Group-Based Key Pre-Distribution in Wireless Sensor Networks[*]

Donggang Liu
CSE Department
University of Texas at Arlington
dliu@cse.uta.edu

Peng Ning
CS Department
North Carolina State University
pning@ncsu.edu

Wenliang Du
EECS Department
Syracuse University
wedu@ecs.syr.edu

ABSTRACT

Many key pre-distribution techniques have been developed recently to establish pairwise keys for wireless sensor networks. To further improve these schemes, researchers have proposed to take advantage of sensors' expected locations to help pre-distributing keying materials. However, it is usually very difficult, and sometimes impossible, to guarantee the knowledge of sensors' expected locations. In order to remove the dependency on expected locations, this paper proposes a practical deployment model, where sensor nodes are deployed in groups, and the nodes in the same group are close to each other after the deployment. Based on this model, the paper develops a novel group-based key pre-distribution framework, which can be combined with any of existing key pre-distribution techniques. A distinguishing property of this framework is that it does not require the knowledge of sensors' expected locations and greatly simplifies the deployment of sensor networks. The analysis also shows that the framework can substantially improve the security as well as the performance of existing key pre-distribution techniques.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Security, Design, Algorithms

Keywords

Sensor Networks, Key Management, Key Predistribution

1. INTRODUCTION

Recent technological advances have made it possible to develop wireless sensor networks consisting of a large number of low-cost, low-power, and multi-functional sensor nodes that communicate in short distances through wireless links [1]. Such sensor networks are ideal candidates for a wide range of applications such as health monitoring, data acquisition in hazardous environments, and military operations. The desirable features of wireless sensor networks have attracted many researchers to develop protocols and algorithms that can fulfill the requirements of these applications (e.g., [1, 11, 12, 19, 20]).

Security becomes a critical issue to ensure normal network operations as well as the integrity, availability, and at times confidentiality of the data collected by sensor nodes in hostile environments. However, providing security services in wireless sensor networks is quite challenging due to the resource constraints on sensor nodes and the threat of node compromises. In particular, it is usually impractical to establish keys between communicating sensor nodes using traditional methods such as public key cryptography and key distribution centers (KDC).

Key management is the cornerstone of security services such as authentication and encryption in wireless sensor networks. Research seeking low-cost key management techniques that can survive node compromises in sensor networks becomes quite active in the past two, three years, yielding several novel key pre-distribution schemes [5, 6, 8–10, 17, 18, 24, 25].

A basic probabilistic key pre-distribution scheme was proposed in [10]. In this scheme, each sensor node is assigned a random subset of keys from a key pool before deployment. As a result, two sensor nodes have a certain probability to share at least one key after deployment. This scheme was further extended in [6] by requiring two sensor nodes share at least q pre-distributed keys to establish a pairwise key. A random pairwise keys scheme was also developed in [6]. This scheme pre-distributes random pairwise keys between a sensor node and a random subset of other sensor nodes, and has the property that the compromise of sensor nodes does not lead to the compromise of any pairwise key shared directly between two non-compromised sensor nodes. Two similar threshold-based techniques were developed independently in [9, 17]. PIKE was developed by using peer sensor nodes as trusted intermediaries [5]. These three schemes significantly enhance the resilience of key predistribution against node compromises.

However, due to the resource constraints (especially the limited battery power) on sensor nodes and the threat of compromised nodes, none of the above key management schemes can guarantee the security of the keying materials used for the communication between sensor nodes. It is always desirable to improve the security and performance of key management.

In many sensor network applications, long distance peer-to-peer

[*] This work is supported by the US Army Research Office (ARO) under grant W911NF-05-1-0247, and by National Science Foundation (NSF) under grants CNS-0430223 and CNS-0430252.

WiSE'05, September 2, 2005, Cologne, Germany.
Copyright 2005 ACM 1-59593-142-2/05/0009 ...$5.00.

secure communication between sensor nodes is rare. When needed, we can use a secret key to secure the long distance peer-to-peer communication, where the key is established through a number of intermediate nodes if the hop by hop encryption and authentication is available. Thus, the primary goal of secure communication is to provide authentication and/or encryption between neighbor sensor nodes. Therefore, the most important information that can benefit key pre-distribution is the knowledge about *what nodes are the neighbors of each sensor node*.

Several techniques have been proposed to utilize the deployment knowledge of sensor nodes to improve key pre-distribution protocols [8, 14, 18, 24]. However, all these improved schemes assume that *the locations of sensor nodes can be pre-determined to a certain extent*. In practice, it is usually very difficult, and sometimes impossible, to guarantee the knowledge of sensors' expected locations. Moreover, this assumption severely limits the deployment of sensor networks. Thus, an interesting question we may ask is: *can we improve the existing key pre-distribution techniques without using expected location information?*

To answer the above question, this paper identifies a practical deployment model, where sensor nodes are deployed in groups, and the nodes in the same group are close to each other after the deployment. Based on this deployment model, this paper develops a novel group-based key pre-distribution framework. The analysis indicates that the framework indeed improves the security as well as the performance of existing key pre-distribution techniques substantially. Compared to the previous techniques for improving key pre-distribution, this approach has the following two advantages.

1. The proposed framework does not require the knowledge of sensors' expected locations, which is required by all the previous techniques in [8, 14, 18, 24] for improving key pre-distribution. This improvement greatly simplifies the deployment of sensor networks.

2. The proposed framework can be easily combined with any of those existing key pre-distribution techniques, while the previous techniques can only be used to improve certain type of key pre-distribution techniques.

The rest of this paper is organized as follows. The next section discusses our group-based deployment model. Section 3 presents our framework and provides detailed analysis. Section 4 reviews related work on sensor network security. Section 5 concludes this paper and points out possible future research directions.

2. GROUP-BASED DEPLOYMENT

In this section, we present a practical deployment model, where sensor nodes are only required to be deployed in groups. The knowledge used to improve the performance of key pre-distribution is the assumption that the sensor nodes belonging to the same group are deployed close to each other. This assumption is generally true, since the sensor nodes in the same group are supposed to be deployed from the same point at the same time. For example, a group of sensor nodes are dropped from the helicopter during the deployment. For the sake of presentation, we call such a group of sensor nodes as a *deployment group*.

We assume that sensor nodes are static once they are deployed. We define the *resident point* of a sensor node as the point location where this sensor node finally resides. Sensors' resident points are generally different from each other. However, we assume the resident points of the sensor nodes in the same group follow the same probability distribution function. The detailed description of the deployment model is given below.

The sensor nodes that are to be deployed are divided into n groups $\{G_i\}_{i=1,\ldots,n}$. We assume that the groups are evenly and independently deployed on a target field. The nodes in the same deployment group G_i are deployed from the same place at the same time with the deployment index i. During the deployment, the resident point of any node in group G_i follows a probability distribution function $f_i(x, y)$, which we call the *deployment distribution* of group G_i. An example of the pdf $f_i(x, y)$ is a two-dimensional Gaussian distribution. Figure 1 illustrates a two-dimensional Gaussian distribution at the center $(150, 150)$.

