Towards a Deterministic Hierarchical Key Predistribution for WSN Using Complementary Fano Plane

Sarbari Mitra, Ratna Dutta, and Sourav Mukhopadhyay

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

Abstract. We propose a key pre-distribution scheme based on the complementary design of a Fano plane. The nodes are arranged hierarchically in the form of a 6-nary tree. Key predistribution follows a deterministic approach. Each node in our scheme requires storing significantly less number of secret keys. Our scheme provides better resiliency compared to other existing schemes and reasonable connectivity as well. It can be found that any two nodes are connected either directly or via a key-path. Moreover, any number of nodes can be introduced in the network by assigning a few keys to the newly joined nodes only, without disturbing the existing set-up of the network.

Keywords: complementary design, Fano plane, key predistribution.

1 Introduction

Sensor nodes are small, mobile, low-cost, battery powered and resource (such as memory, power etc.)-constrained devices. They are deployed with high density in the target region to form a Wireless Sensor Network (WSN). Due to their huge application in many areas (home front to military operation), WSN has become a burgeoning field nowadays. There are two types of WSNs: Distributed and Hierarchical. In Distributed network all the nodes are assumed to be uniform whereas Hierarchical network comprised of sensor nodes with different memory, power, transmission range etc.

The sensor nodes are supposed to collect data from the environment and then transmit them to the base station by communicating with other nodes within the specified transmission range. This communication, when takes place in hostile region, is intended to be secret, for which secret keys need to be given to the nodes. One of the possible methods is online key agreement, but this is practically infeasible as this approach is highly expensive. The other approach is to store the keys to the nodes before their deployment, which is termed as key predistribution. Key predistribution can be of three types: (i) *Probabilistic*- where the keys are chosen randomly from the key pool and given to the nodes so that

M. Rajarajan et al. (Eds.): SecureComm 2011, LNICST 96, pp. 373–388, 2012.

any two nodes share common key with certain probability, (ii) *Deterministic*- the selection and assignment of keys to the nodes follow a certain pattern, and (iii) *Hybrid*- which is a combination of the above two approaches.

The parameters of a key pre-distribution scheme are : (i) *Scalability*- when it is required to introduce a few new nodes to the network, existing set-up should not be disturbed, i.e., key-chains of the existing nodes should not be altered; (ii) *Storage*- less number of keys should be stored in the nodes so that rest of the memory can be used for computation; (iii) *Resilience*- how robust the network is against node capture; (iv) *Connectivity*- most of the nodes should share secret keys so that they can communicate secretly.

There are two extreme key pre-distribution schemes. First, is to store one master key to all the nodes in the network. Connectivity of the resulting network is very high as any two nodes can communicate but the network is not at all resilient. Capture of any single node will reveal the secret master key. As a result, the whole network cease to work. Second is to store a secret key for each pair of nodes in the network. Then connectivity and resiliency both are optimal, but the storage requirement is too expensive which is not affordable. Hence we observe that any of the above cases is not suitable due to the fact that the parameters storage, resiliency and connectivity are contradictory in nature. To achieve a scheme which optimizes all the parameters, authors have tried to get a trade-off between aforesaid parameters. We discuss literature survey in the following subsection.

1.1 Previous Work

Eschenauer and Gligor [7] were first to use random key pre-distribution in WSN. The key distribution scheme proposed by them includes random selection of key chains from the large key-pools and then assigning the keys to the nodes. Any two nodes can communicate if they share a common key. The scheme is referred to as the *basic scheme*. Later Chan, Perrig and Song [5] proposed *q-composite scheme* which is a modified version of the basic scheme: any two nodes can communicate if they share at least q common keys.

The main disadvantage of the aforesaid probabilistic schemes is that sharing of common keys between any two nodes is not certain. On the contrary, the schemes based on deterministic approach using combinatorial designs increases the probability of key sharing between nodes to a greater extent. Naturally, *Combinatorial Design* has become a useful technique of key pre-distribution. Mitchell and Piper [11] were first to apply combinatorial design as one of the key distribution techniques whereas Camptepe, Yener [1] introduced combinatorial design for key predistribution in wireless sensor network. In this paper [1] two combinatorial designs are considered: first is the symmetric $(p^2 + p + 1, p + 1, 1)$-BIBD (or finite projective plane of order p) and the second is generalized quadrangles. The advantage of this deterministic approach is that any two nodes certainly share a common key, which improves the connectivity of the network to significantly. The authors observed that the main drawback of deterministic approach is that the scheme is not scalable as the network size N should satisfy $N \leq p^2 + p + 1$; if one wants to introduce some new nodes to the network which

