A Combinatorial Approach to Key Predistribution for Distributed Sensor Networks

Invited Paper

Jooyoung Lee
Department of Combinatorics and Optimization
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada

Douglas R. Stinson
School of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada

***Abstract*— In this paper, we discuss the use of combinatorial set systems in the design of deterministic key predistribution schemes for distributed sensor networks. We concentrate on analyzing combinatorial properties of the set systems that relate to the connectivity and resilience of the resulting distributed sensor networks.**

I. Introduction

Distributed Sensor Networks (DSNs) are ad-hoc mobile networks that include sensor nodes with limited computation and communication capabilities. They are mainly used for military purposes but they also have wide applications in civilian areas. In military operations, sensor nodes are distributed in a hostile territory in order to monitor and collect various information (e.g., acoustic, seismic, magnetic). Since they are typically carried by soldiers or spread from airplanes, we assume that sensor nodes have no information on where they are located, that is, they are distributed in a random way. Once deployed, they operate unattended for extended periods without any movement. They have no external power supply during their operation. Therefore the most essential requirement is that each sensor should consume as little power as possible.

The sensor nodes in DSNs should be able to communicate with each other in order to relay or accumulate secret information. There are three ways to establish pairwise keys between sensor nodes. First is to establish secret keys using a public-key infrastructure (PKI). However, asymmetric cryptographic primitives are not suitable due to expensive computational cost as well as storage constraints in each node. In other strategies, a sensor node is chosen to be a *trusted authority* (TA) or there is an explicit *base station*, which all nodes in the network are assumed to trust. The TA or base station shares a long-lived key with every node and transmits session keys to sensor nodes on request. This method can result in expensive costs for message relay. Arbitrated protocols are also vulnerable to a compromise of the TA or the base station. Therefore it is natural that we are interested in *key predistribution schemes* (or *KPSs*), where a set of secret keys is installed in each node before the sensor nodes are deployed.

Recently, Eschenauer and Gligor [5] proposed a randomized key predistribution scheme. Their scheme consists of three phases: *key predistribution*, *shared-key discovery*, and *path-key establishment*. We briefly describe these phases since our schemes follow the same framework. In the key predistribution phase, a large pool of keys and their key identifiers are generated. Every sensor node is loaded with a fixed number of keys chosen from the key pool, along with their key identifiers. After deployment of the DSN, the shared-key discovery phase takes place, where two nodes in a wireless communication range look for their common keys. If they share one or more common keys, they can pick one of them as their secret key for cryptographic communication. The path-key establishment phase takes place if there is no common key between a pair of nodes in a wireless communication range. Then, they look for multiple secure links (hops) to reach each other, so that one of them can choose an arbitrary key and then relay it through the links in encrypted form to the destination node.

The communication capabilities of a DSN can be modeled as the intersection of a physical layer and a network layer. Due to resource constraints, a sensor node can communicate with nodes only within a limited radius. Sensor nodes are deployed randomly within a certain physical space, so the *physical layer* is represented by a *random geometric graph*. On the other hand, the *network layer* is represented by the *network graph*, in which two nodes are adjacent if they share a common key. The network graph is determined by the structure of the KPS, and it is independent of the physical distribution of the sensor nodes. In order for two sensor nodes to communicate, the two nodes must be connected by a path in both the geometric graph and the network graph.

A. Our Contributions

The Eschenauer-Gligor KPSs ([5]) are randomized schemes: every node is assigned a random subset of keys from a given pool of keys. In this paper, we focus on combinatorial constructions for deterministic key predistribution schemes. The rest of this paper is organized as follows. In Section II, we define some basic types of combinatorial set systems (designs), and how they can be used to set up a KPS for a DSN. In Section III, we introduce "configurations" and discuss their influence on the local connectivity of the DSN. Section IV characterizes the configurations that yield DSNs in which any two nodes share a common key. Section V introduces the new concept of a "μ-common intersection design", and Section VI discusses the existence of two-hop paths in the corresponding DSNs. Section VII examines a subclass of configurations that are optimal with respect to their common intersection properties. Section VIII treats resiliency of the DSNs in the presence of failed or compromised nodes, and Section IX mentions one efficient method for shared-key discovery.

