Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 490202, 17 pages
http://dx.doi.org/10.1155/2014/490202

Research Article

GenDep: Location-Aware Key Management for General Deployment of Wireless Sensor Networks

JongHyup Lee[1] and Taekyoung Kwon[2]

[1] *Department of Software, Korea National University of Transportation, Chungju 380-702, Republic of Korea*
[2] *Graduate School of Information, Yonsei University, Seoul 120-749, Republic of Korea*

Correspondence should be addressed to Taekyoung Kwon; taekyoung@yonsei.ac.kr

Received 26 October 2013; Revised 28 February 2014; Accepted 30 March 2014; Published 15 May 2014

Academic Editor: Frank Ehlers

Limited capabilities of tiny sensor node make it difficult to build a secure wireless sensor network. To cope with this problem, location-aware key predistribution schemes have been studied but without considering irregularity of real sensing fields. They were designed for ideal sensing fields divided into uniform regular polygons but false is the real sensing field, where we face irregular obstacles, undulations, and boundaries. In this paper, we tackle this problem from practical sense and first introduce the general deployment model, comparable to the previous uniform deployment model. To construct a secure WSN in the general deployment model, we present a new framework called GenDep, which consists of two phases: group placement and key management. In the group placement phase, we find an optimal position for sensor groups so that both connectivity and global coverage are satisfied for secure communications. In the key management phase, we devise a new structure called extended groups to partition the whole deployment area into irregular subareas. GenDep can build secure and flexible WSNs by using the existing key predistribution schemes even designed in the uniform deployment model. To ensure practicality of GenDep, we also conduct rigorous analysis and simulation in this paper.

1. Introduction

Wireless sensor networks (WSNs) play an important role in mission critical applications, such as military systems, healthcare services, and industrial systems. In such applications, *security* is a crucial requirement. Unfortunately, it is quite challenging to construct secure WSNs because tiny sensor nodes are very constrained in the capacity of computation, communications, storage, and power management. In particular, key management, which is the most fundamental part of security mechanisms, must be considered in the design phase of WSN with regard to the limited capabilities of tiny sensor nodes.

Since Eschenauer and Gligor have introduced the first proposal [1], a variety of approaches have been studied for secure key management based on the key predistribution concept [2–7]. Among them, location-aware key management schemes [5–7] were remarkable in their efficiency in terms of computation and communication costs (Section 2.1). Indeed, the efficiency came from the preknowledge of the location that helps sensor nodes share secret keys effectively with their (likely) adjacent nodes. However, previous approaches are unrealistic. (1) The previous deployment models assumed a sensing field to be a flat area that is dividable into uniform, regular grids. Such a *uniform deployment model* is, however, far from the real sensing field, which is actually not uniform in its geographical features and so highly inefficient. (2) The previous location-aware key management schemes are not applicable to a *general deployment model*, which covers various geographic features in the real sensing field. Their target application is so limited; nonuniform density was not allowed for placing the sensor nodes (Section 2.2).

To cope with these problems in this paper, we first present the general deployment model for WSNs and then propose a new location-aware key management framework called GenDep so as to be effective in that model. We can summarize our contribution as follows.

(i) We present the *general deployment model* (Section 3). We address that the conventional uniform models are unrealistic because geographical features are irregular in the real world. The general deployment model relaxes the constraints of sensing fields and implements the conceptual WSNs to be practical.

(ii) On the general deployment model, we construct the new framework called GenDep that actually enables practical location-aware key management. GenDep consists of two phases. First, in the group placement phase (Section 4), we find optimal positions of sensor groups by fulfilling the requirements of WSN applications and geographical features. After placing groups of sensor nodes, we partition a whole sensing field into a number of nonuniform polygons, called *zones*. Second, in the key management phase (Section 5), we can employ the existing key predistribution schemes (even designed in the uniform deployment model) for our general deployment model. To do so, we devise a new structure of *extended groups*, each of which consists of a pair of adjacent zones. The extended groups are organized for every pair of zones; thus they overlap each other and connect all neighbors. We treat every extended group as an independent group. Therefore, we can employ any key predistribution scheme for each extended group.

(iii) We evaluate GenDep via numerical analyses and simulations (Section 6). In particular, the numerical analyses handle connectivity, security, and storage overhead of sensor nodes with general probability distributions.