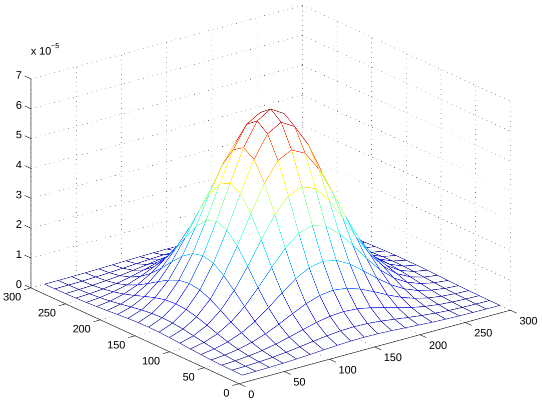

Figure 1: Deployment Distribution

Note that the actual deployment distribution is affected by many factors. For simplicity, we model the deployment distribution as a Gaussian distribution (also called Normal distribution) since it is widely studied and proved to be useful in practice. Although we only employ the Gaussian distribution, our methodology can be applied to other distributions as well.

We assume that the deployment distribution for a node in group G_i follows a two-dimensional Gaussian distribution centered at a *deployment point* (x_i, y_i). *Different from the deployment models in [8, 14], where the deployment points of groups are pre-determined, we do not assume any prior knowledge of such deployment points.* In fact, we only assume the existence of such deployment points. The mean of the Gaussian distribution μ equals (x_i, y_i), and the pdf for any node in group G_i is the following:

$$f_i(x, y) = \frac{1}{2\pi\sigma^2} e^{-[(x-x_i)^2+(y-y_i)^2]/2\sigma^2} = f(x - x_i, y - y_i),$$

where $f(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$, and σ is the standard deviation.

3. GROUP-BASED KEY PRE-DISTRIBUTION

According to the deployment model discussed in the previous section, the sensor nodes in the same deployment group have high probability of being neighbors. To take advantage of this observation, the pairwise key pre-distribution techniques should at least benefit the sensor nodes in the same deployment group. Hence, we first employ an *in-group key pre-distribution* method, which enables the sensor nodes in the same deployment group to establish pairwise keys between each other with high probability. To handle the pairwise key establishment between sensor nodes in different deployment groups, we then employ a *cross-group key pre-distribution* method, which enables selected sensor nodes in different deployment groups to establish pairwise keys and thus bridges different deployment groups together .

In the above idea, as long as a key pre-distribution technique can provide pairwise key establishment between sensor nodes in a group, it can be used as the basic building block to construct the group-based scheme. This implies that our framework can be applied to any existing key pre-distribution technique.

3.1 A General Framework

Without loss of generality, let $\mathcal{D}$ denote the key pre-distribution technique used in the framework. This subsection shows how to construct an improved key pre-distribution technique by applying the group knowledge to $\mathcal{D}$. Note that the previous location-based key pre-distribution techniques [8, 14, 18, 24] are not applicable here since the framework does not assume the knowledge of the expected locations of sensor nodes.

A key pre-distribution technique can usually be divided into three phases, *pre-distribution*, which specifies how to pre-distribute keying materials to each sensor node, *direct key establishment*, which specifies how to establish a pairwise key shared between two sensor nodes *directly*, and *path key establishment*, which specifies how to find a sequence of nodes to help two given nodes to establish a temporary session key. The key established in the direct key establishment phase is called the *direct key*, while the key established in the path key establishment phase is called the *indirect key*.

We refer to an instantiation of $\mathcal{D}$ for a group of sensor nodes as a *key pre-distribution instance*. A key pre-distribution instance D includes a set of target sensor nodes G, a set of keying materials K (e.g., keys [5, 6, 10], polynomials [17], or matrixes [9]), and a function g that maps an ID in G to a subset of keying materials in K. In such an instance, each sensor node i in group G is pre-distributed with a set of secrets that are computed from the mapping result of ID i under function g. This set of secrets could be keys [5,6,10], polynomial shares [17], or a row of elements on a matrix [9].

We also define the following *property functions* to characterize the typical properties of a key pre-distribution instance.

- $M(D)$: the memory requirements on sensor nodes for a key pre-distribution instance D.
- $p_{dk}(D)$: the probability of sharing a direct key between any two sensor nodes in a key pre-distribution instance D.
- $p_{cd}(D, x)$: the probability of a direct key between two non-compromised sensor nodes being compromised in a key pre-distribution instance D when the adversary has randomly compromised x sensor nodes.

Our group-based key pre-distribution framework is built upon a number of key pre-distribution instances. For simplicity, we assume there are n equal size deployment groups with m sensor nodes in each of those groups. The description of our framework are described below. For simplicity, we omit the detail of the message format.

3.1.1 Pre-Distribution

For each deployment group G_i, we randomly generate a key pre-distribution instance D_i. The pairwise key establishment between sensor nodes in group G_i is based on instance D_i. For the sake of presentation, these randomly generated instances are called the *in-group (key pre-distribution) instances*.

To handle the pairwise key establishment between sensor nodes in different deployment groups, we further generate m key pre-distribution instances $\{D'_i\}_{i=1,...,n}$. These instances are called the *cross-group (key pre-distribution) instances*. The set of nodes having the same cross-group instance D'_i form a *cross group* G'_i. The requirements on these cross groups $\{G'_1, ..., G'_m\}$ are: (1) each cross group includes exactly one sensor node from each deployment group, and (2) there are no common sensor nodes between any two different cross groups. In other words, for any i and j with $i \neq j$, we have $G'_i \cap G'_j = \phi$ and $|G'_i \cap G_j| = 1$. By doing this, each cross group provides a potential link for any two deployment groups.

In this paper, we propose a simply way to construct deployment groups and cross groups. Basically, each deployment group G_i contains the sensor nodes with IDs $\{(i-1)m + j\}_{j=1,...,m}$, while each cross group G'_i contains the sensor nodes with IDs $\{i + (j-1)m\}_{j=1,...,n}$. By doing this, any sensor node can easily figure out what deployment group and cross group a sensor node belongs to. Figure 2 shows an example of such a construction when $n = 4$ and $m = 3$. In the figure, G'_1 includes node 1, 4, 7 and 10, G'_2 includes node 2, 5, 8 and 11, and G'_3 includes node 3, 6, 9 and 12.

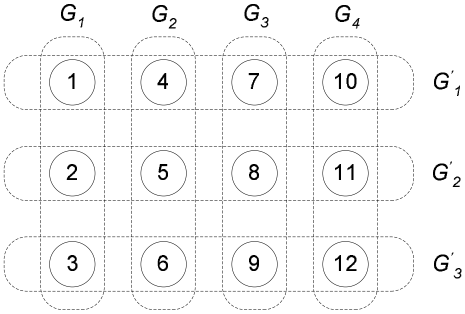

Figure 2: Example of group construction

This approach is similar to the logical grid in the grid-based key pre-distribution scheme [17], which was also used in PIKE recently [5]. However, our research in this paper is focused on using the locality of group deployment to improve the performance of the existing key pre-distribution techniques, and is substantially different from [17] and [5].