B. Related Work

The basic model we are studying is due to Eschenauer and Gligor [5], who studied randomized KPSs for DSNs. Extensions and variations of this approach can be found in Chan, Perrig and Song [2], Du, Deng, Han and Varsheney [4], and Liu and Ning [7]. Papers studying deterministic KPSs for DSNs include Çamtepe and Yener [1], Lee and Stinson [6], and Wei and Wu [9]. The use of combinatorial designs in this context was first proposed in [1].

II. Combinatorial Set Systems and DSNs

A *set system* or *design* is a pair $(X, \mathcal{A})$, where $\mathcal{A}$ is a finite set of subsets of X, called *blocks*. The *degree* of a point $x \in X$ is the number of blocks containing the point x. $(X, \mathcal{A})$ is *regular* (of degree r) if all points have the same degree, r. The *rank* of $(X, \mathcal{A})$ is the size of the largest block. If all blocks have the same size, say k, then $(X, \mathcal{A})$ is said to be *uniform* (of rank k).

Example 1: Let

$$\begin{aligned} X &= \{1,2,3,4,5,6,7,8,9\}, \quad \text{and} \\ \mathcal{A} &= \{123, 456, 789, 147, 258, 369, \\ &\quad 159, 267, 348, 168, 249, 357\}. \end{aligned}$$

Then $(X, \mathcal{A})$ is a set system in which there are nine points and twelve blocks. This set system is regular of degree 4 and uniform of rank 3. □

A set system can be used as a *key predistribution scheme* in a distributed sensor network as follows. Suppose that

$$X = \{x_i : 1 \leq i \leq v\}$$

and

$$\mathcal{A} = \{A_j : 1 \leq j \leq b\}.$$

Let the sensor nodes be denoted $N_1, \ldots, N_b$. That is, we identify the b blocks in $\mathcal{A}$ with the b sensor nodes. Further, the points in X are identified with a set of v keys, as follows: For $1 \leq i \leq v$, a key L_i is randomly chosen from some specified key-space, say $\mathcal{L}$.

Then, for $1 \leq j \leq b$, the sensor node N_j receives the set of keys

$$\{L_i : x_i \in A_j\}.$$

That is, the block A_j of the set system is used to specify which keys are given to the node N_j.

It is useful and convenient if every node receives a constant number of keys and every key is assigned to a constant number of sensor nodes ([9]). Therefore, from now on, we will only consider regular and uniform set systems. Such a set system is called a (v, b, r, k)-1-*design*, where $|X| = v$, $|\mathcal{A}| = b$, r is the degree and k is the rank. A necessary condition for existence of such a set system is that $bk = vr$. It is well-known that this necessary condition is sufficient for existence of a (v, b, r, k)-1-design (see, for example, [8]).

The correspondences between the parameters of a set system and the related key predistribution scheme for a DSN are summarized in Table I (note that "two-hop paths" will be described in detail in Section V).

III. Configurations and Local Connectivity of the DSNs

In this section, we address the *local connectivity* of the network. Our desire is that any two nodes that are in close physical proximity to each other should be able to establish a secure channel. Observe that two nodes, say N_i and N_j, share a common key if and only if $A_i \cap A_j \neq \emptyset$. In this case, they can use any key $L \in A_i \cap A_j$ as a secret key for cryptographic communication.

We can evaluate the connectivity of the network layer by studying the *block graph* $G_{\mathcal{A}}$ of the set system $(X, \mathcal{A})$. The graph $G_{\mathcal{A}}$ has vertex set $\mathcal{A}$, and two vertices (blocks), say A_i and A_j, are adjacent in $G_{\mathcal{A}}$ if $A_i \cap A_j \neq \emptyset$. Two nodes, say N_i and N_j, share a common key if and only if A_i and A_j are adjacent in the block graph.