exceeds the bound then p has to be raised to the next prime number (as the existence of projective planes of order p is confirmed for only prime values of p), which results in a much more larger network than what is required, and the key-chains at each node have to be changed. It is also observed that generalized quadrangles induce better scalable network and provide better resilience than projective planes [2]. For the scalability, they have proposed a hybrid scheme which improves the resilience, but 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 almost 0.99995 portion of the nodes can communicate either directly or via intermediate nodes. Chakrabarti et al. [3] provided an example to show that out of 2401 nodes in a network 18% of the links will be destroyed if only 10 nodes are captured. This is the main disadvantage of this scheme. Later, in 2008, the authors had developed quadratic schemes [9] based on Transversal designs and referred the method described in [8] as linear schemes. Their work suggests that the quadratic scheme provides best resilience unless the number of compromised nodes is high. If the number of compromised nodes increases beyond 20, then linear scheme is preferred to quadratic scheme for better resilience. Quadratic schemes in general provide better connectivity than linear schemes. Both linear and quadratic schemes are preferred to 2-composite scheme if shared-key-discovery is taken into consideration.

In 2005, Chakrabarti et al. [3] proposed a probabilistic key predistribution scheme. Construction of the blocks were in the same manner as proposed by Lee and Stinson in [8]. The sensor nodes are then formed by random merging of the blocks, which consequently increases the probability of sharing common keys between sensor nodes. Their scheme provides better resiliency as compared to the Lee-Stinson scheme at the cost of large key-chain size in each node. Dong et al. [6] proposed a scheme by considering 3-design as the underlying design. Keys are assigned to the sensor nodes in the network by M$\ddot{o}$bius Planes. This scheme provides better connectivity than that of the scheme proposed by Lee-Stinson [9] and better storage as compared to Camptepe-Yener scheme [1]. The prime drawback of the scheme is that resiliency reduces rapidly with the increasing number of compromised nodes.

Ruj and Roy [12] 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.

It is observed that the schemes based on deterministic approach provide high connectivity, but the storage is also very expensive and the schemes are not scalable in most of the cases. On the contrary, the probabilistic schemes are scalable but do not confirm high connectivity. Our target is to develop a scheme which gives scalability in deterministic approach and also provides better values for the other parameters.

1.2 Our Contribution

Here we present a deterministic key pre-distribution scheme. We have used the complementary design of the Fano plane, i.e., a symmetric $(7, 4, 2)$ - BIBD as our basic building block and map it repeatedly to design the whole network. The network thus formed is heterogeneous, i.e., the nodes are assumed to be placed hierarchically on the basis of computation power, the chance of getting compromised etc.

The storage requirement for this scheme is significantly less (better) than majority of the existing schemes. Storage is an important factor as we all know that once the nodes are deployed to the target region, any external source of power is not available. Moreover, increased memory consumption for storage will decrease the computation power.

We emphasize that apart from storage-efficiency, this scheme provides reasonable connectivity. The whole network is divided into 7 sub-networks each of which forms a 6-nary tree-hierarchical structure. Most of the nodes in the same sub-network are directly connected, but nodes from the different sub-networks may be connected directly or via a key-path through the *level* 1 nodes (in the worst possible case).

Apart from being cost-effective, storing significantly less number of keys leaks very less information (in the form of secret keys) when captured. This leads to improve the resilience of the network. Obtained results support the fact that our scheme provides better resilience than the other similar schemes. Unlike the existing deterministic 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 in the following manner. Some definitions are given in Section 2, the proposed scheme is discussed in detail in Section 3. Obtained results are included in Section 4. Section 5 and Section 6 respectively provides the connectivity and performance of the scheme following the concluding remarks in Section 7.

2 Preliminaries

Combinatorial Design is one of the mathematical tools used for key predistribution to the nodes. Some useful definitions from combinatorial designs are given below:

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, λ) block design (with $t \leq k \leq v$) such that the following are satisfied (i)* $\mathrm{X} = v$ *, (ii) each block contains*

k points, (iii) for any set of t points there are exactly λ blocks that contain all these points.

Definition 2.03. *A t-design with $t = 2$ is known as (v, k, λ)-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.*

Definition 2.06. *The Fano Plane is the projective plane of smallest order i.e., of order 2. It is a (7,3,1) BIBD and it can also be considered as a Steiner system.*

Therefore, all the projective planes are Steiner systems.

