Product Construction of Key Distribution Schemes for Sensor Networks

R. Wei and J. Wu
Department of Computer Science
Lakehead University
Thunder Bay, Ontario P7B 5E1, Canada

May 7, 2004

Abstract

Wireless sensor networks is composed of a large number of randomly deployed sensor nodes with limited computing ability and memory space. These characteristics gives rise to much challenge to key agreement. General key agreement schemes like KDC, PKI and Diffie-Hellman key exchange schemes are not applicable to the sensor networks. Several key distribution schemes have been proposed specifically for sensor networks, aimed to provide high connectivity and resilience while keep low memory usage in the sensor nodes.

In this paper, we formularize and analysis these methods, and deduce the general condition for a scheme to be optimal in term of connectivity, resilience and memory usage. The result provides guide line to design optimal schemes. Based on the result, we proposed 2 schemes that can achieve optimal connectivity and resilience with the restriction of memory space.

1 Introduction

A distributed sensor network is composed of a large number of sensor nodes that are densely deployed. The position of sensor nodes usually are not predetermined. This allows random deployment in inaccessible terrains or disaster relief operations. This means that sensor network protocols and algorithms must possess self-organizing capabilities. In general, a sensor node is battery powered and equipped with integrated sensors, data processing capabilities, and short-range radio communications. Examples of sensor network protocols include SmartDust [9] and WINS [1]. There is a wide range of applications for sensor networks. Some examples of the application areas are health, military, and smart environment (see, i.e., [2]).

To secure communications for a sensor network is extremely important, as the network is prone to different types of malicious attacks when it is deployed in a hostile environment. An adversary can compromise sensor nodes much easier than compromise computers. However, since the limitation in both the memory resources and computing capacity of a sensor node, it is impractical to use public-key cryptosystems to secure sensor networks. Using a traditional internet style key exchange and key distribution protocols which based on trusted third parties are also impractical because the topology of a sensor network changes very frequently and sensor nodes are limited in transmission power, which only provide short distance communications.

To solve the key management problem for sensor networks, several researchers considered special key pre-distribution schemes. In [7], Eschnauer and Gligor used random methods to distribute keys. In their scheme, each sensor node received a random subset of keys from a large key pool before deployment. Any two nodes able to find one common key within their respective subsets can use that key as their shared secret to initiate communication. Some theory of random graphs was used to analysis their scheme. Based on this scheme, Chan, Perrig and Song in [4] proposed a q-composite random key pre-distribution schemes. In their scheme, q common key instead of just one common key are used to establish secure communications between two nodes, which increases the security (resilience) of the network. Recently, Du, Deng, Han and Varshney in [6] and Liu and Ning in [10] used a new method to construct key distribution schemes, which we will call it product construction. In their method, they combined traditional pairwise key distribution scheme with other schemes to construct new key distribution schemes. Their methods improved network resilience comparing to previous key pre-distribution schemes. The purpose of this paper is to formularize and analysis their methods in order to optimize this method. Upon these analysis, some combinatorial methods are then used to improve their constructions.

When we design a key distribution scheme for a distributed sensor network, the following key characteristics of the design must be considered.

- *Small key size*: Since the limited resource of a sensor node, key storage should be small. For example, if there are b nodes in the network, then we cannot expect that a node can store $b-1$ keys to share a secrete key with each of other nodes.

- *Resilience of the network*: Even a quite amount of sensor nodes are compromised by an adversary, the communications between other nodes should be still secure. In other words, a coalition of certain number of sensor nodes cannot computer other secrete keys used by other sensor nodes.

- *Local connectivity*: A sensor node should be able to securely communicate to its local neighbours. Here a local neighbour means a sensor node physically located within transmission range.

- *Global connectivity*: Any two nodes of the sensor network are connected. So for any two nodes u and v in the network, there are notes $c_1, c_2 \cdots c_t$ such that u and c_1 share a secret key, c_i and c_{i+1} share a secret key for $i = 1, \cdots, t-1$ and c_t and v share a secret key.

We will only consider schemes satisfying all these properties. The main contributions of this paper are as follows. First we use a uniformed method to generalize the methods used in [6, 10]. We define a product of a key distribution scheme and a set system and use that definition to construct new key distribution schemes. Then we use combinatorial methods to analyze the product construction and give some necessary conditions to optimize the product construction. Finally, we propose new constructions which meet all of these necessary conditions.

