On the Construction of Practical Key Predistribution Schemes for Distributed Sensor Networks using Combinatorial Designs

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

November 8, 2005

Abstract

In this paper, we discuss the use of combinatorial set systems in the design of key predistribution schemes (KPSs) for sensor networks. We show that the performance of the KPS can be improved by carefully choosing a certain class of set systems as “key ring spaces”. Especially, we analyze the performance of the KPSs based on two types of transversal designs, which are represented by sets of linear polynomials and quadratic polynomials, respectively. These KPSs turn out to have significant efficiency in a shared-key discovery phase without degrading connectivity and resiliency.

1 Introduction

Distributed sensor networks (DSNs) are ad-hoc networks that are comprised of sensor nodes with limited computation and communication capabilities. Recently, sensor networks have been widely studied due to their applications in civilian areas as well as in military operations. As an example of military applications, sensor nodes might be distributed in a random way in a hostile territory in order to monitor and collect various information (e.g., acoustic, seismic, magnetic). For example, they might be spread from airplanes or carried by soldiers. Once deployed, the sensor nodes 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. For additional discussion of this model, we refer to [4, 17].

The nodes in distributed sensor networks should be able to communicate with each other in order to relay or accumulate secret information. There are three ways to establish pairwise secret keys between sensor nodes, which we briefly review now.

One method is to establish secret keys using public-key protocols such as key agreement schemes. However, public-key cryptography is not suitable in this setting due to expensive computational costs as well as the need for a public-key infrastructure (PKI).

Another strategy is to have a *base station*, which all nodes in the network are assumed to trust. The 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, and it may not be feasible to maintain a secure base station.

The third approach, which we follow in this paper, is to employ *key predistribution schemes* (or KPSs), where a set of secret keys is installed in each node before the sensor nodes are deployed. In the seminal paper by Eschenauer and Gligor [10], a probabilistic approach to key predistribution for sensor networks is proposed: every node is assigned a subset of keys (called a key ring), where the key rings are randomly chosen subsets of a given pool of keys. In this paper, we will consider *deterministic* key rings as an alternative to randomized key rings.

In the Eschenauer-Gligor scheme and in all other schemes we consider, there are three basic operations that need to be implemented: *key predistribution*, *shared-key discovery* and *path-key establishment*. Shared-key discovery refers to an algorithm that two nearby nodes will use to determine if they share a common key, or if they can construct a shared key from their common shared information. When two nodes, say x and y, do not share a common key, they atttempt to find a sequence of one or more "intermediate" nodes, say $z_1, z_2, \ldots, z_t$, such that every pair of adjacent nodes in the path $x, z_1, z_2, \ldots, z_t, y$ share a common key. This is the path-key establishment phase.

It turns out that certain combinatorial properties of the set of key rings are closely related to the performance of the resulting DSNs. Especially, we analyze two types of transversal designs in this context, which are constructed by sets of linear polynomials and quadratic polynomials, respectively. The corresponding DSNs turn out to have significantly improved efficiency in the shared-key discovery and path-key establishment phases — without degrading connectivity and resiliency of the network — as compared to DSNs based on randomized KPSs.

After reviewing and summarizing related work, we present a general framework to construct deterministic key predistribution schemes from set systems. Then we introduce various metrics to evaluate the performance of DSNs. In Sections 3, 4 and 5, we present two classes of set systems that yield DSNs with very good performance. Simulation results and comparisons of several schemes are given in Section 6. Finally, we conclude with some discussions and topics for future research.

1.1 Related work

In this section, we briefly summarize the Eschenauer-Gligor scheme and other schemes related to and/or derived from it.

1.1.1 Eschenauer and Gligor's scheme

Eschenauer and Gligor's key predistribution scheme is called the *basic scheme* in follow-up papers. The scheme is defined by two parameters, one of which is the size k of a key ring, and the other is the size P of the key pool. Every node in the network receives a random k-subset of keys from P.

1.1.2 q-composite scheme

The basic scheme was generalized by Chan, Perrig and Song [6], who stipulated that two nodes will compute a pairwise key only if they share at least q common keys. The integer q is a pre-specified *intersection threshold*. Given that two nodes have at least q common keys, they use all their common keys to compute their pairwise key by means of an appropriate key derivation function (e.g., see Section 2.1.2).

1.1.3 Çamtepe and Yener's scheme

The use of combinatorial designs in key predistribution for sensor networks was first proposed in [2]. They consider two classes of combinatorial designs: symmetric balanced incomplete block designs (in particular, finite projective planes) and generalized quadrangles.[1] The points and blocks in the set systems are associated with the distinct key identifiers and sensor nodes, respectively. Thus a sensor node associated with a block, say B, receives all the keys indexed by the points contained in B. It is possible to achieve good connectivity and resiliency by using these kinds of set systems. One limitation of this approach is that the network size is limited by the number of blocks in the set system. For this reason, they presented a *hybrid approach*, where random key rings are chosen once all the blocks in the set system are exhausted.

1.1.4 Lee and Stinson's scheme

Lee and Stinson [12] also studied the use of combinatorial designs for key predistribution for sensor networks. They introduced a class of designs termed *common intersection designs* and focussed on *transversal designs* defined by linear polynomials modulo p for a prime p.

1.1.5 Chakrabarti, Maitra and Roy's scheme

Chakrabarti, Maitra and Roy [5] also use combinatorial designs for key predistribution for sensor networks. Their main idea is to start with a certain set system (e.g., a transversal design, as proposed in [12]) and then to form key rings by *merging blocks* in the set system. This leads to larger key rings, but some other performance metrics are improved.

1.1.6 Other works

Several other constructions for key predistribution schemes for sensor networks have been proposed. Some schemes have involved "combining" basic schemes (or schemes based on combinatorial designs or graphs) with other key predistribution schemes such as Blom schemes [1]. Works following this approach include Du *et al* [9], Lee and Stinson [11], Liu and Ning [16] and Wei and Wu [20]. These schemes typically have increased storage requirements as compared to the schemes we study in this paper.

Another avenue of research has involved generalizations of the Leighton-Micali key predistribution schemes [15]. The Leighton-Micali schemes are based on hash chains, and modifications have been proposed for use in sensor networks by Lee and Stinson [13] and by Ramkumar and Memon [18].

2 Key ring spaces

2.1 Framework

In this section, we extend the framework of Eschenauer and Gligor's schemes by using combinatorial set systems as key ring spaces. This framework is common to several of the above-mentioned papers, e.g., [2, 12, 5]. We begin with the definition of a set system.

[1] Two general references on combinatorial designs are [7] and [19].

Definition 2.1. *A* set system *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* 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*).*

2.1.1 Key predistribution phase

Once the network size, denoted n, is determined, the center chooses a set system $(X, \mathcal{A})$ having exactly n blocks. The center also determines an integer q, called the *intersection threshold*. Then the set system can be used as a key predistribution scheme in a sensor network having n nodes, as follows.