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

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

By complementary design we mean the design where each block is mapped to another block such that they are mutually exclusive and exhaustive. The complementary design of the Fano plane is a 2-(7,4,2) design,i.e., it is a symmetric $(7, 4, 2)$-BIBD. From the structure it is clear that the design is no longer a Projective plane and obviously not a Steiner system either, as any pair of keys is included in exactly two nodes.

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 Proposed Scheme

3.1 Key Predistribution to Seven Nodes

In this section we will discuss a particular Steiner system that is taken as a basic building block to design our key predistribution in the hierarchical structure of nodes and then explain how it can be mapped to a sensor network. Let us consider a 2 - $(7, 4, 2)$ design where $X = \{1, 2, 3, 4, 5, 6, 7\}$. The blocks are given by the set $A = \{(4, 5, 6, 7), (2, 3, 5, 6), (2, 3, 4, 7), (1, 3, 5, 7), (1, 3, 4, 6), (1, 2, 6, 7), (1, 2, 4, 5)\}$. Note that each block shares exactly two common elements with all other blocks. We map the system to the sensor network by considering X to be the key-pool i.e. all the elements to be the keys and sets (blocks) in A correspond to the key-chains of each sensor node. Here seven elements correspond to seven keys and each block represents a sensor node (key chain of the node). This assigns a set of seven keys to seven nodes such that all nodes together contain exactly seven keys and any two are connected by exactly two common keys.

3.2 Key Predistribution to the Tree Hierarchy

We label all the nodes and all the keys by $1, 2, 3, 4, \ldots$. 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-rings assigned to the nodes $1, 2, 3, 4, 5, 6, 7$ are respectively $\{4, 5, 6, 7\}$, $\{2, 3, 5, 6\}$, $\{2, 3, 4, 7\}$, $\{1, 3, 5, 7\}$, $\{1, 3, 4, 6\}$, $\{1, 2, 6, 7\}$, $\{1, 2, 4, 5\}$. Loosely speaking, the node-set $\{1, 2, 3, 4, 5, 6, 7\}$ of these seven nodes form a 2-$(7, 4, 2)$ design in level 1. We refer the keys $\{1, 2, 3, 4, 5, 6, 7\}$ chosen in level 1 as level 1 keys. In level 2, node 1 forms a 2-$(7, 4, 2)$ design with new six level 2 nodes $8, 9 \ldots 13$. A set of seven keys is required to complete the key set. Note that node

Table 1. Components of 2 - $(7, 4, 2)$ designs formed by level 1 nodes

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

1 already contains the keys $\{4, 5, 6, 7\}$. We choose three new keys, say $\{8, 9, 10\}$ and take the key set $\{4, 5, 6, 7, 8, 9, 10\}$ for the key predistribution among seven nodes. We call the new keys $8, 9, 10$ chosen in level 2 as level 2 keys. This process is repeated for the other nodes of level 1. The 2 - $(7, 4, 2)$ designs corresponding to all the level 1 nodes are explicitly described in Table 1. In level 3, each of level 2 nodes are attached to 6 new level 3 nodes and the corresponding key chain is chosen in the same manner i.e. keeping the four keys same as the level 2 keys contained by level 2 nodes and adding three new level 3 keys. This process is repeated until keys are assigned to all the nodes in the network. We provide below the algorithm KPDistribution for assigning keys to the tree hierarchy as explained above. We consider a hierarchical structure using a 6-nary tree for key predistribution.

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 four keys assigned to N[i] are stored in N[i][1],N[i][2], N[i][3], N[i][4] respectively. Choose $\{u_4, u_5, u_6\ u_7\} \in_R K$, where the symbol $\in_R$ stands for random selection.

Algorithm. KPDistribution

$i := 0$;
N [1][1] := u_4; N [1][2] := u_5; N [1][3] := u_6; N [1][4] := u_7;
procedure KPDistribution (u_1, u_2, u_3, u_4)
$X := \{u_4, u_5, u_6, u_7\}$;
Choose $\{u_1, u_2, u_3\} \in_R B$ where $B \subseteq K - X$, B is the set of unused keys
$X := X \cup \{u_1,\ u_2,\ u_3\}$;

