Key Predistribution Scheme for Grid Based Wireless Sensor Networks using Quadruplex Polynomial Shares per Node

V. Sridhar, V. Raghavendar

Department of Mathematics and Computer Applications, PSG College of Technology, Peelamedu, Coimbatore 641004, India

Email: {vsridhar1729, linkwithraghav}@gmail.com

Abstract

To enable secure communication, pre-distributing keys to sensors in a grid based wireless sensor network has been an efficient alternative. In this paper, we propose a deterministic key predistribution scheme using symmetric bivariate polynomials as key pool in which each node stores four polynomial shares. The \( \tau \)-secure property of \( \tau \)-degree polynomial is exploited to achieve high resilience against both random and selective node capture. Also, a strategy to mount selective node capture on the proposed scheme is rendered. The proposed scheme is scalable and provides comparable connectivity with that of the existing schemes.

Keywords: Key Predistribution; Grid Based Wireless Sensor Network; Symmetric Bivariate Polynomial; Selective Capture of Nodes; Transversal Designs

1. Introduction

Wireless Sensor Network (WSN) is a distributed, self-organizing network of low-powered, memory constrained, inexpensive sensor nodes whose objective is to monitor the region over which it is deployed. Secure communication of data is a crucial issue which is enabled by employing cryptographic techniques. The inherent key agreement problem in these techniques has been addressed by devising several strategies. Amongst these, the failure of Public Key Infrastructure (PKI) and centralized trust server approaches have paved way for key pre-distribution techniques as an emerging research area, [1], for resolving the key agreement problem. Depending on the degree of control over the location of sensors, Martin in [2] classifies Key Predistribution Schemes (KPS) for a WSN as: Uncontrolled, partially controlled and fully controlled. Eschenauer and Gligor, [3], pioneered the key agreement problem in an uncontrolled sensor network using random graphs. Several KPSs for this mode of deployment are discussed in [4-11]. Imparting deployment knowledge information when devising a KPS can yield better performance in terms of connectivity, resilience, storage, communication and computation overhead, [12]. Key predistribution schemes proposed for such networks is based on the idea: if the probability of two nodes being located close to each other is high, then the probability of that they lie in each other’s communication range is also high. Hence, the probability that they share a cryptographic key should also be high. Schemes with such a motif for partially controlled network are dealt in [13-15]. Such schemes show high resilience, good connectivity for minimum number of keys but are complex in nature.

In fully controlled grid based networks, it is assumed that the points of intersection are accessible and sensors may be placed at those positions with reasonable precision. Such a placement strategy has enormous applications ranging from military, like intrusion detection, to civilian, like monitoring vines in vineyard. In general, it is assumed that the sensing and transmission radii are same, known as Radio Frequency (RF) radius. Ruj et al. in [16, 17] innovated a grid-based deployment KPS in a fully controlled environment, with square RF region and Lee
distance approximation. They pre-distributed the keys using transversal designs, $TD(k,r)$ where each node stores $k$ keys from a key pool of size $kr$. It provides good connectivity, low communication and computation overhead but suffers from limited scalability. It was also observed that the Lee distance approximation had slightly better resilience than square RF region approximation. Blackburn et al. proposed a scalable solution by modeling Distinct-Difference Configurations (DDCs) for predistribution of keys in a square grid, with Lee distance RF region approximation, [12]. Given a specific radius $r$ and maximum storage $m$ available for a sensor, a DDC $DD(m,r)$ is constructed using techniques given in [18]. Though this scheme offers connectivity similar to $TD(k,r)$ scheme, one of the limitations of the scheme is that all the key-sharing neighbors of a node is within the radius $r$. In case a node having radius $r$ gets disconnected (as its key-sharing neighbors could have been compromised), increasing its RF radius does not aid in establishing secure communication with other nodes. Furthermore, this scheme offers weaker resilience than $TD(k,r)$ scheme, [17]. One of the main objectives of imparting deploying knowledge while design KPS is to render high resilience. Presently, these schemes for fully controlled grid network offer low resilience. For example, consider a grid of size $23 \times 23$ in which each node stores 15 keys with RF radius, $\rho = 7$. Compromise of $s = 5$ nodes in a $TD(k,r)$ scheme affects more than 20% of the total number of links.

In this paper, we consider a homogenous fully controlled sensor network, i.e. all nodes have equal capability, deployed on a square grid with square RF region approximation. With an objective to provide higher resilience than the existing schemes, a scalable deterministic KPS using symmetric bivariate polynomial key pool is proposed. The connectivity of each node is established and a strategy to selectively capture nodes for proposed scheme is shown. Also, a theoretical upper bound on its resilience against random node capture is presented and is supported through experimental results. The rest of the paper is organized as follows. The proposed scheme is presented in section 2. The connectivity of each node is provided in section 3 and the resilience of the proposed scheme under selective and random node capture pattern is furnished in section 4. Finally, we conclude in section 5 and discuss our future work.