2. Background and Problem

2.1. Related Work

Sensor Node Placement. Previous studies on the node placement have focused on how to place a sensor node at an optimal location *individually.* Meguerdichian et al. addressed the coverage problem in WSN [8]. They calculated the maximum breach path and the maximum service path of nodes to find a position that maximizes coverage. In [9], Zou and Chakrabarty introduced a greedy algorithm called virtual force algorithm (VFA). VFA assumes targets have virtual attractive forces and obstacles have virtual repulsive forces. Sensor nodes can find their position under the influence of the forces. Zou and Chakrabarty [10] also proposed setting a virtual backbone from active nodes to improve coverage and connectivity in ad hoc networks. Takahara et al. [11] evaluated the coverage of sensors deployed in the grid pattern for uniform and normal errors. Wang et al. [12] proposed a deployment strategy when sensors are deployed in Gaussian distribution without concerns on the connectivity. Recently, interesting researches have been carried out to enhance coverage in mixed WSNs, which has the heterogeneous setting of static and mobile nodes. Ghosh and Das presented a good survey on the coverage and connectivity issues in [13]. Static sensor nodes are arranged to maximize the coverage (blanket/barrier coverage of [13]) and mobile sensor nodes move carefully to fill the coverage holes (sweep coverage). Lambrou and Panayiotou proposed collaborative scheme to improve coverage in mixed sensor networks [14, 15]. In particular, considering the tradeoff between coverage and energy consumption (communication), they pursue an autonomous sensor system to improve coverage effectively by mobile sensor nodes. Emphasizing the role of mobile sensor nodes for coverage, Das and Roy presented an algorithm to maximize coverage with continuous movement of sensor nodes in [16]. The proposed algorithm lets the mobile sensor nodes cover the sensing field fully within a fixed time. Mahboubi et al. employ Voronoi polygons in [17] to find the coverage holes. Each sensor node has its Voronoi polygon that is derived by relative position to adjacent nodes. Thus the difference between the sensing area and the Voronoi polygon can show the effectiveness of coverage. The authors proposed node-based and vertex-based strategies to relocate sensor nodes. He et al. presented curve-based deployment for barrier coverage in [18]. They modeled sensor nodes along curved lines to enhance the suboptimal coverage of line-based coverage. To sum up, the previous researches focused on placing individual sensor nodes effectively and achieved their goal. However, in large-scale WSNs, where sensors are deployed in groups, we have to take into account group placement. In that sense, GenDep focuses on placing sensor groups rather than individual sensor nodes.

Key Management in Wireless Sensor Networks. In [1], Eschenauer and Gligor proposed a random key predistribution (RKP) scheme for WSNs. A randomly selected subkey pool is preinstalled to each node before deployment. Chan et al. enhanced RKP (q-composite RKP, qRKP). qRKP achieves better resilience by restricting the minimum number of shared keys to connect. They also proposed the random-pairwise key scheme (RP) in [19], where RP selects a random pair and gives a pairwise key. In [3], Du et al. proposed a structured key-pool RKP scheme (skRKP) by adding randomness to the Blom scheme [20]. If two sensors share a key space, they can generate a key through shared key matrix. The location-aware schemes enhance key managements in WSN. They partition a whole sensing field into fixed square areas, called "grid." We classify it as *grid deployment*. Du et al. proposed the location-aware RKP, where sensors share a key pool with other sensors in the same and adjacent grids only. Liu and Ning proposed a similar approach using bivariate polynomials [6]. In the grid-group key predistribution scheme (GGD) [5], Huang et al. employ skRKP to connect nodes in the same grid but RP to connect nodes in adjacent grids. Fanian et al. applied a mixed key from polynomial-based and random key predistribution schemes to WSNs in [21]. Grid-based predeployment knowledge on location is used to preinstall the hybrid keys to sensor nodes. Our previous work improves the connectivity in key management with perfect resilience via the key offering method [22].

2.2. Problems.

Previous location-aware approaches [2, 3, 5, 6] assume the uniform deployment model, where a sensing field

is a flat area with a uniform geographical feature. In real world, however, a target sensing field is nonuniform. We hardly meet the ideal sensing field that fits to the uniform deployment model. Furthermore, the real-world field has irregular geographical features. The WSN applications also have different sensing interests. Of course we can apply the uniform deployment model to the real-world field by force but it must be extremely inefficient.

Figure 1 shows an example of the real-world sensing field. Suppose that we would like to install WSN for a plain field only (not on the mountain and lake). We cannot make an optimal design to partition the sensing field with grids. To provide a practical WSN for real-world sensing fields, we should be able to divide the sensing field into any polygons like the dotted lines.

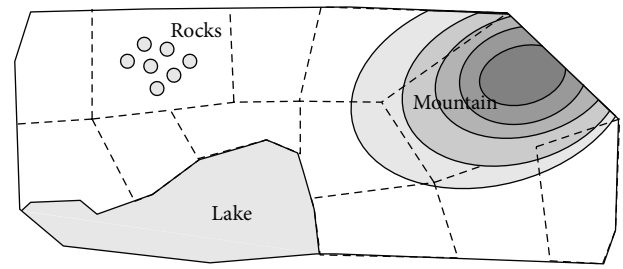


Figure 1: An example of the sensing field in the real world.

3. GenDep Overview

In this section, we overview the proposed method, GenDep, which is a novel location-aware key management scheme in the general deployment model. The detailed explanations on GenDep will be given in later sections, Sections 4 and 5.

3.1. General Deployment Model. Large-scale WSNs benefit from group-based deployment, that is, placing sensors by the unit of a group [23], for their scalability. Since sensor groups should be placed in a fixed pattern, however, the uniform deployment model is not appropriate for large-scale WSNs in the real world. To cope with this impracticality, we present the *general deployment model*. Any form of deployment area can be used for group-based deployment by dividing it into nonuniform subareas, which the existing uniform deployment model cannot support. The general deployment model relaxes the constraints and resolves this problem. Unfortunately, the previous location-aware key management schemes have been considered in the uniform model only, and so they are not quite workable in the general model. It is desirable to develop a new location-aware key management method in the general deployment model.