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

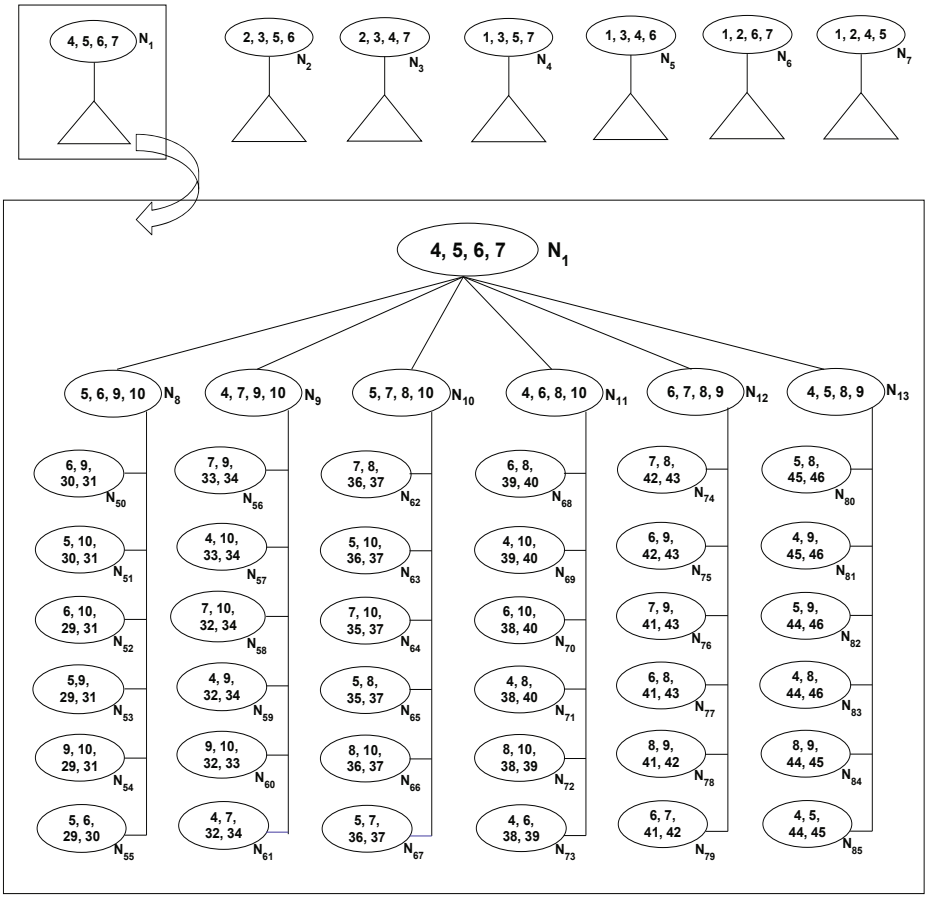

Fig. 1. Key predistribution to a sub tree upto *level* 3

4 Results

Theorem 4.01. *(a) Number of nodes in level j is $n_j = 7\times 6^{j-1}$, $\forall\ j\ \in\ \{1, \ell\}$ where l denotes the total number of levels present in the network.*
(b) Total number of nodes in the network is $T_{nodes}\ =\ \frac{7}{5}(6^{\ell}-1)$.
(c) Number of keys that are used for the first time in level j is

$$k_j = 21 \times 6^{j-2},\ \forall j \geq 2;\quad k_1 = 7;$$

(d) Total number of keys in the network is $K\ =\ 7\ [1 + \frac{3}{5}(6^{\ell-1}-1)]$.
(e) Number of nodes to which a level i key is assigned to, is $N_i\ = 2\times\{3^{l+1-i}-1\}$
.

(f) The maximum level required to accommodate T_{nodes} number of nodes in the network is $l = \lceil log(\frac{5}{7}T_{nodes} - 5) \rceil$.

Proof:

(a) The result holds trivially for $i = 1$.
Let us consider the following notations:
Level 1 nodes are denoted by $N^{(1)}_{i_1}$, $i_1 \in \{1, 2, ... 7\}$. Level 2 nodes are denoted by $N^{(2)}_{i_1,i_2}$, $i_1 \in \{1, 2, ... 7\}$, $i_2 \in \{1, 2, ... 6\}$, where $N^{(2)}_{i_1,i_2}$ represents the i_2^{th} child at level 2 of i_1^{th} node at level 1. Level t nodes are denoted by $N^{(t)}_{i_1,i_2,...i_t}$, $i_1 \in \{1, 2, ... 7\}$, $i_2, i_3, ..., i_t \in \{1, 2, ... 6\}$.
Clearly total number of nodes in t^{th} level is $7 \times (6 \times 6 \times ... \times 6)$ $(t-1$ times).