3.1.2 Direct Key Establishment

After the pre-distribution step, each sensor node belongs to two key pre-distribution instances, an in-group instance and a cross-group instance. Hence, the direct key establishment between two sensor nodes is simple and direct. If they are in the same deployment group, for example, G_i, they can follow the direct key establishment of the in-group instance D_i. If they are not in the same deployment group but belong to the same cross group G'_j, they can follow the direct key establishment of the cross-group instance D'_j. To determine if two sensor nodes are in the same deployment group or the same cross group, they only need to exchange the IDs of groups that they belong to. In our framework, they only need to know the ID of the other party due to our group construction method.

3.1.3 Path Key Establishment

If two nodes cannot establish a direct key, they have to go through path key establishment to find a number of other sensor nodes to help them establish an indirect key. Similar to the direct key establishment, if two nodes are in the same deployment group G_i, they can follow the path key establishment in D_i. The indirect keys between sensor nodes in the same group are called the *in-group indirect keys*. When two nodes belong to two different groups G_i

and G_j, we use a different method to establish an indirect key. Basically, we need to find a “bridge” between these two deployment groups in order to setup a *cross-group indirect key*. A bridge between group G_i and G_j is defined as a pair of sensor nodes $\langle a, b\rangle$ ($a \in G_i$ and $b \in G_j$) that belong to the same cross group G'_k ($a, b \in G'_k$). A bridge is valid when the two sensor nodes involved in this bridge can establish a direct key.

According to the pre-distribution, there are m potential bridges (one from each cross group) that can be used to establish an indirect key. In addition, due to our group construction method, a sensor node can easily compute all possible bridges between any two deployment groups. Specifically, the possible bridges between group G_i and G_j are $\{\langle (i-1)m+k, (j-1)m+k\rangle\}_{k=1,\ldots,m}$. For example, there are 3 bridges between group G_1 and G_4 in Figure 2: $\langle 1, 10\rangle$, $\langle 2, 11\rangle$, and $\langle 3, 12\rangle$.

Assume every message between two sensor nodes is encrypted and authenticated by the pairwise key established between them. The path key establishment for the sensor nodes in different deployment groups works as follows.

1. The source node u first tries the bridge involving itself to establish an indirect key with the destination node v. Assume this bridge is $\langle u, v'\rangle$. Node u first sends a request to v' if it can establish a direct key with v'. If node v' can also establish a (direct or indirect) key with the destination node v, node v' forwards this request to the destination node v to establish an indirect key.

2. If the first step fails, node u tries the bridge involves the destination node v. Assume the bridge is $\langle u', v\rangle$. In this case, node u sends a request to node u' if it can establish a (direct or indirect) key with u'. If node u' can establish a direct key with node v, it forwards the request to the destination node v to establish an indirect key. Note that if node u and v are in the same cross group, this step can be skipped, since step 1 and step 2 compute the same bridge.

3. When both of the above steps fail, node u has to try other bridges. Basically, it randomly choses a bridge $\langle u', v'\rangle$ other than the above two, assuming u' is in the same deployment group with u, and v' is in the same deployment group with v. Node u then sends a request to u' if it can establish a (direct or indirect) key with u'. Once u' receives this request, it forwards the request to v' in the bridge if they share a direct key. If v' can establish a (direct or indirect) key with the destination node v, it forwards the request to node v to establish an indirect key.

To show an example, we use the same configuration as in Figure 2. When node 1 wants to establish a pairwise key with node 12, it first tries the bridge $\langle 1, 10\rangle$. If this fails, it tries the bridge $\langle 3, 12\rangle$. If both bridges fail, it needs to try the bridge $\langle 2, 11\rangle$. If none of these bridges works, the path key establishment fails. In our later analysis, we will see that it is usually very unlikely that none of those bridges works.

Note that in the above approach, the path key establishment in a cross-group instance has never been used. The reason is that the sensor nodes in a cross group usually spread over the entire deployment field, which may introduce significant communication overhead in path key establishment.

3.2 Performance Analysis

For simplicity, we assume all in-group and cross-group key pre-distribution instances have the same property functions ($M(D)$, $p_{dk}(D)$, and $p_{cd}(D, x)$). Indeed, this assumption is true for all the key pre-distribution techniques in [5, 6, 9, 10, 17] given the same storage overhead, group size, and keying material size. Throughout this paper, we use M, p_{dk}, and $p_{cd}(x)$ to represent the three property functions, respectively. For simplicity, the analysis focuses on the probability of establishing keys between sensor nodes. Table 1 lists the notations that are used frequently in our analysis.

Table 1: Notations

n	number of deployment groups
m	number of nodes in a deployment group
c	number of compromised sensor nodes
M	memory required for one key pre-distribution instance
p_{dk}	probability of having a direct key in a key pre-distribution instance
$p_{cd}(x)$	probability of a direct key being compromised in a key pre-distribution instance when the adversary has randomly compromised x nodes
p_{gdk}	probability of having a direct key in the group-based scheme
$p_{gcd}(x)$	probability of a direct key being compromised in the group-based scheme when the adversary has randomly compromised x nodes
$p_{gci-in}(x)$	probability of an indirect key between two nodes in the same deployment group being compromised when the adversary has randomly compromised x nodes
$p_{gci-cr}(x)$	probability of an indirect key between two nodes in different deployment groups being compromised when the adversary has randomly compromised x nodes

3.2.1 Overhead

This paper provides a method to establish pairwise keys between sensor nodes. The overhead of using such keys in security protocol (e.g., encryption or authentication) depends on the real application. Thus, in this paper, we only focus on the overhead involved in establishing such keys.

Obviously, the storage overhead on a sensor node can be estimated as $2M$. The communication overhead to establish a direct key is the same as the communication overhead to establish a direct key in an in-group or cross-group key pre-distribution instance. When two nodes need to establish an indirect key, there are two cases. If these two nodes are in the same deployment group, the path key establishment only involves the sensor nodes in this deployment group. If these two nodes are in different deployment groups, the path key establishment only involves those in the same deployment group with the source node or the destination node. In other words, the communication is limited in two deployment groups. In addition, we also note that if two sensor nodes in two deployment groups are neighbors, the corresponding deployment groups have high probability of being close to each other, which may reduce the overall communication overhead significantly in their path key establishment.