Let the sensor nodes be denoted $U_1, \ldots, U_n$. Let

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

and let

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

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 $1 \leq j \leq n$, the sensor node U_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 U_j. In this context, we call $(X, \mathcal{A})$ a *key ring space.*[2]

Remark 2.1. *If there is no "convenient" set system having exactly* n *blocks, then we can use any set system having* $b \geq n$ *blocks, and randomly select* n *of the* b *blocks (sampling without replacement). However, since sensor nodes are inexpensive and the exact network size is not important, we would probably choose* n *to be equal to the number of blocks in a convenient set system.*

2.1.2 Shared-key discovery phase

Each point x_i acts as the identifier of the key L_i. Therefore, two nodes within wireless communication range must be able to compute the list of common points in the two blocks assigned to them. If the block B_i assigned to a sensor node U_i contains k points, then node U_i could broadcast the k identifiers of these points to each of its neighbours. This would allow the neighbours of U_i to determine the common keys it shares with U_i. However, we will later see that certain classes of set systems allow us to reduce this communication overhead significantly.

Suppose two sensor nodes, say U_1 and U_2, discover common points $\{x_{i_1}, \ldots x_{i_t}\} \in X$. If the number of the common points is at least q, i.e., if $t \geq q$, then they establish a secret key,

$$K_{1,2} = h(L_{i_1} \parallel \ldots \parallel L_{i_t}),$$

using a public *key derivation function* h, which has suitable input and output sizes. (Such key derivation functions are typically constructed from a suitable public hash function.)

[2]In this model, the basic scheme is obtained when the key ring space is a set of n random k-subsets of X.

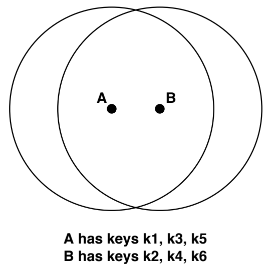

Figure 1: Two nodes with no common key

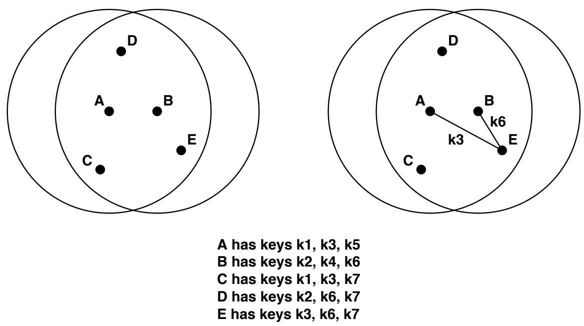

Figure 2: Nodes in a common neighborhood and a two-hop path

2.1.3 Path-key establishment phase

Path-key establishment takes place if two nodes in wireless communication range fail to find a sufficient number of common keys in the shared-key discovery phase. In this case, they look for multiple secure links (or *hops*) to reach each other, so that one of them can choose an arbitrary key and then securely relay it through the links in encrypted form to the destination node.

This is illustrated in Figures 1 and 2. In Figure 1, we have two nodes A and B within wireless communication range that have no common keys. Assume that the intersection threshold is $q = 1$. Suppose that there are three other nodes in the intersection of the neighbourhoods of A and B, namely, nodes C, D and E, as illustrated in Figure 2. Of these three nodes, only E is a suitable intermediate node for a secure two-hop path because it is the only node that has a common key with node A and a common key with node B. This permits the establishment of a secure two-hop path A, E, B, as shown in Figure 2.

2.2 Metrics for the evaluation of KPSs

In this section, we introduce several metrics to evaluate the performance of key predistribution schemes for sensor networks.

2.2.1 Key storage

The manufacturing cost of sensor nodes should be economical since a large number of nodes are deployed in a given area and the sensor nodes need to be very small and inconspicuous. Therefore, it is desirable for each sensor node to have small memory size. This limits the number of keys that can be stored in a sensor node. Perhaps 30-50 keys is a reasonable number, though some authors have considered larger numbers of keys stored in each node. In general, we would assume that each key is 128 bits in length, since these keys are to be used for symmetric-key cryptography.

2.2.2 Global connectivity

The communication capabilities of the network can be modeled as the intersection of a *physical level* and a *network level*. The physical level is represented by the *physical graph*, in which two nodes are joined by an edge if they are within wireless communication range.[3] Since sensor nodes are deployed randomly within a certain physical space, the physical graph is modelled as a *random geometric graph*, in which each vertex is in a random position in a bounded region of the euclidian plane, and two vertices are adjacent if their distance is bounded above by some fixed parameter. Observe that the physical graph does not depend on the structure of the key rings.

The network level is represented by the *network graph*, in which two nodes are adjacent if they share at least q common keys. The network graph is determined by the structure of the key ring space, and it is independent of the physical distribution of the sensor nodes. One main feature of key predistribution in sensor networks (as opposed to conventional key predistribution) is that we permit certain pairs of nodes to share (or be able to compute) no common key. Therefore, the network graph might not be a complete graph.

In order for two sensor nodes to communicate directly, the two nodes must be adjacent in both the physical graph and the network graph. We would like for the intersection of the physical graph and the network graph to be connected, so that any node can relay a message to any other node through a secure path in the network.

2.2.3 Local connectivity

We also consider the local connectivity of the network. Define $\mathbf{Pr}_1$ to be the probability that two random nodes deployed within wireless communication range share at least q common keys. We will show a bit later that the value of $\mathbf{Pr}_1$ depends only on the network graph, and it often can be computed easily from certain parameters that define the key ring space (at least in the deterministic case).

If there do not exist q common keys for a pair of nodes in wireless communication range, they should look for multiple secure links (or hops) in the path-key establishment phase. We will focus on the probability that two sensor nodes with wireless communication range can establish a *two-hop path* (since we would like the length of the path to be as short as possible in order to reduce communication overhead or complexity). We define $\mathbf{Pr}_2$ to denote the probability that two nodes U_i and U_j in wireless communcation range do not share q common keys, but there exists a node U_h in the intersection of their neighborhoods such that U_h shares q keys with both U_i and U_j. In general, $\mathbf{Pr}_2$ depends on the structure of the key ring space, as well as on the number of nodes in the common neighborhood of U_i and U_j. In general, we will asssume that there are at least η nodes in this neighbourhood, where η is a parameter that depends on the physical graph.

[3] Due to power constraints, a sensor node can communicate with nodes only within a limited radius.

2.2.4 Resiliency

In a typical attack scenario, an adversary captures a number of sensor nodes at random. In this case, we assume that all the keys or information stored in the nodes are revealed to the adversary. The adversary might use this information to eavesdrop on other links between noncompromised nodes.

For example, Suppose that U_h, U_i, U_j are three nodes, where U_i and U_j hold q common keys and these q keys are also held by U_h. Furhter, suppose that node U_h is compromised. Then we conclude that U_i and U_j can no longer communicate directly in a secure way. In such a situation, we say that the compromise of U_h *affects* the link from U_i to U_j.

