# A Hierarchical Deterministic Key Pre-distribution for WSN Using Projective Planes

Sarbari Mitra, Ratna Dutta, and Sourav Mukhopadhyay

Indian Institute of Technology, Kharagpur, India
sarbarimitra@gmail.com, {ratna,sourav}@maths.iitkgp.ernet.in

**Abstract.** We present a deterministic key pre-distribution scheme using projective planes where the nodes are organised hierarchically through a structure of $(p^2+p)$-nary tree, where $p$ is prime. Our scheme is incumbent to more efficient resilience and connectivity compared to the existing schemes. Each node in our scheme requires to store significantly less number of keys. Furthermore, any number of nodes can be intrinsically inserted in the system by attributing a very few keys to the recently introduced nodes only. Another interesting feature of our scheme is that such node insertions are done without interfering the normal functioning of the existing organised network.

**Keywords:** t-design, Steiner system, projective planes, key pre-distribution.

## 1 Introduction

Wireless Sensor Network (WSN) consists of a large number of wireless sensor nodes, which are typically small mobile devices with limited memory and computation power to transmit data within a specified range. Sensor nodes are usually plotted in sensitive regions (e.g. military area, hospital etc.) to gather information through sensors and transmit the collected data by communicating among themselves or with some other source. The process of assigning keys to the nodes prior to their deployment in the target region is termed as key pre-distribution. Usually keys are chosen from a large key-pool and then they are loaded at the nodes.

Key pre-distribution to the sensor nodes has drawn the attention of researchers over the years. Till date combinatorial design is one of the most commonly used mathematical tools for the key pre-distribution. Key pre-distribution in wireless sensor network can be broadly categorized into the following [11]:

- *Probabilistic*: Keys are chosen randomly from the key-pool and are assigned to the nodes so that any two nodes are connected (i.e. share a common key) with some definite probability.
- *Deterministic*: Keys are distributed to the nodes following a fixed manner and it can be determined with absolute certainty that which nodes are sharing common keys.
- *Hybrid*: A combination of both probabilistic and deterministic approaches.

D. Simplot-Ryl et al. (Eds.): ADHOCNETS 2011, LNICST 89, pp. 16–31, 2012.

A key pre-distribution scheme involves three main steps [8]:

(a) *Key pre-distribution*: The process of loading secret keys to the sensor nodes.
(b) *Shared key discovery*: Any two nodes wishing to communicate between themselves check whether they have a common key or not. This process is known as shared key discovery.
(c) *Path-key establishment*: If two nodes don't share any common key, they look for neighbouring node(s) sharing a common key with both of them. This is known as path-key establishment.

The salient features of a good key pre-distribution scheme includes the following:

(i) Scalability - whether new nodes can be introduced without much disturbance in the existing set-up;
(ii) Efficiency - less memory, less computation and greater connectivity
(iii) Resilience - how the network is affected when some of the nodes are captured.

The above parameters are conflicting to each other. If one common key is stored at all the nodes, the probability of two nodes sharing a common key is 1, which implies the best connectivity. The memory requirement met as well since the nodes need to store only one key. But the network becomes extremely vulnerable as capture of one single node makes the whole network to cease which leads the resilience to become zero. Another possibility is to store a key for each pair of the nodes. In this case, we get desirable connectivity (probability of any two nodes sharing a common key is 1) and significant resilience as capture of any node will destroy the links of other nodes with the compromised nodes. Whereas the rest of the network will remain undisturbed. The drawback of this method is that each node is required a huge memory to store $N-1$ keys for a network consisting of $N$ nodes. Hence one needs to obtain a trade-off between these parameters.

### 1.1 Previous Work

Eschenauer and Gligor [7] were first to use random key pre-distribution in WSN. Their scheme is known as *basic scheme.* Later Chan, Perrig and Song [5] proposed a modified version of the basic scheme.

Camptepe, Yener [1] were first to introduce combinatorial designs as one of the key pre-distribution techniques. They have considered two combinatorial designs: one is the symmetric $(n^2+n+1, n+1, 1)$-BIBD (or, finite projective plane of order $n$) and the other is generalized quadrangles. The advantage of this deterministic approach is any two nodes share a common key which improves the connectivity of the network to a greater extent. The main drawback of deterministic approach is that the scheme is not scalable as the network size $N$ should satisfy $N \leq n^2+n+1$. If one wants to introduce some new nodes to the network which exceeds the bound, then $n$ has to be raised to the next prime number as the existence of such designs for a non prime power value of n is not certain. This results in a much more larger network than required, and the key-chains at each node have to be changed. It is also observed in [2] that the generalized quadrangles induce better scalable network and provide better resilience than finite projective planes. To improve the scalability, authors have proposed a hybrid scheme in [2]. In this scheme, keys are assigned to the major part of the network according to projective plane, (i.e., following a deterministic approach) and the remaining nodes or newly joined nodes (which could not be accommodated by projective planes) get keys in a completely random manner. This improves the resilience and scalability. However, the probability of any two nodes sharing a common key is reduced.

In 2005, Lee and Stinson [8] proposed a scheme on group-divisible design or Transversal design. It is noticed that the expected proportion that any two nodes can communicate directly is 0.6 and the same for two nodes communicating directly or via intermediate nodes is almost 0.99995. Chakrabarti et al. [3] showed by an example that out of 2401 nodes in a network, if only 10 nodes are captured, then 18% of the links will be destroyed. This is the main disadvantage of this scheme. Later, in 2008, quadratic schemes were developed in [10] based on Transversal designs and the method described in [8] was referred as linear schemes. This work suggests that the quadratic scheme provides best resilience unless the number of compromised nodes is high. Quadratic schemes in general provides better connectivity than linear schemes. Both linear and quadratic schemes are preferred over 2-composite scheme [5] if shared key discovery is taken into consideration.

