Chapter 3

Random Key Pre Distribution

3.1 OVERVIEW

In this chapter, a random key pre-distribution scheme for secured routing in Wireless Sensor Network (WSN) is proposed. After initial deployment of the nodes in the network, the sink divides the nodes into multiple classes and utilizes information from one hop neighbor node to distribute keys. The sink randomly selects \((n/2)\) nodes of a neighboring node and allocates \((k+m)\) keys whereas other remaining nodes acquire \((k)\) keys. As soon as pre-distribution of keys is achieved, each node constructs all possible direct and indirect key paths to their neighbors. With the help of constructed paths the pairwise key is established between neighbors. For data transmission the proposed scheme makes use of Non-Repetitive Random Propagation (NRRP) routing algorithm. When a node desires to transmit data to the sink, the next hop is randomly selected from the secured minimum hop neighbors. In the event such as when there is the presence of more than a secure minimum hop links, then weight function is assigned and the best path is elected.

Before we discuss about the pre key distribution we will discuss the evolution of symmetric key management towards random key pre distribution in the next section.

3.2 SYMMETRIC KEY MANAGEMENT

Blom, R. (1985) There has been enough discussion about lack of memory, power supply, and processing power in Wireless Sensor Networks. This limitation has resulted in the preference for symmetric key management by the wireless sensor with the sender and the receiver opting to have the same key for encryption and decryption. Over a period of time the symmetric key management has evolved and the variations of symmetric key management is briefed below.
3.2.1 EVOLUTION OF SYMMETRIC KEY MANAGEMENT

Network Wide shared key: Blundo et al (1983) The simplest method of all, a single secret key is deployed for all the sensor nodes for usage in case of encryption and decryption. The system is preferred for WSN due to its less usage of memory, power consumption, and processing time, it offers very poor security. A single node compromise can lead to the capture of an entire network using the master key. Even though the connectivity was very high, the immunity from
56 attack was very low. Considering the variety of attacks taking place this method was rejected altogether.

\begin{center}
\includegraphics[width=\textwidth]{network_keys.png}
\end{center}

\textbf{k1= network wide common key}

\begin{center}
Figure 3.2 Network wide shared keys
\end{center}

\textbf{Trusted base station:} Blundo et al (1983): In this system a secret key is shared by all the nodes. Internodal communication is established when the base node distributes session keys to every node. This methods gives base node prominence over other nodes as it is the sole provider of session key. Provision of tamper proof security is possible for base node. In this system capture of single node does not compromise the entire network.
thereby increasing the resilience manifold. However it is not popular in deployment as it question the nature of Wireless Sensor Network. Unlike WSN where the nodes have the freedom to move, form clusters and change topology, this form of trusted base station method makes all the nodes dependent on the base node for their communication including their neighbor node, thereby losing their dynamic nature and topology. This particular feature makes them unpopular in the real life scenario.

Figure 3.3 Trusted Base station model.
**Pair wise key establishment:** Du et al (2005), the aim is to combine the resilience of base station method and preserving the dynamic nature of the Wireless Sensor Network. This can be achieved with pair wise keys. This modality provides a separate secret key to communicate with every member node in the network. Capture of a node or even a group of node will not compromise nodes in the network. Resource constraints acts as a major setback in deployment in Wireless Sensor Network. Assuming “N” number of node availability the key pairs required will be “N-1” in the wireless network, making this unaffordable overhead in the Wireless Sensor Network.
In the above figure, the value of N (number of nodes) = 4, implying each and every node requires N-1 keys for communication establishment. In an atypical Wireless Sensor Network, the value of N being very high, requires high overhead for memory and energy consumption. To achieve a balance between high resilience and lesser overhead random pre key distribution is opted for. This forms the basis of this chapter.

**Random key pre distribution:** Eschenauer et al (2002) This method requires deployment of random pool of keys for every node. If node has to communicate with its neighbors it should find a common key among their key pools. The common key acts as a session key between the two nodes, availability of which is dependent on the random generation factor. Failure in finding a common key will result in generation of intermediate path between the two nodes. It has three different phases which will be explained in detail, during the explanation of our pre distribution work.

**Need for variation in key distribution:** In wireless sensor network every node communicates the data, and all these data gets aggregated at the sink node. Node closer to the base node play much more vital role as they are much more involved in routing protocols when compared to the normal nodes. Attackers prefer to attack these significant nodes rather than the normal nodes, as it leads to compromise of the whole network at one go. These types of attacks are characterized as smart attack.