Fig. 2 illustrates the detailed hierarchical tree structure upto level 5.

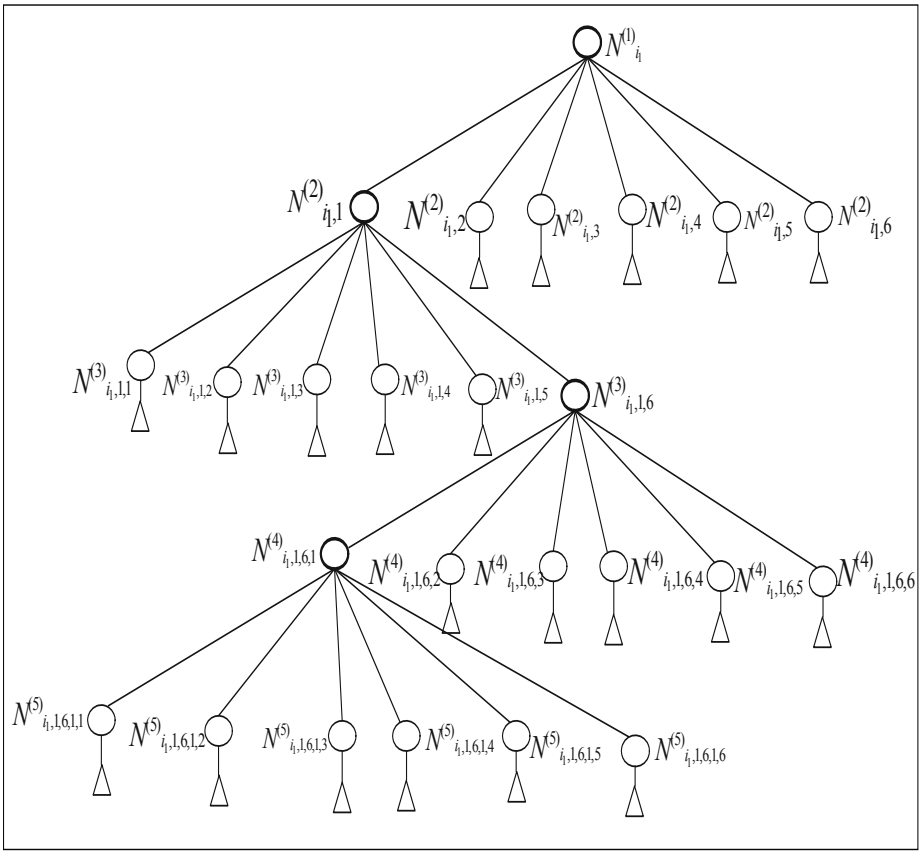

Fig. 2. Hierarchical tree structure upto level 5

(b) As the levels of the nodes are exhaustive and disjoint, we have

$$T_{nodes} = n_1 + n_2 + ... + n_l,$$

where l represents the total number of levels in the network. Thus

$$T_{nodes} = \sum_{j=1}^{j=t} n_j = 7 \sum_{j=0}^{l-1} 6^j.$$

Hence the result follows.

(c) $k_1 = 7$ holds trivially, as level 1 contains only one Complementary Fano plane consisting of seven keys.
Observed that, from level 2 onwards, each Complementary Fano plane in a level includes six new nodes and three new keys in the following level. Hence

$$k_j = 3 \times n_{j-1} \ , \ \forall \ j \ \in \ \{ \ 2 \ , \ \ell \ \},$$

where n_i denotes the number of nodes in level i. The result follows on substitution of the expression for n_i from (a).

(d) As the keys appearing for the first time in a particular level are exhaustive and disjoint, we have

$$K = k_1 + k_2 + ... + k_l,$$

l being the total number of levels in the network. Thus

$$K = \sum_{j=1}^{j=t} k_j = 7 + 21 \sum_{j=0}^{l-2} 6^j$$

Hence the result follows.

(e) The key that appears for the first time in level i is contained in only one Complementary Fano plane and hence goes to four nodes in level i. In $(i+1)^{th}$ level, that key goes to each of the four Complementary Fano planes corresponding to each of the previous level nodes and in each system, the key is contained in three new nodes. Thus we observe that the nodes to which a level i key is contained, form four ternary trees with their root in level i. The number of nodes to which a level j key is assigned to is given by $\sum_{i=j}^{l} 4 \times 3^{i-j}$. Hence the result follows.