3.2. GenDep Framework. The new method proposed in this paper is called GenDep. In the general deployment model, GenDep divides and assigns subareas in irregular form to sensor groups and performs location-aware key management effectively.

Figure 2 illustrates the basic procedure of GenDep. Given the information of a sensing field and application requirements in the general deployment model, we firstly define the *sensing requirements*, which imply the required density of sensors at a particular point. We then apply GenDep in two phases as follows.

(i) Phase 1: in this phase, we make a group placement plan by finding optimal positions of "sensor groups." We first set reference points according to the sensing requirements and then set deployment points of sensor groups based on the reference points. Finally, we adjust the planned location and distribution of sensor groups so as to enhance the intergroup connectivity.

(ii) Phase 2: in this phase, we preinstall key sets and conduct location-based key establishment in the general deployment model. We first build group structures incorporating zones and extended groups. Second, we preinstall key sets to every sensor node so as to deploy sensor groups to the target sensing field according to the group placement plan. Finally, after the real deployment, all the sensor nodes conduct pairwise key establishment.

4. Group Placement (Phase 1)

The group-based deployment phase consists of three steps: (1) setup reference points, (2) setup deployment points, and (3) adjust distributions. Notations section summarizes the notations to be used in this paper.

4.1. Step 1: Setup Reference Points. In this step, we set reference points according to the sensing requirements. First of all, we represent a target sensing field as a two-dimensional Cartesian plane and place reference points (x, y), as illustrated in Figure 3(a). The reference points are the points that hold sensing requirements at their position and are placed by following the uniform distribution. Note that they are not actual deployment points but only used for deriving local sensing requirements based on the prior knowledge about the terrain and WSN application.

From [9, 10, 12, 24], we generalize the sensing requirements for the general deployment model. It is required that an event taking place at a certain reference point (x, y) should be detected by at least $b(x, y)$ numbers of sensor nodes with the probability of $p_d(x, y)$. Hence, the sensing requirements are defined at a specific point (x, y) and parametrized as follows:

(i) $p_d(x, y)$: detection probability of a sensor desired at (x, y),

(ii) $b(x, y)$: number of sensors required to detect an event at (x, y).

We then combine these parameters to the sensor density desired at (x, y), $U(x, y)$. First we compute the area of the sensing region where a sensor node can detect an event with higher probability than $p_d(x, y)$. We assume probabilistic sensing model, in which the detection probability decays exponentially as the power of distance. When we suppose d is the distance of a sensor from an event, $e^{-\alpha d}$ would be

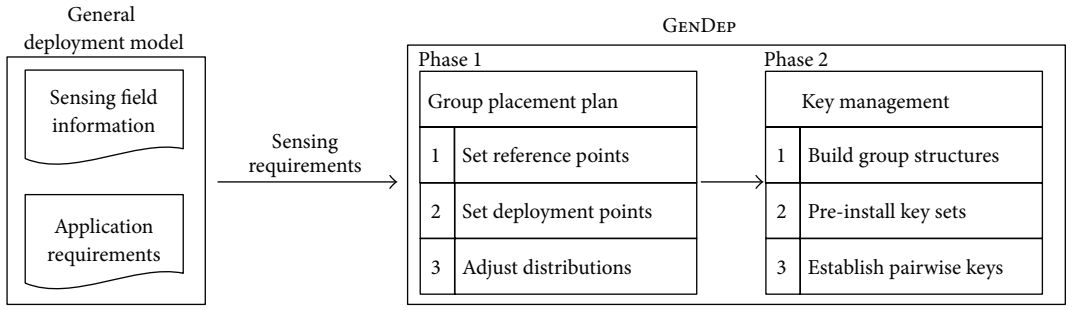
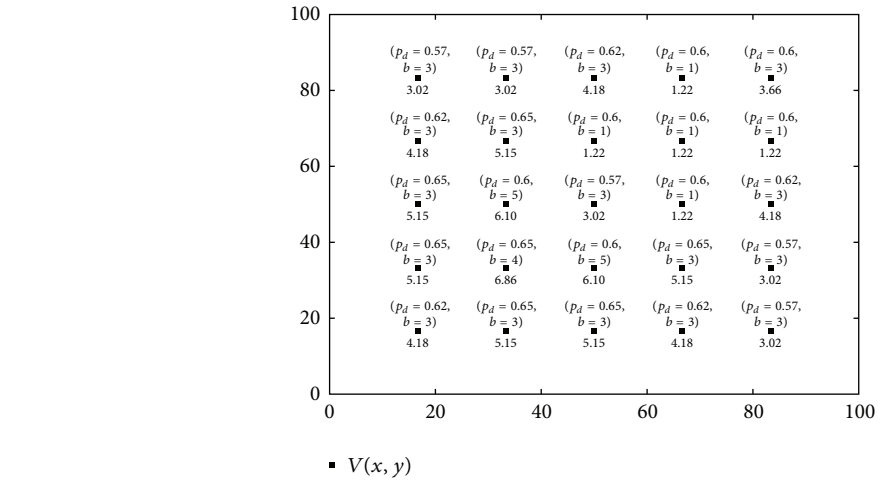
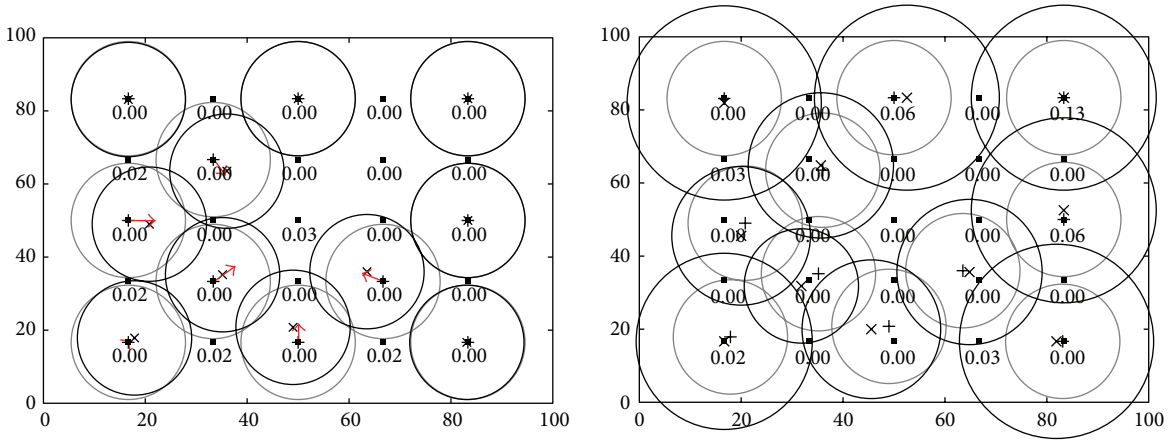