The rest of this paper is organized as follows. Section 2 defines production construction. In Section 3, set system used in the production construction is analyzed using combinatorial methods. Section 4 describes our propose schemes which is then compared with the previous schemes. Section 5 concludes the paper.

2 Product construction

In this section, we give a generalized description of the schemes in [6] and [10], which used a similar method to construct key pre-distribution schemes for sensor networks. We start with a definition of a *pairwise key pre-distribution scheme*.

Definition 2.1 A pairwise key pre-distribution scheme D is a triple $(\mathcal{U}, \mathcal{F}, \mathcal{K})$, where $\mathcal{U}$ is a set of nodes, $\mathcal{F}$ is a set of algorithms and $\mathcal{K}$ is a set of keys, which satisfies the following conditions:

1. For each $u \in \mathcal{U}$ an $f_u \in \mathcal{F}$ is assigned to u;
2. For any $u, v \in \mathcal{U}$ there is a unique key $K_{u,v} \in \mathcal{K}$ shared between u and v, which can be obtained from f_u and from f_v;
3. For any other $w \in \mathcal{U}$, no information about $K_{u,v}$ can be obtained by f_w.

The above definition shows that we are considering unconditional secure schemes (not for computational secure ones).

If a pairwise key pre-distribution scheme has the property that even λ nodes are compromised the system is still secure, then we say that the scheme is λ-secure, or the scheme is λ resilient. More formally, in a λ-secure pairwise key pre-distribution scheme, for any $w_1, w_2, \cdots, w_\lambda \in \mathcal{U}, K_{u,v}$ cannot be computed by $f_{w_1}, f_{w_2}, \cdots, f_{w_\lambda}$ where u, v are different from $w_1, w_2 \cdots, w_\lambda$.

Note that it is not necessary to use a pairwise key pre-distribution scheme to a sensor network, because even two nodes shared a common key, they may not be able to communicate each other when their distance is beyond transmission range. For a sensor network, local communications are more important. So we will consider both the local connectivity and the global connectivity of a key distribution scheme for our purpose.

An example of λ-secure key pre-distribution scheme is the Blom's scheme [3] in which each node stores $\lambda + 1$ keys.

The Blom's scheme can be described as follows. Suppose there are b nodes $u_1, u_2, \cdots, u_b$ in a network. To distribute keys, an *authorized center* (AC) chooses a random bivariate symmetric polynomial in a finite field $GF(q)$:

$$f(x, y) = \sum_{i=0}^{\lambda} \sum_{j=0}^{\lambda} a_{i,j} x^i y^j,$$

where $a_{i,j} = a_{j,i}$. Then the CA gives $P_i(x) = f(x, i)$ to u_i as its personal key. The common key between u_i and u_j is $P_i(j) = P_j(i) = f(i, j)$. It is proved using a linear algebra method that a Blom's scheme is λ-secure.

To formularize the methods used in [6, 10], we need some concepts from combinatorics which we introduce below.

A *set system* S is a pair $(X, \mathcal{B})$ where X is a set of points and $\mathcal{B}$ is a collection of k-subsets (called blocks) of X. For our purpose, same blocks are allowed in a set system.

Suppose there is a map from the set of nodes $\mathcal{U}$ to $\mathcal{B}$ of a set system so that for a $u_i \in \mathcal{U}$ there is a unique $B_i \in \mathcal{B}$ corresponding to it. Then we can define a product of D and S as follows.

Definition 2.2 Suppose $D = (\mathcal{U}, \mathcal{F}, \mathcal{K})$ is a pairwise key predistribution scheme and $(X, \mathcal{B})$ is a set system, where $|\mathcal{B}| \geq |\mathcal{U}|$. Suppose there is also a map from $\mathcal{U}$ to $\mathcal{B}$ such that a $u_i \in \mathcal{U}$ is mapped to a $B_i \in \mathcal{B}$. A product of D and S, $D \times S$, is defined as a triple $(\mathcal{U}, \mathcal{F} \times \mathcal{B}, \mathcal{K} \times X)$ such that the algorithm assigned to u_i is $f_{u_i} \times B_i$.

