Deterministic Key Predistribution Schemes for Distributed Sensor Networks

Jooyoung Lee[1] and Douglas R. Stinson[2]

[1] Department of Combinatorics and Optimization
[2] School of Computer Science,
University of Waterloo, Waterloo, Ontario, Canada N2L 3G1
{j3lee, dstinson}@uwaterloo.ca

Abstract. It is an important issue to establish pairwise keys in distributed sensor networks (DSNs). In this paper, we present two key predistribution schemes (KPSs) for DSNs, ID-based one-way function scheme (IOS) and deterministic multiple space Blom's scheme (DMBS). Our schemes are deterministic, while most existing schemes are based on randomized approach. We show that the performance of our schemes is better than other existing schemes in terms of resiliency against coalition attack. In addition we obtain perfectly resilient KPSs such that the maximum supportable network size is larger than random pairwise keys schemes.

1 Introduction

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

The sensor nodes in DSNs should be able to communicate with each other in order to relay or accumulate secret information. There are three ways to establish pairwise keys between sensor nodes. First is to establish secret keys using a public key infrastructure. However, asymmetric cryptographic primitives are not suitable due to expensive computational cost as well as storage constraints in each node. In another strategy, a sensor node is chosen to be a *trusted authority* (TA), which all other nodes in the network are assumed to trust. The TA shares a long-lived key with every node and transmit session keys between sensor nodes on request. This method can result in expensive costs for message relay. Arbitrated

H. Handschuh and A. Hasan (Eds.): SAC 2004, LNCS 3357, pp. 294–307, 2005.

protocols are also vulnerable to a single compromise of the TA. Therefore it is natural that we are interested in key predistribution schemes (KPSs), where a set of secret keys is installed in each node before each sensor node is deployed. After being deployed, it sets up a secret key with every node in certain neighborhood using their common information.

There are two simple strategies for KPSs. One is to use a single secret key over the entire network. This scheme is obviously efficient in terms of the cost of computation and memory. However the compromise of only a single node exposes all communications over the entire network, which is a serious deficiency. The other approach is to use distinct keys for all possible pairs of nodes. Then every node is preloaded with $n-1$ keys, where n is the network size. This scheme guarantees perfect resiliency in that links between noncompromised nodes are secure against any coalition of compromised nodes. However this scheme is not suitable for large networks since the storage required per node increases linearly with the network size. In a classic paper by Blom [1], a tradeoff between key storage and security is presented. Given a security parameter $1 < t < n$, each node is deployed with $t+1$ keys. This scheme provides perfect security against any coalition of up to t compromised nodes, while the compromise of $t+1$ nodes would totally break the system. We briefly review this scheme in Sect. 5.

Recently, Eschenauer and Gliger [6] proposed a probabilistic key predistribution scheme. This scheme consists of three phases: *key predistribution*, *shared-key discovery*, and *path-key establishment*. We briefly describe these phases since our scheme also follow the same framework. In key predistribution phase, a large pool of keys and their key identifiers are generated. Every sensor node is equipped with a fixed number of keys randomly chosen from the key pool with their key identifiers. After deployment, the shared-key discovery phase takes place, where two nodes in a wireless communication range look for their common keys. If they share common keys in their key rings, they can pick one of them as their secret key. Sensor nodes can exchange the key identifiers of their keys, for example, to discover if they share a common key. The path-key establishment phase takes place in case there is no common key between a pair of nodes in a wireless communication range. They look for multiple secure links (hops) to reach each other so that one of them can choose an arbitrary key and relay it through the links. In our paper, we focus on the key predistribution phase which is given by a deterministic way.

The Eschenauer-Gliger scheme is generalized by Chan, Perrig and Song [3], where two nodes compute a pairwise key only if they share at least q common keys. They also presented a random-pairwise keys scheme, where a random graph is generated as the network layer and each link receives a unique key. In [5] and [9], the Eschenauer-Gligor schemes are combined with Blom's schemes, resulting in better performance compared with existing schemes.