We want node captures to affect as small a part of the entire network as possible. The resiliency of the network is estimated by $\mathsf{fail}(s)$, which is the probability that a link between two noncompromised nodes is affected after s nodes are compromised at random. We will compute upper bounds on $\mathsf{fail}(s)$ that are independent of the physical level of sensor networks.

2.2.5 Communication complexity of shared-key discovery and path-key establishment

By its nature, a randomized key predistribution scheme, such as the basic scheme, has no "structure" because the key rings are random subsets of the set of all keys in the key pool. As a consequence, shared-key discovery between two nodes U_i and U_j typically requires the two 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. In this situation we say that shared-key discovery has *complexity* $\Theta(k)$, where k denotes the number of keys stored in each node. If k is large, this may adversely affect the communication complexity of the protocol, decrease battery life, etc. By choosing carefully the key ring space, we can obtain a compact and efficient algebraic description of the key rings. This may yield efficient algorithms for shared-key discovery, in which only a constant amount of information needs to be broadcast (i.e., independent of the size of the key rings). If this can be achieved, then we say that shared-key discovery has complexity $\Theta(1)$. Similar analyses apply to path-key establishment.

2.2.6 Scalability

In practice, it might be very difficult to predict the number of nodes that have to be deployed in the sensor network. Moreover, power depletion or node captures might require the deployment of new nodes without any kind of reconfiguration of the system. Usually it is suggested that key predistribution schemes should be able to support a sufficiently large number of sensor nodes, approximately of orders 1,000 to 10,000. Key ring spaces derived from combinatorial designs typically have an upper bound on the size of the network they support; this upper bound is just the number of blocks in the associated set system. So we need to ensure that the set systems we construct have a sufficiently large number of blocks.

3 Common intersection designs

In this section, we briefly summarize some desirable properties of set systems for key ring spaces for distributed sensor networks when the intersection threshold is defined to be $q = 1$. Set systems known as "common intersection designs" are very useful in this context. This section is based mainly on [12].

First, 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 (see, for example, [12, 20]). Therefore, 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.

The following combinatorial lemma is elementary.

Lemma 3.1. *([12]) Any block A_j in a (v, b, r, k)-1-design intersects at most $k(r-1)$ other blocks. Further, every block in the set system intersects exactly $k(r-1)$ other blocks if and only if $|A_i \cap A_j| \leq 1$ for all $A_i, A_j \in \mathcal{A}$, $i \neq j$.*

The property in the above lemma leads to the following definition.

Definition 3.1. *A (v, b, r, k)-1-design is called a (v, b, r, k)-*configuration *if any two points occur in at most one block.*

Recall that we set the intersection threshold to be $q = 1$. If the key ring space is a (v, b, r, k)-1-design, then it is easy to see that the value of $\mathbf{Pr}_1$ is maximized precisely when the design is a configuration. For this reason , we will focus on (v, b, r, k)-configurations throughout this section.

When two nodes U_i and U_j in close proximity share no common key, the two nodes U_i and U_j can communicate via a "two-hop path" provided that there is a node U_h (which is physically close to both U_i and U_j) such that

$$A_i \cap A_h \neq \emptyset \quad \text{and} \quad A_j \cap A_h \neq \emptyset. \tag{1}$$

Ideally, we would like there to be many choices for an intermediate node that satisfies (1). This would increase the chance that at least one of these "good" intermediate nodes is physically close to both U_i and U_j.

The above discussion motivates the following definition which was given in [12].

Definition 3.2. *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 $(v, b, r, k; \mu)$-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 $A_i \cap A_j = \emptyset$.

In general, given parameters (v, b, r, k) such that a (v, b, r, k)-configuration exists, it is of interest to construct a (v, b, r, k)-configuration with μ as large as possible. This maximum value of μ will be denoted $\mu^*(v, b, r, k)$. Several results on $\mu^*(v, b, r, k)$ can be found in [12, 14].

Now, suppose we use a $(v, b, r, k; \mu)$-CID for key predistribution in a sensor network (where we set $q = 1$). We can analyze the local connectivity of the network using a method similar to that used in [11]. Suppose that U_i and U_j are two nodes that are in each other's neighborhood. The probability that U_i and U_j share a common key is

$$\mathbf{Pr}_1 = \frac{k(r-1)}{b-1}. \tag{2}$$

Let η denote the number of nodes in the intersection of the neighborhoods of the two nodes U_i and U_j. The probability $\mathbf{Pr}_2$ is estimated as follows:

$$\begin{aligned}
\mathbf{Pr}_2 &\approx (1-\mathbf{Pr}_1) \times \left(1 - \frac{\binom{b-2-\mu}{\eta}}{\binom{b-2}{\eta}}\right) && (3)\\
&\approx (1-\mathbf{Pr}_1) \times \left(1 - \left(\frac{b-2-\mu}{b-2}\right)^{\eta}\right) && (4)\\
&= (1-\mathbf{Pr}_1) \times \left(1 - \left(1-\frac{\mu}{b-2}\right)^{\eta}\right). && (5)
\end{aligned}$$

Then the probability that U_i is connected to U_j via a path of length one or two is roughly $\mathbf{Pr}_1+\mathbf{Pr}_2$.

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

$$\mathsf{fail}(s) = 1 - \left(1 - \frac{r-2}{b-2}\right)^{s}. \tag{6}$$

4 Linear schemes: transversal designs with intersection threshold one

There are many known classes of common intersections designs (see [12, 14]). In [12], it was suggested to use a certain class of common interesection designs, which are known as transversal designs, as key ring spaces. Here, we give a detailed analysis of the performance of transversal designs as key ring spaces.

Let m and k be positive integers such that $2 \le k \le m$. A *transversal design $TD(k,m)$* is a triple $(X, \mathcal{H}, \mathcal{A})$, where X is a finite set of cardinality km, $\mathcal{H}$ is a partition of X into k parts (called *groups*[4]) of size m and $\mathcal{A}$ is a set of k-subsets of X (called *blocks*), which satisfies the following properties:

- $|H \cap A| = 1$ for every $H \in \mathcal{H}$ and every $A \in \mathcal{A}$, and
- every pair of elements of X from different groups occurs in exactly one block in $\mathcal{A}$.

Transversal designs are closely related to other combinatorial structures. For example, a $\mathrm{TD}(k,m)$ is equivalent to a set of $k-2$ *mutually orthogonal Latin squares* of order m (see [19, Theorem 6.44]).

For a prime p, a $\mathrm{TD}(k,p)$ can be easily constructed, as seen in the following theorem.

Theorem 4.1. *Suppose that p is prime and $2 \le k \le p$. Then there exists a $TD(k,p)$.*

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

[4]Here, the use of the term "group" is historical. In particular, the groups are not algebraic groups.

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 $(a,b) \in \mathbb{Z}_p \times \mathbb{Z}_p$, define a block

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

Let

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

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

Remark 4.1. *Theorem 4.1 can be extended to the case where p is a prime power, by performing all operations in the finite field of order p.*

Here is a small example to illustrate the construction of a transversal design.