In 2005, Chakrabarti et al. [3] proposed a probabilistic key pre-distribution scheme. They have constructed the blocks as proposed by Lee et al. [8]. The sensor nodes are then formed by merging blocks randomly. This increases the chance of sharing common keys between two nodes. The scheme in [3] provides better resilience as compared to the Lee-Stinson scheme [8] at the cost of large key-chain size in each node.

3-design is considered to be the underlying combinatorial design of the key pre-distribution scheme proposed by Dong et al. in [6]. Keys are assigned to the sensor nodes in the network by Möbius Planes. This scheme provides better connectivity than the scheme proposed by Lee-Stinson [10] and better memory requirement as compared to Camptepe-Yener scheme [1]. The prime drawback of the scheme is that resilience reduces rapidly with the increasing number of compromised nodes.

Ruj et al. [11] proposed a deterministic key pre-distribution scheme based on Partially Balanced Incomplete Block Design. The authors claim that this scheme gives better resilience than that of [8] storing less than $\sqrt{N}$ keys to the nodes where $N$ is the network size. But to store that many keys to the nodes, for a very large network is also expensive.

### 1.2 Our Contribution

We propose a storage-efficient key pre-distribution scheme adapting a deterministic approach. We have used a typical Steiner system as our basic combinatorial design for key pre-distribution. We emphasize that apart from storage efficiency, our design also provides better resilience and reasonable connectivity as compared to the existing schemes. Nodes are arranged using a hierarchical tree structure. The whole network is divided into $(p^2+p+1)$ sub-networks each of which forms a $(p^2+p)$-nary tree-hierarchical structure, where $p$ is prime. All the nodes in the same sub-network are connected directly or via a chain of intermediate nodes. Moreover, two nodes from two different sub-networks can establish a key-path via *level* 1 nodes which means that all the nodes in the network are connected.

We claim that our scheme provides much better resilience even in the worst possible condition, which is supported by our experimental results provided in the paper. As resilience and connectivity are contradictory in nature, we choose the order of the projective plane suitably to meet the requirement for both the resilience and connectivity. Consumption of power and memory should be minimal since there will be no external supply of power to the nodes once they are deployed. Increased memory consumption will decrease the power available for computation. Storing significantly less number of keys to the nodes is not only cost-effective but also it leaks less information (in the form of keys) when the nodes are captured. Unlike the existing key pre-distribution schemes, our scheme is flexible in the sense that insertion of a large number of nodes can be done by adding only a few keys to the newly joined nodes without disturbing the previously assigned nodes.

Rest of the paper is organized as follows: section 2 includes preliminaries, we discuss the proposed scheme in section 3, which is explained in detail with a particular example in section 4. Section 5 and 6 provide obtained results and performance respectively followed by concluding remarks in section 7.

## 2 Preliminaries

**Definition 2.01.** *A design is defined as a pair* (X, A) *such that (i)* X *is a set of points or elements, (ii)* A *is a subset of the power set of X (i.e. Collection of non-empty subsets of X)*

**Definition 2.02.** *A t-design is defined as a* $t$ - $(v,k,\lambda)$ *block design (with* $t \leq k \leq v$*) such that the following are satisfied (i)* X $= v$ *, (ii) each block contains k points, (iii) for any set of t points there are exactly* $\lambda$ *blocks that contain all these points.*

**Definition 2.03.** *A t-design with* $t = 2$ *is known as* $(v,k,\lambda)$*-Balanced Incomplete Block Design[BIBD].*

**Example 2.01.** *A* $(10,4,2)$*-BIBD has* $X = \{0,1,2,3,4,5,6,7,8,9\}$*,*
$A = \{(0,1,2,3);(0,1,4,5);(0,2,4,6);(0,3,7,8);(0,5,7,9);(0,6,8,9);(1,2,7,8);$
$(1,3,6,9);(1,4,7,9);(1,5,6,8);(2,3,5,9);(2,4,8,9);(2,5,6,7);(3,4,5,8);$
$(3,4,6,7)\}$

**Definition 2.04.** *A t-design with $\lambda = 1$ is known as Steiner system.*

**Example 2.02.** *A* $(9,3,1)$*-design has* $X = \{1,2,3,4,5,6,7,8,9\}$*,* $A = \{(1,2,3); (4,5,6); (7,8,9); (1,4,7); (2,5,8); (3,6,9); (1,5,9); (1,6,8); (2,4,9); (2,6,7); (3,4,8); (3,5,7)\}$

**Definition 2.05.** *Finite symmetric projective plane of order n is defined as a pair of set of $n^2 + n + 1$ points and $n^2 + n + 1$ lines, where each line contains $n + 1$ points and each point occurs in $n + 1$ lines.*

**Example 2.03.** *Projective plane of order 2, a* $(7,3,1)$*-BIBD, which is also known as the Fano plane has* $X = \{1,2,3,4,5,6,7\}$*,* $A = \{(1,2,3); (1,4,7); (1,5,6); (2,4,6); (2,5,7); (3,4,5); (3,6,7)\}$.

**Example 2.04.** *Projective plane of order 3, a* $(13,4,1)$*-BIBD is:*
$X = \{1,2,3,4,5,6,7,8,9,10,11,12,13\}$
$A = \{(1,2,3,4); (1,5,6,7); (1,8,9,10); (1,11,12,13); (2,5,8,11); (2,6,9,13); (2,7,10,12); (3,5,10,13); (3,6,8,12); (3,7,9,11); (4,5,9,12); (4,6,10,11); (4,7,8,13)\}$.

