of low end sensors to a group. This strategy reduces the number of keys stored in the lower end sensors to a large extent. It is enough for a lower end sensor node to hold one unique key to communicate with its group head and one common key for group communication. In the proposed methodology the number of intra-group communication is expected to be less. But it can be achieved by establishing pair wise keys between the nodes in a group. In order to achieve inter-group communication, the keys stored in the cluster head are pair-wise.

### 5.3.1 Possible communications

- Base-Station → Cluster Head
- Cluster Head → Base Station
- Cluster Head → Cluster Head
- Cluster Head → Group Head
- Group Head → Cluster Head
- Group Head → Group Head
- Group Head → Low-End Sensors
- Low-End Sensors → Group Head

The Figure 5.4 explains the sequence in which commands are passed. This diagram depicts the behaviour of the system and its environment and also shows the
interaction of different system components arranged in a time sequence. The vertical dimension represent the time and the horizontal the system components. The vertical line is called the components life line. The life line of the component represents its existence during the interaction.

Figure 5.4 Communication Model of The Proposed Scheme

5.3.2 Message Format (every field is hashed)

(a) Intra-level Communication

Format (a.1). \text{RECEIVER ID} || \text{SENDER ID} || \text{ENCRYPTED ROW VALUE} || \text{INDEX(BASE)} || \text{INDEX(SHIFT)}

Format (a.2). \text{SENDER ID} || \text{RECEIVER ID} || \text{AUTHENTICATED BIT} || \text{ENCRYPTED ROW VALUE} || \text{KEY} || \text{INDEX(BASE)} || \text{INDEX(SHIFT)}

Format (a.3). \text{ENCRYPTED WITH KEY(SENDER ID} || \text{RECEIVER ID} \text{ AUTHENTICATED BIT} || \text{MESSAGE})

(b) Inter-level Communication
Format (b.1).  GROUP ID || ENCRYPTED ROW VALUE || GROUP KEY

|| INDEX (BASE) || INDEX (SHIFT)

Format (b.2).  GROUP ID || RECEIVER ID || AUTHENTICATED BIT ||

ENCRYPTED ROW VALUE || INDEX (BASE) || INDEX

(SHIFT)

Format (b.3).  ENCRYPTED WITH GROUP KEY (GROUP ID ||

AUTHENTICATED BIT || MESSAGE

(c) Node Revocation

Format (c.1).  RECEIVER ID || SENDER ID || ATTACKED NODE ID ||

NEW GROUP ROW || CHANGED HASH ARRAY

INDICES || INDEX(BASE) || INDEX (SHIFT)

The proposed scheme is a matrix based scheme. A positive definite (Hermitian) symmetric matrix [57] is decomposed into two matrices, such that one of them is lower triangular and the other upper triangular, this is mainly done to reduce storage and complex computation, because at least half of the matrix is filled with zero. The proposed scheme decomposes a symmetric matrix using Cholesky [56] factorization, which yields two matrices: lower triangular matrix and its transpose (upper triangular matrix). This reduces the storage almost to half of that consumed in LU scheme [36] as one need not store the column in each node. Thus it is enough if one row of lower triangular matrix is stored in each node. Since memory being expensive must not be utilized unnecessarily, that too, in lower end sensor nodes. For n nodes, the storage is of order \((n^2)\) i.e storage \((n) = (n/2)^{*(n+1)}\).

5.3.3 Pre-deployment Phase [49]: Considering the following scenario, let the number of clusters be c varying from 5 to 10; number of groups in a cluster be g
varying from 10 to 15; number of nodes in a group be n varying from 20 to 25. Thus one cluster matrix with 17 (11+1+5) rows, c group matrices with 22 (16+1+5) rows, g (for a cluster) node matrices with 32 (26+1+5) rows. All the above matrices are stored in the base station. All the parameters that are mentioned in the system and components phase are loaded to the appropriate nodes. In the proposed KMS the notations used are tabulated in Table 5.1.

