Combinatorial Aspects of Key Distribution for Sensor Networks

Douglas R. Stinson

David R. Cheriton School of Computer Science
University of Waterloo

SAC 2013

Wednesday, August 14, 2013

This talk is based on joint work with Kevin Henry, Jooyoung Lee and Maura Paterson.

Wireless Sensor Networks

- sensor nodes have limited computation and communication capabilities
- a network of 1000 – 10000 sensor nodes is distributed in a random way in a possibly hostile physical environment
- the sensor nodes operate unattended for extended periods of time
- the sensor nodes have no external power supply, so they should consume as little battery power as possible
- usually, the sensor nodes communicate using secret key cryptography
- a set of secret keys is installed in each node, before the sensor nodes are deployed, using a suitable key predistribution scheme (or KPS)
- nodes may be stolen by an adversary (this is called node compromise)

Fundamental Problems for WSNs

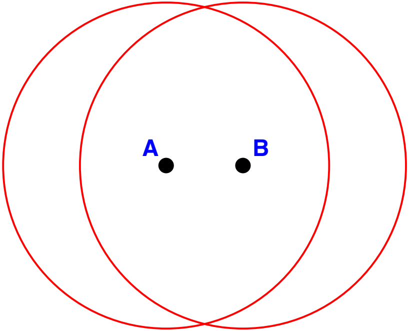
Eschenauer and Gligor (2002) introduced the following problems:

Key predistribution
: How do we assign keys to sensor nodes? We do not want to use a single key across the whole network due to the possibility of node compromise. So each node will receive a moderate sized key ring.

Shared-key discovery
: Two nodes can communicate directly only if they are in close physical proximity and they have a common key. We need an efficient method to determine if two nodes share a common key.

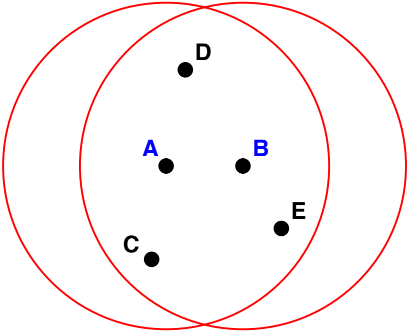
Path-key establishment
: Nodes that cannot communicate directly should be able to communicate via a multi-hop path. We need an efficient method for two nodes to determine a secure multi-hop path. (The preferred solution is a two-hop path.)

Shared-key Discovery



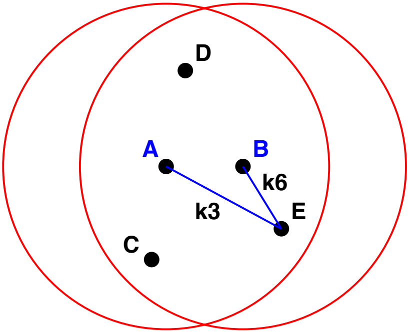
A has keys k1, k3, k5
B has keys k2, k4, k6

Path-key Establishment



A has keys k1, k3, k5
B has keys k2, k4, k6
C has keys k1, k3, k7
D has keys k2, k6, k7
E has keys k3, k6, k7

Path-key Establishment (cont.)



A has keys k1, k3, k5
B has keys k2, k4, k6
C has keys k1, k3, k7
D has keys k2, k6, k7
E has keys k3, k6, k7

Deployed WSNs

- Nodes in a WSN are often deployed in a random way over a large physical area.
- We already observed that two nodes can communicate if and only if they have a common key and they are within wireless communication range.
- Two nodes are joined by an edge in the physical graph if they are within wireless communication range.
- Two nodes are joined by an edge in the key-sharing graph if they have a common a key.
- The communication graph is the intersection of the physical graph and the key-sharing graph.
- In this talk, we focus on the key-sharing graph. (Equivalently, we can assume that all pairs of nodes are within wireless communication range.)

Two Trivial Schemes

1. If every node is given the same secret master key, then memory costs are low. However, this situation is unsuitable because the compromise of a single node would render the network completely insecure.
2. For every pair of nodes, there could be a secret pairwise key given only to these two nodes. This scheme would have optimal resilience to node compromise, but memory costs would be prohibitively expensive for large networks because every node would have to store $n-1$ keys, where n is the number of nodes in the WSN.

The Eschenauer-Gligor Scheme

- In 2002, Eschenauer and Gligor proposed a probabilistic approach to key predistribution for sensor networks.
- For a suitable value of k, every node is assigned a random k-subset of keys chosen from a given pool of v secret keys.
- Suppose that nodes $\mathbf{N_i}$ and $\mathbf{N_j}$ have exactly $\ell \geq 1$ common keys, say $\mathbf{key}_{a_1}, \ldots, \mathbf{key}_{a_\ell}$, where $a_1 < a_2 < \cdots < a_\ell$.
- Such a pair of nodes is termed an ℓ-link.
- Then $\mathbf{N_i}$ and $\mathbf{N_j}$ can each compute the same secret key,

$$K_{i,j} = h(\mathbf{key}_{a_1} \parallel \ldots \parallel \mathbf{key}_{a_\ell} \parallel i \parallel j),$$

  using a public key derivation function h.
- h could be constructed from a secure hash function.