2. Proposed Scheme

As exemplified above, the existing grid based schemes in [12, 16] renders large number of uncompromised links insecure on capture of very few nodes. To improve resilience, we exploiting the $t$-secure property of a polynomial with degree $t$, [19]: if the nodes captured by an adversary has $t$ or lesser shares of a polynomial $f(x,y)$, then all the uncompromised links formed by $f(x,y)$ is secure; but if $(t+1)$ or more shares of the polynomial is captured, then the adversary can interpolate to reconstruct $f(x,y)$ and hence, learn all the keys.

2.1. Key Distribution

Key distribution phase commences with the KMA generating a total of $6(m-1)$ symmetric bivariate polynomials of degree $t$ over a finite field $\mathbb{F}_q$ where $q$ is a prime power, $m = \sqrt{N}$, $N$ is the number of nodes. These polynomials constitute the key pool known as polynomial key pool, $F$, [5]. Let $n_i$ be the node located at $(i,j)$ in the grid. The polynomials are partitioned into four different types: row, column, left diagonal and right diagonal. The different polynomials along with set of nodes sharing these polynomials are enumerated below:

1. **Row polynomials**, $f_{i}^{r}(x,y)$, is shared by nodes $n_{i,1}, n_{i,2}, \ldots, n_{i,m}$, i.e., by nodes located in row $i$. Here, $i = 1, 2, \ldots, m$ and hence, there are $m$ such polynomials.

2. **Column polynomials**, $f_{j}^{c}(x,y)$, is shared by nodes $n_{1,j}, n_{2,j}, \ldots, n_{m,j}$ i.e., by nodes located in column $j$. Here, $i = 1, 2, \ldots, m$ and hence there are $m$ such polynomials.

3. **Left diagonal polynomials** are denoted by $f_{(i,j)}^{ld}(x,y)$ where $(i',j')$ is the starting location of the polynomial. Generally, $(i',j')$ is either of the form $(i,1)$ or $(1,j)$, where $i = 1, 2, \ldots, m - 1$ and $j = 2, \ldots, m - 1$. Such $2m - 3$ polynomials are shared by nodes $n_{i,j}, n_{i+1,j+1}, \ldots, n_{k,l}$, where

$$ (k, l) = \begin{cases} (m, m - i + 1), & \text{if polynomial is of the form } f_{(i,1)}^{ld}(x,y) \\ (m - j + 1, m), & \text{if polynomial is of the form } f_{(1, j)}^{ld}(x,y) \end{cases} $$

4. **Right diagonal polynomials** are denoted by $f_{(i,j)}^{rd}(x,y)$ where $(i',j')$ is the starting location of the polynomial. Generally, $(i',j')$ is either of the form $(1,j)$ or $(i,m)$, where $j = 2, 3, \ldots, m$ and $i = 2, 3, \ldots, m - 1$. Such $2m - 3$ polynomials are shared by nodes located at $n_{i,j}, n_{i-1,j-1}, \ldots, n_{k,l}$, where

$$ (k, l) = \begin{cases} (j, 1), & \text{if polynomial is of the form } f_{(1,j)}^{rd}(x,y) \\ (m, i), & \text{if polynomial is of the form } f_{(i,m)}^{rd}(x,y) \end{cases} $$
Figure 1 depicts the assignment of polynomial shares to each node. To make the figure presentable, the polynomials \( f_a^b(x, y) \) is abbreviated as \( f_a^b \) where \( a = r, c, ld, rd \).

During the generation of these polynomials, the KMA assigns unique IDs for each of polynomial as follows:

1. For each row polynomial, \( f_r^i \), assign its unique ID as \( i_b \), where \( i_b \) is the binary representation of \( i \) of length \( \log_2 m \). Repeat this process for column polynomials also.

2. For each left diagonal polynomial, \( f_{ld}^{i,j} \), assign an unique value for \( (i, j) = d_t \in [1, 2m - 3] \), and let \( d_{rb} \) is the binary representation of a unique \( d_t \) of length \( \log_2 2m - 3 \).

3. For each right diagonal polynomial \( f_{rd}^{i,j} \), assign an unique value for \( (i, j) = d_r \in [1, 2m - 3] \), and let \( d_{rb} \) is the binary representation of a unique \( d_r \) of length \( \log_2 2m - 3 \).