Lemma 1: Any vertex (block) A_j in the block graph $G_{\mathcal{A}}$ of a (v, b, r, k)-1-design, $(X, \mathcal{A})$, has degree at most $k(r-1)$. Further, all vertices in $G_{\mathcal{A}}$ have degrees equal to $k(r-1)$ if and only if $|A_i \cap A_j| \leq 1$ for all $A_i, A_j \in \mathcal{A}$, $i \neq j$.

Proof: Fix a block $A_j \in \mathcal{A}$. For any $x \in A_j$, define

$$\mathcal{B}_x = \{A_i \in \mathcal{A} : x \in A_i\} \backslash \{A_j\}.$$

Clearly, $|\mathcal{B}_x| = r - 1$ for all $x \in A_j$. The degree of A_j in $G_{\mathcal{A}}$, which we denote by $\mathsf{deg}(A_j)$, is equal to the number of blocks intersecting A_j, so

$$\mathsf{deg}(A_j) = \left| \bigcup_{x \in A_j} \mathcal{B}_x \right| \leq \sum_{x \in A_j} |\mathcal{B}_x| = k(r-1).$$

Furthermore, it is easy to see that $\mathsf{deg}(A_j) = k(r-1)$ if and only if the sets $\mathcal{B}_x$ $(x \in A_j)$ are disjoint. This is equivalent to saying that $|A_i \cap A_j| \leq 1$ for all $A_i \in \mathcal{A}$, $i \neq j$. ■

A (v, b, r, k)-1-design is called a (v, b, r, k)-*configuration* if any two distinct blocks intersect in zero or one point. (For a brief survey on configurations, see [3, pp. 253–255]). Lemma 1 asserts that, among all the (v, b, r, k)-1-designs, the (v, b, r, k)-configurations have block graphs which are regular graphs and in which the vertex degrees are maximized. This means that the connectivity of the network layer is as large as possible (for the given parameter values). For these reasons, we will focus on (v, b, r, k)-configurations in this paper.

TABLE I
CORRESPONDENCES BETWEEN TERMINOLOGY FOR DISTRIBUTED SENSOR NETWORKS AND SET SYSTEMS

KPS for a Distributed Sensor Network	Set System	Parameter
network size	number of blocks	b
size of key pool	number of points	v
number of keys per node	block-size (rank)	k
number of nodes per key	degree of a point	r
number of two-hop paths connecting two nodes	number of blocks intersecting two disjoint blocks	μ

The following elementary lemma records some basic facts about (v, b, r, k)-configurations (see [3]).

Lemma 2: A (v, b, r, k)-configuration exists only if $bk = vr$ and $v - 1 \geq r(k - 1)$.

Remark: We noted above that the equation $bk = vr$ holds for any (v, b, r, k)-1-design.

The *deficiency* of a (v, b, r, k)-configuration is the quantity $d = v - 1 - r(k - 1)$. It follows from Lemma 2 that $d \geq 0$.

A. Some Examples of Configurations

In this section, we mention some types of configurations that have been extensively studied in the combinatorial literature.

A (v, b, r, k, λ)*-BIBD* (or *balanced incomplete block design*) is a (v, b, r, k)-1-design in which every pair of points occurs in exactly λ blocks. The set system presented in Example 1 is a $(9, 12, 4, 3, 1)$-BIBD.

The following necessary conditions are well-known (see, for example, [8]).

Lemma 3: A (v, b, r, k, λ)-BIBD exists only if $\lambda(v - 1) = r(k - 1)$, $bk = vr$ and $b \geq v$.

The relation between BIBDs with $\lambda = 1$ and configurations is stated in the following lemma.

Lemma 4: ([3]) A (v, b, r, k)-configuration having deficiency $d = 0$ exists if and only if a $(v, b, r, k, 1)$-BIBD exists.

The so-called finite projective planes are BIBDs of particular interest. A *projective plane* of order $n \geq 2$ is an $(n^2 + n + 1, n^2 + n + 1, n + 1, n + 1, 1)$-BIBD. It is well-known that a projective plane of order n exists if n is a prime or a prime-power; see, for example [8].