The methods used in [6] and [10] for key establishment of sensor networks actually is the above product method. Both of the papers used Blom's scheme as D. [10] proposed two set systems. One is random subset assignment. In this assignment, each node gains a random τ-subset of X, so $\mathcal{B}$ contains u random τ-subsets (by this setting, repeated blocks are allowed). The other proposed set system in [10] is grid-based system. In this system, $X = M_1 \cup M_2 \cup \cdots \cup M_t$, where $M_1, M_2, \cdots M_t$ are disjoint m-sets for $m \geq u^{1/t}$. The set $\mathcal{B}$ contains all the subsets $\{(i_1, i_2, \cdots, i_t) : i_1 \in M_1, \cdots, i_t \in M_t\}$. [6] used the random subset assignment.

As an example, in the following we give a brief description of the random subset assignment used in both [6] and [10].

To distribute keys, the AC chooses a set X where $|X| = v$. Then for each element $i \in X$, the AC generates a random symmetric polynomial $f_i(x, y)$ as in a Blom's scheme. So v polynomials are generated. For a node $u_j \in \mathcal{U}$, the AC chooses a random k-subset of X. Denote the subset as $B = \{i_1, i_2, \cdots, i_k\}$. The keys given to u_j are $f_{i_s}(x, j)$, for $1 \leq s \leq k$. The choice of v and k depends on the connectivity of the network. To form a secret key between u_i and u_j, they will try to find a common element in the subset assigned to them. If they found the element, say i_0, then the secret key is $f_{i_0}(i, j)$.

Sometimes we also can view the product construction as using different copies of a pairwise key distribution scheme D and denote it as $D \times X$. So (D, i) will be used to denote the ith scheme. It's also called a *key space* in this paper.

The main purpose of using a set system is to add resilience of the key distribution scheme. However, [6] and [10] only discussed the specific set systems used in their schemes. In next section, we will discuss how a set system effect the resilience and connectivity of the product scheme in general.

3 Analysis of the set system

For a better key predistribution system, we should consider several things: the resilience of the system, the space requested for a node, the connectivity of the network, etc. Basically, the storage space requested for a node depends on the size of a block in the set system. In this section, we discuss how the set system used in the product construction effect the resilience and connectivity of the network.

3.1 Resilience

Suppose the resilience of the original key predistribution scheme D is λ. When a set system S is used to the product construction, we need to consider the probability that one of the schemes in the product system $D \times S$ is broken. For example we consider the probability that $(D, 1)$ is broken (We denote this event as D_1). Let p_j^1 denote the probability that exact j blocks out of s blocks contain 1. Let C_s denote the event that s nodes were compromised. Then we have

$$Pr(S_1|C_s) = \sum_{j=\lambda+1}^{s} p_j^1. \tag{1}$$

Therefore we want to keep p_j^1 as small as possible. On the other hand, since D_i and D_j are independent and we want the probability that any space is broken as small as possible, we have the following result about the structure of D, which gives some necessary condition for D.

Theorem 3.1 *In a product scheme $D \times S$, suppose D, the size of X, the size of $\mathcal{B}$ and the size of a block are fixed. Then each element of X should appear in equal number of blocks to keep the optimal resilience of the scheme.*

Proof. Suppose all the parameters of the set system mentioned in the theorem are fixed. Let $b = |\mathcal{B}|$, k be the size of a block. Suppose $X = \{1, 2, \cdots, v\}$ and $i \in X$ appears in r_i blocks. Then the probability that exact j out of s blocks contain i is

$$p_j^i = \frac{\binom{r_i}{j}\binom{b-r_i}{s-j}}{\binom{b}{s}},$$

which depends on the value of r_i. Since $\sum_{i=1}^{v} r_i = kb$ is fixed, if there are some i such that the value of $\sum_{j=\lambda+1}^{s} p_j^i$ is small, then there must be some t such that $\sum_{j=\lambda+1}^{s} p_j^t$ is larger. That means (D, t) is easier to break. ∎

In intuition, if an element appears more blocks, then the corresponding key space is a weak space. So we want the elements distributed evenly.

It is easy to check that the grid-based system satisfies the condition of Theorem 3.1 (However, we will see later that its local connectivity is not good). Theoretically, the random subset assignment also satisfies the condition of Theorem 3.1 in a sense of probability. However, in practice the random subset assignment may violate that condition. For example the worst case of the random subset assignment will not fit the condition of Theorem 3.1. So we want some deterministic method to find a set system that has even distributions of elements.