For a node, \( n_{i,j} \), which has to posses the polynomial share of the polynomials with ids \( \{i_b, j_b, d_{lb}, d_{rb}\} \) is given a unique ID as, \( ID(n_{i,j}) = i_b \ | \ j_b \ | \ d_{lb} \ | \ d_{rb} \), where ‘|’ is used to represent the concatenation of binary strings. Hence, the total memory required for a node to store its ID is \( 2 \times (\log_2 m + \log_2 (2m - 3)) \). On assigning a unique ID to each node, KMA then assigns the polynomial shares \( f_{r}^i(\text{ID}(n_{i,j}), y), f_{c}^i(\text{ID}(n_{i,j}), y), f_{ld}^{i,j}(\text{ID}(n_{i,j}), y), f_{rd}^{i,j}(\text{ID}(n_{i,j}), y) \) to each node \( n_{i,j} \) where \((i', j')\) and \((i'', j'')\) correspond to the unique values \( d_{lb} \) and \( d_{rb} \) respectively. It can be seen that the proposed scheme is highly scalable, since if new nodes are to be added, depending on their position in the grid network, they can be given the corresponding polynomial shares. Since each node stores 4 polynomial shares, the proposed KPS is known as Quadruplex Polynomial Shares per Node (QPSN) scheme.

After assigning the corresponding polynomial shares to all the \( N \) sensor nodes, they are transported to the surveillant environment and distributed in accordance with the predetermined locations.

### 2.2. Share Key Discovery Phase

On deploying the nodes in the surveillant environment, they trigger the shared key discovery phase. This phase assists in ascertaining each node’s key-sharing neighbors. In this phase, each node broadcasts its node IDs to its physical neighbors in its RF radius, \( \rho \). As square RF region is consider, the physical neighbors of a node \( n_{i,j} \) is the set of all nodes which are encompassed by a square region of dimension \( 2\rho \times 2\rho \) centered at \((i, j)\). On the receiving end, each node partitions the node ID into four parts: \( i_b, j_b, d_{lb}, d_{rb} \) of size \( \log_2 m \) bits, \( \log_2 m \) bits, \( \log_2 2m - 3 \) bits, \( \log_2 2m - 3 \) bits respectively, in this order. Partitioning is followed by comparing it with its corresponding polynomial ids. If a match occurs, the corresponding polynomial is used to generate the secret key. If \( w \) is the ID of the matched polynomial, then the receiving node \( n_{i',j'} \) calculates the key \( K = f_w^p(\text{ID}(n_{i',j'}), \text{ID}(n_{i,j})) \) by substituting \( \text{ID}(n_{i,j}) \) in its polynomial share. Similarly the source node \( n_{i,j} \) computes the key \( K = f_w^p(\text{ID}(n_{i,j}), \text{ID}(n_{i',j'})) \) by substituting \( \text{ID}(n_{i',j'}) \) in its polynomial share, where \( p \) is the type of the matched polynomial. It can be observed that a secure link is created due to the polynomial shares of at most one polynomial.
In the process, communication incurs an overhead of $2 \cdot (\lceil \log_2 m \rceil + \lceil \log_2 (2m - 3) \rceil) \approx 2 \cdot \log_2 N$ bits, where $m = \sqrt{N}$. The computational complexity is incurred during the polynomial evaluation to derive the secret key. Such an evaluation involves $t$ modular multiplications and $t$ modular additions over finite field $\mathbb{F}_q$. The inputs for the polynomials, the node IDs, are of size approximately $2 \cdot \log_2 N$. Since, modular multiplication contributes maximum to the complexity, the computation complexity is $O(\lceil \log_2^2 N \rceil) \approx O(\lceil \log_2^2 m \rceil)$. TD($k, r$) scheme requires $\log_2 N$ bits for communication and $O(\log_2^2 r)$ computational complexity where $r$ is prime power. The communication and computation complexity of QPSN scheme, though efficient, is not better than that for TD($k, r$).

Let us now evaluate the performance of QPSN scheme using some standard performance indicators.

3. Key Connectivity

Key connectivity evaluates the scheme by furnishing the probability that two nodes are connected. Adopting the technique in [16], the local connectivity of a node $n_{ij}$ can be computed as, $R_p = \frac{|A_{ij}^{(l,j)}|}{|A_{ij}^{(l,j)}|}$ where, $A_{ij}^{(l,j)}$ is the set of key-sharing neighbors of $n_{ij}$ and $P_{ij}$ is the set of physical neighbors of $n_{ij}$. In the same paper, the authors have shown that, for any node $n_{ij}$, $|B_{ij}^{(l,j)}| = (\min(i, r - 1 - i, \rho) + \rho + 1) \cdot (\min(j, r - 1 - j, \rho) + \rho + 1) - 1$ and in particular, for an interior node $n_{ij}$, $|B_{ij}^{(l,j)}| = 4\rho(\rho + 1)$.

3.1. Determining the exact value of $|A_{ij}^{(l,j)}|$ for the proposed scheme

The following theorem determines the number of key-sharing neighbors of $n_{ij}$ i.e., $|A_{ij}^{(l,j)}|$ when $\rho \leq \frac{m}{2}$.