3.2.2 Establishing Direct Keys

Consider a particular sensor node u in the deployment group G_i at position (x', y'). Let A denote its *communication area* in which any other sensor node can directly communication with node u. In this paper, we assume A is a circle centered at (x', y') with radius R, where R is the radio range of a sensor node. Thus, the average number of sensor nodes in the deployment group G_j that finally reside in A can be estimated as

$$n_{i,j}(x', y') = m \iint_A f(x - x_j, y - y_j)\mathrm{d}x\mathrm{d}y.$$

For any deployment group G_j other than G_i, we know that there is only one sensor node u' in G_j that shares the same cross group G'_k with node u. Thus, the probability of this node u' being deployed in A can be estimated as $\frac{n_{i,j}(x',y')}{m}$. This indicates that among all those sensor nodes deployed in A, the average number of senor nodes that belong to the deployment groups other than G_i but share the same cross group G'_k with node u can be estimated as

$$n'_i(x', y') = \frac{\sum_{j=1,j\neq i}^{n} n_{i,j}(x', y')}{m}.$$

When the sensor nodes are evenly distributed in the deployment field, it is possible to further simplify the above equation. Suppose the average number of sensor nodes in the communication range of a sensor node is n_A. We have $\sum_{j=1,j\neq i}^{n} n_{i,j}(x', y') = n_A - n_{i,i}(x', y')$. Thus,

$$n'_i(x', y') = \frac{n_A - n_{i,i}(x', y')}{m}.$$

In addition, the probability of having a direct key between u and any sensor node that shares the same key pre-distribution instance with u is p_{dk}. Thus, the average number of sensor nodes in A that can establish direct keys with node u can be estimated as $(n_{i,i}(x', y') + n'_i(x', y')) \times p_{dk}$. This means that the probability of u having direct keys with its neighbor nodes can be estimated as

$$p_i(x', y') = \frac{(n_{i,i}(x', y') + n'_i(x', y')) \times p_{dk}}{n_A}.$$

Hence, for any node in group G_i, the probability of having direct keys with its neighbor nodes can be estimated as

$$p_{gdk} = \iint_S f(x - x_i, y - x_i)p_i(x, y)\mathrm{d}x\mathrm{d}y,$$

where S denotes the entire deployment field.

p_{gdk} can also be used to estimate the probability of any node in any deployment group having a direct key with its neighbor node when S is an infinite field. For a given deployment field S, we simply configure the deployment point of G_i as its geometric centroid, and use the probability of a node in G_i having a direct key with its neighbor node to represent the probability of having a direct key between any two neighbor nodes.

To evaluate our approach when it is combined with a particular key pre-distribution technique (e.g., the random pairwise keys scheme), we use the following configuration throughout this paper. we assume there are totally 10,000 sensor nodes deployed on a $1000m \times 1000m$ area. These sensor nodes are divided into 100 deployment groups with 100 sensor nodes in each group ($n = m = 100$). We assume sensor nodes are evenly distributed in the deployment field so that the probability of finding a node in each equal size region can be made approximately equal. In other words, the density of sensor nodes is approximately one sensor node per 100 square meter. We always assume the radio range is $R = 40m$. Thus, there are $\frac{\pi \times 40 \times 40}{100} \approx 50.27$ sensor nodes on average in the communication range of a given sensor node. We also set $\sigma = 50$m in all those deployment distributions $\{f_i(x, y)\}_{i=1,\ldots,n}$.

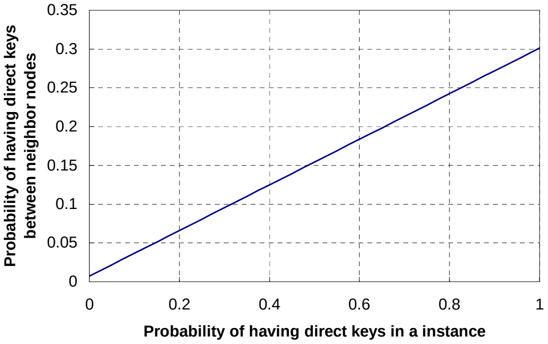

Figure 3: Probability of having a direct key between two neighbor nodes.

Figure 3 shows the probability of having a direct key between two neighbor nodes under the above configuration. We can see that the probability p_{gcd} increases almost linearly as p_{dk} increases. Since p_{dk} can be made quite large with small storage overhead for a small group of nodes, we expect that the group-based schemes can improve the performance of existing key pre-distribution techniques significantly. To illustrate this point, we investigate the improvements we can achieve by combining the framework with the basic probabilistic key pre-distribution scheme in [10], the random pairwise keys scheme in [6], and the polynomial-based key pre-distribution in [3]. The result of such combination generates three novel key pre-distribution schemes: a *group-based EG* scheme, which combines the framework with the basic probabilistic scheme, a *group-based RK* scheme, which combines the framework with the random pairwise keys scheme, and a *group-based PB* scheme, which combines the framework with the polynomial-based scheme.

For the basic probabilistic key pre-distribution scheme, we assume the key pool size is $100,000$. This key pool is divided into 200 small equal size key pools in the group-based EG scheme (500 keys in each small key pool). Each key pre-distribution instance uses a unique key pool. Each sensor node selects the same number of keys from the key pools in its in-group instance and cross-group instance. Figure 4 shows that the group-based EG scheme improves the probability of having a direct key between two neighbor sensor nodes significantly when there are severe memory constraints (e.g., 50 keys on each sensor node).

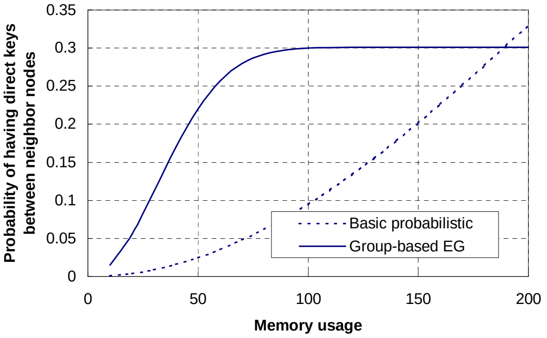

Figure 4: Probability of having a direct key between two neighbor sensor nodes. Memory usage is measured by counting the number of keys stored on each node.

Figure 5 compares the probability of having direct keys between neighbor nodes for the random pairwise keys scheme in [6] and the group-based RK scheme under the same memory constraint. We can clearly see that our framework can significantly improve the probability of having a direct key between two neighbor sensor nodes for the random pairwise keys scheme. This indicates that the group-based RK scheme can support larger sensor networks than the random pairwise keys scheme given the same configuration.

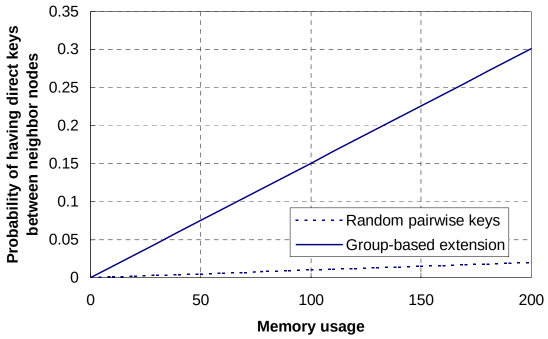

Figure 5: Probability of having a direct key between two neighbor sensor nodes. Memory usage is measured by counting the number of keys stored on each node.