Figure 2: Procedure of GenDep.



(a) Setup $V(x, y)$ at each reference point



(b) Setup deployment points over the reference points ($\delta_d = 0.1$ m and $T_V = 0.001$)

(c) Adjust distributions and deployment points ($C = 1.5$, $N_D = 11$, $p_1^* = 0.33$, $T_V = 0.02$, and $\delta_s = 0.05$)

Figure 3: Three steps of the group placement scheme. (All sensor groups are modeled as the Gaussian distribution with $\sigma = 10.0$.)

Input: A set $Q = \{G(i) \mid 1 \geq i \geq N_D\}$ of sensor groups and a set
$W = \{\forall ((x, y), V(x, y))\}$ of tuples of a reference point (x, y) and its $V(x, y)$ for every reference point
Output: A set $B = \{[x, y]_i \mid 1 \geq i \geq N_D\}$ of deployment points assigned to sensor groups
$B \leftarrow \emptyset$
for $G(i) \in Q$ **do**
Find a reference point (x, y) of highest $V(x, y)$ in W.
$[x, y]_i \leftarrow (x, y)$.
Add $[x, y]_i$ to B.
Update $V(x, y)$s in W.
end
return B

Algorithm 1: Initial placement: initial placement for unassigned sensor groups.

the detection probability of the sensor, where α is a positive parameter that represents the physical characteristics of the sensor node and it is determined by the sensor type [9, 12]. Hence a sensor can detect an event taking place within $-(\ln p_d(x, y)/\alpha)$ with higher probability than $p_d(x, y)$ and the area of the sensing region is $\pi\{-(\ln p_d(x, y)/\alpha)\}^2$ when we assume sensors are omnidirectional. In other words, an event at (x, y) can be detected by sensors located in the sensing region of $\pi\{-(\ln p_d(x, y)/\alpha)\}^2$ with the probability of $p_d(x, y)$. $U(x, y)$ also requires at least $b(x, y)$ number of sensors to be located in the sensing region. Thus,

$$U(x, y) = \frac{b(x, y)}{\pi\{\ln p_d(x, y)/\alpha\}^2}. \quad (1)$$

However, $U(x, y)$ is a goal of density to achieve when deploying sensor nodes. When placing sensor nodes as a group, it is likely that there exists a gap between the goal $U(x, y)$ and the actual density of sensor nodes at (x, y). We represent the gap, $V(x, y)$, as follows:

$$V(x, y) = \max\left(U(x, y) - \sum_{i=1}^{N_D} \frac{N_i(x, y, -(\ln p_d(x, y)/\alpha))}{\pi\{-(\ln p_d(x, y)/\alpha)\}^2}, 0\right), \quad (2)$$

where N_D is the total number of sensor groups and $(N_i(x, y, -(\ln p_d(x, y)/\alpha)))/(\pi\{-(\ln p_d(x, y)/\alpha)\}^2)$ is the density of sensor nodes belonging to ith sensor group, $G(i)$ at (x, y). Let us defer the explanation of $N_i(\cdot)$ to Section 6.2.4.

Figure 3(a) shows an initial state of $V(x, y)$ when the target sensing field is covered by 25 reference points. With $\alpha = 1$, $U(x, y)$ is derived from the sensing requirements given to the reference point (x, y), that is, $p_d(x, y)$ and $b(x, y)$. Since no sensor groups are deployed yet, $V(x, y)$ is the same as $U(x, y)$ for now.