**Theorem 1:** In a $m \times m$ grid, on employing the QPSN scheme, the number of key-sharing nodes for a node $n_{ij}$ having RF radius $\rho \leq \frac{m}{2}$, is given by

$$|A_{ij}^{(l,j)}| = \begin{cases} 8\rho, & \text{if } 1 \leq (i + \rho, i - \rho) \leq m \text{ and } 1 \leq (j + \rho, j - \rho) \leq m \\ 3(i - 1) + 5\rho, & \text{if } 1 \leq (i + \rho, i - \rho) \leq m \text{ and } j - \rho < 1 \\ 3(m - j) + 5\rho, & \text{if } 1 \leq (i + \rho, i - \rho) \leq m \text{ and } j + \rho > m \\ 3(i - 1) + 5\rho, & \text{if } 1 \leq (i + \rho, i - \rho) \leq m \text{ and } j - \rho < 1 \\ 3(m - i) + 5\rho, & \text{if } 1 \leq (j + \rho, j - \rho) \leq m \text{ and } i - \rho < 1 \\ \min(i - 1, j - 1) + 2(i - 1) + 2(j - 1) + 3\rho, & \text{if } i - \rho < 1 \text{ and } j - \rho < 1 \\ \min(m - i, j - 1) + 2(m - i) + 2(j - 1) + 3\rho, & \text{if } i + \rho > m \text{ and } j - \rho < 1 \\ \min(i - 1, m - j) + 2(i - 1) + 2(m - j) + 3\rho, & \text{if } i - \rho < 1 \text{ and } j + \rho > m \\ \min(m - i, m - j) + 2(m - i) + 2(m - j) + 3\rho, & \text{if } i + \rho > m \text{ and } j + \rho > m \\ \end{cases}$$

The total number of secure links in the network is given by $T = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} |A_{ij}^{(l,j)}|$

**Proof.** Let, $l_{ij} = \{(i - \rho, i + \rho) \times (j - \rho, j + \rho) \times (i, j)\}$ for any point $(i, j)$ in the grid. Any node sharing a key with node $n_{ij}$ must lie in any one of the straight path connecting location $(i, j)$ to $(i', j') \in l_{ij}$. Let $(k, l)$ be called a valid location if the point $(k, l)$ lies within the grid.

![Figure 2: The key-sharing neighbors of node n_{ij} with radius \rho.](image)
Since every node has a radius $\rho$, there is exactly $\rho$ nodes on the straight path connecting $(i, j)$ with a valid location in $L_{i,j}$. Figure 2 projects the location of such nodes for node $a$ in $L_{a,j}$. We shall now prove the theorem by considering all the cases individually.

**Case 1:** $1 \leq [i, p, i - \rho] \leq m$ and $1 \leq [j, p, j - \rho] \leq m$.

In this case, all the locations in $L_{i,j}$ are valid locations. There is a total of $\rho$ nodes in the straight path from $(i, j)$ to $(i', j')$, including $(i', j')$. Here, $|L_{a,j}| = 8$, giving a total of $8\rho$ key-sharing nodes.

**Case 2:** $1 \leq [i, p, i - \rho] \leq m$ and $j - \rho < 1$.

It has been assumed that $\rho \leq \frac{m}{2}$ and hence, $j + \rho$ is a valid location. Therefore, all the locations in the set $V = \{(i + p, i, i) \times (j, j + \rho)\} \cap (i, j)$ are valid locations. $V \subset L_{i,j}$ with $|V| = 5$, there are $5\rho$ key-sharing neighbors. Since $(i', j - \rho)$ is not a valid location, $n_{i,j}$ can share keys only with nodes $n_{i', j'}$ where $f' = 1, 2, ..., j - 1$ and $i' \in [i + p, i - \rho]$. Hence, for each of the remaining 3 cases, there is $j - 1$ nodes with which $n_{i,j}$ shares a key.

**Case 3:** $1 \leq [i, p, i - \rho] \leq m$ and $j + \rho > m$.

As discussed in earlier case, since $\rho \leq \frac{m}{2}$, $j - \rho$ is a valid location. Therefore, all the locations in the set $V = \{(i + p, i, i) \times (j, j + \rho)\} \cap (i, j)$ are valid locations. Again, $V \subset L_{i,j}$ with $|V| = 5$ and hence there are $5\rho$ key-sharing neighbors. Since $(i', j + \rho)$ is not a valid location, $n_{i,j}$ can share keys only with nodes $n_{i', j'}$ where $j' = j + 1, j + 2, ..., m$ and $i' \in [i + p, i - \rho]$. Hence, for the remaining 3 cases, there is $m - j$ nodes with which $n_{i,j}$ shares a key.