Let g, u and k be positive integers such that $2 \leq k \leq u$. A *group-divisible design* of *type* g^u and *block-size* k is a triple $(X, \mathcal{H}, \mathcal{A})$, where X is a finite set of cardinality gu, $\mathcal{H}$ is a partition of X into u parts (called *groups*) of size g, and $\mathcal{A}$ is a set of subsets of X (called *blocks*), which satisfy the following properties:

1) $|H \cap A| \leq 1$ for every $H \in \mathcal{H}$ and every $A \in \mathcal{A}$.
2) Every pair of elements of X from different groups occurs in exactly one block in $\mathcal{A}$.

Remark: The groups $H \in \mathcal{H}$ are not algebraic groups; they are just disjoint subsets of points that form a partition of X. The use of the term "groups" in this context is historical.

For information on group-divisible designs, see [3, pp. 185–193].

Lemma 5: ([3]) If there exists a group-divisible design of type g^u and block-size k, then there exists a (v, b, r, k)-configuration with $v = gu$,

$$b = \frac{g^2 u(u - 1)}{k(k - 1)} \quad \text{and} \quad r = \frac{g(u - 1)}{(k - 1)}.$$

The deficiency of this configuration is $d = g - 1$.

Proof: The points and blocks of the group-divisible design comprise the desired configuration. ■

It is easy to verify that a $(v, b, r, k, 1)$-BIBD yields a group-divisible design of type 1^v and block-size k (just define the groups to be v singleton sets).

Here is another special type of group-divisible design that is particularly useful: A *transversal design* TD(k, n) is a group-divisible design of type n^k and block-size k. It follows that $|H \cap A| = 1$ for any block A and any group H in a transversal design.

Remark: A TD(k, n) is equivalent to a set of $k-2$ mutually orthogonal Latin squares of order n. See [3, pp. 111–142] for more information.

We pause to present one easily constructed class of transversal designs. We will make use of these transversal designs a bit later.

Theorem 6: Suppose that p is prime and $2 \leq k \leq p$. Then there exists a TD(k, p).

Proof: Define

$$X = \{0, \ldots, k - 1\} \times \mathbb{Z}_p.$$

For $0 \leq x \leq k - 1$, define

$$H_x = \{x\} \times \mathbb{Z}_p,$$

and then define

$$\mathcal{H} = \{H_x : 0 \leq x \leq k - 1\}.$$

For every ordered pair $(i, j) \in \mathbb{Z}_p \times \mathbb{Z}_p$, define a block

$$A_{i,j} = \{(x, ix + j \bmod p) : 0 \leq x \leq k - 1\}.$$

Let

$$\mathcal{A} = \{A_{i,j} : (i, j) \in \mathbb{Z}_p \times \mathbb{Z}_p\}.$$

Then it is easy to prove that $(X, \mathcal{H}, \mathcal{A})$ is a TD(k, p). ■

B. Dual Set Systems

Suppose that $(X, \mathcal{A})$ is a set system, where

$$X = \{x_i : 1 \leq i \leq v\}$$

and

$$\mathcal{A} = \{A_j : 1 \leq j \leq b\}.$$

The *dual set system* of $(X, \mathcal{A})$ is any set system isomorphic to the set system $(Y, \mathcal{B})$, where

$$Y = \{y_j : 1 \leq j \leq b\}$$

and

$$\mathcal{B} = \{B_i : 1 \leq i \leq v\},$$

and where

$$y_j \in B_i \Leftrightarrow x_i \in A_j.$$

The following results are easy to prove.

Lemma 7: If $(Y, \mathcal{B})$ is the dual set system to $(X, \mathcal{A})$, then $(X, \mathcal{A})$ is the dual set system to $(Y, \mathcal{B})$.

Lemma 8: If $(X, \mathcal{A})$ is a (v, b, r, k)-1-design, then the dual set system is also a (b, v, k, r)-1-design.

Lemma 9: If $(X, \mathcal{A})$ is a (v, b, r, k)-configuration, then the dual set system is also a (b, v, k, r)-configuration.

Lemma 10: If $(X, \mathcal{A})$ is a projective plane of order n, then the dual set system is also a projective plane of order n.