When an attacker achieves in executing a smart attack on Wireless Sensor networks, he can gather and compromise the network much faster. So it is essential to differentiate the normal nodes and crucial nodes. Security in these significant nodes must be much higher than the normal peripheral nodes. This can be achieved by pre-distributing more keys to significant nodes, whereas the normal nodes have lesser number of keys. This will make smart attack difficult to execute because even if he captures the significant nodes, more
resilience added to those nodes by means of extra keys will make this smart attack much more difficult to execute.

Figure 3.5 crucial nodes.

3.3 Network Architecture of random key pre distribution

Consider the Wireless Sensor Network with \( N_i \) number of sensor nodes, where \( i = 1, 2 \ldots n \) and a sink node \( S \). The \( N_i \) nodes are randomly distributed in the network. We assume that the sink (\( S \)) is preinstalled with a set of \( K \) keys in the key pool. The \( S \) is in control
for randomly distributing keys to all nodes in the network. The schematic diagram of our network is shown in figure-3.1.

Figure 3.6 Random pre key distribution – basic architecture.

With deployment of nodes in the network, every node explores its one hop neighbors by sending HELLO messages. Nodes that receive HELLO messages respond back to the sender. By collecting replies, node $N_i$ constructs neighbor table (NT) and the format of NT is given below in table-3.1,

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Sequence Number</th>
<th>Neighbor Node ID</th>
<th>Key ID</th>
</tr>
</thead>
</table>

Table 3.1 Format of HELLO message
In the above table, Key ID will be discussed in later section. Once NT tables are constructed by nodes, it sends the collected neighborhood information to the sink.

3.3 DISTINGUISHED KEY PRE-DISTRIBUTION SCHEME

Our distinguished key pre-distribution scheme supposes that nodes are secured and they cannot be compromised before pre-distribution of keys. The proposed distinguished key pre-distribution scheme consist of two phases, they are initial key pre-distribution and pair wise key generation.

3.3.1 INITIAL KEY PRE-DISTRIBUTION

Initially, the sink (S) partitions the nodes into H number of classes such that \( N_i \) \((1 \leq i \leq H)\) nodes. The technique describes the node according to their class names. For instance, nodes in \( i^{th} \) class are defined as class i nodes and so on. Then, the sink selects distinct key \( K_i \) \((1 \leq i \leq H \text{ and } K_1 \geq K_2 \geq ... \geq K_H)\) from the key pool of size K and shares it with nodes in class i.

In order to distribute secret keys to nodes in a group, the sink makes use of NT information of a node. As we described in section-3.2, each node constructs NT by collecting information from one hop neighbor. Consider node \( N_i \) of \( i^{th} \) class has \( n \) one hop neighbor, and then the sink randomly selects \( (n/2) \) one hop neighbor and shares \( (K_i+m) \) keys, where \( m \) is a changeable variable. Other nodes in NT table will obtain \( K_i \) keys. The nodes are selected by the sink randomly and it does not depend on any parameters.

Since, \( (n/2) \) nodes are selected randomly to distribute \( K_i+m \) keys; it is difficult for an adversary to disclose all keys by compromising a single node. This process is repeated by the sink until nodes in the entire group are distributed with predistributed secret keys. Nodes that possess \( K_i+m \) keys have high resilience against failures. Further, these nodes assure an extra reliability and security when compared with nodes with \( K_i \) keys.
Therefore, while transmitting data between any pair of source and destination, nodes with $K_i+m$ keys are preferred to other nodes.

Figure 3.7 Key Distribution Phase
Consider the sketch given in figure 3.2, the sink (S) invokes the key predistribution phase as nodes are distributed in the network. Figure 2 includes the nodes that correspond to class-1. To distribute keys, initially the S exploits NT of node N\textsubscript{1}. The node N\textsubscript{1} has six neighboring nodes namely N\textsubscript{2}, N\textsubscript{3}, N\textsubscript{4}, N\textsubscript{5}, N\textsubscript{6}, and N\textsubscript{9}. Among six neighbors (n=6), the S randomly chooses (6/2) (i.e) three nodes namely N\textsubscript{2}, N\textsubscript{6}, and N\textsubscript{9} and distributed with K\textsubscript{i+m} keys. Other nodes namely N\textsubscript{3}, N\textsubscript{4}, and N\textsubscript{5} obtain K\textsubscript{i} keys.