**Case 4:** $1 \leq [j, p, j - \rho] \leq m$ and $i - \rho > 1$.

Again, as $\rho \leq \frac{m}{2}$, $i + \rho$ and $j - \rho$ is a valid location. Consequently, all the locations in the set $V = \{(i, j) \times (j + p, j)\} \cap (i, j)$ are valid locations with $|V| = 3$. Hence, there are $3\rho$ nodes which shares a key with $n_{i,j}$. Furthermore, there are $i - 1$ valid locations in the straight path from $(i, j)$ to $(i - \rho, j + \rho)$ and $(i, j - \rho)$. Likewise, in each of the straight path from $(i, j)$ to $(i + p, j, j - \rho)$ and $(i, j - \rho)$ there are $j - 1$ valid locations. It can also be observed that there is a total of $\min(i - 1, j - 1)$ valid locations in the straight path from $(i, j)$ to $(i - \rho, j + \rho)$.

Similar to the proof for case 6, the remaining cases can be proved. Hence, we have obtained the total number of key-sharing neighbors for a node $n_{i,j}$. The corresponding key-sharing graph, $[1]$, consists of vertices representing the nodes and an edge which denotes a secured link i.e., the two nodes share a key. The degree of each vertex, $n_{i,j}$ is $\delta_p^{(i,j)}$. By handshaking lemma [20], the total number of edges/links in the network is, $T = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \delta_p^{(i,j)}$.

Table 1 illustrates the connectivity ratio of the QPSN scheme and the $TD(k, r)$ scheme for an interior node as RF radius varies. It can be seen that the proposed QPSN scheme performs better than the $TD(k, r)$ scheme. In both the schemes, the proportion of nodes to which an interior node is connected diminishes as $\rho$ increases, the decrease in the $TD(k, r)$ scheme is much more than the former. But, it has also been observed that, increasing the size of the key chain improves the connectivity of the $TD(k, r)$ scheme while it has no effect on QPSN scheme’s connectivity.

### Table 1: Connectivity ratio $R_p$ for interior nodes with change in RF radius, for a 47x47 grid with 7 keys per node, for different schemes.

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TD(k, r)$</td>
<td>0.5000</td>
<td>0.4167</td>
<td>0.3333</td>
<td>0.3000</td>
<td>0.2500</td>
<td>0.2381</td>
<td>0.2054</td>
<td>0.2014</td>
<td>0.1889</td>
</tr>
<tr>
<td>QPSN</td>
<td>1.0000</td>
<td>0.6667</td>
<td>0.5000</td>
<td>0.4000</td>
<td>0.3333</td>
<td>0.2857</td>
<td>0.2500</td>
<td>0.2222</td>
<td>0.1818</td>
</tr>
</tbody>
</table>

### 4. Resilience against Capture of Nodes

The environment, in which the sensors are deployed, is susceptible to capture of nodes by adversary. As the sensors are not tamper resistant, on compromise of a node, all the cryptographic keys it possesses are exposed. An immediate consequence is that all the communication channels which uses any of those keys become insecure. The metric which quantifies the damage caused by the exposure of keys, $E(s)$, is defined as given in [7],

$$E(s) = \frac{\text{Number of links exposed after capture of } s \text{ nodes}}{\text{Number of links present before compromise}}$$

Generally, it is assumed that the adversary is powerful enough to eavesdrop on any communication channel in the grid network and also possesses infinite resources in terms of software, hardware and computation power. Being so powerful, there are two types of capture patterns which the adversary can follow.
1. **Selective capture of nodes** – In this type of attack, the adversary wisely captures a small set of nodes to expose all the keys from the key pool. Generally, such an attack exploits the manner in which the keys are distributed to the nodes. In this case, $E(x) = 1$.

2. **Random capture of nodes** – In this type of attack, the adversary does not follow any specific methodology to capture nodes i.e., it is random.

We shall analyze the resilience of the proposed scheme under these two types of attacks.

### 4.1. Selective Capture of Nodes

To reveal all the keys from the key pool of $TD(k,r)$ scheme, the attacker only has to remove all the nodes from any one of the rows. Hence, for a grid of dimension $r \times r$ ($r$ is a prime power), the proportion of nodes to be removed under such an attack is,

$$P_s = \frac{r}{r^2} = \frac{1}{r}$$

In this section, the total number of nodes to be removed and guidelines on which nodes are to be removed to successfully mount this attack on the proposed QPSN scheme is discussed and a comparative study is tabulated.

Consider a grid of dimension $m \times m$. For such a grid, the proposed scheme generates $6(m-1)$ polynomial of degree $t$. To completely reveal all the keys from the key pool, the adversary must capture nodes such that the complete polynomial key pool can be reconstructed. Due to the $t$-secure property of each polynomial, the captured nodes must contain $t+1$ polynomial shares of all the polynomials. It must be noted that for polynomials which are shared by less than $t+1$ nodes, all the nodes sharing the polynomial must be captured to reveal the pair-wise keys. The steps to be followed, along with the explanation of how many polynomials, are given below.