Suppose each element appears in r blocks. Then we have

$$p_j^i = \frac{\binom{r}{j}\binom{b-r}{s-j}}{\binom{b}{s}}. \tag{2}$$

Therefore the value of $Pr(D_1|C_s)$ is determined by the values of r and b.

To obtain a set system satisfying the condition of Theorem 3.1, we need a definition from combinatorial design theory. For general information about combinatorial design theory used in this paper, see [11].

Definition 3.2 *A 1-design $S_r(1, k, v)$ is a set system $(X, \mathcal{B})$ such that each element $x \in X$ is contained in exactly r blocks, where $v = |X|$, k is the block size.*

The following construction is from [11, Theorem 9.10].

Theorem 3.3 *There exists an $S_r(1, k, v)$ with b blocks if $b = vr/k$ is an integer.*

Proof. Let $u = \gcd(k, r)$. Then $r = ur'$ and $k = uk'$ where $\gcd(r', k') = 1$. Since $b = vr/k = vr'/k'$ and $\gcd(r', k') = 1$, it must be the case that $v \equiv 0 \pmod{k'}$. Let $v = sk'$ where s is a positive integer. Then $b = sr'$.

Let Y be a set of cardinality k', and define $X = Y \times \mathbb{Z}_s$. Let $A_1, A_2, \cdots, A_{r'}$ be r' arbitrary u-subsets of $\mathbb{Z}_s$. For $1 \leq i \leq r'$, define $B_i = Y \times A_i$. Then each B_i is a k-subset of X. Now for each B_i, we develop s blocks B_i^j as follows. Suppose $B_i = Y \times \{s_1, s_2, \cdots, s_{r'}\}$. Then for each $j, 1 \leq j \leq s-1$, let $B_i^j = Y \times \{s'_1, s'_2, \cdots, s'_{r'}\}$, where $s'_t = s_t + j \pmod{s}, 1 \leq t \leq r'$. The result is an $S_r(1, k, v)$. □

When $r = \binom{v}{k}$, we have a easy way to construct a 1-design.

Theorem 3.4 *There exists an $S_r(1, k, v)$ for $r = \binom{v-1}{k-1}$ and $b = \binom{v}{k}$.*

Proof. Let the set of blocks contains all the k-subsets of a v-set. □

Suppose we fix $|X| = v$, and the size of block is k. Then we want each element belongs to $\frac{kb}{v}$ blocks so that the condition of Theorem 3.1 is satisfied. In practice, we let b be multiples of v and $r = \frac{kb}{v}$.

3.2 Connectivity

To consider the connectivity of a sensor network, we consider the graph $G(\mathcal{U}, E)$, where two nodes are connected by an edge if and only if these two nodes share at least one common secret key. Following from the method used in [7], we view a sensor network as a random graph. Since the connectivity of a random graph is a monotone property (when the number of nodes are fixed, the probability of connectivity is increasing when the number of edges is increasing), according to a theory of [8], the expected node degree d can be computed as follows:

$$d = \frac{b-1}{b}(\ln b - \ln(-\ln P_c)),$$

where $b = |\mathcal{U}|$ and P_c is the probability that the random graph is connected. Therefore the connectivity of the network depends on the degree d when the number of nodes are fixed.

In the product construction, two nodes share a common secret key if and only if their blocks have at least one common element. Suppose in the set system, each block intersects t other blocks. For a given density of sensor network deployment, if the expected value of number of neighbours is n, then $d = \frac{nt}{b}$. So we have the following result.

Lemma 3.5 *The connectivity of the product scheme depends on the number of blocks which share at least one element with a block in the set system.*

From Lemma 3.5, we know that the connectivity of the scheme from grid-based system in [10] is not good. In that system, each block intersects $tv^{1/t} - t$ other blocks. However, we will see later that using other set system will improve the connectivity of the network a lot.