DSNs can be regarded as superposition of a physical layer and a network layer. Due to resource constraints, a sensor node can communicate with only nodes within a limited radius. Hence the physical layer is represented by a *random geometric graph*. On the other hand, the network layer is represented by a graph

such that two nodes are adjacent if they share a secret key, which is called a *network graph*. The network graph is determined by the KPS, independent of the distribution of sensor nodes. The network graphs have been given by random graphs since Eschenauer and Gliger's work. In this paper, we propose to use strongly regular graphs as network graphs. This means that the assignment of keys is deterministic. We can reduce the storage per node without loss of resiliency by introducing public one-way functions in our KPSs. We describe this method in Sect. 3 and 4. In Sect. 5, we modify Blom's scheme by allowing asymmetric matrices when generating keys, which yields a tradeoff between the connectivity of the network and the resiliency. In a similar way as Du-Deng-Han-Varsheney/Liu-Ning schemes [5], we use the modified Blom's schemes on strongly regular graphs at a network layer. Our schemes show better resiliency than Du-Deng-Han-Varsheney/Liu-Ning schemes.

2 Preliminaries

In this section we present some basic terminologies and facts on combinatorial objects. These notions turn out to be useful to describe deterministic KPSs.

2.1 Set Systems and KPSs

We begin with the following definition.

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 the point x. $(X, \mathcal{A})$ is* regular *(of degree d) if all points have the same degree, d. The* rank *of $(X, \mathcal{A})$ is the size of the largest block. If all blocks have the same size, say r, then $(X, \mathcal{A})$ is said to be* uniform *(of rank r).*

Balanced incomplete block designs (BIBDs) are widely studied set systems. For extensive survey, we refer to [4] and [10].

Definition 2.2. *A (v, r, λ)-*BIBD *is a uniform set system $(X, \mathcal{A})$ of rank r with $|X| = v$ such that every pair of points in X occurs in exactly λ blocks.*

In the context of KPS, the set X corresponds to a key pool and each block to a sensor node. Thus a node is loaded with the keys in the corresponding block. If any two blocks have nonempty intersection, then they can establish their secret key. We can obtain Eschenauer-Gliger schemes choosing blocks of the same size randomly from a key pool. Each block is required to have size as small as possible in view of limited memory of a sensor node. In a KPS based on a regular set system of degree d, the exposure of one key in a node compromises d nodes. Hence we also wish the degree of each node to be as small as possible.

Example 2.1. An $(n^2 + n + 1, n + 1, 1)$-BIBD is called a *projective plane* of order n. A projective plane of order n exists for a prime power n. It is a symmetric BIBD, which means that the number of blocks is equal to the number of points. If

the network chooses a projective plane of order 32 for KPS, it can accommodate 1057 nodes. Each node has 33 keys loaded in it. This scheme is deterministic and needs no path-key establishment.

For a set system $(X, \mathcal{A})$, the network graph of the corresponding KPS is given by the *intersection graph* $(\mathcal{A}, E)$ of the set system, where two blocks are adjacent if they have nonempty intersection. In the above example, the intersection graph is a complete graph.

2.2 Strongly Regular Graphs and KPSs

Once a set system is defined, we can check the connectivity of the corresponding KPS through its intersection graph. On the other hand, we can first specify an intersection graph, and then construct a corresponding KPS as follows: Given a graph G on n nodes, we use $E(G)$ as a key pool. A set of keys

$$K(v) = \{e \in E : \ e \text{ is incident with } v\}$$

are predeployed in a node v. In this scheme, each node has a set of $\leq \Delta(G)$ keys, where $\Delta(G)$ is the maximal degree of G. No matter how many nodes are captured, any link between noncompromised nodes remains secure. When we take G as a random graph on n nodes, the KPS is reduced to the random pairwise keys scheme [3]. We want small degrees at the nodes and short paths between nonadjacent nodes of G. For this reason, we are interested in strongly regular graphs [4] (though they have stronger properties than we require).

Definition 2.3. *A* strongly regular graph *with parameters* (n, r, λ, μ) *is a graph on* n *vertices, without loops or multiple edges, regular of degree* r *(with* $0 < r < n - 1$*), and such that any two distinct vertices have* λ *common neighbors when they are adjacent, and* μ *common neighbors when they are nonadjacent.*

Any pair of nonadjacent nodes in a strongly regular graph are connected by μ paths of length two. There are various ways to construct strongly regular graphs using combinatorial objects. We define an orthogonal array, a latin square and mutual orthogonality [10] to describe a construction.