(f) Follows directly from (b) □

The results in (b),(d) and (f) establish inter-relationship between the total number of nodes, total number of keys and the maximum number of levels required to accommodate all the nodes. Thus, when any two of them are known or given, third one can be obtained. Result(e) helps us to calculate resilience, as will be seen later in Section 6.

5 Connectivity

We now discuss how the nodes are connected by single-hop (direct) paths. From the key distribution pattern among the nodes in the network, it is observed that any two nodes can share at most 2 (i.e. 0 or 1 or 2) keys. So we summarize below the possible cases as the following: Let A be any node from level $\leq k$ in the network.

Case 0: The node B be chosen from level k. A shares **0** keys with B in level k and hence A is not connected to any of its descendants in level $\geq k$ either.

Case 1: The node B be chosen from level k. A shares **1** key with B in level k. In $(k+1)^{th}$ level, B has six children namely : B_1, B_2, B_3, B_4, B_5 and B_6. A is connected to exactly three of them by sharing only one key with each. Without loss of generality, let us assume that A is connected to B_1, B_2, B_3 and is not connected to B_4, B_5, B_6. To get the connectivity of the node A with the grand children of B, i.e. B_{ij} for $i, j \in \{1, 2 \cdots 6\}$ in level $k+2$, we observe the following subcases:

Subcase 1.1: B_1, B_2 or B_3 falls under Case 1 and same arguments hold as in Case 1 with $B := B_i$, for $i \in \{1, 2, 3\}$; $k := k + 1$.

Subcase 1.2: B_4, B_5 or B_6 falls under Case 0.

Case 2: The node C be chosen from level k. A shares **2** keys with C in level K. In $(k+1)^{th}$ level, C has six children namely, C_1, C_2, C_3, C_4, C_5 and C_6. A shares only one key with exactly four of them, only one key with exactly one and no key with the remaining one. Without loss of generality, let us assume that A shares exactly one key with C_1, C_2, C_3, C_4, only one key with C_5 and does not share any key with C_6.
To observe how node A is connected with the grand children of C, i.e., C_{ij} for $i, j \in \{1, 2 \cdots 6\}$ in level $k + 2$, we have the following sub cases:

Subcase 2.1: C_1, C_2, C_3 or C_4 fall under Case 1 and same arguments hold as in Case 1 with $B := C_i$, for $i \in 1, 2, 3, 4$; $k := k + 1$.

Subcase 2.2: C_5 falls under Case 2 and same arguments hold as in Case 1 with $C := C_5$; $k := k + 1$.
Subcase 2.3: C_6 falls under Case 0.

Example:

Let us discuss here how we observe the connectivity of a particular node. According to above discussion we assume that the network consists of 4 levels of nodes and keys.

Connectivity of a level 1 node (say N_1). All the nodes of level 1 form a 2-$(7, 4, 2)$ design, so each node is connected to the other six nodes, and each pair of nodes shares exactly two common keys. Hence, N_1 shares two keys with all other six nodes at level 1, i.e. with N_2, N_3, N_4, N_5, N_6 and N_7.

Now in level 2, N_1 is connected to all its own six children by sharing two common keys with each of them i.e. N_1 shares two keys with its children in level 2. There are exactly one child of each level 1 node with which N_1 shares two keys. Therefore, number of nodes in level 2 with which N_1 shares two common keys is 12 and the number of nodes in level 2 with which N_1 shares exactly one common key is given by 24. Out of total 42 nodes in level 2, N_1 is connected to 36 nodes.

In level 3, the number of nodes with which N_1 shares two common keys is 12 and the number of nodes with which N_1 shares exactly one common key is 120. Thus, out of total 252 nodes in level 3, N_1 is connected to 132 nodes.

Similarly in level 4, N_1 is connected to 12 nodes by sharing two common keys and 408 nodes by sharing exactly one key.

Hence out of total 1512 nodes in level 4, N_1 is connected to 420 nodes. Total number of nodes to which N_1 is connected is $= 6+36+132+420 = 594$. As all the level 1 nodes are uniform, any level 1 node is connected to 594 nodes out of total 1813 nodes in the network. This implies that only one level 1 node is directly connected to almost 32% of the nodes in the whole network. So, intuitively we can say that all the nodes in the network is connected to at least one level 1 node.

Connectivity of a level 2 node (say N_8). We note that out of seven nodes in level 1, N_8 shares two keys with exactly two nodes, no key with one node, and only one key with the remaining four nodes. Therefore six nodes of level 1 are connected to N_8.