IV. Configurations having Complete Block Graphs

Suppose that we use a (v, b, r, k)-configuration, $(X, \mathcal{A})$, for key predistribution in a DSN. Recall that the block graph $G_{\mathcal{A}}$ is a regular graph on b vertices having degree $k(r-1)$. $G_{\mathcal{A}}$ is a complete graph, i.e., the graph K_b, if and only if $k(r-1) = b-1$. In this case, any two nodes in the DSN share a (unique) common key. This situation can be characterized in terms of certain dual designs.

Theorem 11: Suppose that $(X, \mathcal{A})$ is a (v, b, r, k)-configuration. Then the block graph $G_{\mathcal{A}}$ is a complete graph if and only if the $(X, \mathcal{A})$ is the dual design of a $(b, v, k, r, 1)$-BIBD.

We present a small example.

Example 2: Define X and $\mathcal{A}$ as in Example 1. We already observed that $(X, \mathcal{A})$ is a $(9, 12, 4, 3, 1)$-BIBD. The dual design of $(X, \mathcal{A})$ is given by $(Y, \mathcal{B})$, where

$$\begin{aligned} Y &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, T, E, V\}, \quad \text{and} \\ \mathcal{B} &= \{147T, 158E, 169V, 249E, 257V, \\ &\quad 268T, 348V, 359T, 367E\}. \end{aligned}$$

$(Y, \mathcal{B})$ is a $(12, 9, 3, 4)$-configuration whose block graph is a complete graph K_9 (because any two distinct blocks in $\mathcal{B}$ intersect in exactly one point in Y). □

Clearly it is desirable that any two nodes share a common key. Therefore, we investigate a bit further to determine parameter situations when this goal can be achieved. Theorem 11 states that $(X, \mathcal{A})$ is the dual design of a $(b, v, k, r, 1)$-BIBD. This BIBD must satisfy the necessary conditions of Lemma 3. Note that the parameters b and v have been interchanged, as have r and k, and $\lambda = 1$. Because $bk = vr$ and $v \geq b$, it follows that $k \geq r$. Then,

$$b - 1 = k(r-1) \leq k(k-1), \tag{1}$$

so $b \lessapprox k^2$. That is, the number of keys per node is (roughly) at least as big as the square root of the number of nodes.

There are infinite classes of configurations in which (1) is met with equality. In fact, these are just the finite projective planes, which we introduced a bit earlier. From a projective plane of order n, we obtain the following key predistribution scheme, introduced by Çamtepe and Yener in [1].

Theorem 12: Suppose that q is a prime or a prime power. Then there exists a key predistribution scheme for a DSN having q^2+q+1 nodes, in which every node receives exactly $q+1$ keys, and in which any two nodes share exactly one key.

The schemes of Theorem 12 might be perfectly suitable for DSNs containing a relatively small number of nodes. For example, taking $q = 31$, we get a scheme on 993 nodes in which every node receives 32 keys. However, for larger DSNs, the storage requirement of these "projective plane schemes" might be too large. Suppose, for example, that we want to construct a KPS for a DSN having 20000 nodes. The smallest prime or prime-power q such that $q^2 + q + 1 \geq 20000$ is $q = 149$. The resulting KPS would assign 150 keys to every node, which may not be practical.

V. μ-Common Intersection Designs

As noted in the previous section, it may not be practical to employ KPSs in which any two nodes share a common key, at least for large DSNs. In this section, we consider alternatives.

When $G_{\mathcal{A}}$ is not a complete graph, it can happen that two nodes N_i and N_j in close proximity share no common key. In this case, the two nodes N_i and N_j can communicate via a "two-hop path" provided that there is a node N_h (which is physically close to both N_i and N_j) such that

$$A_i \cap A_h \neq \emptyset \quad \text{and} \quad A_j \cap A_h \neq \emptyset. \tag{2}$$

Equivalently, in the block graph, we are looking for a common neighbor of A_i and A_j.

Ideally, we would like there to be many choices for an intermediate node that satisfies (2). This would increase the chance that at least one of these "good" intermediate nodes is physically close to both N_i and N_j.