Definition 2.4. *An* orthogonal array $OA(t, n)$ *is an* $n^2 \times t$ *array* A *on an alphabet* X *of* n *symbols such that every ordered pair of symbols occur in every set of two columns of* A *exactly once.*

Definition 2.5. *A* latin square *of order* n *is an* $n \times n$ *array* L *on an alphabet* X *of* n *symbols such that every symbol occurs exactly once in each row and each column of* L*.*

Definition 2.6. *Let* L *and* M *be two latin squares of order* n *on alphabets* X *and* Y*, respectively.* L *and* M *are* orthogonal *if their superposition contains every ordered pair of symbols. A set of latin squares* $L_1, \ldots, L_s$*, all of the same order* n *are* mutually orthogonal *if* L_i *and* L_j *are orthogonal for all* $i \neq j$*.*

The *block graph* of a (t, n)-orthogonal array A is a graph with the rows of A as vertices, where two rows are adjacent if there exists a position in which they have the same symbol. Such a graph is an $(n^2, t(n-1), n+t^2-3t, t(t-1))$-strongly regular graph. The following results are well-known.

Theorem 2.1. *An $OA(t+2, n)$ exists if and only if t mutually orthogonal latin squares (MOLS) of order n exist, for positive integers n and t.*

Theorem 2.2. *Let $N(n)$ denote the largest number of MOLS of order n. Then $N(n) \leq n-1$, and if n is a prime power, then $N(n) = n-1$.*

We can construct $n-1$ MOLS of prime power order n from a projective plane of order n. The construction of a projective plane and the corresponding MOLS and orthogonal array is described in [10] in detail. To summarize, we have

Construction 2.3. *Let n be a prime power and let $3 \leq t \leq n+1$. Then we can construct an $(n^2, t(n-1), n+t^2-3t, t(t-1))$-strongly regular graph.*

Consider a KPS whose intersection graph is an (n, r, λ, μ)-strongly regular graph G. We assume that sensor nodes are distributed on a plane in a random way and the range where a node can reach physically forms a circle, as shown in Fig. 1. We call this circle a *neighborhood* of the sensor node. The probability that a node shares a common key with another node in a neighborhood is $p = r/(n-1)$. Let d denote the average number of nodes in a neighborhood and d' the number of nodes in the common neighborhood of two nodes u and v within a wireless communication range. The probability that u and v are connected within two hops is given by

$$\begin{aligned} p^2(u,v) &= p + (1-p)\left(1 - \binom{n-\mu-2}{d'-2} \Big/ \binom{n-2}{d'-2}\right) \\ &\approx p + (1-p)\left(1 - \left(1 - \frac{\mu}{n}\right)^{d'-2}\right) \\ &\geq p + (1-p)\left(1 - \left(1 - \frac{\mu}{n}\right)^{\frac{d}{3}}\right). \end{aligned}$$

The last inequality follows from the fact that two circles of the same radius has the intersection whose area is at least $1/3$ the area of the circle if the distance between the centres is less than the radius.

Example 2.2. Suppose that 1000 nodes are to be distributed and each neighborhood contains about $d = 40$ nodes. By taking $n = 32$ and $t = 14$ in Construction 2.3, we obtain a $(1024, 434, 186, 182)$-strongly regular graph G. In the corresponding KPS, we have $p^2(u, v) \geq 0.9547$ for any two nodes u and v within a wireless communication range.

Example 2.3. Consider a *complete bipartite graph* $K_{n,n}$. It is a $(2n, n, 0, n)$-strongly regular graph. In the corresponding KPS, we have

$$p^2_{K_{n,n}}(u,v) \approx 0.5 + 0.5(1 - (0.5)^{d'-2}) = 1 - (0.5)^{d'-1}$$

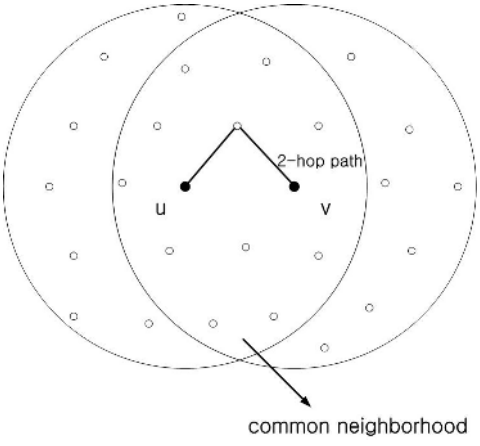

Fig. 1. A 2-hop path between two sensor nodes u and v

for any two nodes u and v within a wireless communication range. If we choose a random graph $G_{2n,p}$ with $p = 0.5$ as a network graph, the network layer has the same local connectivity. However, we have

$$p^2_{G_{2n,p}}(u,v) = 0.5 + (1-0.5)(1-(1-(0.5)^2)^{d'-2}) = 1-(0.5)(0.75)^{d'-2},$$

which is smaller than $p^2_{K_{n,n}}(u,v)$. Hence the complete bipartite graph $K_{n,n}$ performs better than a random graph.