N_8 is one of the child of N_1, therefore N_8 shares two keys with the other five children of N_1 in level 2. Also N_2 in level 1 shares two keys with N_8, hence, out of the six children of N_2, one shares two keys, one no key and remaining only one key with N_8. Thus number of nodes with which N_8 shares two keys in level 2 is 6 and the number of nodes with which N_8 shares only one key in level 2 is 16. Thus total 22 nodes of level 2 are connected to N_8.

Following similar arguments, N_8 shares two keys with 12 nodes in level 3 and the number of nodes with which N_8 shares a common key is given by 72. Hence N_8 is connected to 84 nodes in level 3.

In level 4, N_8 shares two keys with 12 nodes and one key with 264 nodes. Thus N_8 is connected to 276 nodes in level 4. Therefore N_8 is connected to 376 nodes in the whole network. As all the level 2 nodes are uniform, any level 2 node is connected to 376 nodes in the network.

Similarly, we can calculate these values for other level nodes also, and intuitively we can predict that this scheme has reasonable connectivity.

6 Performance

We calculate resilience by the following formula proposed 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 number of 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 and s is the total number of compromised nodes. Therefore we must have $\sum_{i=1}^{l} s_i = s$.

In our scheme, the nodes are arranged hierarchically in the network, i.e., the lower level nodes (which are very less in number) are more powerful and hence are less liable of getting compromised than higher level nodes (which are much more in number).

The average values of $fail(s)$ corresponding to certain values of s has been listed in Table 2, which describes how the network collapses with increasing number of compromised nodes. This table shows that the proposed scheme provides reasonable resilience.

Table 2. Network collapses with increasing number of compromised nodes

s	*fail(s)*	*s*	*fail(s)*	*s*	*fail(s)*
10	0.017549	110	0.238873	450	0.718262
20	0.032655	120	0.252230	500	0.752019
30	0.056979	130	0.290375	550	0.800994
40	0.072504	140	0.302058	600	0.825032
50	0.112959	150	0.314306	650	0.850413
60	0.135263	200	0.396464	700	0.868336
70	0.149500	250	0.469364	750	0.884240
80	0.169968	300	0.574162	800	0.907102
90	0.207049	350	0.635533	850	0.918233
100	0.220104	400	0.679556	900	0.938538

In Table 3, 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 [12] and Lee-Stinson quadratic scheme [9], where T_{nodes} denotes total number of nodes in the network and T_{keys} denotes total number of keys present in each node.

The comparison between the schemes has been shown graphically in Fig. 3 and Fig. 4. In Fig. 3 we show the comparison of our scheme with Lee-Stinson linear scheme [8], Chakrabarti et al. scheme [3], Ruj-Roy scheme [12] and Lee-Stinson quadratic scheme [9] for less number of compromised nodes i.e. $1 - 10$ nodes. In

Table 3. Comparison with some of the existing schemes

	[8]	[3]	[12]	[9]	Ours
T_{nodes}	1849	2550	2415	2197	1813
T_{keys}	30	≤ 28	136	30	4
$fail(10)$	0.201070	0.213388	0.0724	0.297077	0.017549