Figure 6 shows the probability of having direct keys between neighbor sensor nodes for the group-based PB scheme, the random subset assignment scheme [17], and the grid-based scheme [17]. For all these schemes, we assume the same number of bivariate polynomials in the system and the same number of polynomial shares stored on each sensor node. Specifically, there are 100 deployment groups and 100 cross groups for the group-based PB scheme. Each of these groups is assigned one unique bivariate polynomial for the corresponding key pre-distribution instance. Each sensor node gets assigned the polynomial shares on its in-group instance and cross-group instance. Similarly, there are 200 bivariate polynomials in the polynomial pools of the random subset assignment scheme and the grid-based scheme. The random subset assignment scheme assigns the polynomial shares of 2 randomly selected polynomials from the pool to each sensor node, while the grid-based scheme arranges 200 polynomials on a 100 × 100 grid. We can clearly see that the probability of having a direct key between two neighbor sensor nodes in the group-based PB scheme is much higher than that in the random subset assignment scheme and the grid-based scheme.

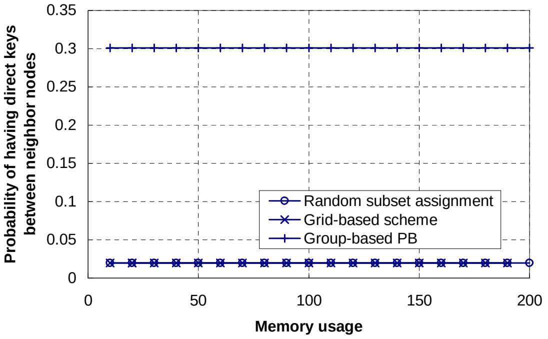

Figure 6: Probability of having a direct key between two neighbor sensor nodes. Memory usage is measured by counting the number of polynomial coefficients stored on each node.

3.2.3 *Establishing Indirect Keys*

In the following, we estimate the probability of having an indirect key between two neighbor sensor nodes if they cannot establish a direct key.

Obviously, if two neighbor sensor nodes are in the same deployment group G_i, they can follow the path key establishment of D_i to establish an indirect key. We note that a deployment group usually has a limited number of sensor nodes (e.g., 100). Since the nodes in the same deployment group are usually close to each other, a sensor node can easily contact most of other nodes in the same deployment group. For example, a sensor node can launch a *group flooding*, where only the sensor nodes in the same group participate in the flooding, to contact other nodes. Note that the group flooding is much more efficient than the network-wide flooding since most of the nodes in a group are located in the same small local area.

Therefore, we believe that it is usually possible to configure the key pre-distribution instance for a deployment group with small storage overhead so that any two sensor nodes in this group can either share a direct key or establish an indirect key at a very high probability with reasonable communication overhead. For example, we employ the random pairwise keys scheme in [6] for a group of 100 sensor nodes, and assign 50 keys to each sensor node. In this case, a sensor node can establish a direct key with its neighbor node at a probability of 0.5. After contacting half of the sensor nodes in this group, the probability of finding one node that shares direct keys with both the source and destination nodes can be estimated as $1-(1-0.5\times 0.5)^{50}\approx 0.999999$. Hence, we always assume two sensor nodes in the same deployment group can always establish an indirect key in this paper.

The situation becomes more complicated if two sensor nodes are in different deployment groups. In this case, they have to find a valid bridge between these two deployment groups to establish an indirect key. Since there are m cross groups, there are m potential bridges. As long as one of them works, the source node can establish an indirect key with the destination node through this bridge. The probability that none of these bridges works can be estimated as $(1-p_{dk})^m$. Thus, the probability that at least one bridge works, which is equivalent to the probability of having an indirect key between two neighbor nodes in different deployment groups, can be estimated as $1-(1-p_{dk})^m$.

Figure 7 illustrates the probability of having an indirect key between two neighbor sensor nodes that are in different deployment groups, assuming the same configuration as in Section 3.2.2 for the group-based EG scheme, the group-based RK scheme, and the group-based PB scheme. We can see that two neighbor sensor nodes in different deployment groups can usually establish an indirect key even if there are severe memory constraints on sensor nodes (e.g., 10 keys per sensor node).

3.3 Security Analysis

The main threat we consider in the security analysis is the compromise of sensor nodes. We assume an adversary randomly compromises c sensor nodes in the network. This subsection focuses on the impact of compromised sensor nodes on the direct key establishment and the path key establishment.

Similar to the analysis in the previous subsection, we investigate the security of the proposed framework after combining it with the basic probabilistic key pre-distribution scheme in [10], the random pairwise keys scheme in [6], and the polynomial-based key pre-distribution in [3].

It is easy to see that the grid-based scheme in [17] can be considered as a group-based PB scheme if a row or a column of sensor nodes in the grid are deployed in the same group. This means that

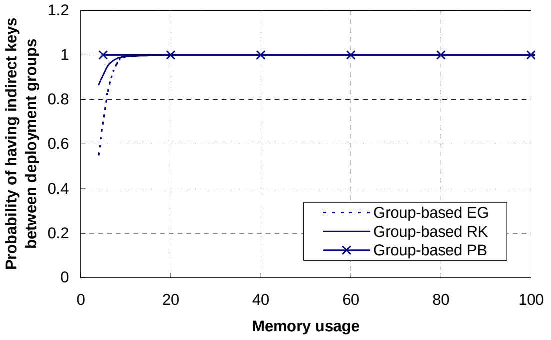

Figure 7: Probability of having indirect keys between sensor nodes in different deployment groups. Memory usage is measured by counting the number of keys or polynomial coefficients stored on each node.

the grid-based scheme and the group-based PB scheme have the same security performance against node capture attacks given the same configuration (e.g., storage overhead, network size). Thus, in our later security analysis, we simply skip the security comparison between the grid-based scheme and the group-based PB scheme. On the other hand, we noticed in Figure 6 that the group-based PB scheme can achieve much higher probability of establishing direct keys between neighbor sensor nodes than the grid-based scheme. This implies that the group-based PB scheme is more desirable than the grid-based scheme when the group-based deployment model is made possible.

During the evaluation, we always assume that the memory usage at each sensor node is equivalent to store 100 cryptographic keys. According to the previous configuration, there are $10,000$ sensor nodes in the network, and $n = m = 100$. Thus, for the random pairwise keys scheme, the probability of having a direct key between two neighbor nodes is 0.01, while for the group-based RK scheme, the probability of having a direct key between two neighbor nodes is 0.15 as shown in Figure 5.

In addition to the above key pre-distribution schemes, we configure all other schemes in such a way that the probability of having a direct key between two neighbor sensor nodes is 0.3.