Example 4.1. *Suppose we take $p = 5$ and $k = 4$; then we construct a TD$(4,5)$. Suppose we write the ordered pairs $(i,j) \in \{0,1,2,3\} \times \mathbb{Z}_5$ in the form ij. Then the groups are*

$$\{00,01,02,03,04\} \quad \{10,11,12,13,14\} \quad \{20,21,22,23,24\} \quad \{30,31,32,33,34\}$$

and the blocks are

$$\begin{array}{lllll}
\{00,10,20,30\} & \{01,11,21,31\} & \{02,12,22,32\} & \{03,13,23,33\} & \{04,14,24,34\} \\
\{00,11,22,33\} & \{01,12,23,34\} & \{02,13,24,30\} & \{03,14,20,31\} & \{04,10,21,32\} \\
\{00,12,24,31\} & \{01,13,20,32\} & \{02,14,21,33\} & \{03,10,22,34\} & \{04,11,23,30\} \\
\{00,13,21,34\} & \{01,14,22,30\} & \{02,10,23,31\} & \{03,11,24,32\} & \{04,12,20,33\} \\
\{00,14,23,32\} & \{01,10,24,33\} & \{02,11,20,34\} & \{03,12,21,30\} & \{04,13,22,31\}.
\end{array}$$

Theorem 4.2. *If there exists a TD(k,m), then there is a $(km, m^2, m, k; k^2-k)$-CID.*

Proof. Let $(X, \mathcal{H}, \mathcal{A})$ be a TD(k,m). It is easy to verify that the set system $(X, \mathcal{H}, \mathcal{A})$ is a (km, m^2, m, k)-1-design. Now, let A and B be two disjoint blocks. Suppose $A = \{x_1, \ldots, x_k\}$ and $B = \{y_1, \ldots, y_k\}$, where $x_i, y_i \in H_i$ for $1 \leq i \leq k$, and $\mathcal{H} = \{H_1, \ldots, H_k\}$. There is no block containing the pair $\{x_i, y_i\}$, $1 \leq i \leq k$. However, there is a unique block containing any pair $\{x_i, y_j\}$, where $i \neq j$. Hence the design is a (k^2-k)-common intersection design. $\square$

The following result is an immediate corollary of Theorems 4.1 and 4.2 and Remark 4.1.

Corollary 4.3. *For any positive integer k and any prime power m such that $2 \leq k \leq m$, there is a $(km, m^2, m, k; k^2-k)$-CID.*

4.1 Analysis

The key predistribution schemes derived from Corollary 4.3 will be termed *linear schemes*. In this section, we analyze various performance metrics for the linear schemes.

4.1.1 Local connectivity

First, the values of $\mathbf{Pr}_1$ and $\mathbf{Pr}_2$ can be computed using equations (2) and (5). We obtain the following:

$$\mathbf{Pr}_1 = \frac{k(p-1)}{p^2-1} = \frac{k}{p+1} \tag{7}$$

and

$$\mathbf{Pr}_2 \approx \left(1 - \frac{k}{p+1}\right) \times \left(1 - \left(1 - \frac{k(k-1)}{p^2-2}\right)^{\eta}\right). \tag{8}$$

Example 4.2. *Suppose we use a TD$(30, 49)$ as a key ring space. This transversal design yields a $(1470, 2401, 49, 30)$-1-design. We can support up to 2401 nodes in the network, 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 (7) and (8), we have*

$$\begin{aligned} \mathbf{Pr}_1 &= 0.6, \\ \mathbf{Pr}_2 &\approx 0.39995, \quad \textit{and} \\ \mathbf{Pr}_1 + \mathbf{Pr}_2 &\approx 0.99995. \end{aligned}$$

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

Even for smaller values of η, we achieve very good local connectivity:

η	$\mathbf{Pr}_1$	$\mathbf{Pr}_2$	$\mathbf{Pr}_1 + \mathbf{Pr}_2$
1	0.6	0.145	0.745
2	0.6	0.237	0.837
3	0.6	0.296	0.896
4	0.6	0.334	0.934
5	0.6	0.358	0.958
10	0.6	0.396	0.996
15	0.6	0.3995	0.9995
20	0.6	0.39995	0.99995

4.1.2 Resiliency

Next, we obtain estimates for $\mathsf{fail}(s)$ immediately from (6) by setting $r = p$ and $b = p^2$:

$$\mathsf{fail}(s) = 1 - \left(1 - \frac{p-2}{p^2-2}\right)^s. \tag{9}$$

Example 4.3. *We return to Example 4.2 and consider the resiliency of the network. Recall that we are using a TD$(30, 49)$, so $\mathsf{fail}(10) \approx 0.1795$, which means any given link is affected with a probability of about 18% when 10 random nodes are compromised.*

For smaller values of s, we achieve even better resiliency:

s	$\mathsf{fail}(s)$
1	0.0196
2	0.0388
3	0.0576
4	0.0761
6	0.1119
8	0.1464

4.1.3 Communication complexity

Now we consider the communication complexity of shared-key discovery and path-key establishment. Suppose p is prime and we are using a linear scheme based on a transversal design $\mathrm{TD}(k,p)$. In the resulting network, each node is identified by an ordered pair $(a,b) \in \mathbb{Z}_p \times \mathbb{Z}_p$. We will show that it is sufficient for two nodes to exchange their identifiers. Suppose that the two nodes are denoted $U_{(a,b)}$ and $U_{(a',b')}$. These two nodes can independently determine if they share a common key in $\Theta(1)$ time (as a function of k), as follows:

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

Next, we briefly describe the method to establish two-hop path keys, in the case that two nodes $U_{(a,b)}$ and $U_{(a',b')}$ find no common key. Suppose $U_{(a,b)}$ and $U_{(a',b')}$ broadcast the messages (a,b,a',b') and (a',b',a,b), respectively. Any node $U_{(a'',b'')}$ that receives both messages must be located within the common neighborhood of $U_{(a,b)}$ and $U_{(a',b')}$. Now $U_{(a'',b'')}$ checks if it shares keys with both $U_{(a,b)}$ and $U_{(a',b')}$, using the method described above. If so, then $U_{(a'',b'')}$ generates an arbitrary session key SK, and transmits it to $U_{(a,b)}$ and $U_{(a',b')}$ in encrypted form. Then the session key SK can be used to encrypt information exchanged between $U_{(a,b)}$ and $U_{(a',b')}$ via the intermediate node $U_{(a'',b'')}$.

4.1.4 Scalability

Finally, we consider the scalability of the resulting scheme. Suppose we begin by specifying desired values for the two parameters k (where k denotes the number of keys to be stored in each node) and $\mathbf{Pr}_1$. In order to apply Corollary 4.3, it follows from (7) that we need to choose a prime (or prime power) p such that $p + 1 \leq k/\mathbf{Pr}_1$. Then the maximum network size supported by the corresponding KPS is p^2.