Table 5.1 Notations Used

<table>
<thead>
<tr>
<th>Parameter List</th>
<th>Notations</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node ID</td>
<td>N_ID</td>
<td>Unique identity for a node</td>
</tr>
<tr>
<td>Group ID</td>
<td>G_ID</td>
<td>Unique identity for a group, mainly used in group communication</td>
</tr>
<tr>
<td>Cluster ID</td>
<td>C_ID</td>
<td>Unique identity for a cluster, mainly used in inter-cluster communication</td>
</tr>
<tr>
<td>Row values</td>
<td>R_n (node name) – for nodes</td>
<td>These are the rows stored in clusters or groups or nodes from the corresponding matrices allotted to them. These are used for key generation</td>
</tr>
<tr>
<td>Base</td>
<td>B</td>
<td>Pseudo-randomly generated number for base conversions. This is mainly used for converting individual row value in decimals to corresponding base and vice versa.</td>
</tr>
<tr>
<td>Shift value</td>
<td>S</td>
<td>Pseudo-randomly generated number for shifting purpose. This is mainly used for right shifting (while encryption) and left shifting (while decryption) the clubbed row values.</td>
</tr>
<tr>
<td>Substitution array</td>
<td>SA</td>
<td>It holds the indices for a specific range pseudo randomly generated numbers. Let the pseudo randomly generated numbers be between 80 to 110 and the hashing array for that be {008, 018, 028, 038……098, 009, 019,…….099, 001, 101, 201,…….901, 011}</td>
</tr>
<tr>
<td>Key pool list</td>
<td>KL</td>
<td>This holds the possible set of key values using which a node can communicate. Key lists are unique for unique nodes.</td>
</tr>
</tbody>
</table>
5.3.4 Key Establishment Phase [49]: After successful deployment of the nodes, establishing connectivity becomes the crucial step. This is done using keys. In simple words two nodes can communicate if and only if they share a common key.

1. Pair wise Key Establishment [34]: Steps involved in pair wise key establishment between two nodes:

- The sender node A sends its row \( R_n \) using format a.1 as mentioned in section 5.3.2 above.

- Node B receives the message and retrieves the row values of A. It computes the Key \( K_{AB} \) and checks it presence in the key pool. If it is present then the node B sends its row, checked bit, hash of the key using format a.2 mentioned in section 5.3.2.

- Node A receives the message and retrieves row values \( R_n \) of B, key value \( K_{AB} \), and computes the key value, \( K_{BA} \) using \( R_n \) and \( R_n \). Then it checks whether \( K_{BA} \) is present in its key pool and also whether \( K_{BA} \) matches with \( K_{AB} \). If it matches node A sends the message to B using the computed key using format a.3 mentioned in section 5.3.2.

A simple illustration for node-node communication:

Let \( b=97 \) be the (pre-deploy) pseudo randomly generated base; \( c=3 \) be the pseudo randomly generated shift number. Let the row value in A, \( R_n \) be \((4000\ -2000\ 1000\ 0)\) and that of B, \( R_n \) be \((-1000\ 0\ 0\ 3000)\).

STEP1: A sends \( R_n \) to B.

\[
(4000)_{by7}=04123\quad (-2000)_{by7}=12060\quad (1000)_{by7}=01030
\]

Row value is clubbed and then left shifted \( s \) times.
041231206001030 on right shifting become 231206001030041.

Let the hashing array index of base $b = 97$ be 79 and that of $s = 3$ be 30.

Let the shifted IDs of the sender node A and receiver node B to the base $b$ be $c1g1na$ and $c1g1nb$ respectively.

Thus A sends $c1g1nb c1g1na 2312060010300417930$ to B.

B receives and extracts node ID of A from $c1g1na$, row value of A 231206001030041, hashing indices of base-79 and shift-30 and its ID from $c1g1nb$. If the extracted ID matches with its ID then the following processes occur.

Now B right shifts the row value by $s$ digits which is obtained from the hashing index value of the last two digits. Thus the row values are 041231206001030. This is split into four numbers each with 5 digits.

Thus the row will be 04123 12060 01030

These values are converted to decimals using the base value retrieved from the third and fourth last bit of message received.

So the row values become 4000 -2000 1000 and the remaining values are zero.

This row values are multiplied with the row values of B, $R_n (-1000 0 0 3000)$

Thus the key value is -4000. This key value is checked from the key pool B for its presence. This is done as a step for authentication. If it is matched (here it is matched), then B sends the following message to A;

Let the base generated be 87 and shift be 5; let their hashing indices be 78 and 50;

\[( -1000)_{87} = 11143 \quad 0_{87} = 00000 \quad 0_{87} = 00000 \quad (3000)_{87} = 03442 \]
The row value is 1114300000000003442.