- *Basic probabilistic scheme in [10]*: The key pool size is 28,136. Each sensor node randomly selects 100 keys from this pool.
- *Random subset assignment scheme in [17]*: The polynomial pool size is 13, and each polynomial has the degree of 49. Each sensor node randomly selects 2 polynomials from the pool and stores the corresponding polynomial shares.
- *Group-based EG scheme*: The key pool size in each instance is 500. Each sensor node randomly selects 50 keys from its in-group instance and 50 keys from its cross-group instance.
- *Group-based PB scheme*: Each instance includes a 49-degree bivariate polynomial. Each sensor node gets assigned the polynomial shares from its in-group instance and cross-group instance.

3.3.1 Impact on Direct Key Establishment

Consider a direct key between two non-compromised nodes in the same deployment group G_i. Since there are totally c compromise sensor nodes, the probability of j sensor nodes in group G_i being compromised can be estimated as $\frac{c!}{(c-j)!j!}\frac{(n-1)^{c-j}}{n^c}$ for $j \leq m-2$. When j sensor nodes in group G_i are compromised, the probability of this direct key being compromised can be estimated as $p_{cd}(j)$. Hence, the probability of any direct key between two non-compromised sensor nodes in a deployment group being compromised can be estimated as

$$p_{gcd}(c) = \sum_{j=0,j<=c}^{m-2} \frac{c!}{(c-j)!j!}\frac{(n-1)^{c-j}}{n^c} p_{cd}(j)$$

Since $n = m$, the above $p_{gcd}(c)$ can also be used to estimate the probability of a direct key between two non-compromised sensor nodes in the same cross group being compromised.

Figure 8 compares the probability of a direct key between two non-compromised sensor nodes being compromised for the basic probabilistic key pre-distribution scheme in [10] and the group-based EG scheme. We can see that the security of direct keys can be significantly improved by applying our framework.

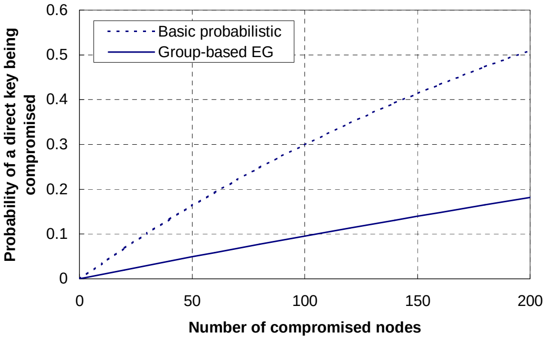

Figure 8: Probability of a direct key between two non-compromised nodes being compromised. Assume the probability of having a direct key between two neighbor nodes is 0.3.

For the random pairwise keys scheme [6], the compromise of sensor nodes does not affect any of the direct keys established between non-compromised sensor nodes ($p_{cd}(j) = 0$), since every key is generated randomly and independently. Thus, if we apply our framework to the random pairwise keys scheme, the resulting scheme still has the perfect security guarantee against node capture attacks ($p_{gcd}(c) = 0$), which means that the compromise of sensor nodes does not affect direct keys between non-compromised nodes. Together with the result in Figure 5, we can conclude that our framework can improve the probability of having direct keys between neighbor sensor nodes significantly without sacrificing the security of direct keys.

Figure 9 shows the probability of a direct key between two non-compromised sensor nodes being compromised for the group-based PB scheme and the random subset assignment scheme in [17]. We can see that the group-based PB scheme has much better security performance than the random subset assignment scheme in terms of the compromised direct keys.

3.3.2 Impact on Path Key Establishment

In the following, we first study the impact of compromised sensor nodes on the indirect keys established between sensor nodes in the same deployment group (in-group indirect keys), and then study the impact of compromised sensor nodes on the indirect keys established between sensor nodes in different deployment groups (cross-group indirect keys).

Note that when the compromised sensor nodes can be detected, two non-compromised nodes can always re-establish an indirect

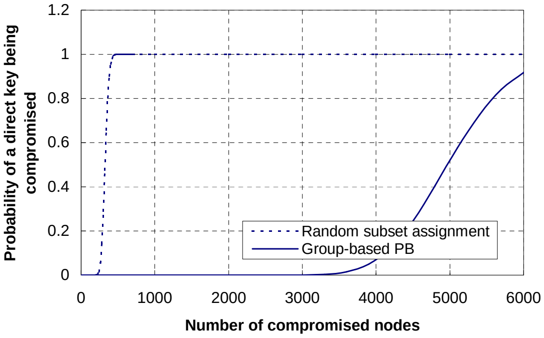

Figure 9: Probability of a direct key between two non-compromised nodes being compromised. Assume the probability of having a direct key between two neighbor nodes is 0.3.

key through path key establishment and avoid those compromised sensor nodes or compromised key pre-distribution instances. However, it is usually very difficult to detect compromised sensor nodes. When the compromised nodes cannot be detected, the indirect key between two non-compromised nodes may be disclosed to the attacker without being noticed. In the following analysis, we focus on the probability of a given indirect key between two non-compromised sensor nodes being compromised when the node capture attacks cannot be detected.

Probability of in-group indirect keys being compromised: When there are c compromise nodes, the probability of a particular sensor node being compromised can be estimated as $\frac{c}{nm-2}$. According to our earlier analysis, the probability of establishing an in-group indirect key that only involves one intermediate node is usually very high. For simplicity, we assume the in-group indirect key can always be established through one intermediate node. Thus, the establishment of an in-group indirect key involves an intermediate node, a direct key for the link between the source node and the intermediate node, and a direct key for the link between the intermediate node and the destination node. Thus, if the intermediate node and the two direct keys are not compromised, the indirect key is still secure. This means that the probability of an in-group indirect key between two non-compromised nodes being compromised can be estimated as

$$p_{gci-in}(c) = 1 - (1 - \frac{c}{nm-2})(1 - p_{gcd}(c))^2$$

Figure 10 shows the probability of in-group indirect keys between non-compromised nodes being compromised for the group-based EG scheme. It also includes the probability of a given indirect key (involving only one intermediate node) between two non-compromised nodes being compromised for the basic probabilistic scheme in [10]. We can see that the group based EG scheme has higher security guarantee for the indirect keys between the sensor nodes in the same deployment group.

For the group-based RK scheme, since $p_{gcd}(c) = 0$, we have $p_{gci-in}(c) = \frac{c}{nm-2}$. This means that given the same network size, the probability of an in-group indirect key being compromised for the group-based RK scheme will equal to the probability of a given indirect key (involving only one intermediate node) being compromised in the random pairwise keys scheme in [6]. However, we note the probability of having a direct key between two neighbor nodes in the random pairwise keys scheme is much lower than that in the group-based RK scheme. In fact, given a large sensor network and small storage overhead, it is very difficult and expensive for the random pairwise keys scheme to establish an indirect key (not to mention the indirect key that involves only one intermediate node) between two neighbor nodes. On the other hand, according to the analysis in Section 3.2.3, we know that the probability of having an indirect key between two neighbor nodes is almost 1 for the group-based RK scheme even if there are severe memory constraints on sensor nodes. Hence, in later discussion, we will also skip the security comparison between these two schemes.