3 Basic ID-Based One-Way Function Schemes

In this section we use a public one-way hash function h in order to reduce the number of keys stored in a node. The KPSs presented here are ID-based since a unique ID is assigned to each sensor node and the ID is used to compute secret keys. First we determine a network graph G and construct a key pool $\mathcal{K} = \{K_v : v \in G\}$. Next we decompose the edges of graph G into star-like subgraphs. A sensor node u receives a secret key K_u and 'hashed' keys $h(K_v \parallel ID(u))$ if it is contained in a star-like subgraph centred at v. Since a node v can compute $h(K_v \parallel ID(u))$ by evaluating the public one-way function h at the concatenation of its unique key K_v and public ID, $ID(u)$, both of u and v can establish their secret key $h(K_v \parallel ID(u))$. In case v is a leaf of a star-like subgraph centred at u, $h(K_u \parallel ID(v))$ is established as their secret key.

Now we consider an edge decomposition of a regular graph into star-like subgraphs. We begin with the following definition.

Definition 3.1. *An* Euler circuit *of G is a circuit in a graph G containing all the edges.*

Theorem 3.1. *A nontrivial connected graph has an Euler circuit if and only if each vertex has even degree.*

There is an algorithm to find Euler circuits in $O(|E|)$-time [7].

Theorem 3.2. *A connected regular graph G of order n and even degree r has an edge decomposition into star-like subgraphs such that each vertex is a centre of one star and a leaf of $r/2$ distinct stars.*

Proof. By using an Euler circuit, we will find an edge colouring of G such that the edges with the same colour form a star-like subgraph.

Note that $|E(G)| = nr/2$. Let $v_0E_0v_1E_1 \cdots v_{\frac{nr}{2}-2}E_{\frac{nr}{2}-2}v_{\frac{nr}{2}-1}(= v_0)$ be an Euler circuit of G. We use a set of colours labeled by vertices in G. Now we colour each edge E_i with colour v_i. Then the edges coloured by v is the $r/2$ edges coming from the vertex v in the Euler circuit, which form a star-like subgraph centred at v. Thus this colouring yields an edge decomposition of G into star-like subgraphs such that each vertex is a centre of one star and a leaf of $r/2$ distinct stars. □

Each node v stores one secret key K_v and $r/2$ hashed keys for the nodes u such that v is contained in a star-like subgraph centred at u. Therefore the total number of keys stored in a sensor node is given by $r/2 + 1$. This scheme reduces the number of keys per node by almost 50% as compared with the method discussed in the previous section.

Security Analysis. When a node u is revealed to an adversary, he obtains K_u as well as $h(K_{v_i} \parallel ID(u))$ for $r/2$ adjacent nodes v_i. It is infeasible to compute K_{v_i} even though he knows the key $h(K_{v_i} \parallel ID(u))$ since h is a one-way function. It follows that an adversary cannot compromise any link between two noncompromised nodes. Under the restriction of perfect resiliency, random pairwise keys schemes [3] exhibited the highest performance in terms of maximum supportable network size. However the basic ID-based one-way function schemes (IOSs) with regular network graphs (of even degree r) have maximum supportable network size two times larger than the random pairwise keys scheme, for a fixed probability p of sharing a common key, as shown in Fig. 2. Assuming each node contains k secret keys, the maximum supportable network size n is estimated as

$$n = \frac{2(k-1)}{p} + 1 \approx \frac{2k}{p},$$

since $p = r/(n-1)$ and $k = r/2 + 1$. In random pairwise keys schemes, the maximum supportable network size is given by $n = k/p$.