Any design $(X, A)$ can be mapped to a sensor network where the elements of the set $X$ represent the keys and the blocks of the set $A$ correspond to sensor nodes.

## 3 Our Generalized Scheme

In this section we shall discuss how a projective plane can be mapped to the network. We consider a particular projective plane of order $p$, ($p$ is considered to be prime so as to ensure the existence of the projective plane), as our basic building block design. Initially we label all the nodes by $\mathsf{Node}[1]$, $\mathsf{Node}[2]$, $\mathsf{Node}[3]$, $\cdots$, $\mathsf{Node}[N]$; where $N$ is the total number of nodes. Similarly we label the keys as $\mathsf{key}[1]$, $\mathsf{key}[2]$, $\mathsf{key}[3]$, $\cdots$, $\mathsf{key}[K]$; where $K$ is the size of the key-pool. A set of $p^2 + p + 1$ keys are chosen from the large key pool and then distributed to a set of $p^2 + p + 1$ nodes so that each node gets $p + 1$ keys and each key is assigned to $p + 1$ nodes. Without loss of generality, we assume that the nodes chosen are given by $\mathsf{Node}[1]$, $\mathsf{Node}[2]$, $\mathsf{Node}[3]$, $\cdots$, $\mathsf{Node}[\mathsf{p}^2 + \mathsf{p} + 1]$. We call these nodes as *level* 1 nodes, the keys used in *level* 1 are termed as *level* 1 keys. The set of nodes and keys used in *level* 1 thus forms a Steiner system. In *level* 2, $\mathsf{Node}[1]$ forms a Steiner system with $p^2 + p$ new nodes. We know that total $p^2 + p + 1$ keys are required to form a Steiner system. As there are $p + 1$ keys already stored in the first node, we need only $p^2$ nodes to complete the Steiner system. In the similar manner, Steiner systems are produced corresponding to each of the *level* 1 nodes. All the nodes that are included in *level* 2 are referred to as *level* 2 nodes and all the keys that are used for the first time in *level* 2 are called *level* 2 keys. Steiner systems are created corresponding to each of the *level* 2 nodes to complete *level* 3. This method is repeated until keys are distributed to all the nodes in the network. This completes the key pre-distribution phase.

## 4 Fano Plane Scheme

We explain the key distribution procedure described above for a Fano plane. Fig.1 illustrates the 2 - $(7, 3, 1)$ Steiner system i.e. the Fano Plane under consideration.

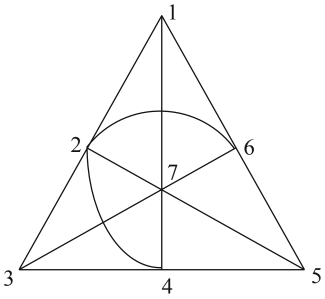

**Fig. 1.** The Fano Plane

Here seven nodes correspond to seven keys and each line represents a sensor node (key chain of the node). This assigns a set of 7 keys to 7 nodes such that all nodes together contain exactly 7 keys and any two are connected by exactly one common key. We label all the nodes and all the keys by $1, 2, 3, 4, \ldots$ for convenience. In *level* 1, seven keys $\{1, 2, 3, 4, 5, 6, 7\}$ are distributed to the first seven nodes as described above. Thus the key-chains assigned to the nodes $1, 2, 3, 4, 5, 6, 7$ are respectively $\{1, 2, 3\}$, $\{1, 4, 7\}$, $\{1, 5, 6\}$, $\{2, 4, 6\}$, $\{2, 5, 7\}$, $\{3, 4, 5\}$, $\{3, 6, 7\}$.

The Steiner systems corresponding to all the *level* 1 nodes are explicitly described in Table 1.

In *level* 3, each of *level* 2 nodes are attached to six new *level* 3 nodes to form a Steiner system and the corresponding key chain is chosen in the same manner, i.e., keeping the first three keys same as the *level* 2 keys contained by *level* 2 nodes and adding four new *level* 3 keys. This process is repeated until keys are assigned to all the nodes in the network. We provide the algorithm Key Pre-Distribution for assigning keys to the tree hierarchy as explained above. We consider a hierarchical structure using a 6-nary tree for key pre-distribution.

Let us consider a network having maximum $N$ nodes. Let $K$ denote the total key-pool and $l$ denote the maximum level in the hierarchical tree structure. The three keys assigned to Node[i] are stored in Node[i][1], Node[i][2], Node[i][3]. Choose $\{u_1, u_2, u_3\} \in_R K$, where the symbol $\in_R$ denotes random selection.

**Table 1.** Components of Steiner systems formed by *level* 1 nodes

| Node | Node-set | Key-set |
|---|---|---|
| node 1 | $\{1,8,9,10,11,12,13\}$ | $\{1,2,3,8,9,10,11\}$ |
| node 2 | $\{2,14,15,16,17,18,19\}$ | $\{1,4,7,12,13,14,15\}$ |
| node 3 | $\{3,20,21,22,23,24,25\}$ | $\{1,5,6,16,17,18,19\}$ |
| node 4 | $\{4,26,27,28,29,30,31\}$ | $\{2,4,6,20,21,22,23\}$ |
| node 5 | $\{5,32,33,34,35,36,37\}$ | $\{2,5,7,24,25,26,27\}$ |
| node 6 | $\{6,38,39,40,41,42,43\}$ | $\{3,4,5,28,29,30,31\}$ |
| node 7 | $\{7,44,45,46,47,48,49\}$ | $\{3,6,7,32,\ 33,34,35\}$ |