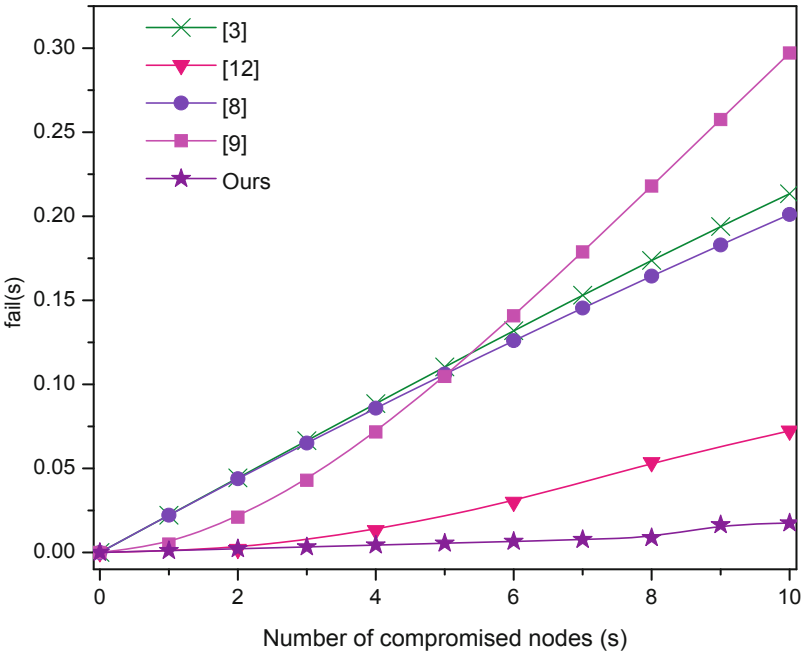

Fig. 3. Comparison of resilience for small number of compromised nodes

Fig. 4 we provide the comparison with Lee-Stinson linear scheme [8] and Lee-Stinson quadratic scheme [9] for a large number (i.e., 10-200) of compromised nodes. It is very clear from the figures that the networks based on other schemes collapses very fast compared to ours.

Remarks. *We feel that generalizing the scheme by considering the complementary design of any projective plane (instead of Fano plane) will improve the connectivity of the network. This is due to the fact that complementary design of a* $(p^2 + p + 1, p + 1, 1)$ *projective plane is in the form of a symmetric* $(p^2 + p + 1, p^2, p^2 - p)$*-BIBD, i.e., a set of* p^2 *keys are shared between* $p^2 - p$ *nodes in the network, which increases by a greater extent with increasing values*

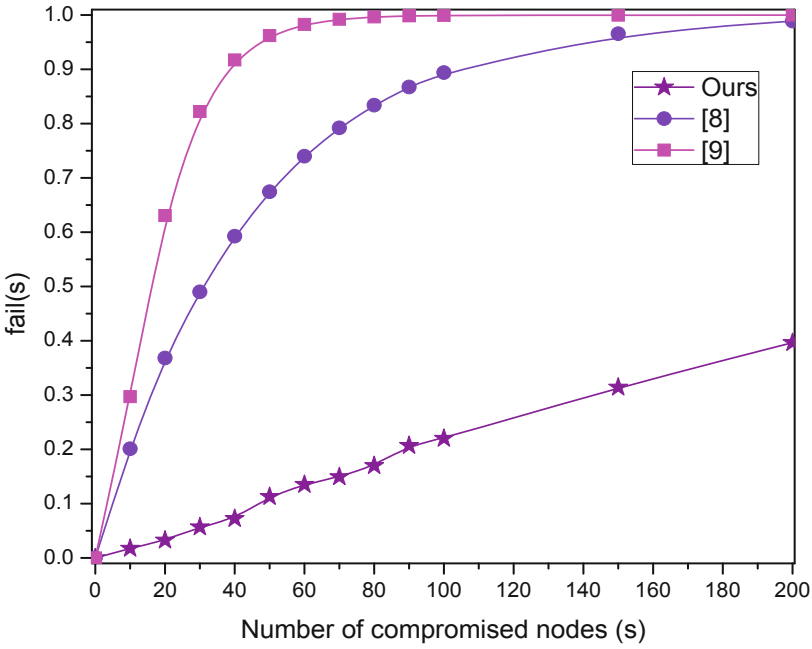

Fig. 4. Comparison of resilience for large number of compromised nodes

of p (a prime number). Thus, key-sharing between the nodes can be achieved to the desired level for complementary design of higher order projective planes. But we have to analyse the resilience of those schemes also. As our future work we would like to achieve improved connectivity with a reasonable trade off with resilience.

7 Conclusion

In this paper we have introduced a key predistribution scheme for wireless sensor network based on Complementary Fano plane. Our approach is deterministic and the sensor nodes are arranged hierarchically in the form of a 6-nary tree structure. The proposed scheme is significantly storage-efficient and has the flexibility of introducing new sensor nodes by adding only a few keys to the joining nodes without disturbing the existing set-up. We have analysed the connectivity of our scheme and it was noticed that all the nodes in the network are well-connected. It is observed that any node shares two keys with a considerable portion of the network. Obtained results support that the resilience of the resulting network is found better than some of the similar combinatorial design based schemes.

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. IEEE/ACM Trans. Netw. 15(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. IEEE Computer Society (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 Conference on Computer Communications Security, 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 (WCNC), pp. 1200–1205 (2005)
9. 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)
10. Lee, J., Stinson, D.R.: Common Intersection Designs. International Journal of Combinatorial Designs 14, 251–269 (2006)
11. Mitchell, C.J., Piper, F.: Key Storage in Sensor Networks. Discrete Applied Mathematics 21, 215–228 (1988)
12. 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)