The shifted row value is 00000000000344211143

Hash value of key = \((-4000)_{87}\) = 14585

Let the shifted IDs of the sender node A and receiver node B to the base b be C1G1NA and C2G2Nb respectively.

Thus the message is: \(c1g1nac1g1nb50000000000034421114314585178\) is sent to A.

Now the node A receives the message and does the same decryption and obtains the row values of B, \(R_n (-1000 0 0 3000)\); key -4000, base 87 and shift 5.

These row values are multiplied with \(R_n\) and the key value is found. If it matches (here it matches, key = -4000) the node A checks it with its key pool and again if there is a match then A sends the message to B using the key in an encrypted manner.

This methodology is followed everywhere in the proposed scheme, i.e., at different layers of proposed scheme. This ensures partially pair wise key establishment (fully connected graph at each hierarchy level).

---

Let the node A be the sender and node B be the receiver.

**Sender A**

| ID_B||ID_A||Encrypted Row of A||index (b)||index(s) |

**Receiver B**

| ID_A||ID_B||Authenticated bit||Encrypted Row of B|| Key (AB)||Index(b)||Index(S) |

Encrypted with Key ( ID_A||ID_B||Authenticated BIT||MESSAGE)

---

*Figure 5.5 Models for Unicasting*
2. **Group Key Establishment:**

Group key [52] is mainly used for commanding and controlling the nodes. There is only one group key at any level. Here broadcasting technique is followed.

Steps involved in group key establishment:

1. The group head broadcasts the message to all its nodes using message format b.1 mentioned above.
2. Appropriate nodes receive the message and retrieve the necessary command.

The process of decryption is as same as the process depicted above in pair wise key establishment.

![Figure 5.6 Model of Multicasting](image)

3. **Cluster Key Establishment:** This process is same as group key establishment.
5.4 PERFORMANCE ANALYSIS

1. Key Connectivity:

Key connectivity [35] is a measure of possibility of communication between two nodes in a network. Global connectivity is a measure of connected components in the entire network. For system with high performance key connectivity should be high. This is because with high connectivity probability of multi-hopping reduces. This reduces unnecessary intermediate communications which in turn reduces the transmission power. Thus battery power (power source of sensor nodes) is reserved for processing and hence performance increases with key connectivity.

![Diagram of Performance and Security Parameters]

**Figure 5.7 Parameters for analysis**

In the proposed scheme, 100% key connectivity is achieved at each tier of hierarchy, i.e. the network is a completely connected graph at each level of hierarchy (completely pair-wise). The proposed network (structure) is a connected (not a fully connected) graph. As mentioned the connected components of the graph are fully connected. Generally, a lot of communication happens only within nodes of same level i.e the number of intra-level communications is more when compared to inter-level communication. Thus it is enough if 100% key connectivity is assured within a level
and inter-level communication can be achieved using secondary or ternary neighbours. The proposed scheme uses this strategy.

In Random Pair wise scheme [30 ] in order to reduce the key storage when compared to EG [28 ] scheme, the entire graph is divided into several overlapping connected components (nodes). Though this scheme does not support 100% connectivity but ensures the network is connected.

![Graph showing key connectivity vs number of nodes](image)

**Figure 5.8 Key Connectivity vs Number of Nodes**

Probability of key connectivity: $p = (1/n) \times m$; $P =$ probability of connectivity; $N =$ number of nodes $= 2000$; $M =$ degree of each node

Since all the nodes perform same tasks, frequent communication between them is required. Thus for two nodes, which are far apart, to communicate lot of hopping has to be done, this increases communication overhead. Thus high key connectivity is achieved at the expense of transmission Power.
In Asymmetric Pre-distribution scheme [32 ], the key connectivity is not 100% initially. Whenever two non-connected nodes want to communicate, they first establish a pair wise key between them with their first degree H sensor node. Thus key connectivity gradually reaches 100% at the expense of memory, i.e the storage memory in L sensor nodes increases.

\[
\text{Key Connectivity: } 1 - \left( \frac{(p-m)!(p-l)!}{p!(p-m-l)!} \right); \quad p = \text{pool size}; \quad m = \text{number of keys loaded in the node}; \quad l = \text{number of shared common keys};
\]

\[(m, l) = \{(30, 5), (35, 7), (40, 9)\} \text{ from top to bottom.}\]

In the proposed scheme 100% key connectivity is achieved between primary neighbours, unlike AP scheme. Thus a balance is struck between key connectivity and key storage.
2 Efficiency

(a) Storage: A typical node is subject to memory constraints for better performance i.e a maximum storage capacity of a node is generally small [25]. Also high end nodes have better storage capacity compared to low end sensor nodes. The proposed scheme uses this fact and stores data accordingly, i.e the storage decreases down the hierarchy. Matrix generation and other major storing activities are limited with the top level itself.