Algorithm-1 describes the process of initial key pre-distribution

**Algorithm-1**

1. Let H be the set of classes i, where i=1, 2 … H and S be the Sink

2. Consider N as a set of sensor nodes and n as the number of neighboring nodes

3. Let NT\textsubscript{i} be the neighbor table of node N\textsubscript{i}

4. Consider K\textsubscript{i} and K\textsubscript{i+m} as keys taken from the key pool of size K

5. Nodes are distributed and S is positioned in the network

6. S classifies the nodes into H classes

7. It selects the class i from H and chooses node N\textsubscript{i} from N

8. S acquires NT\textsubscript{i} and calculates n

9. It calculates (n/2)

10. Randomly selects (n/2) nodes and distributes K\textsubscript{i+m} keys to them

11. S distributes K\textsubscript{i} keys to the remaining nodes in NT\textsubscript{i}

12. Steps 5 to 11 are repeated until S reaches the class H
3.3.2 PAIR WISE KEY GENERATION

As soon as the sink completes the pre-distribution of keys to nodes, every sensor node forwards K-info message to their one-hop neighbor. K-info message contains the key-ID’s information. Subsequently, every node receives a set of K-info messages from its one hop neighbors. The key ID information of neighboring nodes is tracked in the Key ID of NT. By utilizing this information, every node generates Direct and Indirect key paths with its neighbors. Here, Direct path refers to one-hop key path and Indirect path denotes two-hop key path.

The construction of Direct and Indirect key paths is as follows:

(i) When node $N_i$ distributes keys with node $N_{i+1}$, then node $N_i$ generates a Direct key path with $N_{i+1}$. The node $N_i$ can construct as many “Direct key paths” to their neighbors.

(ii) When node $N_i$ desires to generate Indirect key path with node $N_{i+2}$, then it first forwards a request message to all its neighbors including the node ID’s of $N_i$ and $N_{i+2}$. Once, the neighboring node (say $N_{i+1}$) receives the request, it checks whether it has shared the predistributed keys with both node $N_i$ and $N_{i+2}$. If so node $N_{i+1}$ replies back to node $N_i$. Thus, $N_i$ generates Indirect key path as,

$$ N_i \rightarrow N_{i+1} \rightarrow N_{i+2} $$

A node can construct many “Indirect key paths” regardless of “Direct key path” along any Specific node
Take into account the illustration in figure-3.3. In the figure, node $N_1$ creates direct and indirect key paths to node $N_3$. It constructs only one direct key path described as DKP and four indirect key paths namely IKP-1, IKP-2, IKP-3 and IKP-4.

After the node ($N_i$) has constructed all possible “Direct and Indirect key paths” to another node ($N_{i+1}$), node $N_i$ will produce multiple random shares and forward every key share on every key path such as Direct and Indirect key paths. With combined use of XOR, all key shares are encrypted/decrypted hop by hop along the path. Eventually, combination of all the key shares is forwarded between node $N_i$ and $N_{i+1}$ can be termed as the pair wise key of those two nodes. This generated pair wise key is used for secured transmission during routing.
3.4 ROUTE SELECTION

To route data from any node to the sink, in this paper we enhance the Non-Repetitive Random Propagation (NRRP) routing algorithm.

3.4.1 WEIGHT FUNCTION COMPUTATION

In order to calculate the weight function of a path, let us consider the distinguished key predistribution scheme given in section 3.3. Assume \( K(i,j) \) as the number of keys shared between node \( N_i \) and \( N_j \). This information can be obtained from the trusted offline authority (OA). Then the number of defense keys (\( D(i,j) \)) between \( N_i \) and \( N_j \) can be given as,

\[
D(i, j) = K(i, j) + \sum_{g=1}^{T} \min(K(i, OA_g), K(OA_g, j)) \quad (1)
\]

In the above expression, \( T \) denotes the number of indirect key paths between \( N_i \) and \( N_j \). The defense keys define strength of a link. Through this, we can describe how resilient the link is.

To perform data communication, node \( N_i \) assigns weight to all its neighboring nodes. The weight function is as follows,

\[
W_{Nj} = \frac{D(i, j)^{0}}{\sum_{k \in M(i)} D(i,k)^{0}} \quad (2)
\]

Here, node \( N_i \) assigns weight only to the nodes to which it has established pair wise keys. Let \( M(i) \) be the set of secure neighboring nodes of node \( N_i \), which are closer to the sink than itself. \( W_{Nj} \) is the weight probability that node \( N_i \) selects \( N_j \) as the forwarder. \( \beta \) is referred as priority variable. The value of \( \beta \) can be assigned as per the requirement of communication as,

*If (\( \beta = 0 \)) then*

*Equal priority will be given to all nodes in \( M(i) \)*

*Else if (\( \beta = \)positive value) then*

*Higher priority will be given to more resilient links*
Else if ( $\beta = \text{infinity}$) then