Suppose $X = \{1, 2, \cdots, v\}$ and $\mathcal{B} = \{B_1, B_2, \cdots, B_b\}$. An incidence matrix of the set system $(X, \mathcal{B})$ is a $b \times v$ 0-1 matrix $A = (a_{i,j})$, where

$$a_{i,j} = \begin{cases} 1 & \text{if } j \in B_i, \\ 0 & \text{otherwise.} \end{cases}$$

Let $C = AA^T = (c_{i,j})$. Then C is a symmetric $b \times b$ matrix. Suppose each element of X appears in r blocks. Then we have $c_{i,i} = k$, $0 \leq c_{i,j} \leq k$ and

$$\sum_{i=1, i \neq j}^{b} c_{i,j} = k(r-1). \tag{3}$$

The number of blocks which intersect block B_i equals to the number of nonzero elements in the ith row of C. So if we want to keep the local connectivity as large as possible, we need to let the number of nonzero elements in C as large as possible. In other words, we want to keep $c_{i,j}$ as small as possible. In intuition, we don't want repeat blocks to avoid the case that $c_{i,j} = k$ for some $i \neq j$.

Remark 3.6 From (2) and (3) we can see that there is a trad-off between the resilience and connectivity of the network. For the connectivity, we want r to be large. However, when r is larger the probability that a scheme is broken is increasing.

The following result indicates that the construction of Theorem 3.4 is optimal.

Theorem 3.7 *When $b = \binom{v}{k}, r = \binom{v-1}{k-1}$ and $v > 2k$, the set system constructed in Theorem 3.4 has largest number of intersections for a block.*

Proof. It is easy to know that a block intersects

$$I = \binom{v}{k} - \binom{v-k}{k} - 1$$

other blocks in the set system of Theorem 3.4. We are going to prove that if there are repeated blocks in an $S_r(1, v, k)$, then a block intersects less blocks.

Suppose there are two identical blocks. Then each element in that block appears in $r-2$ other blocks. So that block can intersect at most $I' = k(r-2)+1$ other blocks. Since $\binom{v}{k} = \frac{v}{k}\binom{v-1}{k-1}$ and $r = \binom{v-1}{k-1}$ we have

$$I = \frac{v}{k}\binom{v-1}{k-1} - \binom{v-k}{k} - 1,$$

and

$$\begin{aligned} I - I' &= \frac{v-k}{k}\binom{v-1}{k-1} - \binom{v-k}{k} + 2k - 2 \\ &= \frac{(v-1)(v-2)\cdots(v-k)}{(v-k)(v-k-1)\cdots(v-2k+1)}\binom{v-k}{k} - \binom{v-k}{k} + 2k - 2 \\ &= \left(\frac{(v-1)(v-2)\cdots(v-k)}{(v-k)(v-k-1)\cdots(v-2k+1)} - 1\right)\binom{v-k}{k} + 2k - 2 \\ &> 0. \end{aligned}$$

The conclusion follws. ∎

In order to use construction of Theorem 3.3, we need to consider how to choose the sets $A_1, A_2, \cdots, A_{r'}$. Suppose $S \subseteq \mathbb{Z}_s$, where $\mathbb{Z}_s$ is the additive group of order s. Define the differences of S as:

$$\bigtriangledown S = \{x - x' \pmod{s} : x, x' \in S, x \neq x'\}.$$

If an element of $\mathbb{Z}_s \setminus \{0\}$ appears t times in $\cup_i \bigtriangledown S_i$ for some subsets S_i, then we say that the element has $t-1$ repeatings. The sum of the repeatings of all elements is called the repeatings of $\cup_i \bigtriangledown S_i$.

Theorem 3.8 *The 1-design constructed from Theorem 3.3 has largest local connectivity, if the collection*

$$\cup_{i=1}^{r'} \bigtriangledown (A_i)$$

contains least repeatings.

Proof. If an element $g \in \mathbb{Z}_s \setminus \{0\}$ has $t-1$ repeatings in $\cup_{i=1}^{r'} \bigtriangledown (A_i)$, then pairs (x_i, x_j) appear in t blocks, where $x_i - x_j = g$. Since each element appears in r blocks, we want to reduce the number of blocks containing a same pair of elements to maximize the number of blocks which intersect a fixed block. □

Definition 3.9 *Let G be an additive abelian group of order v. A set system $(G, \mathcal{B})$ is called a (v, k, λ) difference family if every nonzero element of G occurs λ times in*

$$\cup_{B \in \mathcal{B}} \bigtriangledown B.$$

There are many results about the construction of difference families in literature (see i.e., [5]). From Theorem 3.8 we know that we can use blocks in a (v, k, λ) difference family with smallest λ to construct 1-design and then obtain a good product scheme.