The above discussion motivates the following definition: Suppose that $(X, \mathcal{A})$ is a (v, b, r, k)-configuration. We say that $(X, \mathcal{A})$ is a *μ-common intersection design* (or *μ-CID*) provided that

$$|\{A_h \in \mathcal{A} : A_i \cap A_h \neq \emptyset \text{ and } A_j \cap A_h \neq \emptyset\}| \geq \mu$$

whenever $A_i \cap A_j = \emptyset$. Note that we can take $\mu = \infty$ if $A_i \cap A_j \neq \emptyset$ for all i, j.

In general, given parameters (v, b, r, k) such that a (v, b, r, k)-configuration exists, we would like to construct a (v, b, r, k)-configuration with μ as large as possible. This maximum value of μ will be denoted $\mu^*(v, b, r, k)$.

Now we present a few easy observations:

Theorem 13: $\mu^*(v, b, r, k) = \infty$ if and only if there exists a $(b, v, k, r, 1)$-BIBD.

Proof: This follows immediately from Theorem 11. ■

Theorem 14: Suppose that $k \leq n$. If there exists a TD(k, n), then $\mu^*(nk, n^2, n, k) \geq k(k-1)$.

Proof: Because $k \leq n$, any TD(k, n), say $(X, \mathcal{H}, \mathcal{A})$, contains disjoint blocks, say A_i and A_j. For any $x \in A_i$ and any $y \in A_j$ such that x and y are in different groups in $\mathcal{H}$, there is a unique block containing x and y. Hence, there are $k^2 - k$ blocks intersecting both A_i and A_j and we have a $(k^2 - k)$-CID. ■

VI. Two-hop Paths in the DSNs

Suppose we use a (v, b, r, k)-configuration that is a μ-CID for key predistribution in a DSN. We can analyze the local connectivity of the network using a method similar to that used in [6]. We assume that the sensor nodes are distributed in the Euclidean plane in a random way and the range covered by each node forms a circle of fixed radius whose center is that node. We call this circle a *neighborhood* of the given sensor node.

Suppose that N_i and N_j are two nodes that are in each other's neighborhood. The probability that N_i and N_j share a common key (i.e., A_i is adjacent to A_j in the block graph) is

$$p_1 = \frac{k(r-1)}{b-1}. \tag{3}$$

Let η denote the number of nodes in the intersection of the neighborhoods of the two nodes N_i and N_j. (In general, η depends on the size of the physical area where the nodes are deployed, the distance between nodes, and on the total number of sensor nodes in the DSN). The probability (denoted by p_2) that N_i and N_j do not share a common key, but there exists a node N_h in the intersection of their neighborhoods such that N_h shares a key with both N_i and N_j, is estimated as follows:

$$p_2 \approx \left(1 - \frac{k(r-1)}{b-1}\right) \times \left(1 - \left(1 - \frac{\mu}{b-2}\right)^{\eta}\right). \tag{4}$$

Then the probability that N_i is connected to N_j via a path of length one or two is roughly $p_1 + p_2$.

Example 3: Suppose we use a TD$(30, 49)$ as a key predistribution scheme. From Theorem 14, we see that the transversal design yields a $(1470, 2401, 49, 30)$-configuration which is an 870-CID. We can support 2401 nodes in the resulting DSN, and every node is required to store 30 keys.

Now suppose that nodes are distributed in a physical region in such a way that $\eta \geq 20$. Then, from (3) and (4), we have

$$\begin{aligned} p_1 &= 0.6, \\ p_2 &\approx 0.39995, \quad \text{and} \\ p_1 + p_2 &\approx 0.99995. \end{aligned}$$

Hence, in the resulting DSN, the probability that two nearby nodes are not connected in one or two hops is less than 0.00005. □

VII. Optimal Configurations

First, we state an easy upper bound on μ^*.

Lemma 15: If there exists a (v, b, r, k)-configuration and $\mu^*(v, b, r, k) < \infty$, then

$$\mu^*(v, b, r, k) \leq (r-1)k.$$

In the rest of this section, we consider configurations which satisfy the upper bound of Lemma 15. Thus we suppose that there exists a (v, b, r, k)-configuration that is an $(r-1)k$-CID. We will call such a configuration *optimal*.