4.2. Step 2: Setup Deployment Points. The goal of this step is to set deployment points of sensor groups on the positions that would minimize the sum of all $V(x, y)$, that is, $\min \sum_{\forall(x,y)} V(x, y)$. We first perform initial placement and then adjust the position of the deployment points.

Initial Placement. We conduct the initial deployment as shown in Algorithm 1. Algorithm 1 finds a reference point of highest $V(x, y)$ and assigns an unassigned sensor group at the point iteratively. Since $V(x, y)$ represents the gap between $U(x, y)$ and the sensor density of assigned sensor groups, the value of $V(x, y)$ keeps being updated during the process. As a result, we obtain the set B of assigned deployment points of sensor groups.

We will improve the result of initial placement in the following steps. We first adjust positions in B to minimize the sum of $V(x, y)$. And then we determine an *optimal position* (in Step 3) with regard to the intergroup connectivity.

Position Adjustment. We present a new algorithm, pVFA, which adjusts the assigned deployment position of sensor groups to find more suitable position for achieving the goal. pVFA is an iterative, subgradient method. In an iteration of pVFA, every sensor group calculates a gradient that would decrease the sum of all $V(x, y)$. The gradient is represented as an Euclidean vector (direction and magnitude) in the target sensing field. Sensor groups move along their gradients by a unit step at the end of an iteration.

More specifically, Algorithm 2 explains the detailed process. Let $\vec{F}^A_{G(i)}(x, y)$ be attractive force on a group $G(i)$ from a reference point (x, y). A reference point attracts sensor groups with the power of $V(x, y)$; that is, $|\vec{F}^A_{G(i)}(x, y)| = V(x, y)$. The gradient of $G(i)$, $\nabla_{G(i)}$, is the sum of attractive power from all the reference points, $\nabla_{G(i)} = \sum_{\forall(x,y)} \vec{F}^A_{G(i)}(x, y)$. A sensor group, $G(i)$, updates its assigned deployment point as $[x + \delta_d \cdot \nabla_{G(i)}|_x, y + \delta_d \cdot \nabla_{G(i)}|_y]_i$ by applying its gradient, where δ_d is a unit step for distance. The iteration is repeated until the average of gradients goes below a threshold T_V.

Figure 3(b) shows the results of this step. The circles indicate the effective region of sensor groups (which we will describe in the next step). The initial deployment determines the gray circles while the black circles indicate the position of sensor groups after the position adjustment. We show the gradient $\nabla_{G(i)}$ as a vector in arrow. As a result, we could obtain

Input: A set $B = \{[x, y]_i \mid 1 \geq i \geq N_D\}$ of deployment points assigned to sensor groups
Output: A set $B' = \{[x, y]_i \mid 1 \geq i \geq N_D\}$ of deployment points after position adjustment
repeat
  for $G(i) \in Q$ **do**
    Calculate an attractive force, $\vec{F}^A_{G(i)}(x, y)$, from $V(x, y)$.
    $\left(\vec{F}^A_{G(i)}(x, y)|_x = |V(x, y)| \cos\theta,\ \vec{F}^A_{G(i)}(x, y)|_y = |V(x, y)| \sin\theta\right)$
    Derive the gradient of $G(i)$: $\nabla_{G(i)} = \sum_{\forall(x,y)} \vec{F}^A_{G(i)}(x, y)$.
    Update its position as $[x + \delta_d \cdot \nabla_{G(i)}|_x, y + \delta_d \cdot \nabla_{G(i)}|_y]_i$ of B
    Update $V(x, y)$s in W.
  end
  Copy B to B'.
until $T_V > \dfrac{\sum_{i=1}^{N_D} |\nabla_{G(i)}|}{N_D}$
return B'

Algorithm 2: Position adjustment: adjustment of initial position of sensor groups.

the intermediate deployment points of sensor groups and we will finalize the optimal deployment points in the next step.

4.3. Step 3: Adjust Distributions. The result of the previous step could have satisfied the sensing requirements (initially given by the general deployment model) but may have isolated groups having no *well-connected* adjacent groups because of missing the intergroup connectivity. To avoid such a case, we adjust the assigned deployment point and distribution of sensor groups before accomplishing the group placement plan.

The requirement for achieving "well-connectedness" in the intergroup connectivity is to have an effective, reasonable number of interconnections for two adjacent sensor groups. Thus, we count interconnections between sensor nodes, both of which are effective members in their own sensor groups only. To do so, we first define an *effective region* of a sensor group as the area covered by an effective number (we set the effective number as 80 percent of the number of sensor nodes in a group) of sensor nodes in that group. And we then set the minimum number of interconnections for two sensor groups to become a connected component.