1. Capture nodes $n_{c,i}$ where $i = 1, 2, ..., m$ and $j = 1, 2, ..., t + 1$. At the end of this step, $t + 1$ shares of the following polynomials is exposed:
   - All the $m$ row polynomials.
   - $m - 1$ left diagonal polynomials $f(x, t', i')$, $i' = 2, 3, ..., m - 1$.
   - $m - 1$ right diagonal polynomials $f(x, j', t')$, $j' = 2, 3, ..., m$.
   - The capture of these $(t + 1)m$ nodes renders all the links corresponding to $3m + t - 1$ polynomials insecure.

2. Capture nodes $n_{d,i}$ where $i = 1, 2, ..., t + 1$ and $j = t + 2, ..., m$. At the end of this step, $t + 1$ shares of the following polynomials is revealed:
   - Remaining $m - (t + 1)$ column polynomials $f(x, j')$.
   - Remaining $m - 2$ left diagonal polynomials $f(x, i', j')$, $i' = 2, 3, ..., m - 1$.

3. For every right diagonal polynomials $f(x, m, t)$, $i = \left\lfloor \frac{t+1}{2} \right\rfloor$, $\ldots$, $t + 1$, capture $(t + 1) - 2 \times (t - i + 2)$ nodes possessing the corresponding polynomial share. Therefore, a total of $\sum_{i=1}^{t+1} \left( (t+1) - 2 \times (t - i + 2) \right)$ nodes are capture in the process. At the end of this step, $t$ right diagonal polynomials $f(x, m, t)$, $i = 2, 3, ..., t + 1$ have been compromised.

4. It can be observed that each of $f(x, m, t)$, $i = m - 1, m - 2, ..., m - t$ is distributed among $t + 1$ or lesser nodes. Hence, all the nodes sharing any one of these polynomials have to be captured to gain knowledge of the pair-wise keys used amongst nodes having any of these polynomials. This leads to two cases,
   - If $(m - 2) - t < t + 1$, then the polynomials remaining to be captured has less than or equal to $t + 1$ nodes. Hence all these must be compromised to obtain the secret key which each of nodes uses to communicate securely. This involves capturing $\sum_{i=t+2}^{m-2} (m - i + 1) = \frac{(m-t-1)(m-t)}{2}$ nodes.
   - Else if $(m - 2) - t \geq t + 1$, then there exists uncompromised right diagonal polynomials which have more than $t + 1$ nodes. Hence, $t + 1$ nodes having the polynomial shares $f(x, m, t)$, $i = t + 2, ..., m - t - 1$. Since each node have can have at most one right diagonal polynomial share, the number of nodes captured is $(t + 1)(m - 2t - 2)$. As discussed above, in this case, the polynomials $f(x, m, t)$, $i = m - 1, m - 2, ..., m - t$. 

necessitates all the nodes having polynomial shares of these polynomials must be compromised. In this process, the adversary captures \(\sum_{i=m-1}^{m-1} (m - i + 1) = \frac{t^2 + 3t}{2}\) additional nodes.

Hence, at the end of this step, all the polynomials from the polynomial key pool are compromised. The total number of nodes to be compromised to reveal the entire key pool is given by,

\[
T_s = \begin{cases} 
2m(t + 1) - (t + 1)^2 + \sum_{i=1}^{t+1} [(t + 1) - 2 \cdot (t - i + 1)] + \frac{(m - t - 1)(m - t)}{2}, & \text{if } (m - 2) - t < t + 1 \\
2m(t + 1) - (t + 1)^2 + \sum_{i=1}^{t+1} [t + 1 - 2 \cdot (t - i + 1)] + (t + 1)(m - 2t - 2) + \frac{t^2 + 3t}{2}, & \text{otherwise}
\end{cases}
\]

Therefore, the proportion of nodes to be removed to successfully mount this on the QPSN scheme is, \(P_s = \frac{T_s}{N}\).

The following table compares the resilience of the proposed QPSN scheme with the TD\((k, r)\) scheme for some grid networks. Here \(r = m\) and \(k\) is the number of keys stored in each node of the TD\((k, r)\) scheme. The value of \(t\) is taken such that \(4(t + 1) \approx k\).