4 Proposed scheme

Several recently proposed key distribution schemes used random distribution method [4, 6, 7, 10]. There are reasons that deterministic method should be developed as well. For example, the theoretical analysis shows scheme in [6] can provides good connectivity, resilience and memory consuming attributes to sensor networks. But the random distribution of the keys leaves open issues in practical implementation. We can look this scheme as a $D \times S$ scheme. Here S is a set system (X, B), where $X = \{A_1, A_2, \cdots, A_v\}$, $B = \{B_1, B_2, \cdots, B_b\}$, and each B_i contains k elements randomly selected from X. As we analyzed in Section 3, in a sense of probability this scheme meets Theorem 3.1 and 3.8 to achieve optimal connectivity and resilience. But in real implementation, it may produce worse result. The result depends on the random number generating function used to generate the S set system. Different random number generators may result in different S systems. Some may satisfy the condition to achieve the optimal result, but some may not. It's necessary to design a deterministic distribution scheme that meets Theorem 3.1 and 3.8.

In this section we consider the construction of set systems used in the key distribution scheme. Given the scale and connectivity of the sensor networks, the memory space of each sensor node, we need to determine which set system to use, what are the parameters of the selected set system, and how to construct the set system.

The predefined requirements and restriction on the sensor network include:

- b, the number of nodes in the sensor networks,
- M, the memory space of the sensor nodes to store the keys,
- P_c, the probability that the random graph of the sensor nodes connect, and
- n, the estimated number of neighbors of a sensor node after deployment.

The constructed set systems should meet the above requirements and restrictions and at the same time achieve optimal resilience. Analysis in Section 3 shows that set systems that meet Theorem 3.1 and 3.8 can obtain optimal resilience and connectivity. In the next parts we give construction of 2 such set systems.

4.1 Construction with $(v, k, 1)$ Difference Family

From Definition 3.9 we know that $(v, k, 1)$ Difference Families meets Theorem 3.1 and 3.8. In this parts, we give the construction of $(v, k, 1)$ Difference Families, and analyze its performance.

First we compute $P_{connect}$, which is the probability that a pair of node share at least one common key space:

$$P_{connect} = \frac{b-1}{nb}(\ln b - \ln(-\ln P_c)). \tag{4}$$

Next we choose a proper $(v, k, 1)$ difference family that can provide the desired connectivity. We compute P_{vk1}, the probability that a pair of blocks from the $(v, k, 1)$ shares an element:

$$P_{vk1} = \frac{k^2(r'-1) + k(k-1)}{vr' - 1},$$

where r' is the number of basic blocks of the chosen family. Also we can compute the number of blocks of the $(v, k, 1)$:

$$r = kr'$$

The following table chooses a sample collection of $(v, k, 1)$ difference families from [5] and computes their parameters parameters:

v	k	r'	r	P_{vk1}
25	3	4	100	0.33
27	3	5	135	0.31
31	3	5	155	0.27
33	3	6	198	0.25
37	3	6	222	0.23
39	3	7	273	0.22
43	3	7	301	0.20
40	4	4	160	0.38
49	4	6	196	0.31

Usually there are more than one families whose connectivity are better than and close to $P_{connect}$. We choose the one with least blocks that is larger than b, and assign each block to a sensor node.

The third step is to construct v key spaces. We computer the security threshold λ of the key space:

$$\lambda = \lfloor \frac{M}{k} \rfloor - 1$$

The construction of the key spaces is the same with that in [6]. We briefly introduce an example as follows.

1. Select a primitive element s from a finite field $GF(q)$, where q is the least prime larger than the key size, then generate the following $(\lambda + 1) \times b$ matrix G:

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ s & s^2 & s^3 & \cdots & s^b \\ s^2 & (s^2)^2 & (s^3)^2 & \cdots & (s^b)^2 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ s^\lambda & (s^2)^\lambda & (s^3)^\lambda & \cdots & (s^b)^\lambda \end{bmatrix}$$

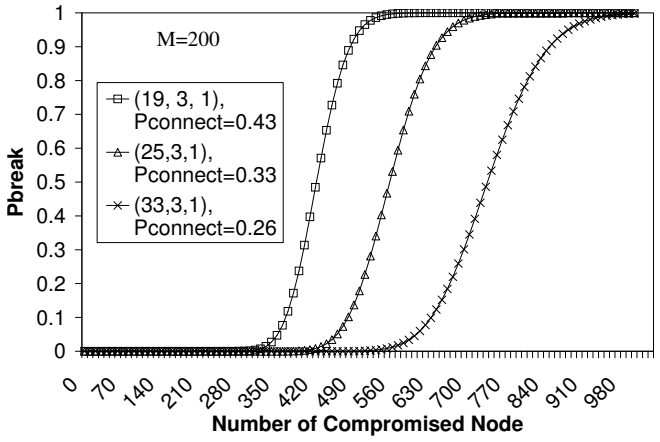

Figure 1: Probability that one key space is broken.