**Algorithm : Key Pre-Distribution**

```
$i := 0$;
Node $[1][1] := u_1$,  Node $[1][2] := u_2$,  Node $[1][3] := u_3$;
procedure Key Pre-Distribution $(u_1, u_2, u_3)$
$X := \{u_1, u_2, u_3\}$;
Choose $\{u_4, u_5, u_6, u_7\} \in_R B$ where $B \subseteq K - X$, $B$ is the set of unused keys
$X := X \cup \{u_4, u_5, u_6, u_7\}$;

    $j := 6i + 2$;
        Node $[j][1] := u_1$,      Node $[j][2] := u_4$,      Node $[j][3] := u_7$;
        Node $[j+1][1] := u_1$,    Node $[j+1][2] := u_5$,    Node $[j+1][3] := u_6$;
        Node $[j+2][1] := u_2$,    Node $[j+2][2] := u_4$,    Node $[j+2][3] := u_6$;
        Node $[j+3][1] := u_2$,    Node $[j+3][2] := u_5$,    Node $[j+3][3] := u_7$;
        Node $[j+4][1] := u_3$,    Node $[j+4][2] := u_4$,    Node $[j+4][3] := u_5$;
        Node $[j+5][1] := u_3$,    Node $[j+5][2] := u_6$,    Node $[j+5][3] := u_7$;
$p := 1$; $r := 1$; $s := 0$; $N := r + s$
while $(p < l)$ do
        $r := r + 6^p$; $s := s + 6^{p-1}$;
        $p++$;
        for $i := N$ to $(r + s - 1)$ do in parallel
            call Key Pre-Distribution (Node[i][1], Node[i][2], Node[i][3])
        end do
    $N := r + s$;
end do
end Key Pre-Distribution
```

The detailed hierarchical tree structure (upto *level* 3) has been depicted in Fig.2

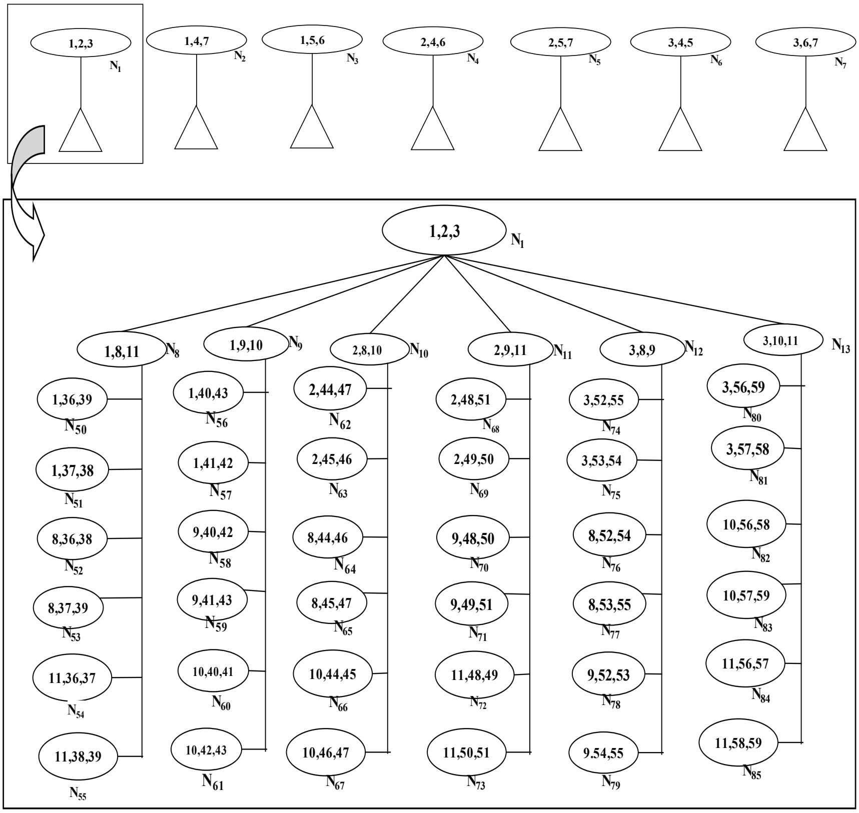

**Fig. 2.** The nodes, denoted by $N_i$ and corresponding key-chains upto *level* 3

## 5 Results

**Theorem 5.01.** *Number of nodes in level $j$ is $n_j = (p^2+p+1)(p^2+p)^{j-1}$ , $\forall j \in \{1, \ell\}$.*

*Proof*: The result holds trivially for $i = 1$.

In *level* 2, each node from *level* 1 forms a projective plane with new $p^2+p$ *level* 2 nodes. Let us refer to these $p^2+p$ nodes as the children of that *level* 1 node. Similarly, all *level* 1 nodes have $p^2+p$ children. Following the same pattern all *level* 2 nodes have $p^2+p$ children in *level* 3 and this pattern continues. Therefore, we note that the nodes are distributed in the form of $(p^2+p)$-nary trees corresponding to each of $p^2+p+1$ nodes in *level* 1. Thus the nodes form $p^2+p+1$ numbers of $(p^2+p)$-nary trees. Hence the result follows. □

**Corollary 5.01.** *Total number of nodes in the network is* $N = \frac{(p^2+p+1)}{(p^2+p-1)}\{(p^2+p)^l - 1\}$

Proof: As the levels of the nodes are exhaustive and disjoint, we have $N = n_1 + n_2 + ... + n_l$, where $l$ represents the total number of levels in the network. Consequently, total number of nodes in the network is $N = \sum_{j=1}^{l}(p^2+p+1)(p^2+p)^{j-1} = \frac{(p^2+p+1)}{(p^2+p-1)}\{(p^2+p)^l - 1\}$. This completes the proof. □

**Theorem 5.02.** *Number of keys that are used for the first time in level* $j$ *is* $k_j = p^j(p+1)^{j-2}(p^2+p+1), \quad \forall\, j \geq\ 2$ *and* $k_1 = p^2+p+1$.