Attack Model

- The most studied adversarial model in WSNs is random node compromise.
- An adversary compromises a fixed number of randomly chosen nodes in the network and extracts the keys stored in them.
- Any links involving the compromised nodes are broken.
- However, this can also cause other links to be broken that do not directly involve the compromised nodes.
- A link formed by two nodes $\mathbf{N_i}$ and $\mathbf{N_j}$, will be broken when a compromised node $\mathbf{N_k} \not\in \{\mathbf{N_i}, \mathbf{N_j}\}$ contains all the keys held by $\mathbf{N_i}$ and $\mathbf{N_j}$, i.e., when $\mathbf{N_i} \cap \mathbf{N_j} \subseteq \mathbf{N_k}$.
- If s nodes, say $\mathbf{N_{k_1}}, \ldots, \mathbf{N_{k_s}}$, are compromised, then a link $\mathbf{N_i}, \mathbf{N_j}$ will be broken whenever

$$\mathbf{N_i} \cap \mathbf{N_j} \subseteq \bigcup_{h=1}^{s} \mathbf{N_{k_h}}.$$

The q-composite Scheme

- In 2003, Chan, Perrig and Song suggested that two nodes should compute a pairwise key only if they share at least η common keys, where the integer $\eta \geq 1$ is a pre-specified intersection threshold.
- This decreases connectivity but increases resilience.
- For now, we will assume $\eta = 1$.

Important Metrics

Storage requirements

The number of keys stored in each node, which is denoted by k, should be "small" (e.g., at most 100).

Network connectivity

The probability that a randomly chosen pair of nodes can compute a common key is denoted by $\mathbf{Pr_1}$. $\mathbf{Pr_1}$ should be "large" (e.g., at least 0.5).

Network resilience

The probability that a random link is broken by the compromise of s randomly chosen nodes not in the link is denoted by $\mathbf{fail_s}$. We want $\mathbf{fail_s}$ to be small: high resilience corresponds to a small value for $\mathbf{fail_s}$. In this talk we mostly consider $\mathbf{fail_1}$.

Local Connectivity of the Eschenauer-Gligor Scheme

- Recall that each node contains a random k-subset of the v keys.
- The probability that a random k-subset B is disjoint from a random k-subset A is

$$\frac{\binom{v-k}{k}}{\binom{v}{k}}.$$

- Therefore,

$$\mathbf{Pr_1} = 1 - \frac{\binom{v-k}{k}}{\binom{v}{k}}.$$

- "Expanding" the binomial coefficients, we have

$$\mathbf{Pr_1} = 1 - \frac{((v-k)!)^2}{k!(v-2k)!}$$

as stated in Eschenauer and Gligor (2002).

Local Connectivity of the E-G Scheme (cont.)

- If $v \gg k$, then we can estimate $\mathbf{Pr_1}$ as follows:

$$
\begin{aligned}
\mathbf{Pr_1} &= 1 - \frac{\binom{v-k}{k}}{\binom{v}{k}} \\
&= 1 - \frac{(v-k)(v-k-1)\cdots(v-2k+1)}{v(v-1)\cdots(v-k+1)} \\
&\approx 1 - \left(\frac{v-k}{v}\right)^k \\
&= 1 - \left(1 - \frac{k}{v}\right)^k \\
&\approx 1 - \left(1 - k \times \frac{k}{v}\right) \\
&= \frac{k^2}{v}.
\end{aligned}
$$

Resilience of the Eschenauer-Gligor Scheme

- Resilience of the E-G scheme was first discussed in Chan, Perrig and Song (2003).
- However, their analysis contained some errors, as noted in Yum and Lee (2012) and Kendall, Kendall and Kendall (2012).
- The probability that a two nodes form an ℓ-link is

$$\mathbf{link}(\ell) = \frac{\binom{k}{\ell}\binom{v-k}{k-\ell}}{\binom{v}{k}}.$$

  (This formula is from Kendall, Kendall and Kendall (2012); it is a simplification of the equivalent formula first given in Chan, Perrig and Song (2003).)

- Note that

$$\mathbf{Pr_1} = \sum_{\ell=1}^{k} \mathbf{link}(\ell).$$

Resilience of the Eschenauer-Gligor Scheme (cont.)