2. Generate v symmetric matrix $D_1, \ldots, D_v$ of size$(\lambda+1) \times (\lambda+1)$, then compute the matrixes $A_i = (D_i \cdot G)^T, i \in [1, v]$. Here we get v key spaces A_1 to A_v. Each key space is λ-secure.

Last, key spaces are assigned to the sensor nodes. For example, if a block {2,3,4} was assigned to node 5, the $5th$ rows of matrixes A_2, A_3 and A_4 are assigned to node 5.

We give an illustration of the resilience of the scheme using $(v, k, 1)$ (we call it $(v, k, 1)$ scheme). We use the $P_{break} = Pr(S_1|C_s)$ defined in (1) as an indication of resilience, and plot it as a function of number of compromised node in Figure 1. In the figure, M is set to 200, 3 schemes with different parameters and connectivity are shown. We see that to achieve a probability of 0.5 to break 1 key space, more than 200 nodes are to be compromised. The lower the $P_{connect}$ of the $(v, k, 1)$, the more compromised nodes are needed. This is the same attribute that the scheme in [6] (we call it random scheme in the following parts) provides.

Then we compare the $(v, k, 1)$ scheme with the random scheme in Figure 2. Figure 2 shows the difference between P_{break} of pairs of $(v, k, 1)$ and random schemes with same M and similar $P_{connect}$. The figure shows the difference is very small, and generally, $(v, k, 1)$ schemes are better. That means with the same M, to achieve the same connectivity, the 2 schemes risk similar resilience compromise.

4.2 Construction with all k-subsets

One potential drawback of the $(v, k, 1)$ scheme is that its number of blocks is limited. So when the network size is large, we consider using all k-subsets which provides large block size easily. As proved in Theorem 3.4 and 3.7, all k-subsets meets Theorem 3.1 and 3.8, and can provide optimal connectivity and resilience. We give construction steps of the all k-subsets as follows.

First, we compute $P_{connect}$ using (3), then we need to find the v and k so that the set system meets the requirement on the scale and connectivity. The next 2 conditions need to

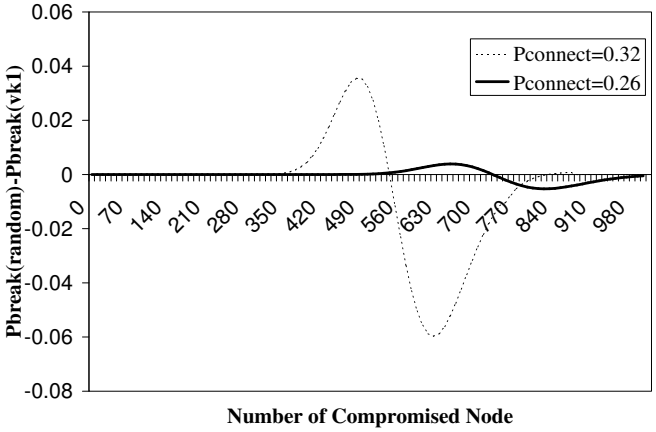

Figure 2: Difference between P_{break} of $(v, k, 1)$ scheme and random scheme.

be meet:

$$P_{connect} \geq 1 - \frac{\binom{n-k}{k}}{\binom{n}{k} - 1}$$

$$b \leq \binom{v}{k}$$

The above functions produce a list of tables for v, k, $P_{connect}$ and b. From the tables, given $P_{connect}$ and b, we can get corresponding v and k. Following is a sample table for $Pconnect = 0.3$:

v	k	b
20	3	1140
21	3	1330
22	3	1540
23	3	1771

With v and k, it's easy to construct all k-subsets.