*Most resilient links are selected for routing*

End if

3.4.2 NON-REPETITIVE RANDOM PROPAGATION (NRRP) ROUTING ALGORITHM

When node $N_i$ intends to pass on data to the base node, it utilizes the $M(i)$ nodes for transmission. $M(i)$ is a set of secure neighboring nodes of $N_i$, where it has already established pairwise secret key with them. At first, it randomly selects a node from $M(i)$ and transmits a data packet. Similarly, data packets are transmitted. To enhance the efficiency of propagation and to assure loop-free routing, NRRP routing algorithm includes the node-in-route (NIR) field. Initially, this field is set to zero. Whenever a data packet is forwarded from a node to another node, the corresponding node ID is appended in NIR field such that the source does not select the same node again and again. Thus, when determining next hop, the node $N_i$ selects a minimum and secure next hop from $M(i)$. In the event such as, when the source has more minimum hop secure paths, then weight function ($W_{ij}$) is utilized. The node $N_i$ assigns weight function to each path and chooses the best path.

![Figure 3.9 NRRP selection of high Resilient nodes](image)

Figure 3.9 NRRP selection of high Resilient nodes
3.5 MERITS OF PROPOSED SCHEME

- In our distinguished key predistribution scheme, as \((n/2)\) nodes are selected randomly to distribute \(K_i+m\) keys; it is difficult for an adversary to disclose all keys by compromising a single node.
- The enhanced Non-Repetitive Random Propagation (NRRP) routing algorithm increases efficiency in propagation and assures loop free routing paths.
3.6. SIMULATION RESULTS

3.6.1 SIMULATION PARAMETERS

The key pool size is K. It is a crucial parameter in the random key distribution scheme. The storage size of reserved key is a present constraint in each node. This forms the size for key ring R, which is again a fixed parameter. With R being set the probability of larger valued K being shared by two key is less. Node initialization is simulated at the bootstrapping phase of the network.

NS-2 simulation is used to evaluate Randomized Secure Routing (RSR). Under consideration is a random network of sensor nodes deployed in the area of 500 X 500m. The amount of nodes can be as 50, 100, 150, 200. Assuming that the base node deployed is 100 m from the specified area, the simulated traffic using Ns2 is CBR with UDP. The cluster is 9 and data transmitted to the sink will be by for cluster heads. Cluster head receive data from three sensor nodes in each cluster.

<table>
<thead>
<tr>
<th>Node Parameters</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area Size</td>
<td>500 X 500</td>
</tr>
<tr>
<td>Mac</td>
<td>802.11</td>
</tr>
<tr>
<td>Routing protocol</td>
<td>RSR</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>25 sec</td>
</tr>
<tr>
<td>Traffic Source</td>
<td>CBR</td>
</tr>
<tr>
<td>Packet Size</td>
<td>512 bytes</td>
</tr>
<tr>
<td>Attackers</td>
<td>10 to 40</td>
</tr>
<tr>
<td>Transmission Range</td>
<td>250m</td>
</tr>
<tr>
<td>Transmit Power</td>
<td>0.395 w</td>
</tr>
<tr>
<td>Receiving power</td>
<td>0.660 w</td>
</tr>
<tr>
<td>Idle power</td>
<td>0.035 w</td>
</tr>
<tr>
<td>Initial Energy</td>
<td>10 Joules</td>
</tr>
</tbody>
</table>

Table. 3.2. Simulation parameters
3.6.2 PERFORMANCE METRICS

The performance of RSR is compared with the Differentiated Key Management scheme. The performance is evaluated mainly based on the following benchmarks.

- **Energy**: Average energy consumed for the data transmitted.
- **Delay**: Average time taken by the packets to reach the destination.
- **Fraction of Compromised communications**: It is the calculated effect on the network’s resilience in the event of node capture. It is calculated as the estimated fraction of compromised communication between the captured and non-captured nodes.
- **Average Packet Delivery Ratio**: The ratio of number of successfully delivered packets and total number of transmitted packets.

3.6.3 RESULTS

In order to observe the effect of an attack, capturing varied number of compromised sensor nodes, the number of compromised nodes can be changed as 10, 15, 20, 25 and 30 in the 100 node scenario.