In LU [36 ] decomposition each and every node stores one row of lower triangular matrix and corresponding column of upper matrix to generate key by matrix multiplication. In the proposed scheme the upper triangular is the transpose of the lower triangular matrix (U = Lᵀ). This reduces the number of rows to be stored in each node to one. Let minimum size of one row be on average 4 bytes and say there are 5000 nodes (as in a typical network); LU utilises 40000 bytes (2*4*5000) whereas the proposed scheme consumes only half the above value i.e 20000 bytes (4*5000). The remaining reserved memory is efficiently utilized for authentication and computational purposes.

In RP [30 ] scheme the voting keys that are stored in low end sensor nodes used for node revocation increases the storage in the nodes. In the proposed scheme any node revocation within a group or a cluster is initiated and taken care by their corresponding group heads of cluster heads, imposing no additional memory consumption. Head nodes being a high end sensor node can store additional information. Thus the network objectives are achieved subject to memory constraints without any degradation in performance.
In Du et al matrix scheme [32] $\tau$ distinct keys spaces from the possible choices (say $w$) are randomly loaded into the nodes. The size of one row is $\lambda+1$, thus for each node $(\lambda+1)\tau$ units are required. In the proposed scheme many entries in lower triangular matrix are zero thus size of one row is far less than that used in Du et al scheme. This strategy helps to reduce memory consumption to a large extent.

In LEAP [48] each and every node is loaded with individual key, pair wise key, group key and cluster key to achieve high connectivity between different levels of hierarchy. The proposed scheme uses only pair wise key and group key to achieve the connectivity that LEAP achieves. This reduces the memory consumption to almost half of that in LEAP.
(b) Computation:

Computation is done at the expense of power consumption. Since a node is expected to work with minimal power consumption too much computation degrades node’s performance. Computation involving complex multiplication and exponents consumes a lot of power.

The proposed scheme basically involves three computations – multiplication, one-level base conversion, shifting. Simple matrix multiplication [36] is done for key generation. It is simple because most of the row values are zero. The base for conversion is chosen in such a way that it terminates at one level itself, thus restricting the number of divisions to one. Shifting being a bit twiddling operation [25] does not consume much power. Thus the computational power consumption is relatively less compared to many schemes and also the proposed scheme does not perform any computation for node authentication.
In Blundo [39] scheme the nodes are supposed to compute their key using \( n \)-degree polynomial functions with two variables which involve computing exponential powers of those variables and their summation. This consumes a lot power. Proposed scheme limits the number of arithmetic computations to one or two and mainly performs simple bit twiddling operations and hence consumes relatively less power.

In Blom [33] and Du et al [34] scheme the nodes compute keys by multiplying rows and columns. To reduce storage on each node, only the seed of the column (Vander Monde matrix with seed \( s \)) is stored. But this imposes computational overhead in generating the column which happens at the expense of power consumption. The proposed scheme has no such overhead in generating column as only rows are stored.

(c) Communication

Communication is directly related to transmission power. Thus for high performance unnecessary communications should be avoided [49].

In the proposed scheme the nodes are loaded with all the possible keys with which it can communicate, keeping in mind the transmission power, before deployment. Many schemes have shared key discovery phase and path key establishment phase. This involves a lot of communication between nodes. The proposed scheme being completely pre-deployed does not involve any communication of this type. Thus saving a lot of transmission power. Further the proposed scheme involves only two communications for key generation and node authentication.
Compared to AP scheme [32], Du et al matrix [34] scheme, Blundo [39 ], Liu, Ning [62] scheme and q-composite [30] scheme this scheme involves both shared key establishment and path key establishment phase which increase communication.

In LU scheme [36] the key computation involves three steps whereas proposed scheme uses only two steps, hence 33% transmission power is saved. Also in LU if there is a key mismatch then authentication mechanism (µ TESLA) [63] is initiated which consumes computation power. The proposed scheme does node authentication and key establishment within the two steps and hence is efficient.