**Proof:** There are $p^2+p+1$ keys in each of the projective planes. As there is only one projective plane is *level* 1, number of keys in *level* 1 is $p^2+p+1$. We observe that when one node from *level* $i$ forms a projective plane with new nodes from *level* $i+1$, it requires $p^2$ new nodes to complete the projective plane as $p+1$ keys are already stored in that node. So number of keys required in *level* $i$ is $p^2$ corresponding to each Steiner system to be formed. If $k_j$ denotes the number of keys in *level* $j$, then $k_j = p^2 \times n_{j-1} = p^j(p+1)^{j-2}(p^2+p+1)$. □

**Corollary 5.02.** *Total number of keys in the network is* $K = (p^2+p+1)\ [1 + \frac{p^2}{p^2+p-1}((p^2+p)^{\ell-1} - 1)]$

**Proof:** As the keys appearing for the first time in a particular *level* are exhaustive and disjoint, we have $K = k_1 + k_2 + ... + k_l$, where $l$ represents the total number of levels in the network. Hence, total number of keys required in the network is $K = (p^2+p+1)\sum_{j=1}^{l} p^j(p+1)^{j-2}$. The result follows on simplification. □

**Theorem 5.03.** *Number of nodes to which a level i key is assigned to, is given by* $N_i = \frac{p+1}{p-1} \times \{p^{l+1-i} - 1\}$, *where* $l$ *denotes the total number of levels present in the network.*

**Proof:** The keys that appear for the first time in *level* $i$ is contained in only one Steiner system and hence goes to $p+1$ nodes in *level* $i$. In the next *level*, i.e. in $(i+1)^{th}$ *level*, that key goes to each of the $p+1$ Steiner systems corresponding to each of the previous *level* nodes and in each system, the key is contained in $p$ new nodes. Thus we observe that the nodes to which a *level* $i$ key is contained, form $(p+1)$ number of $p$-nary trees with their roots in *level* $i$. Therefore, the number of nodes to which a *level* $j$ key is assigned to is $\sum_{i=j}^{l}(p+1)p^{i-j}$. Hence the result follows. □

**Theorem 5.04.** *Number of nodes containing key-chain as* (*level* $j$, *level* $i$, $\cdots$, *level* $i$) *in level* $i$ *is given by*

$$\begin{cases} (p^2+p+1) \times p^{i-1} \times (p+1)^j \ \ for\ j\ =\ 1; \\ (p^2+p+1) \times p^i \times (p+1)^{j-1} \quad \forall\ j\ = 2,\ 3,\ \cdots,\ (i-1). \end{cases}$$

**Proof:** We observe that all the nodes in *level* 1 contain $p+1$ *level* 1 keys. In *level* 2, the nodes contain one *level* 1 and $p$ *level* 2 keys. In *level* 3 we get two types of nodes depending on the key distribution. All *level* 3 nodes contain exactly $p$ *level* 3 keys and the remaining key is from *level* 1 or *level* 2, we call the nodes Type 1 and Type 2 accordingly. We also notice that the ratio of Type 1 and Type 2 nodes is $1 : p$ as shown in the Fig. 3.

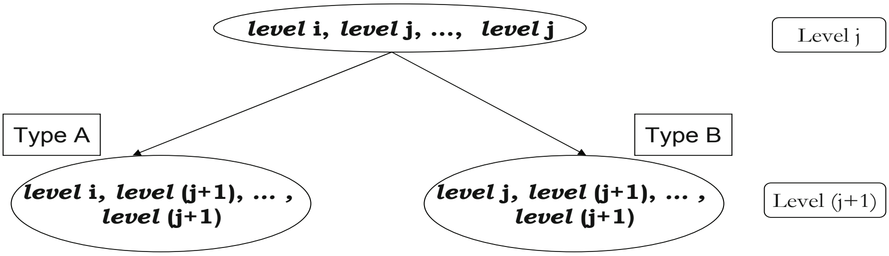
level i, level j, ..., level j
Level j
Type A
Type B
level i, level (j+1), ... , level (j+1)
level j, level (j+1), ... , level (j+1)
Level (j+1)

**Fig. 3.** Type A and Type B nodes in level $j+1$ are in the ratio $1 : p$

In *level* 4 we get three types of nodes: Type 1 with key chain (*level* 1, *level* 4, $\cdots$, *level* 4); Type 2 with key chain (*level* 2, *level* 4, $\cdots$, *level* 4); Type 3 with key chain (*level* 3, *level* 4, $\cdots$, *level* 4); and they are in the ratio $1 : p : p(p+1)$. Continuing this process upto *level* $i$, the nodes are in the ratio $1 : p : p(p+1) : p(p+1)^2 : \cdots : p(p+1)^{i-3}$ . Sum of these ratios is $(p+1)^{i-2}$. Their corresponding proportions are given in Table 2. Now, we know that the number of nodes in *level* $i$ is given by $n_i = (p^2+p+1)(p^2+p)^{i-1}$. Multiplying the proportions of different types of nodes with $n_i$, we get the individual number of them as provided in Table 2. □

Let us now discuss the effect of adversarial interference. We try to estimate how the network gets disturbed when nodes are captured by the opponent. Since the nodes are deployed in hostile regions, they are likely to be captured frequently. When a node gets captured, all the keys contained in it are also compromised. As a result, the set of links which uses keys from the captured nodes are *destroyed* completely. Moreover, the node which contains any compromised key cannot use it for any further communication, though the other keys stored at them are still secret. We refer to these nodes as (partially) *affected* nodes. We notice that capture of each node reveals only $p$ keys at a time, which is very less portion of the total key-pool. Now we analyse below what portion of the nodes and the links are affected when a single node is captured.