From Table 2, it can be seen that the proposed scheme provides significantly higher resilience against this attack than the TD\((k, r)\) scheme, for nearly the same amount of memory utilized for keys. It should also be noted that, for the same grid network a small increase in \(t\) (and hence, memory requirement increases) improves the resilience drastically. Furthermore, for the proposed scheme, the attacker has to compromise nodes from large portions of the grid whilst in TD\((k, r)\) scheme the nodes have to be compromised from a small portion of the grid.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(m = 23), (t = 2, k = 15)</th>
<th>(m = 23), (t = 3, k = 15)</th>
<th>(m = 31), (t = 4, k = 20)</th>
<th>(m = 37), (t = 6, k = 30)</th>
<th>(m = 53), (t = 11, k = 49)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TD((k, r))</td>
<td>0.0435</td>
<td>0.0435</td>
<td>0.0323</td>
<td>0.0270</td>
<td>0.0189</td>
</tr>
<tr>
<td>QPSN</td>
<td>0.3516</td>
<td>0.4518</td>
<td>0.4246</td>
<td>0.4865</td>
<td>0.5635</td>
</tr>
</tbody>
</table>

4.2. Random Capture of Nodes

As stated earlier, in this pattern of attack, the adversary captures the nodes one after another at random. In this process, if the adversary is successful in compromising \(t + 1\) polynomial shares of any polynomial, then all the links formed using this polynomial becomes insecure. Since it is very difficult to determine the exact value of \(E(s)\), degree of damage caused by capture of \(s\) nodes, a theoretical upper bound for \(E(s)\) is derived.

Let \(L_s\) denote the number of links exposed on compromise of \(s\) nodes and \(T\) be the total number of links before compromise of \(s\) nodes. Therefore, \(E(s) = \frac{L_s}{T}\), where \(T\) is computed from Theorem 1. \(L_s\) depends on the number of polynomials compromised and the number of links broken due to every compromised polynomial. Since an exact value for \(T\) is known, determining the upper bound for \(E(s)\) reduces to the determining the upper bound for \(L_s\).

4.2.1. Derivation for \(E(s)\)

The following theorem on the number of secured links formed by a polynomial is useful in determining the upper bound for \(L_s\).

**Theorem 2.** Let \(f\) be any polynomial from the polynomial key pool \(F\) which is shared by \(n\) nodes. If \(\rho\) is the radius, then the number of links formed due to the polynomial \(f\) is given by,

\[
L_f(n, \rho) = \rho n - \frac{\rho(\rho + 1)}{2}
\]

**Proof.** Let \(u_1, u_2, \ldots, u_n\) be the nodes possessing the polynomial shares of the polynomial \(f\), with nodes \(u_1\) and \(u_n\) lying on the sides of the \(m \times m\) square grid (i.e., the end points) and \(u_i\) is a physically one hop away from \(u_{i-1}\) and \(u_{i+1}\), \(i = 2, 3, \ldots, n - 1\). Since every node has a radius \(\rho\), \(u_i\) can establish links with \(u_{i-1}, u_{i+1}\) using the polynomial share corresponding to the polynomial \(f\). Hence, \(u_i\) has \(\rho\) key-sharing neighbors i.e., its degree in the key-sharing graph for polynomial \(f\), thus constructed is \(\rho\). Similarly, \(u_n\) also has degree \(\rho\). Likewise, \(u_2\) and \(u_{n-1}\) have degree \(\rho + 1\) each. Continuing in this manner, nodes \(u_\rho\) and \(u_{n-\rho}\) have degree of \(2\rho - 1\) each. The remaining middle \(n - 2\rho\) nodes have a degree of \(2\rho\) each. By handshaking lemma, the total number of edges/links is given by,
Let $C_p$ be a set of compromised polynomials. 

\[ L_T(n, \rho) = \frac{2^p(n-2\rho)+\sum_{x\in C_p}(2^p-1)}{2} = \rho(n-2\rho) + 2\rho + \rho - \frac{\rho(p+1)}{2} = \rho n - \frac{\rho(p+1)}{2} \]

Let $C_p$ be a set of compromised polynomials. \( L_T(n, \rho) \) is the maximum number of nodes which can share a polynomial in $m$. 

\[ L_T(n, \rho) \leq L_T(m, \rho), \forall f \in F \]

\[ L_T \leq \sum_{f \in C_p} L_T(m, \rho) = |C_p| \cdot L_T(m, \rho) \]

Generally, it is assumed that only a small number of nodes are captured. Hence, we assume that $s \leq (t+1)m$ and adopt the selective capture of nodes strategy to derive a tight upper bound on $|C_p|$. As large number of polynomials is revealed on capturing very small number of nodes, the selective capture of nodes strategy, described in section 4.1, is pursued. Since $s \leq (t+1)m$, in the worst case, the attacker will execute the first step of the procedure resulting in the exposure of large number of polynomials in the network. Since the strategy is well-defined and simple, we present the following theorem, without proof, to furnish the maximum value on $|C_p|$.

**Theorem 3.** If $s$ number of nodes are captured at random, where $s = x(t+1) + i$ and $s \leq (t+1)m$. Following the selective capture of nodes strategy, then the number of nodes captured is given by,

\[ |C_p| = \begin{cases} 
2x - 1, & 1 \leq x \leq t \\
2x, & 1 \leq x \leq t - 1 \\
2t + i, & x = t \\
3x, & t + 1 \leq x \leq m - 2 \\
3x + 1, & t + 1 \leq x \leq m - 2 \\
3x + i + 1, & x = m - 1 \\
\end{cases} \]

Now, it is possible to give a theoretical upper bound for $E(s) = \frac{L_T}{7} \leq \frac{|C_p| L_T(m, \rho)}{7}$, where $|C_p|$ and $L_T(m, \rho)$ can be determined by Theorem 3 and 2 respectively, and $\tau = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} |A_p^{(i,j)}|$ where $|A_p^{(i,j)}|$ is given by Theorem 1.