3. Scalability

This measures the performance of the network on addition of new nodes. For a typical network the performance should not be affected while adding new nodes. That is the network supposed to have extensibility property. In the proposed scheme the order of matrix is set to the maximum having a futuristic view on the scalability. When a new node is added to a particular group, unused row of the matrix allotted to that group is loaded into it along with all other remaining necessary data before deployment. The strategy used here for accommodating large number of new nodes is to group them under new group. This strategy handles scalability to a large extent. If the number of nodes is too large when compared to the number of unused rows in the allotted matrix, then all those nodes are grouped under a new group head which has its own allotted unused matrix taken from the matrix store in the base station. Whenever a new node is added all the data that was persistent earlier must be concealed from the new node. This action is taken care by corresponding group or cluster heads depending on type of the node added, i.e low-end sensor node or group heads respectively.
In Hierarchical LU [36] the rows and columns are randomly loaded into the nodes. When more and more new nodes are added the possibility of two nodes having same rows increases. Thus probability of link compromising increases.

In RP scheme [30] each node is loaded with m identifiers in its vicinity. When a new node is added, its key identity must be updated in that connected component of the network. This imposes communication overhead.

In SHELL [45] scheme if a new node A joins in an EBS [43] (n, k, m). The k keys replace the old keys in the respective rows of A to ensure backward secrecy. The k new keys as well as a new session key S are sent to the new user in a message X encrypted with its personal key. In addition, to let all other nodes know their respective new keys, the server multicasts a message Y encrypted by the old session key. The message Y contains the new keys encrypted by the keys that they replace. This imposes a lot of communication overhead and involves huge processing.

5.5 SECURITY ANALYSIS

1 Resilience

A network should be secure enough so that the entire message passing is done secretly. This ensures no data leakage. Thus a malicious user cannot hack out the information from the nodes in the network.

Resilience is a measure of how quick the system recovers upon node capturing. The recovery depends on the impact of node capturing on the system, i.e it indirectly measures how much remaining nodes and links get compromised on node capturing.
A good wireless sensor network must definitely be resilient otherwise the entire system will be attacked and all the data can be hacked out of it.

2. Message Interception

This is situation where malicious users intercept messages (brute force attacks) in the network by snooping, traffic analysis, modification, masquerading, repudiation, replaying, and denial of service and many other security threats and attacks. A good network is supposed to ensure high data integrity, confidentiality and availability.

The proposed scheme produces cipher messages which are highly encrypted. Thus any malicious user will not be able to retrieve any information from it. By doing this the proposed scheme overcomes the traffic analysis and spoofing threats. Here the node ID and the computed key values which are checked against key pool list acts as the digital signature to provide authentication. Further the proposed scheme uses cryptography and thus ensures data integrity and overcomes masquerading, modification, etc.

Assuming awkward situation where in the attacker retrieved the content of the message and found the key. In the proposed scheme each and every node has a unique pair wise key with each node within a level. The attacker remains helpless with one key as he will not be able to masquerade with other nodes. Thus no link gets compromised. The only link that gets compromised is the link from which the attacked retrieved the information. In such a situation the corresponding group or cluster head initiates node recovery mechanism after identifying malicious network interfaces. Heads replace the row values and substitution array indices (keeping the value of the array same) of all the nodes in the vicinity of the alien interface with new rows from its allotted matrix and corresponding key pool list. Thus for the same index
the array value will be different, thus the malicious user will not be able to decrypt the message. All these changes are done by unicasting all these information to respective nodes.

3. Node and Link Compromising

Here the attacker uses any physical attacks to directly capture nodes. On capturing a node the attacker will get know about all the information that is stored in it. Considering an awkward situation where in a node in the proposed network is captured physically and all the information that is store in it are known to the attacker. Using this node the attacker can easily communicate with all the other nodes in its communication range. If the node is a group head then the attacker will be able to retrieve information from the entire cluster in which it belongs.

