A Hybrid Key Predistribution Scheme for Sensor Networks Employing Spatial Retreats to Cope with Jamming Attacks

Korporn Panyim and Prashant Krishnamurthy

Graduate Program in Telecommunications and Networking
University of Pittsburgh
{kop1,prashk}@pitt.edu

Abstract. In order to provide security services in wireless sensor networks, a well-known task is to provide cryptographic keys to sensor nodes prior to deployment. It is difficult to assign secret keys for all pairs of sensor node when the number of nodes is large due to the large numbers of keys required and limited memory resources of sensor nodes. One possible solution is to randomly assign a few keys to sensor nodes and have nodes be able to connect to each other with some probability. This scheme has limitations in terms of the tradeoffs between connectivity and memory requirements. Recently, sensor deployment knowledge has been used to improve the level of connectivity while using lesser amounts of memory space. Jamming attacks are an easy and efficient means for disruption of the connectivity of sensors and thus the operation of a sensor network. One solution for mobile sensor nodes to overcome the impact of jamming is to perform spatial retreat by moving nodes away from jammed regions. However, deployment based key predistribution schemes may cause a large number of nodes to be cryptographically isolated after they move out of the jammed area. Moved nodes may not be able to reconnect to the network because they do not have any shared secret with new neighbors at new locations. In this paper, we propose a hybrid key predistribution scheme that supports spatial retreat strategies to cope with jamming attacks. Our scheme combines the properties of random and deployment knowledge based key predistribution schemes. In the presence of jamming attacks, our scheme provides high key connectivity (similar to deployment knowledge based schemes) while reducing the number of isolated nodes. We evaluate the performance of our scheme through simulations and analysis.

1 Introduction

Sensor networks applications have been constantly diversifying to include environmental sensing, object detection, structural health monitoring, patient health monitoring, and goods tracking. In many of these scenarios it is important to preserve confidential the data exchanged by sensors. For these purposes, sensor nodes must share cryptographic keys (typically secret (symmetric) keys because

E. Bertino and J.B.D. Joshi (Eds.): CollaborateCom 2008, LNICST 10, pp. 715–731, 2009.

public-key schemes are computationally expensive for sensors). There are several challenges here. At one end of the spectrum, assigning a single master key to every node results in a lack of resilience to node compromise. A single node, if compromised, can enable communications of all pairs of nodes to be compromised. It is difficult to assign and manage *pairwise* secret keys for all pairs of sensor nodes when the number of nodes is large due to the large numbers of keys and limited memory resources of sensor nodes (the number of keys stored is $n - 1$ for a group of n nodes). Pairwise keys also limit deployment of additional sensors. One possible solution is to randomly predistribute a subset of keys from a big pool of keys to sensor nodes and have nodes be able to securely connect to each other with some probability [5]. In this approach sensors in communicating range can securely connect only if they share at least one key from the randomly pre-distributed set they each carry. This probability (a related measure of which is called local connectivity) depends on the key pool size and the number of keys stored in each sensor. Recently, sensor deployment knowledge has been used to improve local connectivity while using a smaller memory space [4] by partitioning the pool of keys such that nodes that are deployed together spatially are more likely to share keys as against nodes that are far away from each other.

Jamming attacks form efficient means for disruption of the connectivity of sensors and thus the operation of a sensor network. One solution for mobile sensor nodes to overcome the impact of jamming is to perform *spatial retreats*[6,11] by moving nodes away from jammed regions. With spatial retreats and deployment based key predistribution a large number of sensor nodes can be isolated from the rest of the network after they move out of the jammed area. This is because moved nodes may not be able to find share secret keys with new neighbors at new locations. The random key predistribution scheme [5] is not affected by movement of nodes, but it has a lower *a priori* connectivity than the one that employs deployment knowledge given the same number of keys stored in sensor nodes. Similar problems of isolation can be anticipated with other techniques to combat jamming (e.g., increasing transmit power to reach nodes that are beyond the jammed region) although we do not consider them in this paper.

In this paper, we propose a *hybrid key predistribution scheme* that supports spatial retreat strategies to cope with jamming attacks. This scheme combines the properties of random and deployment knowledge based key predistribution schemes. In the presence of jamming attacks, the scheme provides high local connectivity (similar to deployment knowledge based schemes) while reducing the number of isolated nodes (like the random scheme) due to node's movement. We evaluate the performance of our scheme through simulations and analysis. We organize our paper as follows: Section 1.2 provides the background of key predistribution schemes for sensor networks and jamming models for attacks; Section 1.3 describes the impact of jamming on key connectivity of sensor nodes that adopt a spatial retreat strategy. We introduce the hybrid key predistribution scheme in Section 1.4; Section 1.5 presents an evaluation of the hybrid key predistribution scheme using simulations; Section 1.6 provides some discussions and limitations of the work; and finally Section 1.7 presents the conclusions.

2 Background

In this section, we present some basic background of key predistribution schemes for wireless sensor networks and an overview of jamming attacks. This section is necessary for understanding the hybrid scheme and its performance.

2.1 Key Predistribution for Sensor Networks

Unique characteristics of wireless sensor networks introduce challenges in providing security services. A sensor has limited size of memory but the number of sensor nodes involved in one application can be large (1,000 to 10,000 nodes). A possible approach for providing security services in wireless sensor networks is to install cryptographic keys in sensor nodes prior to deployment. If a single master key is installed in all sensors (which will then be used to bootstrap secure communications), a single node compromise can impact the entire network. When the number of sensors is large, installing pairwise keys (where each pair of nodes has a unique shared secret key) becomes unmanageable. Each node has to keep $n - 1$ keys in an n-node network and rekeying if nodes are added becomes a problem. Finally, since sensors typically communicate locally with direct neighbors, it may not be necessary to install pairwise keys between all pairs of sensors. However it is hard to determine which sensors will be eventual neighbors after deployment.