4 Multiple ID-Based One-Way Function Schemes

Basic IOSs are not suitable for a network of large size since they can accommodate only $O(k)$ sensor nodes for the node storage of k keys. In this section, we

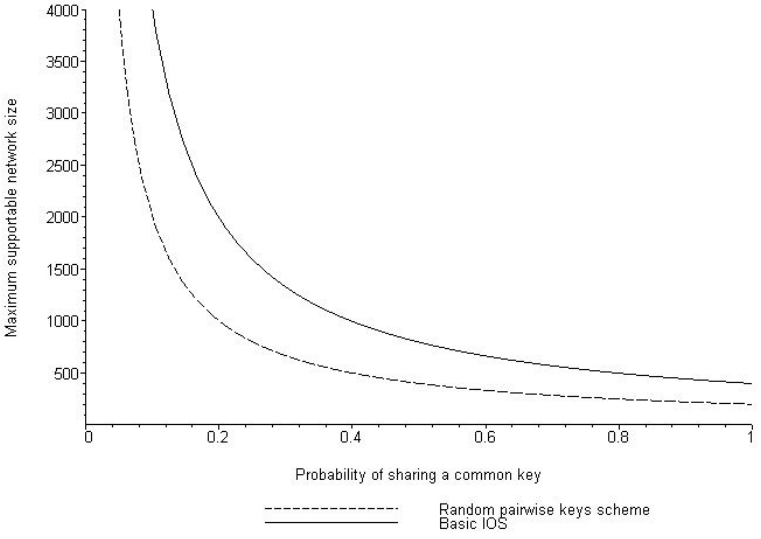

Fig. 2. The relationship between the probability of sharing a common key and the maximum supported network size for perfect resilience against node compromise. Each node is assumed to have $k = 200$ keys

use multiple copies of a single basic IOS to increase the network size relative to available memory. In exchange, the resiliency of multiple IOSs is weakened. In order to deploy $n = ml$ sensor nodes, we first determine an (m, r, λ, μ)-strongly regular graph G which is decomposed by star-like subgraphs. Each vertex u of G corresponds to l sensor nodes, say $u_1, \ldots, u_l$, of the network. We say the sensor nodes $u_1, \ldots, u_l$ are contained in a class u. Every node in a class u receives a common key K_u. If a vertex u is contained in a star-like subgraph centred at a vertex v in G, each sensor node u_i in a class u receives $h(K_v \parallel ID(u_i))$. Since any node v_j in a class v can compute $h(K_v \parallel ID(u_i))$, two nodes u_i and v_j can establish their session key using this hashed key. We assume that the duplicates $u_1, \ldots, u_l$ share no common key (even though we can set up an arbitrary key among them). The number of keys stored in a node is $k = r/2 + 1$, which is $1/l$ times smaller than using a single graph with the same probability of sharing a common key. The probability that two nodes share a common key is given by $p = rl/(n-1) \approx r/m$.

Security Analysis. Consider a DSN of size $n = ml$ which adopts an l-multiple IOS based on an (m, r, λ, μ)-strongly regular graph G. Suppose that an adversary compromises s nodes randomly in the network. We compute the probability that an arbitrary link $u_i v_j$ $(u \neq v)$ between two noncompromised nodes is compromised. It also estimates the fraction of compromised links between noncompromised nodes in the total network. Let u and v be the vertices (classes) of G containing u_i and v_j, respectively, such that v is a leaf of a star-like subgraph

centred at u. Then $h(K_u \parallel ID(v_j))$ is established as a secret key between two nodes u_i and v_j. In order to compute $h(K_u \parallel ID(v_j))$ without capturing u_i or v_j, the coalition have to contain at least one node in class u different from the node u_i. Therefore the probability is estimated as

$$P(s) = 1 - \frac{\binom{n-l-1}{s}}{\binom{n-2}{s}} \approx 1 - \left(1 - \frac{l-1}{n-2}\right)^s \approx 1 - \left(1 - \frac{p}{2k}\right)^s. \quad (1)$$

Figure 3 shows the performance of a multiple IOS compared with other existing schemes.

Example 4.1. Let $G = K_{\frac{m}{2},\frac{m}{2}}$ be a complete bipartite graph, where $4|m$. It is an $(m, m/2, 0, m/2)$-strongly regular graph. Using l copies of the graph G, we can accommodate lm sensor nodes. The number of keys per node is $m/4+1$. If a node u_i in a class u is compromised, then $ml(l-1)/4$ links between noncompromised nodes are compromised. These are the links between the other $l-1$ duplicates in the class u and the nodes whose class is a leaf of a star-like subgraph centred at u in G. Note that we do not consider physical constraints in this analysis.