- Define $\mathbf{fail_s}(\ell)$ to be the probability that an ℓ-link is broken by the compromise of s random nodes not in the link.
- Resilience is given by the formula

$$\mathbf{fail_s} = \frac{1}{\mathbf{Pr_1}} \sum_{\ell=1}^{k} (\mathbf{link}(\ell) \times \mathbf{fail_s}(\ell)).$$

- It is easy to see that

$$\mathbf{fail_1}(\ell) = \frac{\binom{v-\ell}{k-\ell}}{\binom{v}{k}}. \qquad (1)$$

- Kendall, Kendall and Kendall (2012) use inclusion-exclusion to prove a general formula for $\mathbf{fail_s}(\ell)$:

$$\mathbf{fail_s}(\ell) = 1 - \sum_{i=1}^{\ell} (-1)^{i-1} \binom{\ell}{i} \left(\frac{\binom{v-i}{k}}{\binom{v}{k}} \right)^{s}. \qquad (2)$$

Resilience of the Eschenauer-Gligor Scheme (cont.)

- If we substitute $s = 1$ into (2) and apply some binomial identities, we get the formula (1).

- We make a final observation concerning an estimate for $\mathbf{fail_1}$.
- When $v \gg k^2$, most links are 1-links.
- In this situation, we can approximate $\mathbf{fail_1}$ by $\mathbf{fail_1(1)}$.
- We obtain

$$\mathbf{fail_1} \approx \frac{\binom{v-1}{k-1}}{\binom{v}{k}} = \frac{k}{v}.$$

Global Connectivity of the Eschenauer-Gligor Scheme

- Eschener and Gligor appealed to random graph theory to determine parameters that would guarantee (with high probability) that the key-sharing graph is connected.
- They employed the Erdös-Rényi model, where a random graph $G(n, p)$ means that there are n vertices, and any pair of vertices is joined by an edge with probability p.
- Here, $p = \mathbf{Pr_1}$; for simplicity, the approximation $\mathbf{Pr_1} \approx k^2/v$ is often used.
- A fundamental result of Erdös and Rényi (1960) is that a random graph in $G(n, (1+\epsilon)\ln n/n)$ is "asymptotically almost surely" connected.
- This suggests that, when

$$\frac{k^2}{v} > \frac{\ln n}{n},$$

we would expect the key-sharing graph to be connected.

The Problem with this Approach

- The problem with this is approach is that every edge in $G(n,p)$ is chosen independently of every other edge.
- This independence property does not hold in key-sharing graphs, e.g., it is generally not the case that

$$\mathbf{Pr}[\mathbf{N_i} \sim \mathbf{N_j} \mid \mathbf{N_i} \sim \mathbf{N_k} \wedge \mathbf{N_j} \sim \mathbf{N_k}] = \mathbf{Pr}[\mathbf{N_i} \sim \mathbf{N_j}]. \quad (3)$$

- Suppose that $\mathbf{N_i} \cap \mathbf{N_k} \neq \emptyset$ and $\mathbf{N_j} \cap \mathbf{N_k} \neq \emptyset$.
- Let $x \in \mathbf{N_i} \cap \mathbf{N_k}$; then $x \in \mathbf{N_j} \cap \mathbf{N_k}$ with probability at least $1/k$.
- Therefore, $\mathbf{Pr}[\mathbf{N_i} \sim \mathbf{N_j} \mid \mathbf{N_i} \sim \mathbf{N_k} \wedge \mathbf{N_j} \sim \mathbf{N_k}] > 1/k$.
- When $v > k^3$, it holds that $1/k > k^2/v$ and hence (3) is violated.

Random Intersection Graphs

- It is better to model the key-sharing graph as a random intersection graph $G(b, v, k)$.
- The graph has b vertices corresponding to the nodes of a WSN in which each node is given a random k-subset of a set of v possible keys, and $\mathbf{N_i} \sim \mathbf{N_j}$ iff $\mathbf{N_i} \cap \mathbf{N_j} \neq \emptyset$.
- Sufficient conditions for a random graph in $G(b, v, k)$ to be asymptotically almost surely connected can be found in Blackburn and Guerke (2009); these conditions are very similar to the Erdös and Rényi conditions mentioned above.

Shared-key Discovery in the Eschenauer-Gligor Scheme

- Suppose two nearby nodes $\mathbf{N_i}$ and $\mathbf{N_j}$ wish to discover if they have at least one shared key.
- The method proposed in Eschenauer and Gligor (2002) is for the two nodes to broadcast their lists of key identifiers, say L_i and L_j, to each other.
- The broadcast has size $O(k)$.
- If these lists are pre-sorted, then it is possible for both nodes to determine all their shared keys in $O(k)$ time.

Shared-key Discovery in the Eschenauer-Gligor Scheme (cont.)