To overcome the above challenges, Eschenauer and Gligor proposed a random key predistribution scheme (EG scheme) [5]. The EG scheme (also called "basic" random key predistribution) relies on probabilistic key sharing among nodes in a random graph. The EG scheme consists of three phases: key distribution phase, shared-key discovery phase, and path-key establishment phase. In the key distribution phase, an off-line key distribution center generates a key pool consisting of large number of keys. Each node randomly picks k keys from this global key pool S of size $|S|$ and stores them in its memory. Each key is associated with a key identification (key-ID). The set of keys drawn from the key pool with associated key-IDs is called a key ring. In the shared-key discovery phase, each node exchanges, with its neighbor, information used to establish a shared key. The goal of this phase is to find a common key between two neighboring nodes. The common key(s) can be used to establish a secure link between two nodes by encrypting all messages with their shared key (or performing local key establishment using these keys). A secure link exists between two nodes if they share a key and can communicate directly. The simplest way to do this is to have each node broadcast, in clear text, its list of key IDs in the key ring. To add security to exchanged information, a challenge-response protocol can be used to hide key sharing patterns among nodes from an adversary [5]. However, since keys in node's key ring are randomly drawn from the key pool, it is possible that a pair of nodes may not have any common key. The path-key establishment phase allows a pair of nodes that do not have common key to establish a secure path through two or more links. The graph of sensor nodes is connected (securely) if each sensor node has enough neighbors even though k is small compared to $|S|$.

Typically, k is on the order of a hundred while $|S|$ is on the order of several tens or hundreds of thousands. From [5], the probability that any two sensor nodes share a key given $|S|$ and k is:

$$1 - \frac{((|S| - k)!)^2}{(|S| - 2k)!|S|!} \tag{1}$$

The above equation considers the number of possible sets of size k drawn without repetition from a set of size $|S|$ that have no overlap to compute the probability that two nodes do not share a key and subtracts this from 1 to determine the probability that two nodes do share at least one key. We will refer to the fact that two nodes within transmission range share at least one key as constituting "secure connectivity" in this paper.

The use of *deployment knowledge* is proposed as an improvement to the EG scheme. The deployment knowledge based key predistribution scheme (we shall call it *EGD scheme* throughout this paper), proposed by Du, et al [4], is based on the idea that the way that sensor nodes are deployed can be use to improve secure connectivity. One practical way to deploy sensor nodes is to divide sensors into small deployment groups or clusters. These groups are deployed sequentially so that the sensors in groups that are next to each other have a better chance to be within each other's radio transmission range. Knowing which pair of nodes is "likely" to comprise of neighbors is valuable in assigning keys from the key pool. The clustered deployment of sensor nodes is modeled by using probability density functions. In EG scheme, nodes are deployed uniformly in the entire sensor field – therefore there is no information on clustering. Every pair of nodes has the same chance to be neighbors. The EGD scheme uses a two dimensional Gaussian distribution to model node deployment in clusters where a mean (μ) is the targeted deployment point of each group. Next, multiple key pools are used in the EGD scheme as opposed to a single global key pool in the EG scheme. Each deployment group has its associated group key pool of size $|S_c|$ which is generated from the larger key pool of size $|S|$. Keys from the global key pool are assigned to group key pools in a way that the group key pools that are deployed nearby have a certain number of common keys. Overlapping factors denoted by a and b determine the fraction of shared keys between two adjacent group key pools. Assuming that clusters of sensors are arranged in a grid, of the $|S_c|$ keys in a given group key pool, $a|S_c|$ keys are shared between its horizontal and vertical neighboring clusters. The number of keys shared with its diagonal neighbors is $b|S_c|$. If two clusters are not neighbors, the group key pools do not share any keys. Given a global key pool of size $|S|$, number of deployment group, and overlapping factor, one can calculate $|S_c|$ by using a method described in [2]. For a memory size of k, a node randomly picks k keys from its associated *group* key pool of size $|S_c|$. The scheme has been shown to improve the network connectivity over the EG scheme for the same number k of keys installed in each node's memory. The probability of finding at least one common key between two nodes n_i and n_j that belong to deployment groups G_i and G_j respectively can be determined as in [4] as follows. Let $\delta(i, j)$ denote the number of shared keys between the deployment groups G_i and G_j and the overlapping factors between vertical-horizontal and diagonal groups be a and b respectively. The value of $\delta(i, j)$ changes as follows:

- When $i = j$, $\delta(i, j) = |S_c|$
- When i and j are horizontal or vertical group neighbors, $\delta(i, j) = a|S_c|$
- When i and j are diagonal group neighbors, $\delta(i, j) = b|S_c|$
- When i and j are not neighbors, $\delta(i, j) = 0$

The probability that two nodes share at least one key is:

$$1 - \frac{\sum_{m=0}^{min(k,\delta(i,j))} \binom{\delta(i,j)}{m} \binom{|S_c|-\delta(i,j)}{k-m} \binom{|S_c|-m}{k}}{\binom{|S_c|}{k}^2} \tag{2}$$

The computation of the above probability again considers the chance that two sets of k keys (now drawn differently as described) have no overlap (and subtract this probability from 1). To calculate Pr[two nodes do not share any key], first sensor node with a key ring of size k selects m keys from the intersecting keypool of size $\delta(i, j)$ and $k - m$ keys from its non-intersecting group key pool. The second node, in order to avoid selecting k keys that already selected by the first node, can pick only $|S_c| - m$ keys from its group key pool where m is the number of overlap keys between both node's group key pool that already picked by first node.