Table 3 illustrates the comparative study of the resilience of the proposed scheme with the $TD(k, r)$ scheme against random capture of nodes. The practical values for $TD(k, r)$ scheme, with square RF region, $E_{TP}(s)$ and for Lee distance approximation, $E_{TP}(s')$ has been extracted from Table 2 of paper in [17]. Experimental values are obtained by taking average for 100 runs.

<table>
<thead>
<tr>
<th>$r = m$</th>
<th>Storage</th>
<th>$\rho$</th>
<th>$s$</th>
<th>$E_{TP}(s)$</th>
<th>$E_{TP}(s')$</th>
<th>$s'$</th>
<th>$E_p(s')$</th>
<th>Upper bound on $E_p(s')$</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>$k = 15, t = 3$</td>
<td>7</td>
<td>5</td>
<td>0.1990</td>
<td>0.2006</td>
<td>15</td>
<td>0.0013</td>
<td>0.1066</td>
</tr>
<tr>
<td>23</td>
<td>$k = 15, t = 3$</td>
<td>5</td>
<td>5</td>
<td>0.1981</td>
<td>0.2008</td>
<td>15</td>
<td>0.0018</td>
<td>0.1044</td>
</tr>
<tr>
<td>31</td>
<td>$k = 20, t = 4$</td>
<td>7</td>
<td>5</td>
<td>0.1526</td>
<td>0.1516</td>
<td>50</td>
<td>0.0117</td>
<td>0.1750</td>
</tr>
<tr>
<td>31</td>
<td>$k = 25, t = 5$</td>
<td>7</td>
<td>5</td>
<td>0.1528</td>
<td>0.1513</td>
<td>50</td>
<td>0.0018</td>
<td>0.1400</td>
</tr>
<tr>
<td>37</td>
<td>$k = 30, t = 6$</td>
<td>7</td>
<td>5</td>
<td>0.1289</td>
<td>0.1283</td>
<td>75</td>
<td>0.0020</td>
<td>0.1426</td>
</tr>
<tr>
<td>53</td>
<td>$k = 49, t = 11$</td>
<td>7</td>
<td>5</td>
<td>0.0913</td>
<td>0.0915</td>
<td>250</td>
<td>0.0011</td>
<td>0.2939</td>
</tr>
<tr>
<td>53</td>
<td>$k = 49, t = 11$</td>
<td>7</td>
<td>10</td>
<td>0.1756</td>
<td>0.1736</td>
<td>500</td>
<td>0.1342</td>
<td>0.6025</td>
</tr>
</tbody>
</table>

When $s \in [0, 10]$, the proposed QPSN scheme offered very high resilience (and hence, insignificant values for $E_p(s)$) against this attack. Hence, higher number of compromised nodes is considered to tabulate the resilience of the proposed scheme, $E_p(s)$. From Table 3, it can be inferred that the resilience, against random node capture, of proposed scheme clearly outperforms that of $TD(k, r)$ scheme. For a $53 \times 53$ grid network, removal of just 10 nodes disrupts 17% of the network links in the $TD(k, r)$ scheme whilst even after the removal of 500 nodes from the same grid network; only 13% of the links are only broken in the proposed scheme. Furthermore, on increasing the memory storage in both the schemes marginally, the proposed QPSN scheme exhibits drastic improvement in its resilience in comparison with the $TD(k, r)$ scheme. Overall, the proposed scheme offers better resilience against any form of attack than the $TD(k, r)$ scheme on a grid based wireless sensor network.

5. Conclusion and Future Work
In this paper, a scalable KPS has been proposed using symmetric bivariate polynomial key pool with an objective to improve resilience. Each node stores four polynomial shares and an efficient shared key discovery phase is devised. Also, the number of key-sharing neighbors of all nodes, the number of nodes to be selectively captured to reveal the key pool, and a theoretical upper bound for the resilience against random node capture has been mathematically proven. Experimental results demonstrate the supremacy of this scheme over the existing schemes.

There are several directions for future work. Amidst the fact that the proposed scheme assures 2-hop connectivity, 1-hop connectivity is low. Moreover, as the expression is dependent only on \( \rho \), increasing the storage does not aid in improving the connectivity. We would like to work towards incorporating additional features to the design which enables better connectivity. Mounting the selective capture of nodes attack on the \( DD(m,r) \) scheme and evaluating its resilience is another study of interest.

References