- An alternative approach is to use a PRNG to generate the key identifiers for each node from a seed stored in that node.
- Then a node $\mathbf{N_i}$ would only need to broadcast $\mathbf{seed}_i$ during shared-key discovery.
- Given $\mathbf{seed}_i$, node $\mathbf{N_j}$ would perform the following operations:
  1. using $\mathbf{seed}_i$, generate the list L_i,
  2. sort L_i (and L_j, if it is not already sorted), and
  3. search for common key identifiers in L_i and L_j.
- This approach takes time $O(n \log n)$, but the broadcast size is reduced to $O(1)$.

Deterministic Key Predistribution Schemes

- The Eschenauer-Gligor schemes are randomized schemes, in that the keys assigned to each node are chosen randomly.
- In 2004, deterministic KPS were proposed independently by Camtepe and Yener; by Lee and Stinson; and by Wei and Wu.
- In a deterministic scheme, the assignment of keys to nodes is done in a deterministic fashion.
- A suitable set system (i.e., a design) is chosen, and each block is assigned to a node in the WSN (the design and the correspondence of nodes to blocks is public).
- The points in a block are the indices (i.e., the identifiers) of the keys given to the corresponding node.

Combinatorial Set Systems (aka Designs)

- A set system is a pair $(X, \mathcal{A})$, where the elements of X are called points and $\mathcal{A}$ is a set of subsets of X, called blocks.
- As stated above, we pair up the blocks of the set system with the nodes in the WSN, and the points in the block are the key identifiers of the keys given to the corresponding node.
- 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; then each key occurs in r nodes in the WSN.
- If all blocks have size k, then $(X, \mathcal{A})$ is said to be uniform (of rank k); then each node is assigned k keys.

Configurations and BIBDs

- A (v, b, r, k)-configuration is a set system $(X, \mathcal{A})$ where $|X| = v$ and $|\mathcal{A}| = b$, that is uniform of rank k and regular of degree r, such that every pair of points occurs in at most one block.
- In a configuration, it holds that $vr = bk$.
- A (v, b, r, k, λ)-BIBD is a set system $(X, \mathcal{A})$ where $|X| = v$ and $|\mathcal{A}| = b$, that is uniform of rank k and regular of degree r, such that every pair of points occurs in exactly λ blocks.
- "BIBD" is an abbreviation for balanced incomplete block design.
- A BIBD with $\lambda = 1$ is a configuration.
- Examples of BIBDs with $\lambda = 1$ include finite projective planes, finite affine planes and Steiner triple systems.

Toy Example

We list the blocks in a $(7, 7, 3, 3)$-configuration (this is a projective plane of order 2, i.e., a $(7, 7, 3, 3, 1)$-BIBD) and the keys in a corresponding KPS:

node	block	key assignment
$\mathbf{N_1}$	$\{1, 2, 4\}$	k_1, k_2, k_4
$\mathbf{N_2}$	$\{2, 3, 5\}$	k_2, k_3, k_5
$\mathbf{N_3}$	$\{3, 4, 6\}$	k_3, k_4, k_6
$\mathbf{N_4}$	$\{4, 5, 7\}$	k_4, k_5, k_7
$\mathbf{N_5}$	$\{1, 5, 6\}$	k_1, k_5, k_6
$\mathbf{N_6}$	$\{2, 6, 7\}$	k_2, k_6, k_7
$\mathbf{N_7}$	$\{1, 3, 7\}$	k_1, k_3, k_7

The actual values of keys are secret, but the lists of key identifiers (i.e., the blocks) are public.

In this example, $\mathbf{Pr_1} = 1$ and $\mathbf{fail_1} = 1/5$.

Possible Advantages of Deterministic KPS

Deterministic KPS have several possible advantages:

Simpler set-up

No random number generator is required for key assignment; simple formulas dictate which keys are given to which nodes.

No need to verify expected properties of the WSN

Randomized KPS have desirable properties with high probability, but there are no guarantees, e.g., due to a "bad" choice of random numbers.

Simpler shared-key discovery and path-key establishment

The complexity of these algorithms can be significantly reduced, sometimes to $O(1)$ time, (as compared to $O(k)$ or $O(k \log k)$ time required in the randomized case).