**Theorem 5.05.** *Let $\phi(i,l)$ be the number of affected nodes when an $i^{th}$ level node is compromised, $l$ being the total number of nodes (and keys) present in the network. Then*

$$\phi(i,l) = \frac{p+1}{p-1}\left(p^{l-i+2}-1\right) - \frac{p^l}{(p+1)^{i-3}} - p$$

**Table 2.** Proportion of different types of nodes

| Type | Key-chain | Proportion | Number of Nodes |
|---|---|---|---|
| 1 | (*level* 1, *level* $i$, $\cdots$ , *level* $i$) | $1/(p+1)^{i-2}$ | $p^{i-1}(p+1)(p^2+p+1)$ |
| 2 | (*level* 2, *level* $i$, $\cdots$ , *level* $i$) | $p/(p+1)^{i-2}$ | $p^{i}(p+1)(p^2+p+1)$ |
| 3 | (*level* 3, *level* $i$, $\cdots$ , *level* $i$) | $p/(p+1)^{i-3}$ | $p^{i}(p+1)^2(p^2+p+1)$ |
| 4 | (*level* 4, *level* $i$, $\cdots$ ,*level* $i$) | $p/(p+1)^{i-4}$ | $p^{i}(p+1)^3(p^2+p+1)$ |
| $j$ | (*level* $j$, *level* $i$, $\cdots$ , *level* $i$) | $p/(p+1)^{i-j}$ | $p^{i}(p+1)^{j-1}(p^2+p+1)$ |
| $i-1$ | (*level* $i-1$, *level* $i$, $\cdots$ , *level* $i$) | $p/(p+1)$ | $p^{i}(p+1)^{i-2}(p^2+p+1)$ |

**Proof :** We note that all the *level* $i$ nodes contain $p$ *level* $i$ keys and the remaining key may belong to any of the previous $(i-1)$ levels. Depending on this we divide the nodes into types. Therefore there are $(i-1)$ types of *level* $i$ nodes.

Let $n_{ij}$ denote the number of nodes affected when a *level* $i$ node of Type j, (for $j = 1, 2, ...i-1$) is compromised. A *level* $i$ node of Type j contains exactly one *level* $j$ key and $p$ *level* $i$ keys. A *level* $i$ node is attached to $\frac{p+1}{p-1} \times \{p^{l+1-i} - 1\}$ nodes. Hence we have, $n_{i1} = \{1 \times \left(p^l - 1\right) + p \times \frac{p+1}{p-1} \left(p^{l+1-i} - 1\right)\} - p$. Here $p$ is subtracted as a set of $p+1$ keys is contained in exactly one node (i.e., the compromised node here), which we count once for every key repeating it $p$ extra times. Similarly, we have

$$n_{ij} = \{1 \times \left(p^{l+1-j} - 1\right) + p \times \frac{p+1}{p-1} \left(p^{l+1-i} - 1\right)\} - p, \quad \forall j \in 1, 2, \cdots, i-1.$$

Simplified expressions for $n_{ij}$ are listed in Table 3.

The proportions of the different types of nodes given in Table. 2. Thus we have $\phi(i,l) = \frac{n_{i1} \times 1}{(p+1)^{i-2}} + \frac{n_{i2} \times p}{(p+1)^{i-2}} + \cdots + \frac{n_{ij} \times p}{(p+1)^{i-j}} + \cdots \frac{n_{i(i-1)} \times p}{(p+1)}$. Substituting the values for $n_{ij}$, from Table 3, we get the desired expression. □

*Remark 1. If we wish to know the effect of the adversarial attack on the network, $\phi(i,l)$ will provide the average number of the affected nodes (and hence their proportion) when an $i^{th}$ level node is compromised, i.e., $\phi(i,l)$ gives an estimate for resilience involving affected nodes. Resilience involving destroyed links is discussed in the following section.*

**Table 3.** Values of $n_{ij}$

| | |
|---|---|
| $n_{i1}$ | $\frac{p+1}{p-1}(p^{l}-1)+\frac{p+1}{p-1}(p^{l-i+2}-p)-p$ |
| $n_{i2}$ | $\frac{p+1}{p-1}(p^{l-1}-1)+\frac{p+1}{p-1}(p^{l-i+2}-p)-p$ |
| $n_{i3}$ | $\frac{p+1}{p-1}(p^{l-2}-1)+\frac{p+1}{p-1}(p^{l-i+2}-p)-p$ |
| $n_{i4}$ | $\frac{p+1}{p-1}(p^{l-3}-1)+\frac{p+1}{p-1}(p^{l-i+2}-p)-p$ |
| $n_{ij}$ | $\frac{p+1}{p-1}(p^{l-j+1}-1)+\frac{p+1}{p-1}(p^{l-i+2}-p)-p$ |
| $n_{i(i-1)}$ | $\frac{p+1}{p-1}(p^{l-i+2}-1)+\frac{p+1}{p-1}(p^{l-i+2}-p)-p$ |

## 6 Performance

We determine resilience mathematically by the following formula put forward by Lee-Stinson [8]:

$$fail(s) \;=\; 1 \;-\; \prod_{i=1}^{l}\left(1 \;-\; \frac{N_i-2}{N-2}\right)^{s_i}$$

where $fail(s)$ denotes the portion of total link failure when $s$ nodes are compromised, $N_i$ denotes the number of nodes to which a *level* $i$ key is assigned to, $s_i$ is the number of compromised nodes in the $i^{th}$ *level.* Therefore, $\sum_{i=1}^{l} s_i = s$.

Unlike other key pre-distribution schemes based on combinatorial designs, our scheme is heterogeneous, i.e., the nodes are not distributed uniformly. Therefore it is not possible to present the exact value of $fail(s)$. Instead we provide the average value of $fail(s)$.