Effective Region and Interconnections. We model sensor nodes deployed in a sensor group by omnidirectional probabilistic distributions, for example, Gaussian distribution. Hence the effective region is a circular region whose radius we calculate from the inverse cumulative distribution (CDF) [25] of the probabilistic distribution. CDF of the distribution of a sensor group returns the portion of member nodes for a given region of the sensor group; thus the inverse CDF provides an area (or a radius) that holds the given portion of sensor nodes. For example, we calculate the radius of the effective region as $P^{-1}(0.8)$, where $P^{-1}()$ is the inverse CDF. We then regard interconnections of sensor nodes located in intersection of the effective regions of two adjacent sensor groups as effective interconnections. Let w_i and w_j, respectively, denote the number of sensor nodes from $G(i)$ and $G(j)$ in the intersection region between $G(i)$ and $G(j)$.

Minimum Number of Interconnections. To become a connected component, a sensor group must have such a minimum degree of graph as $\lceil C \log n_z \rceil$, where C is a tunable radio transmission range parameter ($C \geq 1.5$ will increase the global connectedness to be close to 1) [26] and n_z is the number of sensor nodes in a sensor group. Similarly, two sensor groups must have a minimum degree of graph, $\lceil C \log 2n_z \rceil$. Therefore, we define a set of well-connected neighbor groups, $\widehat{N}_i$, for group $G(i)$:

$$\widehat{N}_i = \left\{ j \mid p_1^* \cdot w_j \geq \lceil C \log 2n_z \rceil,\ \forall j, i \neq j \right\}, \tag{3}$$

where p_1^* is the probability that two nodes of different groups can be connected. It indicates that two nodes are within each other's communication range and also share keys because only sensor nodes sharing the same key can communicate with each other [5]. Finally, to avoid the case that isolated groups exist, the following condition should be satisfied:

$$\left|\widehat{N}_i\right| \geq \lceil C \log N_D \rceil. \tag{4}$$

pVFA with an Extending Option. We improve the intergroup connectivity of the isolated sensor groups by extending their effective regions. For this purpose, we employ pVFA with an *extending option*. Algorithm 3 shows the process of this step. First we compute the effective region, denoted by F_i, of a group $G(i)$ and calculate intergroup connections between every pair of neighboring sensor groups, which have overlapped effective region in common. We then identify the isolated groups by using the condition of (4). When extending the effective region, we control the parameter of probability distribution of the sensor group. Here we suppose that we model the probability distribution as the two-dimensional Gaussian distribution. Thus we increase the standard deviation (std), σ_i, of $G(i)$ by multiplying δ_s, which is a unit step for std. During extending the effective region, we can improve the connectivity but we could also experience the lower density of sensor nodes in the sensor group at the same time. It may bring back $V(x, y)$. Thus pVFA also adjusts the position by applying the algorithm PositionAdjustment of

Input: A set $B' = \{[x, y]_i \mid 1 \geq i \geq N_D\}$ of deployment points after position adjustment
Output: Deployment plan: a set $K = \{([x, y]_i, \sigma_i) \mid 1 \geq i \geq N_D\}$ of deployment points and distribution for each sensor group
$I \leftarrow \emptyset$
repeat
  foreach $G(i)$ **do** Compute the effective region of $G(i)$, F_i
  for $\forall(G(i), G(j))$, $F_i \cup F_j \neq \emptyset$ **do**
    Calculate inter-group connections between $G(i)$ and $G(j)$
  end
  Identify isolated groups as I
  for $G(i) \in I$ **do**
    $\sigma_i \leftarrow \delta_s \cdot \sigma_i$
  end
  Apply PositionAdjustment
until $I \neq \emptyset$ **and** $T_V > \dfrac{\sum_{i=1}^{N_D} |\nabla_{G(i)}|}{N_D}$
return K

Algorithm 3: Final adjustment: adjustment of position and distribution for intergroup connectivity.

Algorithm 2. This process is repeated until there is no isolated group and the average magnitude of the gradient is below T_V. The result is a deployment plan for a given sensing field. The deployment plan indicates the position of sensor groups (i.e., deployment points) and its distribution parameter (i.e., std).

In Figure 3(c), the gray circle and the black circle indicate the effective region, respectively, before and after the adjustment. We could observe that isolated groups have extended their effective regions while several groups have moved to minimize $V(x, y)$ in Figure 3(c). Compared with the result of Step 2, the fraction of connected groups has been significantly improved from 0.455 to 1.0 at the cost of $V(x, y)$, of which the average increased from 0.0043 to 0.0132.

5. Key Management (Phase 2)

In this section, we describe the key management phase of GenDep. On the basis of the result of the group placement phase, we preinstall key sets to sensors for establishing pairwise keys in the key management phase. To do so, we describe building group structures in Step 1, preinstalling key sets to sensors in Step 2, and establishing pairwise keys in Step 3 (after deployment).

5.1. Step 1: Build Group Structures. In order to provide the location awareness to the key management schemes, we need to define a zone for a group. The key management schemes assume zones are nonoverlapped; thus we divide the sensing field into zones by using the Voronoi decomposition method [27], where a Voronoi cell for a seed point is a region consisting of points that are closer to the seed point than other seed points.