Instead of sharing keys, it is possible to share "key spaces" (e.g., using Blom's approach [3][1], that increases the resiliency of the network to multiple node compromise). While the proposed hybrid scheme can be changed to include this, we only consider sharing of keys in this paper. Both (1.1) and (1.2) ignore the fact that two sensor nodes may not be in transmission range. So the probability that two sensor nodes can securely communicate is actually conditional on the fact that they are within range of one another.

2.2 Jamming Attacks

Jamming attacks can disrupt communications in any wireless network quite easily. Xu, et al [11] has classified jammers into the following types: 1) Constant jammers that constantly emit a radio signal 2) Deceptive jammers that constantly inject fake (but valid otherwise) packets into the network without following the medium access protocol 3) Random jammers (also considered energy efficient jammers) that randomly choose a period of time to sleep and a random period of time to jam and 4) Reactive jammers that sense the channel and when they sense valid traffic being exchanged in the network they start jamming. To detect the presence of jamming attacks, [11] proposed to use packet delivery ratios as the main metric along with carrier sensing time and the signal strength. The results are promising, but not conclusive. In this paper we assume that jamming can be detected accurately.

Solutions to cope with jamming attacks include adjusting transmit power [10], data rate, or hopping to another frequency channel [13]. For a sensor node that has an ability to move, one convenient solution is to physically move the sensors away from the jammer [6]. We assume a constant or deceptive jammer and spatial retreats for combatting the attack in this paper.

To the best of our knowledge there is no publication that has looked at the effects of jamming attacks over connectivity of secure links, and how this problem can be solved. In the next section we explain the impact of jamming on secure connectivity and also describe the solution we propose to cope with the low level of secure connectivity due to jamming attacks.

3 Impact of Jamming on Secure Communications in Sensor Networks

In this section, we demonstrate the impact of a constant jammer on the probability of secure links in sensor networks. We use *local connectivity* (defined as the fraction of neighbors with whom at least one key is shared) and *number of moved nodes that are isolated* (nodes that share no keys with any neighboring nodes after spatial retreat) as our performance metrics. Then, we present our hybrid key pre-distribution scheme to cope with jamming attacks.

***Jamming versus node compromise*:** The node compromise attack is usually considered when designing key pre-distribution schemes. When a node is captured, sensitive information including encryption keys stored in node's memory may be disclosed. Jamming attacks may not be able to expose information inside a jammed node. However, in the worst case, it is essentially incommunicado and cannot help in the application objectives. An adversary may find it is more convenient to launch an jamming attack remotely using a powerful transmitter; rather than being in deployment area to capture a node.

A successful jammer can prevent the victim nodes from transmitting and receiving data. It is not necessary that a jammer should jam the whole network. A jammer can launch targeted jamming which focuses only specific victim nodes, links or flows. A jammed node may transmit a signal to a non-jammed node, thus creating an asymmetric link in the network [7]. However, due to MAC protocols that use carrier sensing, jamming attacks may be successful in preventing legitimate nodes from accessing the channel to send data. When a node senses a channel, it will see the channel as busy all the time [8].

***Jamming Attack Model*:** Here we describe the model of the jamming attacks that will be used in this paper.

- The jammer performs constant jamming or deceptive jamming. Any node that lies in jammed area is assumed to be affected completely by the jamming attack.
- The jammed region is assumed to be a circle centered at the jammer's location, the size of jammed region is measured by transmission range of the jamming device.
- The jammer interferes with part of the deployment area. As a result, there will be some nodes that are jammed and some nodes that are not jammed.

We will analyze the performance of the key pre-distribution schemes under this jamming model.

***Strategy for Spatial Retreat*:** The first step is to detect the presence of jamming attacks. We assume that sensor nodes use various statistical methods to detect the presence of jamming [12]. Once jamming is detected, nodes can identify jammed and non-jammed areas and map them [9]. One possible solution to overcome jamming is for a jammed node to evacuate from the jammed area (spatial retreat) [13]. The main goal of the evacuation process is to move jammed nodes out of the jammed region. The solution proposed by [6] is to move the jammed nodes in a random direction out of jammed area. Upon moving, each node continuously runs its detection algorithm until it reaches the border of the jammed region. After the node is outside the jammed area, it tries to connect to the sensor nodes nearby (finding new neighbor nodes). If there is no node within its radio range, the node will move along the jammed perimeter until it connects to other nodes.

We use a simpler strategy for node evacuation. If a node is deployed within a jammed area, the node will move out from the jammed region by randomly selecting its new location within the sensor field (it random picks a new x and y coordinate). This can be accomplished by the node moving a random distance in a random direction. Once the node moves to new location, it will check if its new location is also jammed. If so, it will randomly pick another location. After that, node will try to connect with sensor node nearby. In our simulations, we repeat the move till the node moves out of the jammed area. It is possible to improve the approach by increasing the distance moved from the current location in subsequent tries or to use the original approach in [6].

***Demonstration of the Impact of Jamming on the Secure Connectivity after Spatial Retreat*:** A secure link can be established between two sensor nodes under these two conditions: 1) sensor nodes are within each others' communication range 2) there is a common key between two nodes. After a node moves to its new location, it tries to find whether it has a common key with its new neighbors. A neighbor node that has at least one shared key will be able to establish a secure link with the moved node. The probability of having at least one common key with the new neighbor node depends on the type of key predistribution that was employed. If the sensor nodes select keys from a single key-pool as in the EG scheme, each node will have (on average) the same chance as in (1.1) to have a common key with its neighbor because the keys stored in the node's memory are selected regardless of the location of the nodes.