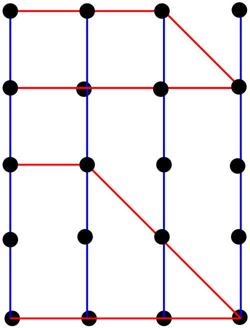
Properties of Configuration-based KPS

- Every block intersects $k(r-1)$ blocks in one point and is disjoint from all the other blocks.
- Therefore
$$\mathbf{Pr_1} = \frac{k(r-1)}{b-1}.$$
- A link L is defined by two blocks that intersect in one point, say x.
- There are $r-2$ other blocks that contain x; the corresponding nodes will compromise the link L.
- Therefore,
$$\mathbf{fail_1} = \frac{r-2}{b-2}.$$
- There is a tradeoff between $\mathbf{Pr_1}$ and $\mathbf{fail_1}$, which can be quantified by computing the ratio $\rho = \mathbf{Pr_1}/\mathbf{fail_1}$:
$$\rho = \frac{k(b-2)(r-1)}{(b-1)(r-2)} \approx k.$$

Transversal Designs

- Lee and Stinson (2005) proposed using transversal designs to construct KPS.
- Let n, k and t be positive integers.
- A transversal design TD(t, k, n) is a triple $(X, \mathcal{H}, \mathcal{A})$, where X is a finite set of cardinality kn, $\mathcal{H}$ is a partition of X into k parts (called groups) of size n, and $\mathcal{A}$ is a set of k-subsets of X (called blocks), which satisfy the following properties:
  1. $|H \cap A| = 1$ for every $H \in \mathcal{H}$ and every $A \in \mathcal{A}$, and
  2. every t elements of X from different groups occurs in exactly one block in $\mathcal{A}$.
- Transversal designs are equivalent to orthogonal arrays, which have been extensively studied in the setting of statistical design of experiments.

Some Blocks in a Transversal Design (Diagram)



Groups are represented as vertical blue lines, and blocks are represented as red lines. Each block is a transversal of the groups.

An Easy Construction for Transversal Designs

- Suppose that p is prime and $t \leq k \leq p$.
- Define
$$X = \{0, \ldots, k-1\} \times \mathbb{Z}_p.$$
- For every ordered t-subset $\mathbf{c} = (c_0, \ldots, c_{t-1}) \in (\mathbb{Z}_p)^t$, define a block
$$A_{\mathbf{c}} = \left\{ \left(x, \sum_{i=0}^{t-1} c_i x^i \right) : 0 \leq x \leq k-1 \right\}.$$
- Let
$$\mathcal{A} = \{A_{\mathbf{c}} : \mathbf{c} \in (\mathbb{Z}_p)^t\}.$$
- Then $(X, \mathcal{A})$ is a TD(t, k, p).
- The construction can be adapted to any finite field $\mathbb{F}_q$, where q is a prime power.
- These transversal designs are equivalent to Reed-Solomon codes.

Example

Suppose we take $p = 5$ and $k = 4$; then we construct a TD$(2, 4, 5)$:

$A_{0,0}$={00,10,20,30} $A_{0,1}$={01,11,21,31} $A_{0,2}$={02,12,22,32}
$A_{0,3}$={03,13,23,33} $A_{0,4}$={04,14,24,34} $A_{1,0}$={00,11,22,33}
$A_{1,1}$={01,12,23,34} $A_{1,2}$={02,13,24,30} $A_{1,3}$={03,14,20,31}
$A_{1,4}$={04,14,24,34} $A_{2,0}$={00,12,24,31} $A_{2,1}$={01,13,20,32}
$A_{2,2}$={02,14,21,33} $A_{2,3}$={03,10,22,34} $A_{2,4}$={04,11,23,30}
$A_{3,0}$={00,13,21,34} $A_{3,1}$={01,14,22,30} $A_{3,2}$={02,10,23,31}
$A_{3,3}$={03,11,24,32} $A_{3,4}$={04,12,20,33} $A_{4,0}$={00,14,23,32}
$A_{4,1}$={01,10,24,33} $A_{4,2}$={02,11,20,34} $A_{4,3}$={03,12,21,30}
$A_{4,4}$={04,13,22,31}

Some Properties of Transversal Designs

- A TD(t, k, n) has kn points and n^t blocks.
- Every block contains k points and every point occurs in n^{t-1} blocks.
- If $t = 2$, then the blocks of a TD(t, k, n) form a configuration.
- The KPS constructed from the "easy" TD$(2, k, p)$ are called linear KPS and the The KPS constructed from the "easy" TD$(3, k, p)$ are called quadratic KPS (Lee and Stinson (2005)).
- This is because the blocks are "defined" by linear (quadratic, resp.) equations.

Properties of the Linear KPS

- A $\mathsf{TD}(2, k, n)$ is an (nk, n^2, n, k)-configuration.
- Therefore

$$\mathbf{Pr_1} = \frac{k(n-1)}{n^2-1} = \frac{k}{n+1} \quad \text{and} \quad \mathbf{fail_1} = \frac{n-2}{n^2-2}.$$

- Since $v = nk$ in a $\mathsf{TD}(2, k, n)$, we have

$$\mathbf{Pr_1} \approx \frac{k}{n} = \frac{k^2}{v} \quad \text{and} \quad \mathbf{fail_1} \approx \frac{1}{n} = \frac{k}{v}.$$

- Recall that the Eschenauer-Gligor scheme has

$$\mathbf{Pr_1} \approx \frac{k^2}{v} \quad \text{and} \quad \mathbf{fail_1} \approx \frac{k}{v}$$

  when $v \gg k$.
- So the two schemes have very similar properties.