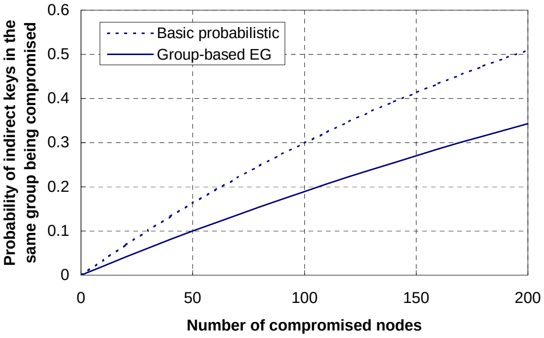

Figure 10: $p_{gci-in}(c)$ for the group-based EG scheme and the probability of an indirect key being compromised for the basic probabilistic scheme. Assume the probability of having a direct key between two neighbor nodes is 0.3.

Figure 11 shows the probability of in-group indirect keys between non-compromised nodes being compromised for the group-based PB scheme. It also includes the probability of a given indirect key (involving only one intermediate node) between two non-compromised nodes being compromised for the random subset assignment scheme in [17]. We can see that the group-based PB scheme has much better security performance than the random subset assignment scheme in terms of the compromised indirect keys between nodes in the same deployment group.

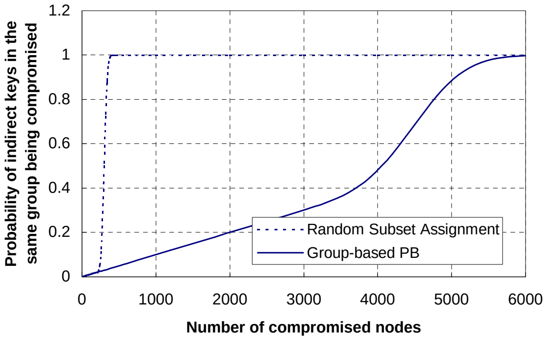

Figure 11: $p_{gci-in}(c)$ for the group-based PB scheme and the probability of an indirect key being compromised for the random subset assignment scheme. Assume the probability of having a direct key between two neighbor nodes is 0.3.

Probability of cross-group indirect keys being compromised: Though the establishment of an in-group indirect key involves one intermediate node, the establishment of an indirect key between sensor nodes in different groups may involve up to four intermediate nodes.

Assume the source node u in group G_i wants to setup an indirect key with the destination node v in group G_j. Assume the indirect key is established through a bridge $\langle u', v' \rangle$, where $u' \in G_i$ and $v' \in G_j$. Since the key established between u and v is an indirect key, we have either $u \neq u'$ or $v \neq v'$. Thus, we need to consider the following three cases:

1. *u and v share the same cross group*: The probability of this case can be estimated as $\frac{1}{m}$. In addition, we also note that $u \neq u'$ and $v \neq v'$. Thus, the probability of the path key establishment involving two intermediate nodes can be estimated as p_{dk}^2, which means that u shares a direct key with u', and v shares a direct key with v'. Similarly, the probability of the path key establishment involving three intermediate nodes can be estimated as $2(1-p_{dk})p_{dk}$, and the probability of the path key establishment involving four intermediate nodes can be estimated as $(1-p_{dk})^2$.

2. *u and v belong to different cross groups with either $u = u'$ or $v = v'$*: The probability of this case can be estimated as $\frac{m-1}{m}(1-(1-p_{dk})^2)$. Similar to the analysis in the first case, the probability of the path key establishment involving one intermediate node can be estimated as p_{dk}, and the probability of the path key establishment involving two intermediate nodes can be estimated as $1-p_{dk}$.

3. *u and v belong to different cross groups with neither $u' = u$ nor $v' = v$*: The probability of this case can be estimated as $\frac{m-1}{m}(1-p_{dk})^2$. Similar to the analysis in the first case, the probability of the path key establishment involving two intermediate nodes can be estimated as p_{dk}^2, the probability of the path key establishment involving three intermediate nodes can be estimated as $2(1-p_{dk})p_{dk}$, and the probability of the path key establishment involving four intermediate nodes can be estimated as $(1-p_{dk})^2$.

Consider an indirect key established between two sensor nodes in different deployment groups. Let p_i denote the probability of the establishment of this key involving i intermediate nodes, we have

$$\begin{cases} p_1 = & \frac{m-1}{m}[1-(1-p_{dk})^2]p_{dk} \\ p_2 = & \frac{1}{m}p_{dk}^2 + \frac{m-1}{m}[(1-(1-p_{dk})^2)(1-p_{dk}) \\ & +(1-p_{dk})^2 p_{dk}^2] \\ p_3 = & 2(1-p_{dk})p_{dk}[\frac{1}{m} + \frac{m-1}{m}(1-p_{dk})^2] \\ p_4 = & \frac{1}{m}(1-p_{dk})^2 + \frac{m-1}{m}(1-p_{dk})^2(1-p_{dk})^2 \end{cases}$$

When the path key establishment involves i intermediate nodes, the indirect key will be still secure if all of these i nodes and the related $i+1$ direct keys are not compromised. Thus, for an indirect key that involves i intermediate nodes, the probability of it being compromised can be estimated as $1-(1-p_{gcd}(c))^{i+1}(1-\frac{c}{nm-2})^i$. Hence, the probability of a cross-group indirect key between two non-compromised sensor nodes being compromised can be estimated as

$$p_{gci-cr}(c) = \sum_{i=1}^{4} p_i \times [1-(1-p_{gcd}(c))^{i+1}(1-\frac{c}{nm-2})^i].$$

Figure 12 shows the probability of a cross-group indirect key between two non-compromised sensor nodes being compromised for the group-based EG scheme. It also includes the probability of an indirect key (involving only one intermediate node) between two non-compromised nodes being compromised for the basic probabilistic scheme [10]. We can see that the security of these two scheme are very close to each other in terms of the indirect keys between sensor nodes in different deployment groups.

Figure 13 shows the probability of a cross-group indirect key between two non-compromised sensor nodes being compromised for the group-based PB scheme. It also includes the probability of an indirect key (involving only one intermediate node) between two non-compromised nodes being compromised for the random subset assignment scheme in [17]. We can still see that the group-based PB scheme has much better security performance than the random subset assignment scheme in terms of the indirect keys between nodes in different deployment groups.

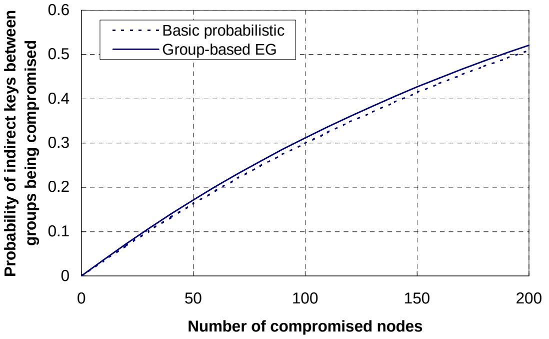

Figure 12: $p_{gci-cr}(c)$ for the group-based EG scheme and the probability of an indirect key being compromised for the basic probabilistic scheme. Assume the probability of having a direct key between two neighbor nodes is 0.3.