First, we would like to deduce how the network accomplishes its function when the order of the underlying projective plane is altered (let us consider a network composed of two levels so that for $p = 2$, $T_{nodes} = 49$, for $p = 3$, $T_{nodes} = 169$, for $p = 5$, $T_{nodes} = 961$ and for $p = 7$, $T_{nodes} = 3249$, where $T_{nodes}$ represents total number of nodes occurring in the network). If we wish to estimate their comparative performances by retaining the number of compromised nodes, it would culminate into inadequate consequences. This is due to the evidential information that if we regard only 10 nodes to be compromised then a major portion of the network is interfered for $p = 2$. In contrast to this, for $p = 7$ a very negligible portion of the network is disrupted from normal functioning. To keep up the uniformity on the compromised nodes, instead of a fixed number, we assume that the number of compromised nodes is a certain percentage of the total number of nodes present in the network.

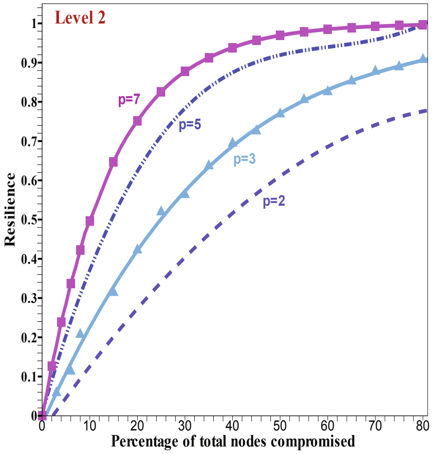

**Fig. 4.** Networks consisting two levels

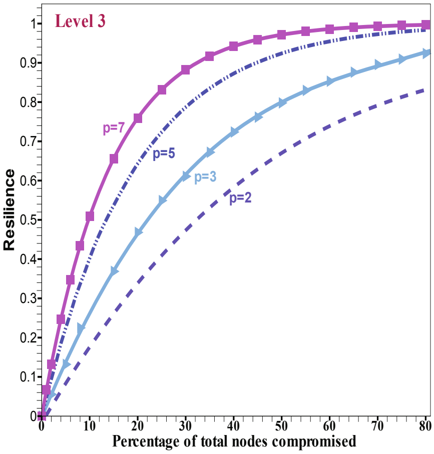

**Fig. 5.** Networks consisting three levels

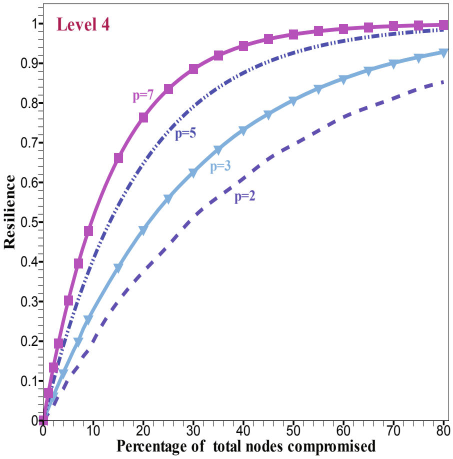

**Fig. 6.** Networks consisting four levels

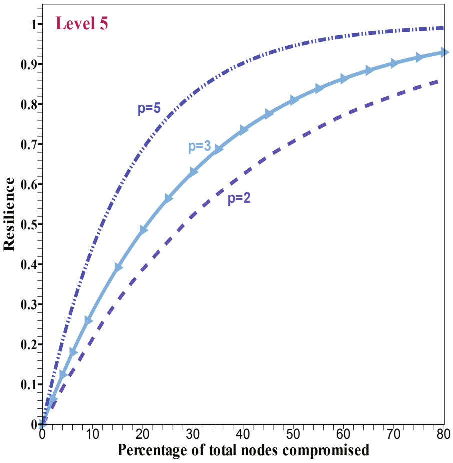

**Fig. 7.** Networks consisting five levels

The supportive figures Fig.4 - Fig.7 relate to the comparison between the performance of our scheme making use of different orders of projective planes when the maximum level in the network is predetermined. The four figures Fig.4, Fig.5, Fig.6 and Fig.7 describe and justify the comparison graphically for the networks composed of two, three, four and five levels respectively. In these figures, the percentage of total nodes in the network, which is compromised is plotted against the portion of the destroyed links. From these figures it can be observed that for any network, the resilience gets adversely affected with increasing order of the projective planes, when number of levels is kept unaltered.

Next we shall concentrate on comparing the performance of our scheme with other existing schemes.

In Table 4, we provide the comparison based on the performance of our scheme with Lee-Stinson linear scheme [8], Chakrabarti et al. scheme [3], Ruj-Roy scheme [11] and Lee-Stinson quadratic scheme [10], where $T_{nodes}$ denotes total number of nodes in the network and $T_{keys}$ denotes total number of keys present in each node. To keep up $T_{nodes}$ in our scheme comparable with other schemes, we consider $p = 2$, $level = 4$. The details have been mentioned in the table.