Key Revocation. If a node u_i is detected as being compromised, a controller node (which has a large communication range and may be mobile) broadcasts $ID(u_i)$ so that secure nodes can stop communicating with u_i. Nevertheless the other duplicates u_j, $i \neq j$ can still use the links established by the keys of the form $h(K_v \parallel ID(u_j))$.

In order to replace the captured node u_i, a new node u_{new} is installed with a new key $K_{u_{new}}$ and $h(K_v \parallel ID(u_{new}))$ for $r/2$ node classes v, where the node classes v are randomly chosen among secure classes. Alternatively, we can choose the same classes v as the hashed keys of the revoked node u_i has. Now the controller node broadcasts $ID(u_{new})$ so that every node v_i can compute $h(K_{v_i} \parallel ID(u_{new}))$. After deployment, the node u_{new} can communicate with a (physical) neighbor v_j of a class v for which u_{new} has $h(K_v \parallel ID(u_{new}))$. Shared-key discovery and path-key establishment phase should be restarted.

5 Deterministic Multiple Space Blom's Schemes

In this section, we briefly describe Blom's KPSs and present modified schemes for DSNs. First we consider original Blom's KPSs which are secure against up to coalition of size t. Let n be the size of a network and q a prime power large enough to assume that keys of $\ln q$ bits in length are secure. In order to accommodate n sensor nodes, the TA constructs a public $(t+1) \times n$ matrix M over $GF(q)$ such that any $t+1$ columns of M are linearly independent. A well-known example of such a matrix M is a Vandermonde matrix

$$M = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ 1 & 2^2 & 3^2 & \dots & n^2 \\ & & \vdots & & \\ 1 & 2^t & 3^t & \dots & n^t \end{pmatrix}.$$

Each node u receives a unique $(t+1) \times 1$ column vector x_u from the matrix M, which is public. Using a Vandermonde matrix, a node given the j-th column vector can store only a seed $j \in GF(q)$ to generate the other elements [5]. In the next step, the TA generates a secret random $(t+1) \times (t+1)$ symmetric matrix D over $GF(q)$ and assigns secret information $K_u = x_u^T D$ to each node u. Any two nodes u and v can compute their secret key $K_{uv} = x_u^T D x_v$ from one's secret information and the other's public column vector. Note that $x_u^T D x_v = x_v^T D x_u$ due to the symmetry of the matrix D.

5.1 Modified Blom's KPSs on Complete Bipartite Graphs

As described above, any pair of nodes can establish a secret key in Blom's schemes. Thus the network layer is represented by a complete graph. We can weaken the connectivity of the network graph in order to improve resiliency against node compromise. We choose a complete bipartite graph K_{m_1,m_2} instead of a complete graph as a network graph in this modification. We divide the set of nodes into two sets U and V such that $|U| = m_1$ and $|V| = m_2$. The initial assignment of public column vectors is the same as the original schemes. The difference is that the TA generates a random $(t+1) \times (t+1)$ matrix D, which is *not necessarily symmetric*. Secret information $x_u^T D$ is assigned to each node $u \in U$ and Dx_v is assigned to each node $v \in V$, given their public column vectors x_u and x_v. Now both of the nodes x_u and x_v can compute their secret key $x_u^T D x_v$. The following theorem supports the stronger resiliency of this modification.

Theorem 5.1. *Let $U = \{u_1, \ldots, u_{m_1}\}$ and $V = \{v_1, \ldots, v_{m_2}\}$ be sets of $(t+1) \times 1$ column vectors over $GF(q)$ such that any $t+1$ vectors, either all in U or all in V, are linearly independent. Let D be a $(t+1) \times (t+1)$ matrix. Then*

1. *D is determined by $t+1$ row vectors $u_{l_i}^T D$, $i = 1, \ldots, t+1$ or $t+1$ column vectors Dv_{l_i}, $i = 1, \ldots, t+1$, and*
2. *for any $t+1$ vectors $u_{l_i} \in U$, $(i = 1, \ldots, t+1)$, and for any $t+1$ vectors $v_{l_i} \in V$, $(i = 1, \ldots, t+1)$, and for any scalar $k \in GF(q)$, there exists a $(t+1) \times (t+1)$ matrix D' such that*

$$u_{l_i}^T D' = u_{l_i}^T D, \text{ and } D' v_{l_i} = D v_{l_i} \ (i = 1, \ldots, t), \text{ and } u_{l_{t+1}}^T D' v_{l_{t+1}} = k.$$

Proof. Let

$$U = \begin{pmatrix} u_{l_1}^T \\ u_{l_2}^T \\ \vdots \\ u_{l_{t+1}}^T \end{pmatrix} \text{ and } V = \begin{pmatrix} v_{l_1} & v_{l_2} & \ldots & v_{l_{t+1}} \end{pmatrix}.$$

Then U and V are invertible $(t+1) \times (t+1)$ matrices over $GF(q)$. Given UD or DV, we can compute D by multiplying by the inverse matrix U^{-1} or V^{-1}, which proves the first part of the theorem.