However, when the key pre-distribution scheme employs multiple key pools with deployment knowledge, each node will select its keys according to its associated key pool which depends on the deployment group that the node belongs to. Two nodes that picked their keys from the same key pool (they are from the same deployment group) will have a greater probability of finding a common key than two nodes that chose their keys from different key-pools (they are from different deployment groups). If the jammed node moves far enough to enter a

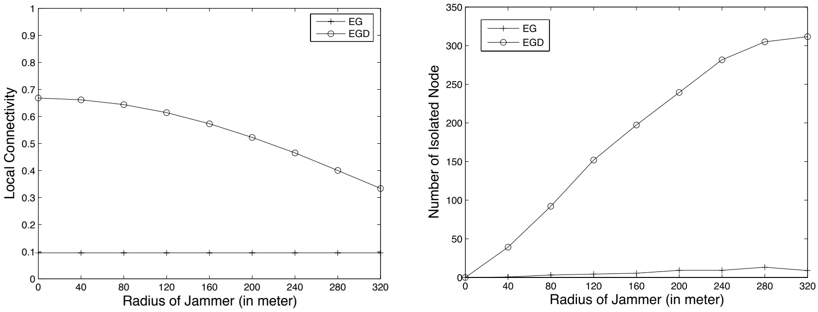

Fig. 1. (a) Local connectivity of EG and EGD schemes and (b) number of moved nodes that are isolated in EG and EGD schemes with different jamming radii

completely different deployment area, the chance of finding some common keys to establish secure links with the new set of neighbors will be reduced.

To see what impact jamming has on the local connectivity and the number of moved nodes that are isolated, we ran simulations that used $|S| = 100000$ keys, $|S_c| = 1760$ keys, number of keys installed in a node's memory $k = 100$ keys, overlap factors $a = 0.15$ and $b = 0.1$ in a 10000 node network in a 1000m × 1000m sensor field. The clusters of sensors in the deployment based multiple key pool approach are arranged a 10 × 10 grid, where each grid cell is of size 100m × 100m. The transmission range of a sensor is 40m. The numbers and scenario used here are very similar to the ones in [4,5]. The jammer is placed at the center of the sensor field.

Figure 1a shows the local connectivity after the nodes evacuate from the jammed region. We show the results of key connectivity for the whole network for different sizes of the jamming region. When the size of jamming hole is 0, it is equivalent to a network with no jamming. We compare the random scheme (EG) with the deployment knowledge scheme (EGD). Under jamming, we calculate the average connectivity of the whole network after all jammed nodes move away from jamming hole. It is clear that the local connectivity with the EGD scheme decreases while connectivity for EG scheme remain at the same level. Note however that the EG scheme already has poor connectivity (in this case, only 10% of neighbors share a key which implies that a high node density is mandatory for a securely connected network).

When a jammed node moves out of its deployment location, it will see a new set of one-hop neighbors at its final destination. With the EGD scheme, a node may travel beyond its deployment group to non-neighboring deployment groups. Nodes will have a slim chance of finding common keys with new neighbors since the selected keys are from non-overlapping group key pools. Thus, these nodes may be isolated from the network as they cannot connect to other sensors securely. By isolated we mean the node that is isolated because of jamming evacuation. Such a node cannot connect because it does not have any shared key with its new neighbors even though it is within their communication range. In Figure 1b, we plot the number of isolated nodes with different sizes of jamming area. When the jamming radius increases, the number of isolated nodes also increases at least up to a jamming radius of 320m (we have more discussion in Section 1.5). The number of isolated nodes with the EGD scheme is significantly larger than the number of isolated nodes with the EG scheme.

4 Hybrid Key Predistribution Scheme

In this section, we present a *hybrid* key predistribution scheme (HB scheme) designed to support the spatial retreat strategy to cope with jamming attacks. It makes use of the beneficial features of both the EG and EGD schemes. The goal of our scheme is as follows: When there is no jamming, the new scheme should show better connectivity compared to the random (EG) scheme. The new scheme should have an acceptable level of local connectivity even when the nodes have moved away from their original locations and fewer nodes should be isolated. All of this must be achieved without increasing the number of installed keys in a sensor node.

We adopt the group-based deployment model as in [4]. A group of N sensor nodes is divided into equal sized groups arranged in a grid of size $t \times n$. A sensor node that belongs to a group $G_{i,j}$ for $i = 1, \ldots, t$ and $j = 1, \ldots, n$ is deployed according to a target deployment point (x_i, y_j). The deployment points are arrange in grid as in [4]. Note that deployment points can be differently arranged depending on the method of deployment and application objective. We use a two-dimensional Gaussian distribution (Normal distribution) as in [4] for modeling deployment where the target deployment point is the mean of the distribution. The actual location of a sensor node will be around the associated target deployment point. The standard deviations are $50m$, which is similar to the number used in [4].

Like other existing key pre-distribution schemes proposed in the literature, the hybrid scheme comprises of 3 phases: a key distribution phase, a shared key discovery phase, and a path-key establishment phase.