**Table 4.** Comparison with some of the existing schemes

| | [8] | [3] | [11] | [10] | Ours |
|---|---|---|---|---|---|
| $T_{nodes}$ | 1849 | 2550 | 2415 | 2197 | 1813 |
| $T_{keys}$ | 30 | $\leq 28$ | 136 | 30 | 3 |
| $fail(10)$ | 0.201070 | 0.213388 | 0.0724 | 0.297077 | 0.010087 |

The comparison between the schemes has been represented graphically through figures Fig. 8 and Fig. 9. In Fig. 8 we demonstrate the comparison of our scheme with Lee-Stinson linear scheme [8], Chakrabarti et al. scheme [3], Ruj-Roy scheme [11] and Lee-Stinson quadratic scheme [10] for a handful (i.e., 1 - 10)

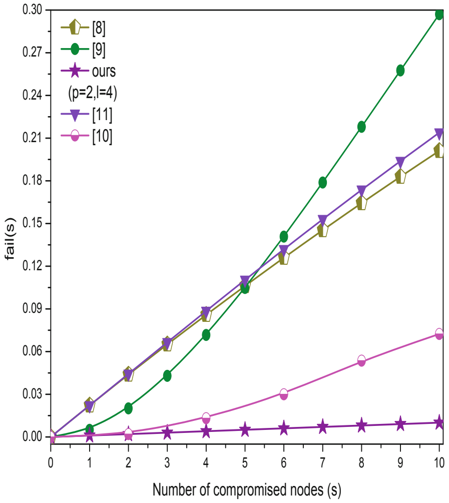

**Fig. 8.** Comparison of resilience with some of the existing schemes for small number of compromised nodes

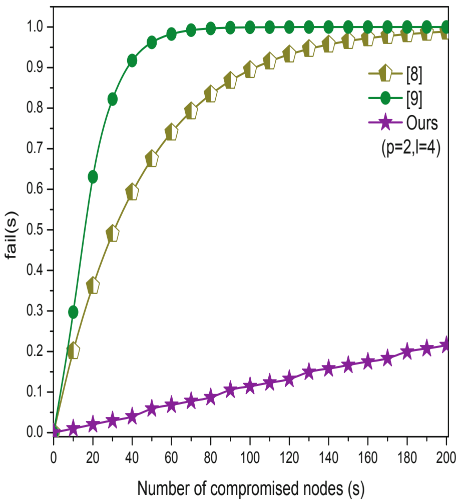

**Fig. 9.** Comparison of resilience with some of the existing schemes for large number of compromised nodes

number of compromised nodes. On the other hand in Fig. 9 we provide the comparison with Lee-Stinson linear scheme [8] and Lee-Stinson quadratic scheme [10] for a large number (i.e., 10 - 200) of compromised nodes. It is very evident from the figures that the networks incorporated on other schemes collapses in no time when compared to ours.

## 7 Discussion and Conclusion

We have proposed a key pre-distribution scheme by applying combinatorial design. The memory prerequisite in each sensor node is appreciably reduced. Further, we perceive that unlike most of the deterministic and combinatorial design based schemes, the proposed scheme sustains scalability, i.e., a number of nodes could be inserted without interfering the present network set-up. The discussed scheme affords reasonable connectivity: any two nodes are connected either directly or via a key-path. We note that it is advantageous to make use of projective planes of small order as they acquire better resilience. However, comparing with other existing schemes, we come across that our scheme offers enhanced resilience for higher order projective planes. Intuitively, we can also declare that the connectivity gets better with increasing order of the projective plane employed as basic building block to design the whole network.

In the present scheme the lower level nodes are more sensitive compared to the higher level nodes. As a future work, we would like to propose a randomized scheme by merging the more sensitive nodes with less sensitive nodes to achieve uniform sensitivity among the nodes in the network, to continue functioning interactively.

## References

1. Çamtepe, S.A., Yener, B.: Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks. In: Samarati, P., Ryan, P.Y.A., Gollmann, D., Molva, R. (eds.) ESORICS 2004. LNCS, vol. 3193, pp. 293–308. Springer, Heidelberg (2004)
2. Camptepe, S.A., Yener, B.: Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks. ACM Trans. Netw. 5(2), 346–358 (2007)
3. Chakrabarti, D., Maitra, S., Roy, B.: A Key Pre-distribution Scheme for Wireless Sensor Networks: Merging Blocks in Combinatorial Design. In: Zhou, J., López, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 89–103. Springer, Heidelberg (2005)
4. Chakrabarti, D., Seberry, J.: Combinatorial Structures for Design of Wireless Sensor Networks. In: Zhou, J., Yung, M., Bao, F. (eds.) ACNS 2006. LNCS, vol. 3989, pp. 365–374. Springer, Heidelberg (2006)
5. Chan, H., Perrig, A., Song, D.X.: Random Key Predistribution Schemes for Sensor Network. In: IEEE Symposium on Security and Privacy, pp. 197–213 (2003)
6. Dong, J., Pei, D., Wang, X.: A Key Predistribution Scheme Based on 3-Designs. In: Pei, D., Yung, M., Lin, D., Wu, C. (eds.) Inscrypt 2007. LNCS, vol. 4990, pp. 81–92. Springer, Heidelberg (2008)

7. Eschenauer, L., Gligor, V.D.: A Key-management Scheme for Distributed Sensor Networks. In: ACM CCS, pp. 41–47. ACM (2002)
8. Lee, J., Stinson, D.R.: A Combinatorial Approach to Key Predistribution for Distributed Sensor Networks. In: IEEE Wireless Communications and Networking Conference, pp. 1200–1205 (2005)
9. Lee, J., Stinson, D.R.: Common Intersection Designs. International Journal of Combinatorial Designs 14, 251–269 (2006)
10. Lee, J., Stinson, D.R.: On The Construction of Practical Key Predistribution Schemes for Distributed Sensor Networks Using Combinatorial Designs. ACM Trans. Inf. Syst. Secur. 11(2) (2008)
11. Ruj, S., Roy, B.: Key Predistribution Using Partially Balanced Designs in Wireless Sensor Networks. In: Stojmenovic, I., Thulasiram, R.K., Yang, L.T., Jia, W., Guo, M., de Mello, R.F. (eds.) ISPA 2007. LNCS, vol. 4742, pp. 431–445. Springer, Heidelberg (2007)