Evaluation of the Linear KPS

- **Benefit**: We can make $\mathbf{Pr_1}$ arbitrarily close to 1 by choosing k to be close to n.
- **Benefit**: Shared-key discovery is very efficient, due to the underlying algebraic structure of the linear TDs (see next slides).
- **Drawback**: The network size is n^2, which may not be large enough for "reasonable" values of n.
- **Drawback**: The ratio $\rho \approx k$ may be on the small side for many applications (however, this applies to any configuration-based KPS).

Shared-key Discovery for Linear Schemes

- An advantage of using deterministic KPS is that they may have a compact and efficient algebraic description
- This may yield efficient algorithms for shared-key discovery, in which very little information needs to be broadcasted.
- These advantages are exemplified by the linear schemes.
- Suppose we use a KPS based on the "easy" transversal design TD$(2, k, p)$ (p is a prime).
- In the resulting WSN, each node is identified by an ordered pair $(i, j) \in \mathbb{Z}_p \times \mathbb{Z}_p$.

Shared-key Discovery for Linear Schemes (cont.)

- It is sufficient for two nodes $\mathbf{N_{(i,j)}}$ and $\mathbf{N_{(i',j')}}$ to exchange their identifiers.
- Then they can each determine if they share a common key in $O(1)$ time, as follows:

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

Path-key Establishment

If two nearby nodes $\mathbf{N_{(i,j)}}$ and $\mathbf{N_{(i',j')}}$ 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 of their neighbourhoods.

Global Connectivity of Linear Key Predistribution Schemes

- Recall for E-G schemes that showing the connectivity of the key-sharing graph was a difficult task.
- In contrast, it is much easier to prove that the key-sharing graph of a linear scheme is (highly) connected.
- Wu and Stinson (2008) showed that the key-sharing graph of a KPS constructed from any TD$(2, k, n)$ is $k(n-1)$-connected
- That is, $k(n-1)$ nodes must be removed from the WSN in order to disconnect the network.
- This is the best possible result we could hope for, as every node is involved in exactly $k(n-1)$ links.

Local Connectivity of Linear Key Predistribution Schemes

- We can also say something about the local connectivity of these KPSs.
- Suppose A and B are two disjoint blocks in a TD$(2, k, n)$.
- It is easy to show that there are $k(k-1)$ blocks that intersect both A and B.
- Therefore there are $k(k-1)$ two-hop paths in the key-sharing graph joining any two non-adjacent nodes.

Properties of KPS from TDs with $t = 3$, $\eta = 2$

- We can base a KPS on a TD$(3, k, n)$ with $\eta = 1$ or 2.
- When $\eta = 2$, we have

$$\mathbf{Pr_1} = \frac{k(k-1)}{2(n^2+n+1)} \quad \text{and} \quad \mathbf{fail_1} = \frac{n-2}{n^3-2}.$$

- **Drawback:** The maximum value of $\mathbf{Pr_1}$ is about $1/2$.
- **Drawback:** Shared-key discovery is less efficient (but still reasonable).
- **Benefit:** The network size is n^3, which is quite large, even for “reasonable” values of n.
- **Benefit:** The ratio $\rho \approx k^2/2$ is now considerably larger.

Properties of KPS from TDs with $t = 3$, $\eta = 1$

- When $\eta = 1$, we have

$$\mathbf{Pr_1} = \frac{k(2n-k+3)}{2(n^2+n+1)}$$

and

$$\mathbf{fail_1} = \frac{2n^3 + (4-2k)n^2 + (k-5)n + 2k - 6}{(2n-k+3)(n^3-2)}.$$

- Drawback: The maximum value of $\mathbf{Pr_1}$ is (still) about $1/2$.
- Drawback: Shared-key discovery is less efficient (but still reasonable).
- Benefit: The network size is n^3, which is quite large, even for "reasonable" values of n.
- Benefit: The ratio ρ is more complex to analyse.