Step 1: Key Distribution Phase: Each sensor node randomly selects keys from 2 types of key pools and installs them into the node's memory. We define two types of keypool. A *global keypool* that consists of large number of cryptographic keys and *group keypools* that consist of subsets of keys selected from a second global keypool. It is possible to create group key pools from the first global keypool, but we keep the two key pools separate to simplify the analysis presented next. Simulations (not shown here) show little difference between the two approaches since the group key pool is typically smaller than the global key pool (by two orders of magnitude – $|S_c| \ll |S|$ – for the 10×10 grid). Each deployment group has one associated group key pool. Given a global keypool S of size $|S|$, we divide S into $t \times n$ group key pools $S_{i,j}$ (for $i = 1, 2, \ldots, t$ and $j = 1, 2, \ldots, n$) each of size $|S_c|$. Each group key pool shares some amount of keys with adjacent group keypool (vertically, horizontally and diagonally) as previously described with overlapping factors a and b (see [4] for more details).

We further define a hybrid threshold τ. This threshold τ indicates the distribution of keys that a node selects from the first global keypool and its group keypool. The value of τ ranges from 0 to 1 ($\tau = 0, \ldots, 1$). When $\tau = 0$, a node will select keys only from its group keypool. This is equivalent to the EGD scheme. When $\tau = 1$, a node will select no key from the associated group key pool but will select all keys from the first global key pool. By doing this, the scheme is converted to the EG scheme (each node selects keys from the same key pool). Our scheme benefits from both key predistribution methods by selecting an appropriate value of τ as seen later in the simulations. Each sensor will select some amount of keys from its group keypool and some portion of keys from global keypool. For instance, given a memory size of $k = 100$ keys, when τ is set to 0.25, a node will select 25 keys from the first global keypool and 75 keys from its group keypool.

***Step 2: Shared Key Discovery Phase*:** After the nodes are deployed, they find some common keys with their neighbors. Each node does this by broadcasting a message containing the indices of the keys that they hold. Each node uses these broadcast messages with its neighbors to find out if they share a common key. If a common key exists between a pair of nodes, both nodes can establish a secure link using the shared key as a link key.

***Step 3: Path-Key Establishment Phase*:** Since the distribution of keys to each node is done randomly, it is possible that some nodes may not be able to find any common key with a subset of neighbors. In this case, as long as the key sharing graph of the entire sensor network is connected, the nodes can always establish secure links with neighbors through their shared-key neighbors. Note that step 2 and 3 are similar to EG and EGD schemes.

***Analyzing Secure Connectivity*:** Given that two sensor nodes are neighbors, we can calculate the probability that they share a key by using (1.1) and (1.2). This is simply 1 minus the probability that two nodes do not share a key from the first global key pool nor do they share a key from the group key pools. A node picks $k\tau$ keys from the first global key pool and $k(1-\tau)$ keys from its group key pool. Since the two key pools are independent, given τ, this probability can be written as:

$$1 - \left\{ \frac{((|S| - k\tau)!)^2}{(|S| - 2k\tau)!|S|!} \right\} \times \left\{ \frac{\sum_{m=0}^{min(k,\delta(i,j))} \binom{\delta(i,j)}{m} \binom{|S_c|-\delta(i,j)}{k(1-\tau)-m} \binom{|S_c|-m}{k(1-\tau)}}{\binom{|S_c|}{k(1-\tau)}^2} \right\} \tag{3}$$

Note that this probability is for the situation when there is no jamming. Under jamming and spatial retreat, the equation will change only in terms of the value of $\delta(i,j)$ which could be 0 in the worst case where nodes are from non-adjacent group or $|S_c|$ in the best case where nodes are from the same group.

5 Performance Evaluation

In this section, we evaluate the performance of the hybrid key predistribution scheme through simulations. The metrics considered are local connectivity and the the number of moved nodes that are isolated after detecting jamming and performing spatial retreat. We compare our results to the random scheme (EG scheme) [5] and the deployment knowledge based scheme (EGD scheme) [4]. Simulation parameters are the same as those in Section 1.3 unless otherwise stated. Each simulation is run 10 times with different seeds of the random number generator, and the results represent the average value of the 10 runs. We consider a range of values for the hybrid threshold τ, namely $\tau = 0, 0.25, 0.50, 0.75, 1$, to assess the performance. Under jamming, nodes perform spatial retreat as previously described in Section 1.3.

***Performance with a Single Jammer*:** Here the jammer is placed at the center of the sensor field. We vary the size of jammer by changing transmission range of jammer from 0 to 320 meters. The simulation results are shown in Figure 2. When $\tau = 1$, all keys stored in the node memory are picked from the first global keypool. Thus, the scheme converts to a random key distribution scheme (EG scheme). The only difference between the original EG scheme and the HB scheme with $\tau = 1$ is the nodes deployment method. The EG scheme uses a uniform deployment method while the HB scheme uses two dimensional gaussian deployment as in the EGD scheme. However, the local connectivity is not impacted by the deployment method as seen in Figure 2(a). At the other end, when τ is equal to 0, the scheme acts like the EGD scheme since all the keys installed in a node's memory are from the node's associated group key pool. Nodes that are from different groups will have a smaller chance of finding common keys as they select keys from different group key pools.