Example 4.4. *For $k = 30$ and various values of ρ, we compute the maximum network size based* *on a TD(k, p), where p is prime and $\mathbf{Pr}_1 \geq \rho$. We obtain the following values:*

ρ	p	p^2 = *maximum network size*
0.9	31	961
0.8	31	961
0.7	41	1681
0.6	47	2209
0.5	59	3481
0.4	73	5329
0.3	97	9409
0.2	149	22201

5 Quadratic schemes: a family of KPSs with intersection threshold two

In this section, we present a new class of key predistribution schemes with intersection theshold $q = 2$. We need to begin with the definition of a generalized transversal design. Let t, m and k be positive integers such that $t \leq k \leq m$. A *transversal design TD*(t, k, m) is a triple $(X, \mathcal{H}, \mathcal{A})$, where X is a finite set of cardinality km, $\mathcal{H}$ is a partition of X into k parts (called *groups*) of size m and $\mathcal{A}$ is a set of k-subsets of X (called *blocks*), which satisfy the following properties:

- $|H \cap A| = 1$ for every $H \in \mathcal{H}$ and every $A \in \mathcal{A}$, and
- every subset of t elements of X from t different groups occurs in exactly one block in $\mathcal{A}$.

Remark 5.1. *A TD(k, m), as defined in Section 4, is identical to a TD($2, k, m$) as defined above.*

The following construction generalizes Theorem 4.1.

Theorem 5.1. *Suppose that p is prime and $t \leq k \leq p$. Then there exists a TD(t, 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 $(t+1)$-tuple $\mathbf{c} = (c_0, \ldots, c_t) \in (\mathbb{Z}_p)^{t+1}$, define a block

$$A_{\mathbf{c}} = \left\{ \left(x, \sum_{i=0}^{t} c_i x^i \bmod p \right) : 0 \leq x \leq k-1 \right\}.$$

Let

$$\mathcal{A} = \{A_{\mathbf{c}} : \mathbf{c} \in (\mathbb{Z}_p)^{t+1}\}.$$

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

Remark 5.2. *The construction given in Theorem 5.1 can be viewed as evaluating every polynomial of degree t (or less) at k specified points in $\mathbb{Z}_p$. This construction can also be extended to prime powers p.*

We record some combinatorial properties of $\mathrm{TD}(3,k,p)$ in the following theorem.

Theorem 5.2. *Suppose $(X,\mathcal{H},\mathcal{A})$ is a $TD(3,k,p)$. Then every point $x \in X$ occurs in exactly p^2 blocks, and every pair of points from different groups occurs in exactly p blocks. Further, any block $A \in \mathcal{A}$ intersects exactly $a_1 = k(p-1)(p-k+2)$ blocks in one point, exactly $a_2 = \binom{k}{2}(p-1)$ blocks in two points, and is disjoint from exactly $a_0 = p^3 - 1 - a_1 - a_2$ blocks.*

Proof. First, let x and y be any two points from different groups. Let H be a group such that $x, y \notin H$. Then, for every $z \in H$, there is a unique block containing x, y and z. These p blocks are all the blocks containing x and y.

Next, let x be any point and let H be any group such that $x \notin H$. For every point $z \in H$, there are p blocks containing x and z. The resulting p^2 blocks are distinct and account for all the blocks containing x.

Now, let A be a block. There are $\binom{k}{2}$ ways to choose two points $x, y \in A$. For each such choice of x and y, there are $p-1$ blocks other than A that contain x and y. The resulting $\binom{k}{2}(p-1)$ blocks are distinct and account for all the blocks that intersect A in exactly two points.

Suppose there are a_i blocks that intersect A in exactly i points, $i = 0, 1, 2$. We have shown above that $a_2 = \binom{k}{2}(p-1)$. Now, suppose that $x \in A$. There are $(p-1)(k-1)$ blocks that contain x and exactly one other point from A; these blocks intersect A in exactly two points. There remain $p^2 - 1 - (p-1)(k-1) = (p-1)(p-k+2)$ blocks other than A that contain x. Since there are k points $x \in A$, it follows that $a_1 = k(p-1)(p-k+2)$.

Finally, since the total number of blocks is p^3, it follows that $a_0 = p^3 - 1 - a_1 - a_2$. □

We will use the transversal designs $\mathrm{TD}(3,k,p)$ that are constructed from Theorem 5.1 as key ring spaces. For these schemes, which we term *quadratic schemes*, we set the intersection threshold to be $q = 2$. Each block in a $\mathrm{TD}(3,k,p)$ has size k, so the number of keys per node is also equal to k in the resulting KPS. Since the total number of blocks is p^3, the corresponding sensor network is able to accommodate up to p^3 sensor nodes.

5.1 Analysis

5.1.1 Local connectivity

The local connectivity of the sensor network is easily computed from the parameter a_2 defined in Theorem 5.2. We obtain the following formula for $\mathbf{Pr}_1$:

$$\mathbf{Pr}_1 = \frac{a_2}{p^3 - 1} = \frac{\binom{k}{2}(p-1)}{p^3 - 1} = \frac{k(k-1)}{2(p^2 + p + 1)}. \tag{10}$$

We now consider how to estimate $\mathbf{Pr}_2$. This will depend on another parameter of the $\mathrm{TD}(3,k,p)$, denoted by μ_2, which we define now. For two blocks A and A', define

$$\mu_2(A, A') = |\{A'' : |A \cap A''| = |A' \cap A''| = 2\}|.$$

Then define

$$\mu_2 = \min\{\mu_2(A, A') : |A \cap A'| \leq 1\}.$$

We have the following formula, which is the same as (5) except that μ has now been replaced by μ_2:

$$\mathbf{Pr}_2 \approx \left(1 - \frac{k(k-1)}{2(p^2+p+1)}\right) \times \left(1 - \left(1 - \frac{\mu_2}{b-2}\right)^{\eta}\right). \tag{11}$$

In order to compute $\mathbf{Pr}_2$, we need to know the value of μ_2 for a given $\mathrm{TD}(3, k, p)$. It seems difficult to find a formula for these values. However, for a wide range of ordered pairs (k, p), we have computed μ_2 for the transversal designs $\mathrm{TD}(3, k, p)$ constructed in Theorem 5.1. The computed values of μ_2 are listed in Table 1.