For the all k-subsets scheme, the resilience is

$$P_{break} = \sum_{j=\lambda+1}^{s} \frac{\binom{s}{j}\binom{x-d}{s-j}}{\binom{x}{y}}$$

where $x = \binom{v}{k}$, $b = \binom{v-1}{b-1}$ and $\lambda = \lfloor \frac{M}{k} \rfloor - l$.

The $P_{connect}$ of all k-subsets scheme is

$$P_{connect} = 1 - \frac{\binom{v-k}{k}}{\binom{v}{k} - 1},$$

which is very close to that of random scheme.

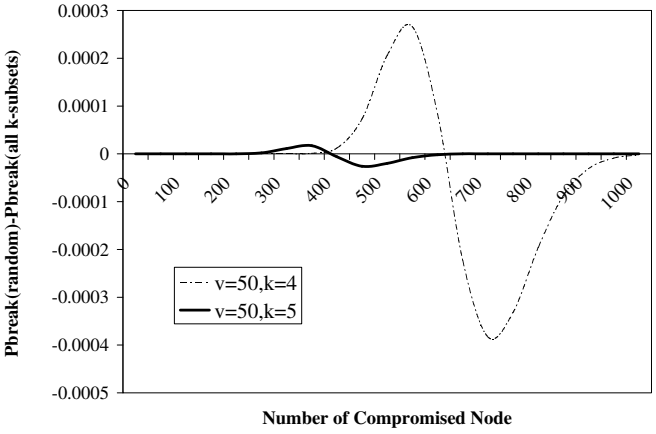

Figure 3: Difference between P_{break} of all k-subsets and random scheme

We compare the all k-subsets scheme with random scheme with the same v and k in Figure 3. The figure shows the difference between P_{break} of all k-subsets and random sets with the same v and k. M is set to 200. We can see that the difference is very limited.

5 Conclusion

In this paper, we introduced a generalized $D \times S$ key pre-distribution scheme for sensor networks. We deduced condition of the set system used in the scheme that can provide optimal connectivity and resilience to the sensor network. Based on the result we analyzed some existing key pre-distribution schemes and evaluated their strength and weakness. Then we proposed 2 specific schemes and their constructions that can achieve optimal connectivity and resilience.

This paper is focused on optimal connectivity and resilience of the key distribution scheme. Another important property of the schemes is scalability. In real implementation, the scale of the sensor networks often impacts connectivity and resilience. In the future research, we are going to focus on scalability and its relationship with connectivity and resilience, and looking for optimal schemes.

References

[1] Wireless Integrated Network Sensors, University of California,

`http://www.janet.ucla.edu/WINS`.

[2] I.F. Akyildiz, W. Su, Y. Sankarasugramaniam and E. Cayirci, A survey on sensor networks, IEEE Communications Magzine, 40(2002), 102-114.

[3] R. Blom, An optimal class of symmetric key generation systems, Advances in Cryptology: EUROCRYP 84 (T. Beth, N. Cot and I. Ingemarsson, eds.) LNCS 209 (1985), 335-338.

[4] H. Chan, A. Perrig and D. Song, Random key predistribution schemes for sensor networks, IEEE Sumposium on Research in Security and Privacy, (2003), 197-213.

[5] C. J. Colbourn and J.H. Dinitz, The CRC Handbook of Combinatorial Designs, CRC Press, 1996.

[6] W. Du, J. Deng, Y. S. Han and P. K. Varshney, A pairwise key pre-distribution scheme for wireless sensor networks, Proc. of the 10th ACM conf. on Computer and communications Security, (2003), 42-51.

[7] L. Eschenauer and V. D. Gligor, A key-management scheme for distributed sensor networks, Proc. of the 9th ACM conf. on Computer and communications Security, (2002), 41-47.

[8] Erdös and Rényi, On random graphs I. Publ.Math. Debrecen, 6(1959), 290-297.

[9] J.M. Kahn, R.H. Katz and K.S.J. Pister, Next century challenges: Mobile networking for smart dust, In: Proceedings of the 5th Annual ACM/IEEE Internation Conference on Mobile Computing and Networking, (1999), 483-492.

[10] D. Liu and P. Ning, Establishing pairwise keys in distributed sensor networks, Proc. of the 10th ACM conf. on Computer and communications Security, (2003), 52-61.

[11] D.R. Stinson, Combinatorial Designs: Constructions and Analysis, Springer, New York, 2003.