From the results in Figure 2(a), the local connectivity level decreases when the size of the jamming radius increases. This is to be expected. It is important to look at the the number of moved nodes that are isolated as well since local connectivity excludes those nodes that cannot connect to any neighbors. The results show that although the EGD scheme or HB scheme with $\tau = 0$ achieve high local connectivity, the the number of moved nodes that are isolated is also high. This is because when the size of the jamming region is increased, the number of jammed nodes increases. Since there are more sensor nodes that

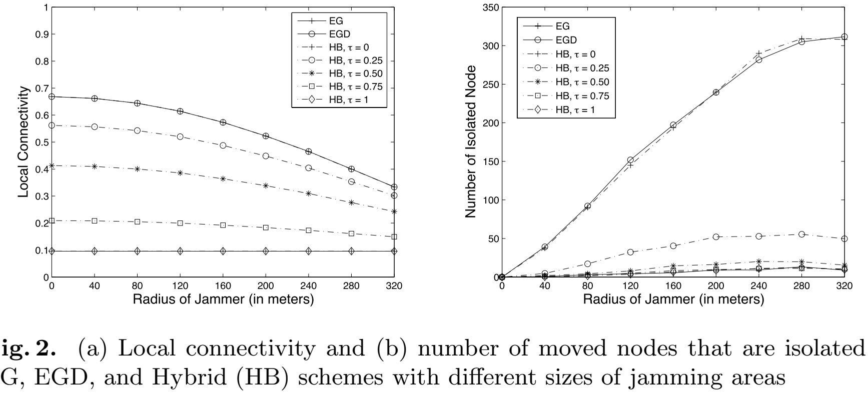

Fig. 2. (a) Local connectivity and (b) number of moved nodes that are isolated for EG, EGD, and Hybrid (HB) schemes with different sizes of jamming areas

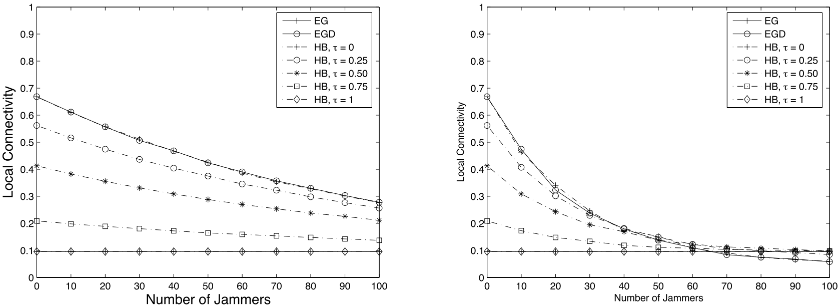

Fig. 3. Local connectivity of EG, EGD, and Hybrid (HB) scheme with multiple jammers (a) radius of jammer = 40m (b) radius of jammer = 80m

need to move out of the jammed area, there will be a larger chance that moved nodes will not be able to find a common key with their new neighbors. If nodes are finally surrounded by neighbors that are from different groups, they will have a small chance of finding common keys with them. However, the hybrid scheme performs in between the EG and EGD schemes depending on the value of τ. Clearly, the hybrid scheme outperforms the EGD scheme in that even with $\tau = 0.25$ when only 25% of the keys installed are from first global key pool, the the number of moved nodes that are isolated is reduced significantly while level of connectivity does not reduce much.

***Performance with Multiple Jammers*:** In the case of multiple jammers, we randomly place jammers in the deployment area (using a uniform distribution). The number of jammers is varied from 0 to 100. In some cases there may be overlap between jammed areas. In such a case, as long as a node is covered by at least one jammer, it is considered to be jammed. Figure 3 shows the local connectivity in the case of multiple jammers for the different schemes. In Figure 3(a), the individual jammers have a jamming radius of 40m (the same as the transmission range of a single sensor). In Figure 3(b), the jamming radius is doubled. Clearly, multiple jammers impact the local connectivity more significantly, especially if they have a larger radius. The performance of the various schemes show a similar trend as that with a single jammer for smaller numbers of jammers (i.e., the HB scheme is in between the EG and EGD schemes). Note that the jammed area could be much larger than the jammed area in the single jammer case, such that for more than 60 jammers with a jamming radius of 80m, the local connectivity of the EGD scheme drops below that of the EG scheme.

The the number of moved nodes that are isolated for the two cases is shown in Figure 4 (a) and (b) respectively. The number of isolated nodes can be as high as 10% of all nodes in the network if only the EGD scheme or HB scheme with $\tau = 0$ are used. Simply changing τ to 0.25 can reduce this number to 2% or lower indicating the benefits of the hybrid scheme. When the jamming radius is 80m and the number of jammers increases, at one point (around 20 jammers), the number of isolated nodes starts to decrease with the EGD scheme and the HB

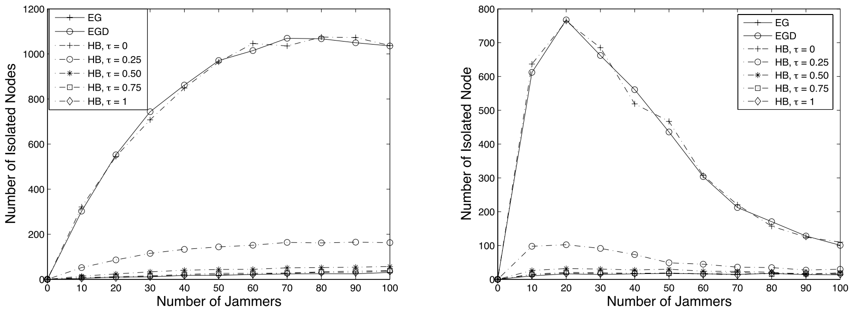

Fig. 4. Number of moved node that are isolated for EG, EGD, and Hybrid (HB) schemes with multiple jammers (a) radius of jammer = 40m (b) radius of jammer = 80m

scheme with $\tau = 0$ and $\tau = 0.25$. This is because the large number of jammers renders the total jammed area to be a significant fraction of the sensor field. Although it is hard to calculate the total jammed area (since the locations of each jammer is random and there could be overlaps), with 20 jammers and and a jamming radius of 80m, the jammed area is approximately $\frac{20 \times \pi \times 80^2}{1000^2} \approx 40.21\%$ of the deployment area. Consequently, sensor nodes are more likely to move close to each other so that the network becomes very dense resulting in a better chance for moved nodes to share keys with some new neighbors. A similar effect is seen with a single jammer when the jamming radius is much larger than 320m (results are not shown here).