Example 5.1. *We compute the local connectivity of a DSN based on a* $TD(3, 23, 23)$ *for various values of* η*. This transversal design has* $\mu_2 = 2420$ *(see Table 1), and each node in the network is required to store* 23 *keys.*

η	$\mathbf{Pr}_1$	$\mathbf{Pr}_2$	$\mathbf{Pr}_1 + \mathbf{Pr}_2$
1	0.458	0.108	0.565
2	0.458	0.194	0.652
3	0.458	0.264	0.721
4	0.458	0.319	0.777
5	0.458	0.364	0.821
10	0.458	0.483	0.941
15	0.458	0.523	0.981
20	0.458	0.536	0.994

5.1.2 Resiliency

Suppose that s random sensor nodes are compromised. Let U_i and U_j be two noncompromised nodes that have two common keys, say L_{P_1} and L_{P_2}. Now we can compute the probability $\mathsf{fail}(s)$ that the link between U_i and U_j is compromised. Let $\mathcal{A}_{P_1}$ be the set of blocks containing the point P_1. $\mathcal{A}_{P_2}$ is also defined similarly. From Theorem 5.2, we have that

$$|\mathcal{A}_{P_1}| = |\mathcal{A}_{P_2}| = p^2$$

and

$$|\mathcal{A}_{P_1} \cap \mathcal{A}_{P_2}| = p.$$

In order for the link between U_i and U_j to remain secure, all the blocks associated with the compromised nodes should be contained either in $\mathcal{A} \setminus \mathcal{A}_{P_1}$ or in $\mathcal{A} \setminus \mathcal{A}_{P_2}$. Therefore the compromise of s random nodes will affect a given link with probability roughly equal to

$$\begin{aligned} \mathsf{fail}(s) &= 1 - \frac{2\binom{p^3-p^2}{s} - \binom{p^3-2p^2+p}{s}}{\binom{p^3-2}{s}} && (12) \\ &\approx 1 - 2\left(1 - \frac{p^2-2}{p^3-2}\right)^s + \left(1 - \frac{2p^2-p-2}{p^3-2}\right)^s. && (13) \end{aligned}$$

Table 1: Values of μ_2 for certain transversal designs $\mathrm{TD}(3,k,v)$

$p=7$											
k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2
3	0	4	0	5	4	6	14	7	36		
$p=11$											
k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2
3	0	4	0	5	0	6	6	7	18	8	40
9	74	10	126	11	200						
$p=13$											
k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2
3	0	4	0	5	0	6	4	7	14	8	32
9	60	10	102	11	162	12	248	13	360		
$p=17$											
k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2
3	0	4	0	5	0	6	2	7	8	8	20
9	42	10	72	11	118	12	182	13	264	14	370
15	510	16	684	17	896						
$p=19$											
k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2
3	0	4	0	5	0	6	2	7	6	8	18
9	36	10	64	11	102	12	158	13	234	14	330
15	450	16	604	17	792	18	1022	19	1296		
$p=23$											
k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2
3	0	4	0	5	0	6	0	7	4	8	12
9	24	10	48	11	82	12	126	13	186	14	264
15	364	16	490	17	644	18	830	19	1052	20	1318
21	1632	22	1998	23	2420						
$p=29$											
k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2
3	0	4	0	5	0	6	0	7	2	8	8
9	18	10	36	11	62	12	94	13	140	14	202
15	280	16	376	17	496	18	644	19	820	20	1028
21	1274	22	1558	23	1888	24	2272	25	2704	26	3200
27	3762	28	4392	29	5096						
$p=31$											
k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2	k	μ_2
3	0	4	0	5	0	6	0	7	0	8	6
9	18	10	34	11	54	12	86	13	130	14	186
15	256	16	348	17	460	18	598	19	760	20	958
21	1186	22	1454	23	1760	24	2118	25	2522	26	2984
27	3504	28	4094	29	4752	30	5486	31	6300		

Example 5.2. *We return to Examaple 5.1, where we compute* $\mathsf{fail}(s)$ *for various values of* s*:*

s	$\mathsf{fail}(s)$
1	0.0017
2	0.0069
3	0.0151
4	0.0259
6	0.0539
8	0.0883
10	0.1274

5.1.3 Communication complexity

Suppose two sensor nodes U_i and U_j in wireless communication range are assigned blocks A_f and A_g, respectively, where $f(x) = ax^2 + bx + c$ and $g(x) = a'x^2 + b'x + c'$. Since the intersection threshold is $q = 2$, the two nodes compute a pairwise key only if they have two common keys. In the shared-key discovery phase, one node, say U_i, broadcasts only its coefficients a, b and c. Once the other node U_j receives the message, it solves the equation $f(x) = g(x)$, or

$$(a - a')x^2 + (b - b')x + (c - c') = 0.$$

Provided that $a \neq a'$, the node U_j will solve the equation by computing the roots

$$x = \left(-b + b' \pm \sqrt{(b - b')^2 - 4(a - a')(c - c')}\right) \left(2(a - a')\right)^{-1} \bmod p.$$

If this equation has two distinct roots which are between 0 and $k - 1$, then a pairwise key can be computed.

Square roots modulo a prime p are easily computed using the following results:

Theorem 5.3. $a \in \mathbb{Z}$ *is a quadratic residue* $\bmod p$ *if and only if* $a^{\frac{p-1}{2}} \equiv 1 \bmod p$.

Theorem 5.4. *If* $p \equiv 3 \bmod 4$ *is prime and* $a \in \mathbb{Z}$ *is a quadratic residue modulo* p*, then the two square roots of* $a \bmod p$ *are* $\pm a^{\frac{p+1}{4}} \bmod p$.

Remark 5.3. *It is well-known that square roots modulo a prime* $p \equiv 1 \bmod 4$ *can be efficiently computed by a randomized algorithm.*

The operations that U_i and U_j will perform to determine if they can compute a pairwise key are listed in Figure 3.

5.1.4 Scalability

Suppose we first choose values for k and $\mathbf{Pr}_1$. (We observe that $\mathbf{Pr}_1 < 0.5$ in (10), because $k \leq p$.) Then

$$p^2 + p + 1 \leq \frac{k(k-1)}{2\,\mathbf{Pr}_1},$$

where p should be a prime or prime power in order to apply Theorem 5.1. The maximum number of nodes in the network is then p^3.

Figure 3: Computing a pairwise key

Input:	Two coefficient vectors, $(a,b,c),(a',b',c') \in (\mathbb{Z}_p)^3$, where $p \equiv 3 \bmod 4$ is prime.
Step 1:	If $a = a'$ then U_i and U_j cannot compute a pairwise key.
Step 2:	Otherwise, compute $\Delta = (b-b')^2 - 4(a-a')(c-c') \bmod p$. If $\Delta^{\frac{p-1}{2}} \bmod p = p-1$ or 0, then U_i and U_j cannot compute a pairwise key.
Step 3:	Otherwise (i.e., if $\Delta^{\frac{p-1}{2}} \bmod p = 1$), then compute $$x_1 = \left(-b + b' + \Delta^{\frac{p+1}{4}}\right)\left(2(a-a')\right)^{-1} \bmod p$$ and $$x_2 = \left(-b + b' - \Delta^{\frac{p+1}{4}}\right)\left(2(a-a')\right)^{-1} \bmod p.$$
Step 4:	If $0 \leq x_1, x_2 \leq k-1$, then U_i and U_j have two common keys, namely $L_{(x_i, a{x_i}^2 + bx_i + c)}$, $i = 1,2$. From these two keys, they compute their pairwise key. (If $x_1 \geq k$ or if $x_2 \geq k$, then U_i and U_j cannot compute a pairwise key.)