Suppose we define a relation $\sim$ on the set of blocks in an optimal configuration as follows:

$$A_i \sim A_j \Leftrightarrow A_i = A_j \text{ or } A_i \cap A_j = \emptyset.$$

In an optimal configuration, there does not exist a subset of three blocks such that two of them intersect and the third one is disjoint from the first two. It follows from this fact that $\sim$ is an equivalence relation.

Let $\mathcal{C}_1, \ldots, \mathcal{C}_m$ be the equivalence classes of blocks. Each $\mathcal{C}_i$ is a set of disjoint blocks, which we call a *partial parallel class*. Further, any two blocks from different partial parallel classes intersect in a unique point.

Since any block in any class $\mathcal{C}_j$ intersects

$$(r-1)k = \sum_{i \neq j} |\mathcal{C}_i|$$

other blocks, it follows that all the equivalence classes contain the same number of blocks, say s blocks. Therefore

$$b = ms, \tag{5}$$

because each of the m classes contains s blocks. Also, every block A_j intersects $(r-1)k$ other blocks, which must equal the number of blocks not in the same class as A_j. Hence, we obtain that

$$s(m-1) = (r-1)k. \tag{6}$$

Two equations (5) and (6) allow us to solve for s and m in terms of the other parameters.

In view of the discussion above, we obtain the following result.

Lemma 16: In any optimal (v, b, r, k)-configuration, the set of blocks can be partitioned into m partial parallel classes, each of which contains s blocks, where

$$s = b - (r-1)k \quad \text{and} \quad m = \frac{b}{b - (r-1)k}.$$

Furthermore, any two blocks from different partial classes intersect in exactly one point.

Now, suppose $(Y, \mathcal{B})$ is the dual set system of an optimal (v, b, r, k)-configuration. Clearly $(Y, \mathcal{B})$ is a (b, v, k, r)-configuration. Two points in $(Y, \mathcal{B})$ are contained in a block if and only if the corresponding blocks in the original configuration are not from the same partial parallel class. ¿From this, it follows that $(Y, \mathcal{B})$ is a group-divisible design of type s^m and block-size r, where m and s are given by Lemma 16.

Conversely, suppose we start with a group-divisible design of type s^m and block-size r. It is straightforward to check that the dual of this design is an optimal (v, ms, r, k)-configuration, where

$$k = \frac{(m-1)s}{r-1}$$

and

$$v = \frac{m(m-1)s^2}{r(r-1)}.$$

The above discussion is summarized in the following theorem.

Theorem 17: An optimal (v, b, r, k)-configuration exists if and only if there exists a group-divisible design of type s^m and block-size r, where m and s are given by Lemma 16.

Example 4: There exists a group-divisible design of type 3^9 and block size four (see [3]). The dual set system is an optimal $(54, 27, 4, 8)$-configuration.

There is extensive research on group-divisible designs, and various necessary and sufficient conditions are known. For example, the following results can be found in [3]:

Theorem 18: Necessary conditions for the existence of a group-divisible design of type s^m and block-size r are as follows:

1) $m \geq r$,
2) $(m-1)s \equiv 0 \pmod{r-1}$, and
3) $m(m-1)s^2 \equiv 0 \pmod{r(r-1)}$.

These necessary conditions are sufficient for $r = 2, 3$ and 4, with the exception of group-divisible designs of types 2^4 and 6^4 and block-size four, which do not exist.

Theorem 18 together with Theorem 17 yields many examples of optimal (v, b, r, k)-configurations with $r = 2, 3, 4$. Also, the dual of a transversal design is an optimal configuration.

VIII. Resiliency of DSNs

If a sensor node is detected as being compromised, then all of the k keys it possesses should no longer be used by any node in the network. This can affect the connectivity of the network. For example, Suppose that N_h, N_i, N_j all have a common key L and N_h is compromised. Assuming that the key predistribution is done using a (v, b, r, k)-configuration, we conclude that N_i and N_j can no longer communicate directly. This is because N_i and N_j hold at only one common key (namely, L) and this key has been compromised. In such a situation, we say that the compromise of N_h *affects* the link from N_i to N_j.