Table 3.3 shows the results of RSR and DIFF-KEY techniques for varying the attackers. Figure 3.11 to 3.14 present the graphical representation of the results for delay, delivery ratio, energy consumption and fraction of Compromised communications, respectively.
<table>
<thead>
<tr>
<th>Attackers</th>
<th>Delay</th>
<th>Delivery Ratio</th>
<th>Energy</th>
<th>Resilience</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DIFF-KEY</td>
<td>RSR</td>
<td>DIFF-KEY</td>
<td>RSR</td>
</tr>
<tr>
<td>10</td>
<td>1.301138</td>
<td>0.884926</td>
<td>0.77333</td>
<td>0.883996</td>
</tr>
<tr>
<td>15</td>
<td>2.918831</td>
<td>1.828129</td>
<td>0.660218</td>
<td>0.783164</td>
</tr>
<tr>
<td>20</td>
<td>2.643398</td>
<td>2.280767</td>
<td>0.644758</td>
<td>0.684641</td>
</tr>
<tr>
<td>25</td>
<td>4.24789</td>
<td>3.253698</td>
<td>0.534258</td>
<td>0.659316</td>
</tr>
<tr>
<td>30</td>
<td>5.743692</td>
<td>3.537087</td>
<td>0.523034</td>
<td>0.596898</td>
</tr>
</tbody>
</table>

Table 3.3 Results for Varying Attackers

![End-to-End Delay](image)

Figure 3.11 End-to-End Delay in RSR
Figure 3.11 displays the delay for both the protocols in the event of an increased number of compromised nodes. It has been observed that the delay increases with proportionate increase in compromised nodes. However a reduced delay of 28% is observed in RSR when compared to DIFF-KEY protocol as it uses NRRP algorithm since it minimizes the time involved in functions like unnecessary route establishment.

Figure 3.12 Packet Delivery Ratio in RSR

Figure 3.12 depicts the decrease in packet delivery ratio in the event of increase in the number of compromised nodes. The delivery ratio degrades when the attackers increase causing a increase in the drop of packets. In the figure the delivery ratio is down up to 0.6 when the number of attackers are 40. RSR method with its random distribution of keys and use of dispersive routing reduces the effect of packet drop. This proves that the delivery ratio of RSR is greater by 13.2% when compared to DIFF-KEY method.
Figure 3.13 shows energy consumed on an average when both the techniques are used. In comparison to DIFF-KEY technique, RSR achieves 18.2% less energy consumption with the use of higher residual energy nodes by the cluster heads.
Figure 3.14 Fraction of Compromised Communications

Figure 3.14 shows the effect of node capture attack. There is a clear indication of an increase in the number in the fraction of communication when there is an increase in compromised nodes. However RSR shows an improved performance at 24% higher resilience in case of an increase in compromised nodes. This is because of the involvement of more shared keys in the key path and a random node selection by the NRRP protocol.
Table 3.4 shows the percentage wise improvement of RSR over DIFF-KEY for varying the number of attackers.

<table>
<thead>
<tr>
<th>Attackers</th>
<th>Delay (%)</th>
<th>Delivery Ratio (%)</th>
<th>Energy (%)</th>
<th>Resilience (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>31.98831</td>
<td>12.51883</td>
<td>18.79081</td>
<td>23.68157</td>
</tr>
<tr>
<td>15</td>
<td>37.36777</td>
<td>15.69863</td>
<td>18.09483</td>
<td>31.23298</td>
</tr>
<tr>
<td>20</td>
<td>13.71837</td>
<td>5.82538</td>
<td>16.29042</td>
<td>25.01759</td>
</tr>
<tr>
<td>25</td>
<td>23.40437</td>
<td>18.96784</td>
<td>19.83914</td>
<td>20.94344</td>
</tr>
<tr>
<td>30</td>
<td>38.41789</td>
<td>12.37464</td>
<td>18.64578</td>
<td>21.4398</td>
</tr>
</tbody>
</table>

Table 3.4 Percentage wise improvement of RSR over DIFF-KEY

3.7 CONCLUSION

In this chapter, a key predistribution scheme for randomized secured routing in WSN is proposed. Initially, using a distinguished key predistribution scheme, the sink utilizes the one hop neighbor information of nodes to pre-distribute keys. It randomly chooses $(n/2)$ nodes and distributes with $(k+m)$ keys and other nodes to obtain $k$ keys. Nodes with more secure keys are considered as high resilient nodes. For data transmission the proposed scheme makes use of Non-Repetitive Random Propagation (NRRP) routing algorithm. When a node desires to transmit data to the sink, the next hop is randomly selected from the secured minimum hop neighbors. In such an event, when there is a presence of more than a secure minimum hop links, then weight function is assigned and the best path is elected. The proposed scheme is simulated in NS-2. Since, $(n/2)$ nodes are selected randomly to distribute $(k+m)$ keys; it is difficult for an adversary to disclose all keys by compromising a single node.