Example 5.3. *For $k = 23$ and various values of $\rho < 0.5$, we compute the maximum network size based on a TD$(3,k,p)$, where p is prime and $\mathbf{Pr}_1 \geq \rho$. We obtain the following values:*

ρ	p	p^3 = *maximum network size*
0.45	23	12167
0.40	23	12167
0.35	23	12167
0.30	23	12167
0.25	31	29791
0.20	31	29791
0.15	37	50653
0.10	47	103823

6 Performance comparisons

In this section, we compare the performance of several schemes. We list the schemes, along with formulas to compute $\mathbf{Pr}_1$, $\mathbf{Pr}_1$ and $\mathsf{fail}(s)$. In all the following formulas, P denotes the size of the key pool and k is the number of keys per node.

1. Basic schemes (Eschenauer and Gligor, [10]).

$$\mathbf{Pr}_1 = 1 - \frac{((P-k)!)^2}{(P-2k)!P!},$$

	Basic/1-comp. scheme	2-composite scheme	Linear scheme	Quadratic scheme
Key pool size	1580	612	2010 ($p = 67$)	930 ($p = 31$)
$\mathbf{Pr}_1$	0.440	0.440	0.441	0.438
maximum network size			4489	29791
Number of keys per node is $k = 30$				

Table 2: Parameters of KPSs

$$\mathbf{Pr}_2 = (1 - \mathbf{Pr}_1)\left(1 - \left(\frac{2\binom{P-k}{k} - \binom{P-2k}{k}}{\binom{P}{k}}\right)^{\eta}\right), \tag{14}$$

and

$$\mathsf{fail}(s) = 1 - \left(1 - \frac{k}{P}\right)^{s}. \tag{15}$$

2. 1-composite and 2-composite schemes ([6]). The local connectivity and the resiliency of a q-composite scheme are estimated by the following formulas:

$$\mathbf{Pr}_1 = 1 - (p(0) + p(1) + \cdots + p(q-1)),$$

and

$$\mathsf{fail}(s) = \sum_{i=q}^{k} \left(1 - \left(1 - \frac{k}{P}\right)^{s}\right)^{i} \frac{p(i)}{p}, \tag{16}$$

where

$$p(i) = \frac{\binom{P}{i}\binom{P-i}{2(k-i)}\binom{2(k-i)}{k-i}}{\binom{P}{k}^2},$$

and

$$p = p(q) + p(q+1) + \cdots + p(k).$$

3. Linear schemes (see Section 4).

4. Quadratic schemes (see Section 5).

In order to compare the performance of different schemes in a similar setting, we fix the number of keys per node and the local connectivity $\mathbf{Pr}_1$ of each scheme. We set $\mathbf{Pr}_1 \approx 0.44$ and the number of keys per node to be $k = 30$. To attain this connectivity and key storage, the conditions in Table 2 should be satisfied.

Now we examine the local connectivity $\mathbf{Pr}_2$, global connectivity and the resiliency of the network by repeated simulations and analysis.

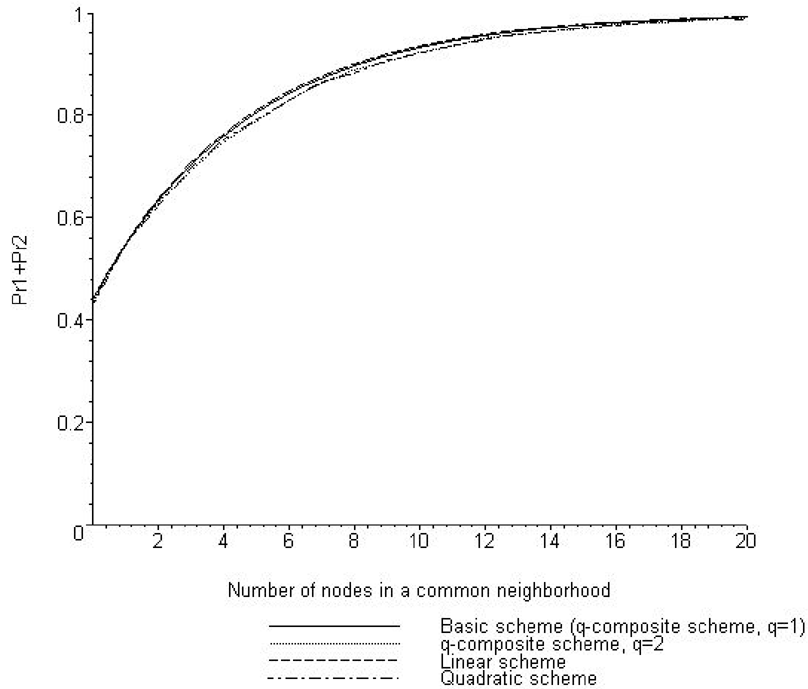

Figure 4: Local connectivity $\mathbf{Pr}_1 + \mathbf{Pr}_2$ vs. number η of nodes in a common neighborhood

6.1 Local connectivity

We compare $\mathbf{Pr}_2$ as the number η of nodes in the common neighborhood of two nodes increases from 0 to 20 by increments of 1. Since the value of $\mathbf{Pr}_1$ is fixed, we compare the values of $\mathbf{Pr}_1+\mathbf{Pr}_2$, as seen in Figure 4. We can compute $\mathbf{Pr}_2$ for basic schemes, linear schemes and quadratic schemes using (14), (5) and Table 1, respectively, while we compute $\mathbf{Pr}_2$ for the 2-composite schemes using 10,000 simulations for each value of the parameter η. The results suggest that all the schemes show almost the same local connectivity.

6.2 Global connectivity

In this simulation, sensor nodes are distributed at random in a square sensor field of length 300. So the i-th sensor node U_i is located at $(x[i], y[i])$, where $0 \leq x[i],\ y[i] < 300$. We will allow only integral coordinates here. The radius of physical communication range is fixed to be 40. The total network size will increase from 100 to 1,000 by increments of 100.

Figure 5 shows the number of globally connected sensor networks among 500 simulations for each network size. The results suggest that all the schemes show almost the same global connectivity.