***Impact of Grid Size and Node Density*:** In the previous results, a 10 × 10 grid of sensor clusters was used in the EGD and hybrid schemes. This means there are 100 group key pools, and each cluster of sensors is deployed in a 100m×100m grid. With a transmission range of 40m, sensors in a cluster (deployment group) will have a good chance of being in each other's transmission

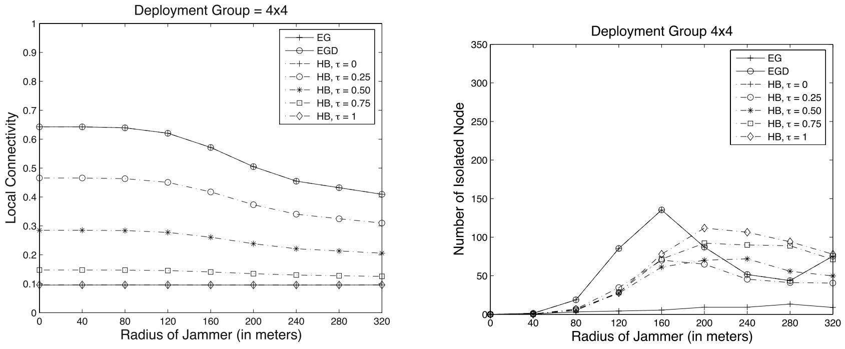

Fig. 5. (a) Local connectivity and (b) number of moved node that are isolated for EG, EGD, and HB schemes with different size of jamming areas for 4×4 grid size

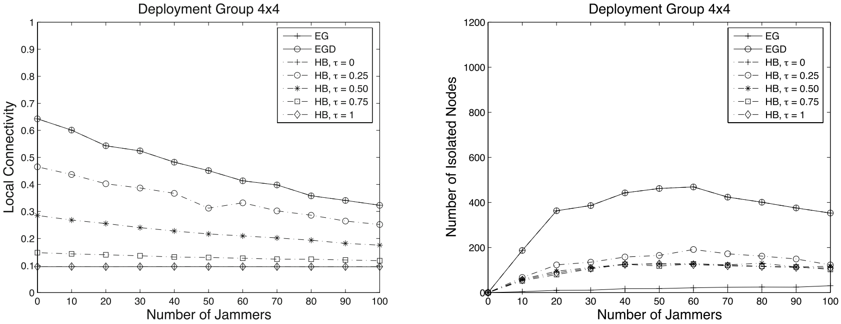

Fig. 6. (a) Local connectivity and (b) number of moved node that are isolated for EG, EGD, and HB schemes with multiple jammers for 4×4 grid size

range. The work in [4] does not look at the sensitivity of the key predistribution scheme to changes in the size of the grid. With the same size of deployment area ($1000m \times 1000m$), we run simulations using a 4×4 grid – there are 16 clusters of sensors and a grid is $250m \times 250m$ in size. The group key pool size increases to $|S_c| = 9433$ keys while it is 1760 keys in the 10×10 grid. There are 10000 sensors deployed in the field as before. We show the average of 5 simulation runs. Figures 5 and 6 show the local connectivity and the number of moved nodes that are isolated for single and multiple jammers respectively for various schemes. The drop in local connectivity of the EGD scheme or HB schemes compared to the 10×10 grid is not significant, and is in fact stable with increase in jamming radius. Moreover, the the number of moved nodes that are isolated is much smaller. This can be expected since a greater number of sensors derive keys from the same key pool (about six times more sensors than before). There is more chance that moved node will still be surrounded by neighbors that are from the same group. It is thus better to deploy fewer clusters of grids to provide resilience to jamming.

The node density will influence the connectivity and the ability to create a securely connected graph in the network. This is an issue that has not received much attention in the literature on key predistribution. We ran simulations to obtain some understanding of the impact of node density. The averages for 5 simulation runs are shown here. Figure 7 shows the results of the local connectivity and the number of moved nodes that are isolated as the number of deployed sensors changes in the 10×10 grid. We picked 50 jammers for illustration and compare the EG, EGD, and HB ($\tau = 0.5$) schemes. An interesting result of the simulations is that the number of moved nodes that are isolated drops as the node density increases with the EG and HB ($\tau = 0.5$) schemes while the EGD scheme continues to perform poorly. This is because the EGD scheme is optimized to exploit deployment and lacks the ability to be robust under changes to the initial deployment.