Now we define

$$(\hat{D})_{i,j} = \begin{cases} k, & \text{if } i = j = t+1, \\ (UDV)_{i,j}, & \text{otherwise.} \end{cases}$$

and

$$D' = U^{-1}\hat{D}V^{-1}.$$

Then we have

$$e_i^T UD'V = e_i^T \hat{D} = e_i^T UDV$$

and

$$UD'Ve_i = \hat{D}e_i = UDVe_i,$$

for every elementary vector e_i (with a "1" in position i and "0"s in all other positions), $i = 1, \ldots, t$. Since $e_i^T UD' = e_i^T UD$ and $D'Ve_i = DVe_i$ for $i = 1, \ldots, t$, it follows that

$$u_{l_i}^T D' = e_i^T UD' = e_i^T UD = u_{l_i}^T D,$$

and

$$D'v_{l_i} = D'Ve_i = DVe_i = Dv_{l_i}$$

for $i = 1, \ldots, t$, and

$$u_{l_{t+1}}^T D'v_{l_{t+1}} = e_{t+1}^T UD'Ve_{t+1} = e_{t+1}^T \hat{D}e_{t+1} = k,$$

as desired. □

Theorem 5.1 means that an adversary cannot obtain any information on the keys of the links between noncompromised nodes unless it compromise at least $t+1$ nodes, either all in U or in V. In the original Blom's scheme with the same threshold parameter t, the compromise of any $t+1$ keys breaks the total system. However, in our modification, the probability of a total break is reduced to

$$P(t+1) = \frac{\binom{m_1}{t+1} + \binom{m_2}{t+1}}{\binom{m_1+m_2}{t+1}}.$$

In general, when s nodes are captured randomly, the probability $P(s)$ of total break is estimated as

$$P(s) = 1 - \frac{\sum_{\substack{i+j=s \\ 0 \le i,j \le t}} \binom{m_1}{i}\binom{m_2}{j}}{\binom{m_1+m_2}{s}}. \tag{2}$$

We will use this modification as building blocks to construct new KPSs in the next section.

5.2 Deterministic Multiple Space Blom's Schemes (DMBSs)

We consider l copies of an (m, r, λ, μ)-strongly regular graph G to accommodate $n = ml$ nodes. We regard each vertex of G as a class of l nodes. Every sensor node u_i receives its public column vector x_{u_i} from a Vandermonde matrix M and every edge e of G is associated with a random $(t+1) \times (t+1)$ matrix D_e, not necessarily symmetric.

Now an arbitrary direction is assigned to every edge of G. For every edge $e \in E(G)$ incident to a vertex (class) u, each node u_i of class u receives row vector $x_{u_i}^T D_e$ if e starts from u, or column vector $D_e x_{u_i}$ if e ends at u. Suppose that an edge $uv \in E(G)$ begins at u. Then two sensor nodes $u_i \in u$ and $v_j \in v$ can compute their secret key $K_{u_i v_j} = x_{u_i}^T D_{uv} x_{v_j}$ using each other's public vector. Since each vector has size equivalent to $t+1$ keys, the total amount of information per node is given by $r(t+1)$. The probability that two nodes share a common key is $p = rl/(n-1) \approx r/m$.

Security Analysis. Suppose that s nodes are captured by an adversary in a random way. Consider a link between two noncompromised nodes u_i and v_j, contained in classes u and v, respectively. In order to compute their secret key $K_{u_i v_j} = x_{u_i}^T D_{uv} x_{v_j}$, the coalition has to contain at least $t+1$ nodes, either all in the class of u or the class of v. Therefore the probability $P(s)$ that the link is compromised is estimated as

$$P(s) = 1 - \frac{\sum_{i=0}^{t} \sum_{j=0}^{t} \binom{l-1}{i} \binom{l-1}{j} \binom{n-2l}{s-i-j}}{\binom{n-2}{s}}. \tag{3}$$

Figure 3 illustrates the graph of $P(s)$ as a function of the number of compromised nodes for various schemes. In this plot, we assume that

1. the total network size is $n = 1200$,
2. each node has 200 pieces of secret information,
3. the probability of sharing a common key between two nodes is $p = 0.5$.