Flexibility of Parameters

- The network size for a TD-based KPS is n^2 when $t = 2$ and n^3 when $t = 3$.
- For the "easy" constructions, we want n to be a prime power.
- The traditional viewpoint with respect to combinatorial KPS is that if a specific network size m is desired, then it suffices to choose parameters to give a scheme for a network of size greater than m and simply discard excess nodes.
- Bose, Dey and Mukerjee (2013) disagree with this viewpoint, saying:

If we then discard the unnecessary node allocations to get the final scheme for use, this final scheme will not preserve the $\mathbf{Pr_1}$ *and* $\mathbf{fail_s}$ *values of the original scheme and hence the properties of the final scheme in this regard can become quite erratic.*

Flexible KPS from TDs with $t = 2$

- When n is a prime power, the "easy" $\mathsf{TD}(2, k, n)$ can be resolved into n parallel classes, each containing n blocks.
- Suppose we take ℓ of the n parallel classes.
- We obtain an $(nk, n\ell, \ell, k)$-configuration.
- Therefore

$$\mathbf{Pr_1} = \frac{k(\ell - 1)}{\ell n - 1} \quad \text{and} \quad \mathbf{fail_1} = \frac{\ell - 2}{\ell n - 2}.$$

- As long as ℓ is not very small, we have a KPS whose values of $\mathbf{Pr_1}$, $\mathbf{fail_1}$ and ρ are similar to what they were before; the value of k is unchanged.
- But we can now accommodate many possible network sizes for a given value of n: any multiple of n from $2n$ to n^2.

Flexible KPS from TDs with $t = 3$

- When n is a prime power, the "easy" TD$(3, k, n)$ can be resolved into n TD$(2, k, n)$'s, each containing n^2 blocks.
- Suppose we take ℓ of these n TD$(2, k, n)$'s.
- When $\eta = 2$, we have

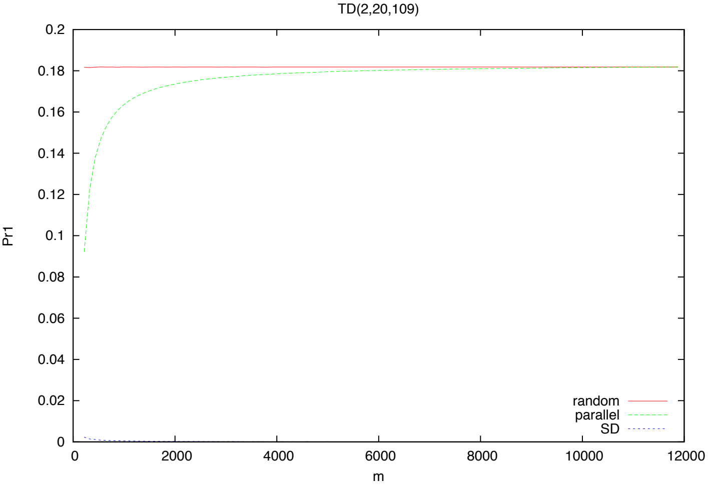
$$\mathbf{Pr_1} = \frac{k(k-1)(\ell-1)}{2(\ell n^2 - 1)} \quad \text{and} \quad \mathbf{fail_1} = \frac{\ell - 2}{\ell n^2 - 2}.$$

- Again, as long as ℓ is not very small, we have a KPS whose values of $\mathbf{Pr_1}$, $\mathbf{fail_1}$ and ρ are similar to what they were before; the value of k is unchanged.
- We can now accommodate many possible network sizes for a given value of n: any multiple of n^2 from $2n^2$ to n^3.

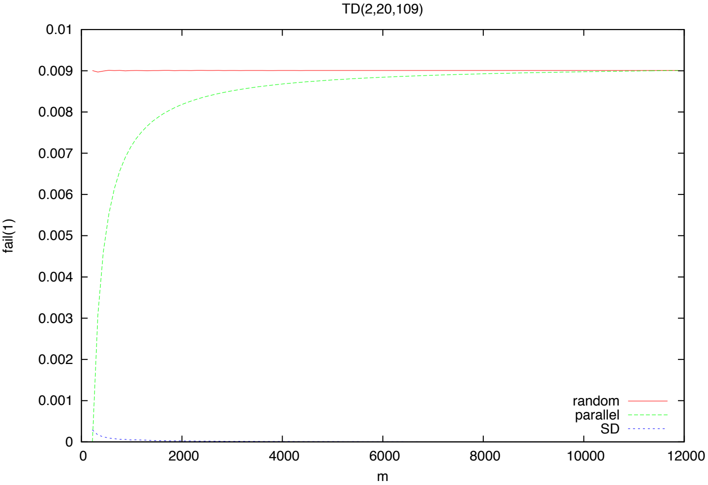
Random Deletion of Nodes from a KPS

- Suppose we randomly delete nodes from a combinatorial KPS.
- **Question:** How are the values of $\mathbf{Pr_1}$ and $\mathbf{fail_1}$ affected?
- **Answer:** Hardly at all. The concerns of Bose *et al.* seem to be unfounded.
- We did large numbers of experiments which showed unequivocally that the "random deletion" approach works very well in practice.
- There is some variation in the values of $\mathbf{Pr_1}$ and $\mathbf{fail_1}$, but the standard deviation is very small.