In general, an arbitrary link (i.e., a common key L held by two given nodes N_i and N_j) is affected with probability $(r-2)/(b-2)$ by the compromise of some other random node, because there are $r-2$ other nodes that contain the key L. More generally, the compromise of s random nodes will affect a given link with probability roughly equal to

$$\mathsf{fail}(s) = 1 - \left(1 - \frac{r-2}{b-2}\right)^s. \tag{7}$$

We will use this as a very rough measure of the resiliency of the DSN. In general, we want $\mathsf{fail}(s)$ to be small, at least when s is small.

Example 5: We return to Example 3 and consider the resiliency of the DSN. Recall that we are using a $(1470, 2401, 49, 30)$-configuration, so we have $b = 2401$ and $r = 49$. Then $\mathsf{fail}(10) \approx 0.17951$, so any given link is affected with a probability of about 18% when 10 random nodes are compromised. □

IX. Shared-key Discovery

By its nature, a randomized KPS has no "structure". As a consequence, shared-key discovery between two nodes N_i and N_j typically requires the nodes to exchange the list of indices of the keys they hold in order for them to be able to determine if they share a common key. This increases the communication complexity of the protocol, decreases battery life, etc.

One advantage of using deterministic KPS based on (v, b, r, k)-configurations (as opposed to a randomized KPS) is that the (v, b, r, k)-configurations may have a compact and efficient algebraic description. This may yield nice algorithms for shared-key discovery, in which very little information needs to be broadcast.

To illustrate, suppose we are using a KPS based on a transversal design $\mathrm{TD}(k, p)$ as constructed in Theorem 6. In the resulting DSN, each node is identified by an ordered pair $(i, j) \in \mathbb{Z}_p \times \mathbb{Z}_p$. We will show that it is sufficient for two nodes to exchange their identifiers. Suppose that the two nodes are denoted $N_{(i,j)}$ and $N_{(i',j')}$. These two nodes can independently determine if they share a common key in $O(1)$ time, as follows:

1) If $i = i'$ (and hence $j \neq j'$) then $N_{(i,j)}$ and $N_{(i',j')}$ do not share a common key.
2) Otherwise, compute $x = (j' - j)(i - i')^{-1} \bmod p$. If $0 \leq x \leq k-1$, then $N_{(i,j)}$ and $N_{(i',j')}$ share the common key $L_{(x,ix+j)}$. If $x \geq k$, then $N_{(i,j)}$ and $N_{(i',j')}$ do not share a common key.

Further, if the two nodes $N_{(i,j)}$ and $N_{(i',j')}$ do not share a common key, then they can easily determine if there are two-hop paths joining them, given the identifiers of all the nodes in the intersection their neighborhoods.

References

[1] S.A. Çamtepe and B. Yener. Combinatorial design of key distribution mechanisms for wireless sensor networks. Technical Report TR-04-10, RPI Dept. of Computer Science, April 2004.

[2] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Research in Security and Privacy*, May 2003, pp. 197–213

[3] C.J. Colbourn and J.H. Dinitz, editors. *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, 1996.

[4] W. Du, J. Deng, Y.S. Han, and P.K. Varsheney. A pairwise key predistribution scheme for wireless sensor networks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS)*, October 2003, pp. 42–51.

[5] L. Eschenauer and V.D. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 41–47, November 2002.

[6] J. Lee and D.R. Stinson. Deterministic key predistribution schemes for distributed sensor networks. To appear in *Lecture Notes in Computer Science* (SAC 2004 Proceedings).

[7] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS)*, October 2003, pp. 52–61.

[8] D.R. Stinson. *Combinatorial Designs: Constructions and Analysis*. Springer-Verlag, New York, 2003.

[9] R. Wei and J. Wu. Product construction of key distribution schemes for sensor networks. To appear in *Lecture Notes in Computer Science* (SAC 2004 Proceedings).