6.3 Resiliency

Figure 6 compares the resiliency of the schemes with the parameters as in Table 2. Note that the resiliency does not depend on the total network size. The quadratic scheme shows better resiliency

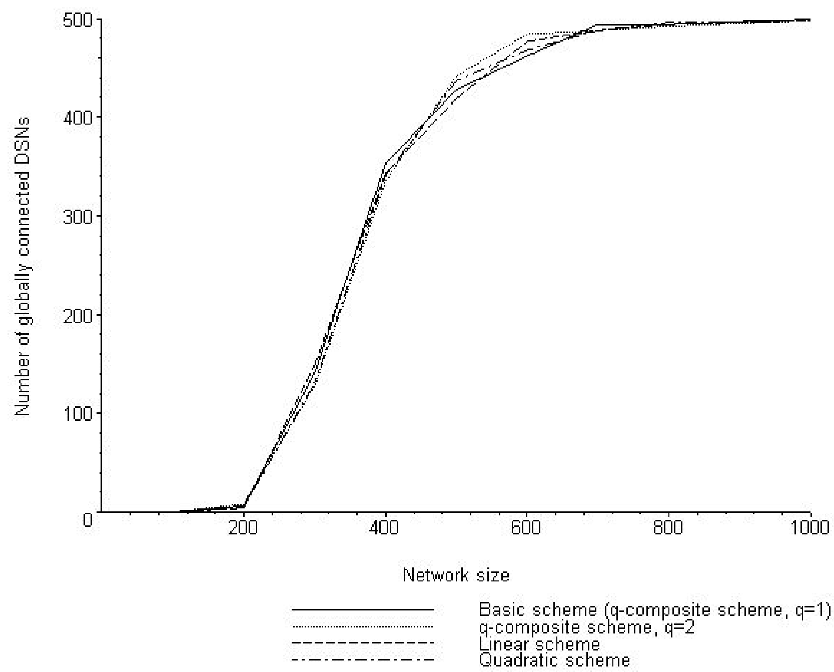

Figure 5: Number of globally connected sensor networks vs. network size

than any other scheme for a small number of compromised nodes, while the linear scheme performs best for a relatively large number of compromised nodes.

7 Conclusions

In this paper, we studied the performance of some key predistributions that are constructed deterministicaly from certain combinatorial set systems. Especially, we analyzed the performance of two types of transversal designs in this context. Several performance metrics were considered, and comparisons were made with randomized schemes. The following points summarize our findings:

- the combinatorial schemes we have presented have significantly improved communication complexity during in shared-key discovery and path-key establishment phases,
- the global and local connectivity of the various schemes are very similar, and
- the linear and quadratic schemes demonstrate better resiliency than the randomized schemes.

The only possible drawback of the combinatorial schemes is that they may not support a network of sufficiently large size. This is true especially for the linear schemes. In [2], it was suggested to use hybrid schemes, which combine the randomized and combinatorial approaches. Here, we have introduced the quadratic schemes, which support considerably larger networks and still achieve most of the other advantages of combinatorial schemes.

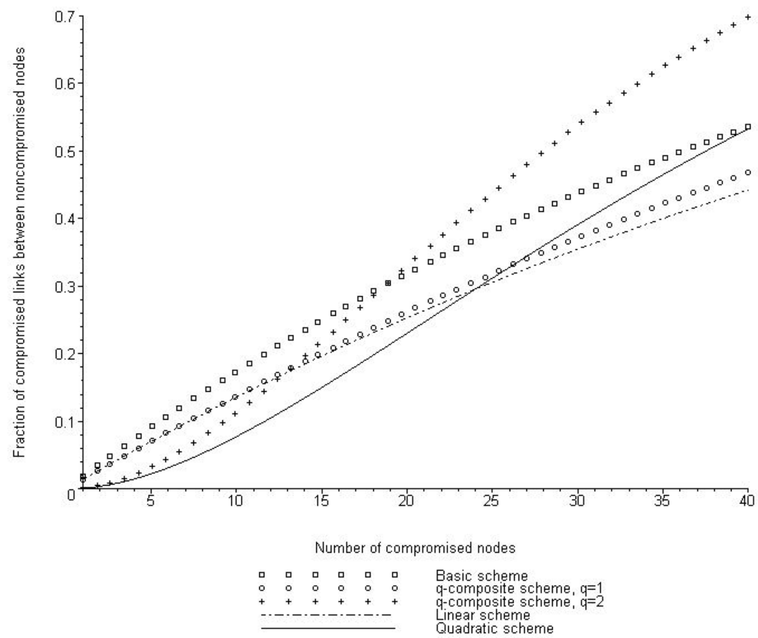

Figure 6: Fraction of compromised links between noncompromised nodes vs. number of compromised nodes

Acknowledgements

D.R. Stinson's research is supported by the Natural Sciences and Engineering Research Council of Canada through the grant NSERC-RGPIN #203114-02.

References

[1] R. Blom. An optimal class of symmetric key generation systems. *Lecture Notes in Computer Science* **209** (1985), 335–338 (EUROCRYPT 1984 Proceedings).

[2] S.A. Çamtepe and B. Yener. Combinatorial design of key distribution mechanisms for wireless sensor networks. *Lecture Notes in Computer Science* **3193** (2004), 293–308 (ESORICS 2004 Proceedings).

[3] D.W. Carmen. New directions in sensor network key management. *International Journal of Distributed Sensor Networks* **1** (2004), 3–15.

[4] D.W. Carmen, P.S. Kruus and B.J. Matt. Constraints and approaches for distributed sensor network security. *NAI Labs Technical Report #00-010*, September 2000.

[5] D. Chakrabarti, S. Maitra and B. Roy. A key pre-distribution scheme for wireless sensor networks: merging blocks in combinatorial design. *Lecture Notes in Computer Science* **3650** (2005), 89–103 (8th International Conference on Information Security, ISC 2005).

[6] 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.

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

[8] R. Di Pietro, L.V. Mancini, A. Mei, A. Panconsei and J. Radhakrishnan. Connectivity properties of secure wireless sensor networks, In *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, October 2004, pp. 53–58.

[9] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz and A. Khalili. A pairwise key predistribution scheme for wireless sensor networks. *ACM Transactions on Information and System Security* **8**, (2005), 228–258.

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

[11] J. Lee and D.R. Stinson. Deterministic key predistribution schemes for distributed sensor networks. *Lecture Notes in Computer Science* **3357** (2004), 294–307 (SAC 2004 Proceedings).

[12] J. Lee and D.R. Stinson. A combinatorial approach to key predistribution for distributed sensor networks. *the IEEE Wireless Communications and Networking Conference*, CD-ROM, 2005, paper PHY53-06.

[13] J. Lee and D.R. Stinson. Tree-based key distribution patterns. To appear in *Lecture Notes in Computer Science* (SAC 2005 Proceedings).

[14] J. Lee and D.R. Stinson. Common intersection designs. *Journal of Combinatorial Designs*, to appear.

[15] T. Leighton and S. Micali. Secret-key agreement without public-key cryptography. *Lecture Notes in Computer Science* **773** (1994), 456–479 (CRYPTO 1993 Proceedings).

[16] D. Liu, P. Ning and R. Li. Establishing pairwise keys in distributed sensor networks, *ACM Transactions on Information and System Security* **8**, (2005), 41–77.

[17] R. Roman, J. Zhou and J. Lopez. On the security of wireless sensor network. *Lecture Notes in Computer Science* **3482** (2005), 681–690 (ICCSA 2005 Proceedings).

[18] M. Ramkumar and N. Memon. An efficient key predistribution scheme for ad hoc network security. *IEEE Journal on Selected Areas in Communications*, **23**, (2005), 611–621.

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

[20] R. Wei and J. Wu. Product construction of key distribution schemes for sensor networks. *Lecture Notes in Computer Science* **3357** (2004), 280–293 (SAC 2004 Proceedings).