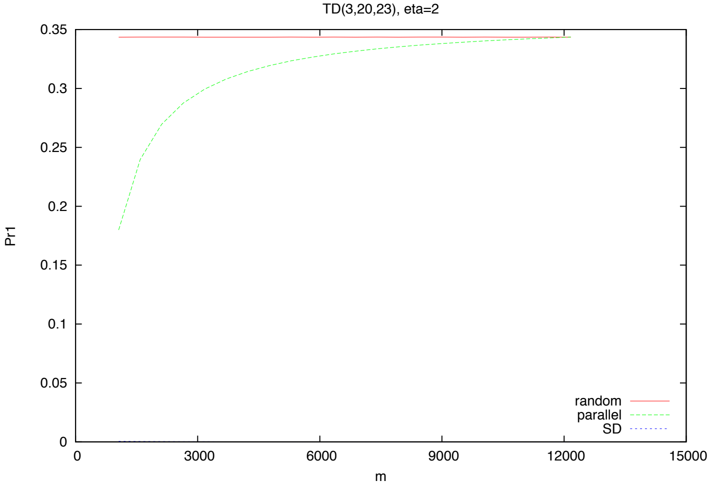
Example: Connectivity of KPS derived from $\mathbf{TD}(2, 20, 109)$



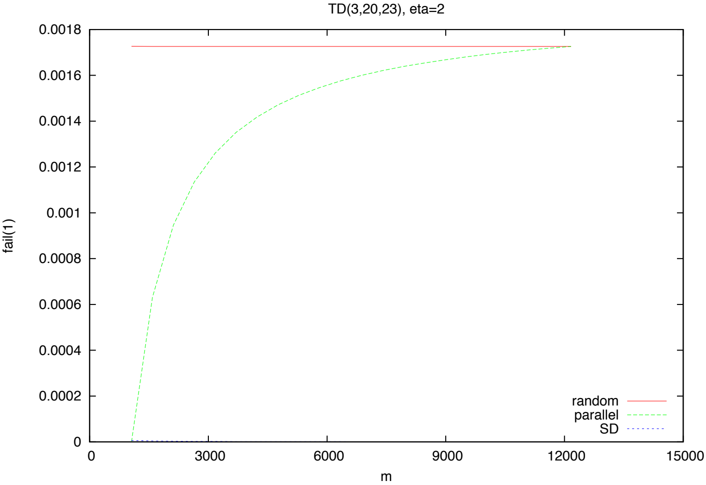
Example: Resilience of KPS derived from $\mathbf{TD}(2, 20, 109)$



Example: Connectivity of KPS derived from $\mathbf{TD}(3, 20, 23)$



Example: Resilience of KPS derived from TD$(3, 20, 23)$



Using Less Regular Set Systems

- We have been employing schemes based on combinatorial structures (TDs, especially).
- **Question:** Could there be any advantage in using less "regular" structures to construct KPS?
- Suppose we use a set system with block size k where the maximum intersection of two blocks equals 1.
- We do not require that every point occurs in the same number of blocks.
- So we are relaxing the requirements of a configuration.
- Suppose that point i occurs in r_i blocks, for $1 \leq i \leq v$.
- Then $\sum r_i = bk$.

Properties of the Resulting KPS

- We have

$$\mathbf{Pr_1} = \frac{\sum_{i=1}^{v} r_i(r_i - 1)}{b(b-1)}$$

and

$$\mathbf{fail_1} = \frac{\sum_{i=1}^{v} r_i(r_i - 1)(r_i - 2)}{(b-2)\sum_{i=1}^{v} r_i(r_i - 1)}.$$

- Therefore,

$$\rho = \frac{(b-2)\left(\sum_{i=1}^{v} r_i(r_i - 1)\right)^2}{b(b-1)\sum_{i=1}^{v} r_i(r_i - 1)(r_i - 2)}.$$

- **Conjecture (?)** Assuming that $\sum_{i=1}^{v} r_i = bk$ is fixed, the value of ρ is maximized when $r_1 = \cdots = r_v = bk/v$.

References

[1] M. Bose, A. Dey and R. Mukerjee. Key predistribution schemes for distributed sensor networks via block designs. *Designs, Codes and Cryptography* **67** (2013), 111–136.

[2] K. Henry, M. B. Paterson and D. R. Stinson. Practical approaches to varying network size in combinatorial key predistribution schemes. *SAC 2013 Proceedings*.

[3] J. Lee and D. R. Stinson. A combinatorial approach to key predistribution for distributed sensor networks. *IEEE Wireless Communications and Networking Conference (WCNC 2005)*, vol. 2, pp. 1200–1205.

[4] M. B. Paterson and D. R. Stinson. A Unified Approach to Combinatorial Key Predistribution Schemes for Sensor Networks. *Designs, Codes and Cryptography*, to appear.

thank you for your attention!