We briefly describe the graphs and parameters used in this plot as follows:

(a) is from Fig. 2 in [9], where we take $s' = 2$, $s = 7$, and $t = 99$.
(b) shows the resiliency of a modified Blom's scheme with threshold parameter $t = 199$ and network graph $K_{600,600}$. We use (2) in Sect. 5.1.
(c) shows the resiliency of a deterministic multiple space Blom's scheme based on 300 copies of a $(4, 2, 0, 2)$-strongly regular graph, where we take threshold parameter $t = 99$. We use (3) in Sect. 5.2.
(d) shows the resiliency of a basic scheme such that 200 keys are chosen from a pool of size 58000 [6].
(e) shows the resiliency of a q-composite scheme with $q = 1$ [3].
(f) is given by (1) in Sect. 4.

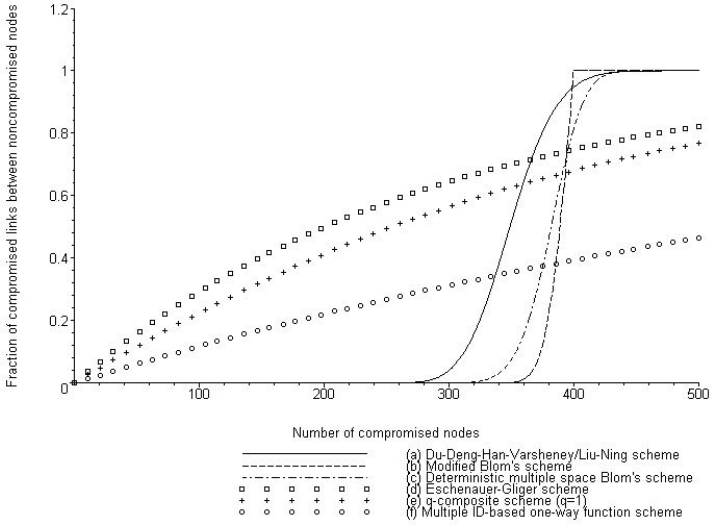

Fig. 3. Fraction of compromised links between noncompromised nodes v.s. number of compromised nodes

6 Conclusion

We presented two KPSs for distributed sensor networks in this paper. We can determine network graphs in both schemes. ID-based one-way function schemes allow each node to reduce the storage by using one-way functions in generating secret keys. Using a single copy of a network graph, we obtain a KPS with perfect resiliency. A basic IOS has the maximum supportable network size larger than a random pairwise keys scheme [3]. A multiple IOS provides a trade-off between node storage (or total network size) and resiliency against coalition attack. MBSs are based on modified Blom's schemes and Du-Deng-Han-Varsheney/Liu-Ning schemes. MBSs show stronger resiliency than Du-Deng-Han-Varsheney/Liu-Ning schemes.

References

1. R. Blom. An Optimal Class of Symmetric Key Generation Systems, In *Advances in Cryptology - Eurocrypt '84*, pages 335-338, 1985. Lecture Notes in Computer Science Volume 209.
2. 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.
3. H. Chan, A. Perrig, and D. Song. Random Key Predistribution Schemes for Sensor Networks, In *IEEE Symposium on Research in Security and Privacy*, pages 197-213, May 2003.

4. C.J. Colbourn, J.H. Dinitz (editors). *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, 1996.
5. W. Du, J. Deng, Y.S. Han, and P.K. Varsheney. A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks, In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS)*, October 2003.
6. L. Eschenauer and V.D. Gligor. A Key-Management Scheme for Distributed Sensor Networks, In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 41-47, November 2002.
7. A. Gibbons. *Algorithmic Graph Theory*, Cambridge Univ. Press, Cambridge, 1985.
8. R.L. Graham, M. Grötschel and L. Lovász (editors). *Handbook of Combinatorics*, vol. 2, North-Holland, Amsterdam, 1995.
9. D. Liu and P. Ning, Establishing Pairwise Keys in Distributed Sensor Networks, In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS)*, October 2003.
10. D.R. Stinson, *Combinatorial Designs: Constructions and Analysis*, Springer-Verlag, New York, 2003.