***Summary*:** By picking appropriate values of τ and the grid size, it is possible to balance the level of local connectivity and the number of moved nodes that are

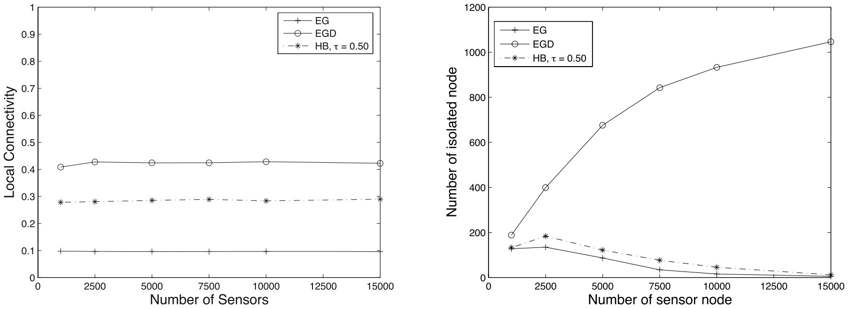

Fig. 7. (a) Local connectivity and (b) number of moved node that are isolated for EG, EGD, and HB (with $\tau = 0.50$) with different size of node density when number of jammers is 50. The jamming radius of each jammer is 40m.

isolated. For example (Figure 3(a) and 4(a)), when there are 50 jammers, the hybrid scheme with τ set to 0.25 has 12.03% lower connectivity than the EGD scheme but has an 85.04% decrease in the number of isolated nodes. Even ignoring the grid size, we can recommend the use of the hybrid scheme with $\tau = 0.25$ for good robustness to jamming and maintaining reasonable local connectivity.

6 Discussions, Limitations, and Ongoing Work

We clarify the limitation of the definition of isolation that we use here which does not guarantee that the network is not partitioned. For instance, two sensor nodes may securely connect to one another as they share common keys, but together, they may not be able to securely connect to any other sensor. Still, it provides a lower bound on the number of nodes that are disconnected from the largest securely connected part of the network. Ongoing work is considering quantifying the partitioning of the network.

Some limitations of this work are as follows. Another measure of connectivity used in [4,5] is the number of hops required to securely reach a direct neighbor. We have not looked at this measure in our work. We also would like to explore performance of our hybrid scheme with different evacuation strategies proposed in literature [6]. Other approaches to overcome jamming (e.g., reducing rate or increasing power) create longer links and the hybrid scheme may be useful there, but the actual tradeoffs are not clear. The assumption that a node that lies in jammed area will be completely affected can be relaxed as sensor nodes that lie at the border of jammed region may be able to retain transmission functionality.

Since a sensor node has limited memory space for storing cryptographic keys, it is desirable to use to the extent possible all of the keys stored in node's memory as link keys to neighbor nodes. A key stored in node's memory that is not useful is wasting memory space. If a link between two node A and B is jammed, it is not necessary for A and B to store a shared key. Information of areas that are more susceptible to jamming could be useful for network operators in predistributing keys to sensor nodes. This is also part of ongoing work.

7 Conclusion

In this paper, we described our study on the performance of key pre-distribution schemes in the presence of jamming attacks. We proposed a solutions for robust key distribution to cope with jamming attacks while maintaining good connectivity even when there is no jamming. We present an analysis and results from our simulations that show the benefits of the proposed scheme. A network operator can use our results to decide an appropriate value of τ that gives a satisfactory level of connectivity and number of isolated nodes under jamming attacks.

Acknowledgments

This work was funded in part by the National Science Foundation Award No. CNS-0721183. The authors would like to thank the anonymous reviewers for their comments and suggestions to improve the paper.

References

1. Blom, R.: An optimal class of symmetric key generation systems. In: Beth, T., Cot, N., Ingemarsson, I. (eds.) EUROCRYPT 1984. LNCS, vol. 209, pp. 335–338. Springer, Heidelberg (1985)
2. Du, W., Deng, J., Han, Y.S., Chen, S., Varshney, P.K.: A key management scheme for wireless sensor networks using deployment knowledge. In: IEEE INFOCOM, vol. 1, pp. 586–597 (2004)
3. Du, W., Deng, J., Han, Y.S., Varshney, P.K.: A pairwise key pre-distribution scheme for wireless sensor networks. In: Proceedings of the 10th ACM conference on Computer and Communications Security CCS 2003 (2003)
4. Du, W., Deng, J., Han, Y.S., Varshney, P.K.: A key predistribution scheme for sensor networks using deployment knowledge. IEEE Transactions on Dependable and Secure Computing 3(1), 62–77 (2006)
5. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: 9th ACM Conference on Computer and Communication Security, pp. 41–47 (November 2002)
6. Ma, K., Zhang, Y., Trappe, W.: Mobile network management and robust spatial retreats via network dynamics. In: 1st International Workshop on Resource Provisioning and Management in Sensor Networks (RPMSN 2005) (November 2005)
7. Noubir, G.: On Connectivity in Ac Hoc Networks under Jamming Using Directional Antennas and Mobility. In: Langendoerfer, P., Liu, M., Matta, I., Tsaoussidis, V. (eds.) WWIC 2004. LNCS, vol. 2957, pp. 186–200. Springer, Heidelberg (2004)
8. Pelechrinis, K., Iliofotou, M.: Denial of service attacks in wireless networks: The case of jammers, `http://www.cs.ucr.edu/~kpele/Jamming.pdf`

9. Wood, A., Stankovic, J., Son, S.: Jam: A jammed-area mapping service for sensor networks. In: 24th IEEE Real-Time Systems Symposium, pp. 286–297 (2003)
10. Xu, W.: On adjusting power to defend wireless networks from jamming. In: MobiQuitous: Mobile and Ubiquitous Systems: Networking and Services, pp. 1–6 (August 2007)
11. Xu, W., Ma, K., Trappe, W., Zhang, Y.: Jamming sensor networks: Attack and defense strategies. IEEE Network 20(3), 41–47 (2006)
12. Xu, W., et al.: The feasibility of launching and detecting jamming attacks in wireless networks. In: MobiHoc 2005: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing, pp. 46–57. ACM, New York (2005)
13. Xu, W., Wood, T., Trappe, W., Zhang, Y.: Channel surfing and spatial retreats: Defense against wireless dinal of service. In: ACM workshop on wireless security, pp. 80–89 (2004)