In the proposed scheme the respective group head or cluster head immediately initiates recovery mechanism after identifying the attacked node, giving no room for the attacker to extract information from other nodes. Once the attacked node is known by the group head it initiates node revocation mechanism. Its first and foremost task is to transmit the node ID of the attacked node to all the remaining nodes in the cluster. This is done by unicasting to the nearest node, the attacked node ID, the new row for group communication and change in hashing index in hashed format (mentioned above). The nearest node retrieves all the necessary information and performs three basic operations. Firstly it passes this message to its neighbour node which is not the attacked one. Secondly, it deletes the attacked node ID and its corresponding key value from the list it stores. It then changes the row for group communication and updates the old key value for group communication in the key list with the new computed group key. Thirdly, the node changes the range of base and shifting number
generated by the pseudo random generator. Thus though the attacker who knows the message format can’t hack it because he doesn’t know to what number the index is referring to. All the other nodes also do the same. Further this ensures resilient property in the network.

In AP scheme [32], Hierarchical LU scheme [37], Du et al matrix [34] scheme probability of using the same key to establish links between different nodes are more because the rows are randomly loaded into the nodes i.e two or more nodes may have same row values for shared key establishment. Thus when one link gets compromised it will also affect all the other links which used the same row for generation.

In RP scheme [32] the voting keys play a crucial role in node revocation. This increases storage in every node. Thus resilience is achieved at the expense of storage. Also voting leads to more communication, which is a communication overhead.

In LU scheme [36] the probability of two nodes to have same row value increases with increase in the number of nodes and number of keys chosen for the scheme. Thus number of link compromised increases with increase in number of key in a node.

4. Node Authentication

Authentication ensures that both the parties at the ends of communication are authenticated. In the proposed scheme, no additional authentication mechanism, like μ-Tesla, is used. Instead a part of key generation mechanism is used for authenticating. First the receiver node that extracts the ID from the message checks with its node ID. If it matches then it is confirmed that the message is from
authenticated node. Further confirmation is done by checking the computed key value with the key pool list it possesses.

If there is a match then authenticated communication takes place between them marked by the randomly generated authentication number. In cases of mismatch in either of the steps, the corresponding group or cluster head is alerted by the node in which mismatch occurred. The heads probe into this issue and finds whether mismatch is due to loss of data or due to malpractices. Thus two level authentications cum key generation reduce communication to a large extent.

In many schemes such as q-composite scheme, SHELL [45] the keys are directly deployed in the nodes. The nodes there directly send messages using those keys. In order to authenticate nodes some additional mechanisms are needed.

**Figure 5.12 Numbers of Keys vs. No of Link Compromised**

\[
\text{Link compromisation: } 1-\frac{(p-x)!^2}{(P!)(P-2x)!}; \quad P = \text{Pool size; } X = \text{Number of nodes}
\]
Many schemes ensure authentication via external authentication mechanisms. The authentication cum key generation is one of the useful techniques to ensure authenticated communication between two nodes with less consumption of transmission power. It also reduces the possibility of intercepting or masquerading the nodes as both key generation and authentication are done together. This poses a double barrier to the hackers who maliciously try to retrieve the information.

5.6 CONCLUSION

In the proposed scheme the achievements with respect to various metrics are tabulated in the Table 5.2 shown below.

<table>
<thead>
<tr>
<th>METRICS</th>
<th>ACHIEVEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key connectivity</td>
<td>100% local key connectivity, using pair wise keys at each level</td>
</tr>
<tr>
<td>Storage</td>
<td>Though LLT is a matrix base scheme, reduced storage is achieved, since most of the values in the matrix are zero and also only one row is stored: Storage(n) = (n/2)(n+1)</td>
</tr>
<tr>
<td>Communication</td>
<td>No shared key and path key establishment phase; also within two steps key is generated and also the nodes are authenticated; thus effective communication</td>
</tr>
<tr>
<td>Computation</td>
<td>Though computation involved in encryption is high, one level division and simple vector multiplication with most of the values zero makes optimizes it. Further shifting being a bit twiddling operation reduces computation overhead.</td>
</tr>
<tr>
<td>Scalability</td>
<td>The trick of the trade is that the generated matrix is of maximum possible order, thus when new nodes are added the unused row</td>
</tr>
</tbody>
</table>
from the matrix is loaded in the new node before deployment and nothing else is done

<table>
<thead>
<tr>
<th>Resilience</th>
<th>No link is compromised on message interception because of pairwise keys, but node eviction is a serious issue which is solved by changing the range of base and shift number which are pseudo randomly generated, thus now the numbers have new indices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>Three tier authentication where node IDs acts as a digital signature.</td>
</tr>
</tbody>
</table>

Thus the proposed scheme is an enhancement of LU scheme in storage, authentication and ultimately performance.