By employing the deployment points of groups as the seed points, we define a zone for the group $G(i)$ is the Voronoi cell based on its deployment point, $[x, y]_i$. Let NV_i denote a set of groups sharing Voronoi edges with the group $G(i)$. Then we could define a set of neighbor groups, $NZ(i)$, as follows:

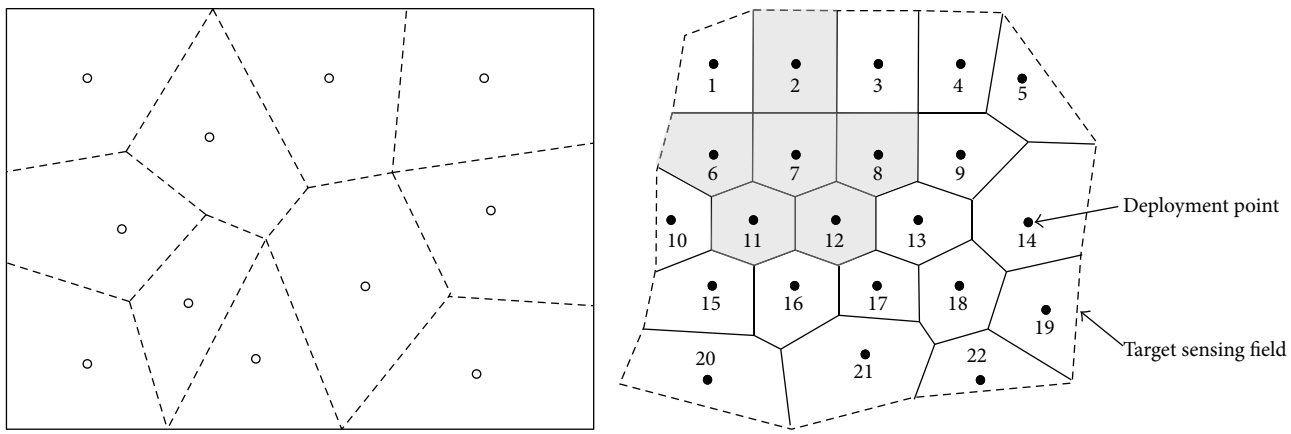
$$NZ(i) = NV_i \cap \widehat{N}_i. \tag{5}$$

Figure 4(a) is an example of the zones based on the deployment points of Figure 3. The sensing field was partitioned into 11 zones. Every zone has different shapes and different neighbors. In order to explain the subsequent process more clearly, we use the field of Figure 4(b) that consists of various polygons, that is, irregular zones.

After partitioning zones, every pair of two neighboring zone groups is merged to be an *extended group*, EG. A group belongs to as many EGs as the number of its neighbor groups.

Algorithm 4 explains the process for setting the neighbor sensor group and assigning extended groups. All the assigned EGs are in E. In addition, Figure 5 illustrates an example of multiple EGs (i.e., 5 EGs) overlapped on the group $G(7)$. Note that EGs could intrinsically support location awareness because of binding a pair of adjacent zones.

5.2. Step 2: Preinstall Key Sets. In this step, we assign key predistribution schemes to EGs and preinstall key sets to be used for key establishment. Before preinstall keys to sensor nodes, we select a key predistribution scheme suitable for each EG. Because each EG is an independent space, we could assign not only homogeneous but also heterogeneous key predistribution schemes to EGs. It is clear that heterogeneous schemes could only affect their own EGs. For example, a compromised sensor node may only contaminate sensors in its local EG. When selecting key predistribution schemes, we could flexibly approach them by considering their distinct features in terms of node-capture resilience, storage overhead, and computational complexity. We could prioritize one of those features in each EG. For instance, we could select RP for prioritizing the perfect resilience against



(a) Voronoi decomposition for Figure 3

(b) Deployment plan example

Figure 4: Voronoi decomposition and deployment plan.

Input: A set Q of sensor groups, a set N_i of well-connected sensor groups of $G(i)$, and a set NV_i of the sensor groups sharing a Voronoi edge with $G(i)$
Output: A set $E = \{\mathsf{EG}(i, j) \mid G(j) \in NZ(i) \vee G(i) \in NZ(j)\}$ of extended groups.
$E \leftarrow \emptyset$
foreach $G(i)$ **do**
  $NZ(i) \leftarrow NV_i \cap \widehat{N}_i$
  for $G(j) \in NZ(i)$ **do**
    if $\mathsf{EG}(i, j) \notin W$ **then**
      $E \leftarrow E \cup \{\mathsf{EG}(i \cdot j)\}$
    end
  end
end
return E

Algorithm 4: Build structure: define the neighbor groups and build a set of extended groups of every pair of neighboring sensor groups.

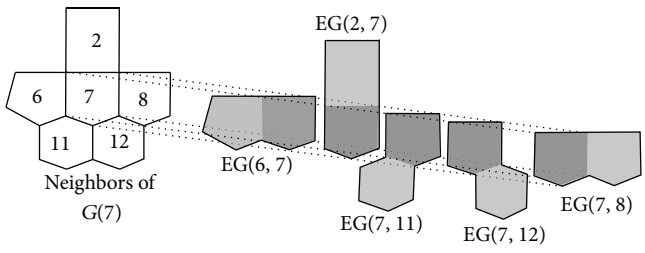


Figure 5: Group structures for group $G(7)$. (Note that $\mathsf{EG}(i, j)$ denotes the extended group of $G(i)$ and $G(j)$.)

node capture attacks or RKP for higher connectivity. No key predistribution scheme is a panacea.