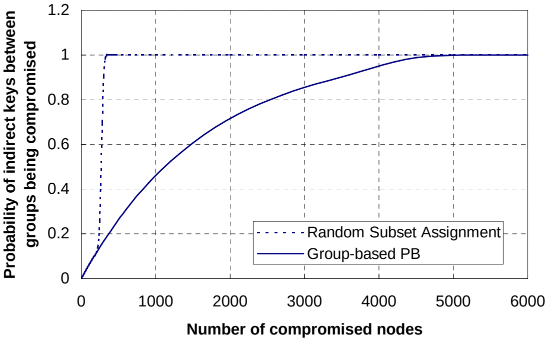

Figure 13: $p_{gci-cr}(c)$ for the group-based PB scheme and the probability of an indirect key being compromised for the random subset assignment scheme. Assume the probability of having a direct key between two neighbor nodes is 0.3.

According to the above security analysis and the performance analysis in the previous subsection, we can easily conclude that the proposed framework can significantly improve the security as well as the performance of existing key pre-distribution techniques.

4. RELATED WORK

A number of techniques have been proposed to establish pairwise keys in resource constrained sensor networks. A basic probabilistic key pre-distribution scheme was introduced in [10] and improved in [6]. The limitation of these approaches is that a small number compromised sensor nodes may affect the secure communication between a large number of non-compromised sensor nodes. A random pairwise keys scheme was proposed in [6]. Although this technique provides perfect security against node capture attacks, it cannot scale to large sensor networks. To improve the resilience of sensor networks against node compromises, two threshold-based key pre-distribution techniques were developed in [9, 17]. A cooperative protocol was developed to enhance the security of pairwise key establishments [21]. The giant component theory was used in [15] to further improve the performance and provide trade-off between connectivity, memory size and security. In this paper, we demonstrate that the performance of these key pre-distribution techniques can be further improved significantly by using our framework.

The grid-based idea was first proposed in [17] to arrange the secrets in sensor networks based on a logical grid. A similar idea was later used in PIKE [5]. However, the grids considered in these two studies are logical grids, while this paper investigates the possibility of using the locality of group deployment to improve the performance of the existing key pre-distribution techniques.

The prior deployment knowledge of sensor nodes has been used to improve the performance of many key pre-distribution protocols [8, 14, 18, 24]. The technique in this paper differs from those approaches in that it does not require the expected location information of sensor nodes, and thus greatly simplifies the deployment of sensor networks.

There are many other studies on sensor network security, including frameworks and evaluation of key management schemes [4, 25], tamper-resistant hardware [2], efficient broadcast authentication [20], secure data aggregation and in-networking processing [7, 13, 22], and vulnerabilities, attacks, and countermeasures [16, 23]. We consider them complementary to ours.

5. CONCLUSION AND FUTURE WORK

In this paper, we developed a general framework that can be used to improve the performance of any existing key pre-distribution scheme. This framework does not require any prior knowledge of sensors' expected locations, and thus greatly simplifies the deployment of sensor networks. The analysis further demonstrates that our technique can improve the security as well as the performance of existing key pre-distribution protocols substantially.

Several research directions are worth further studying, including detailed performance evaluation through simulation, and the implementation of these techniques on real sensor platforms.

Acknowledgment We would like to thank the anonymous reviewers for their valuable comments.

6. REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, 2002.

[2] S. Basagni, K. Herrin, D. Bruschi, and E. Rosti. Secure pebblenets. In *Proceedings of ACM International Symposium on Mobile ad hoc networking and computing*, pages 156–163, 2001.

[3] C. Blundo, A. De Santis, Amir Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-secure key distribution for dynamic conferences. In *Advances in Cryptology – CRYPTO '92, LNCS 740*, pages 471–486, 1993.

[4] D.W. Carman, P.S. Kruus, and B.J.Matt. Constrains and approaches for distributed sensor network security. Technical report, NAI Labs, 2000.

[5] H. Chan and A. Perrig. PIKE: Peer intermediaries for key establishment in sensor networks. In *Proceedings of IEEE Infocom*, March 2005.

[6] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Research in Security and Privacy*, pages 197–213, 2003.

[7] J. Deng, R. Han, and S. Mishra. Security support for in-network processing in wireless sensor networks. In *2003 ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN '03)*, October 2003.

[8] W. Du, J. Deng, Y. S. Han, S. Chen, and P. Varshney. A key management scheme for wireless sensor networks using deployment knowledge. In *Proceedings of IEEE INFOCOM'04*, March 2004.

[9] W. Du, J. Deng, Y. S. Han, and P. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*, October 2003.

[10] 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.

[11] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler. The nesC language: A holistic approach to networked embedded systems. In *Proceedings of Programming Language Design and Implementation (PLDI 2003)*, June 2003.

[12] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D.E. Culler, and K. S. J. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000.

[13] L. Hu and D. Evans. Secure aggregation for wireless networks. In *Workshop on Security and Assurance in Ad Hoc Networks*, January 2003.

[14] D. Huang, M. Mehta, D. Medhi, and L. Harn. Location-aware key management scheme for wireless sensor networks. In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks (SASN '04)*, pages 29 – 42, October 2004.

[15] J. Hwang and Y. Kim. Revisiting random key pre-distribution schemes for wireless sensor networks. In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks (SASN '04)*, pages 43 – 52, October 2004.

[16] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *Proceedings of 1st IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

[17] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. In *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*, pages 52–61, October 2003.

[18] D. Liu and P. Ning. Location-based pairwise key establishments for static sensor networks. In *2003 ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN '03)*, pages 72–82, October 2003.

[19] D. Niculescu and B. Nath. Ad hoc positioning system (APS). In *Proceedings of IEEE GLOBECOM '01*, 2001.

[20] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar. SPINS: Security protocols for sensor networks. In *Proceedings of Seventh Annual International Conference on Mobile Computing and Networks*, July 2001.

[21] R. D. Pietro, L. V. Mancini, and A. Mei. Random key assignment for secure wireless sensor networks. In *2003 ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN '03)*, October 2003.

[22] B. Przydatek, D. Song, and A. Perrig. SIA: Secure information aggregation in sensor networks. In *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys '03)*, Nov 2003.

[23] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 35(10):54–62, 2002.

[24] Z. Yu and Y. Guan. A key pre-distribution scheme using deployment knowledge for wireless sensor networks. In *Proceedings of ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, April 2005.

[25] S. Zhu, S. Setia, and S. Jajodia. LEAP: Efficient security mechanisms for large-scale distributed sensor networks. In *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*, pages 62–72, October 2003.