According to the selected key predistribution schemes, we preinstall key sets to sensor nodes. Algorithm 5 shows the procedure. First we assign unique identifiers, ID_u and $\mathrm{ID}_{\mathsf{EG}}$, to each sensor node and each EG, respectively. For each EG, we generate a key pool, $\mathscr{K}_{\mathsf{EG}(i,j)}$, according to the selected key predistribution scheme for the EG. From the key pool, we select a key set and install it on a sensor node of the EG. The method for generating a key pool `GenerateKeyPool` and selecting a key set from the key pool `SelectKeysFrom` is dependent on the selected key predistribution scheme.

5.3. Step 3: Establish Pairwise Keys. Finally, we could deploy sensor nodes into the sensing field. After deployment, the sensor nodes will then establish pairwise keys by themselves. The procedure to establish pairwise keys is as follows.

(1) *Key Discovery*. Each sensor node broadcasts its ID, ID_u, and key list whereas the key list is composed of the identifiers of its keys, $\langle \mathrm{ID}_k, \mathrm{ID}_{\mathsf{EG}} \rangle$. Sensor nodes could exchange them with their neighbors and compare the lists for key establishment.

(2) *Pairwise Key Establishment*. If a sensor node finds a neighbor node sharing the same key, then it sends a connection request to the neighbor. By using the key agreement method of [3], the two adjacent nodes can generate a pairwise key in a secure way.

(3) *Path Key Establishment*. If there remain neighbors not yet connected, the sensor node broadcasts the IDs of the disconnected neighbors to the connected

Input: A set E of extended groups
foreach a sensor node, u **do**
  `SetNewIdentifier` (u, ID_u)
end
foreach $\mathsf{EG}(i, j)$ **do**
  `SetNewIdentifier` $(\mathsf{EG}(i, j), ID_{\mathsf{EG}})$
  $\mathscr{K}_{\mathsf{EG}(i,j)} \leftarrow$ `GenerateKeyPool`$(\mathsf{EG}(i, j))$
  for a sensor node, $u \in \mathsf{EG}(i, j)$ **do**
    $s \leftarrow$ `SelectKeysFrom`$(\mathscr{K}_{\mathsf{EG}(i,j)})$
    `InstallKeySet`(u, s)
  end
end

Algorithm 5: Preinstall: preinstall a key set for each sensor node.

neighbors, so that the connected neighbors could help them to establish pairwise keys. This step is repeated until the sensor node connects to all the neighbors and cannot find newly connectable nodes.

6. Performance Evaluation

In this section, we analyze the performance of GenDep. We compare our method to the previous location-aware key management schemes based on the fixed deployment model, such as `GGD` [5] and `skRKP` [2].

6.1. System Model for Evaluation. We assume the following.

(i) The number of sensor nodes (n_z) is 100 in a single group.

(ii) To compare the result with the previous schemes fair, we set each zone to be formed as a square: $\forall G(i)$, $L_{G(i)} = 4$.

(iii) GenDep uses three configurations for key predistribution schemes: `G-skRKP`, `G-COMB`, and `G-RP`. For `G-skRKP` and `G-RP`, we select `skRKP` and `RP` as key predistribution schemes, respectively. In `G-COMB`, a combination of `RKP` and `RP` is applied; among 4 EGs of a group, 2 EGs use `RKP` and the other 2 EGs employ `RP`. The key pool size of the `RKP` scheme is 4880 to keep the faction of communication compromised under 0.4 until 50% nodes are compromised [1].

(iv) In order to keep the same level of key storage and to preserve the λ-secure property (for resilience against compromised sensor nodes, the key distribution schemes using the secret key matrix of the Blom scheme, such as `GGD` and `skRKP` [2, 5], have an upper bound on the ratio of the number of sensor nodes and the size of secret matrix; this is called *λ-secure property*; please refer to [3] for the λ-secure property) [3], $\omega = 7$ for all the schemes but $\tau_G = 3$, $\tau_D = 2$, and $\tau_E = 1$ for `GGD`, `skRKP-D`, and `G-skRKP`, respectively. The analysis of key storage is described in Section 6.5.

Additionally, we set the parameters for `GGD` as $\gamma_{\mathtt{GGD}} = 8$ and $\alpha_{\mathtt{GGD}} = 1$ based on [5].

6.2. Key Graph Connectivity. Let $p_{1n}(i, j)$ be the probability that two sensor nodes in $G(i)$ and $G(j)$ share at least one common key. It is assumed that a sensor node of $\mathsf{EG}(i, j)$ randomly picks m_k different keys for $\mathsf{EG}(i, j)$ from a key pool of size M. Then, the $p_{1n}(i, j)$ is

$$\begin{aligned} p_{1n}(i, j) &= 1 - \frac{\binom{M}{m_k}\binom{M-m_k}{m_k}}{\binom{M}{m_k}^2} \\ &= 1 - \frac{\{(M - m_k)!\}^2}{M! \cdot (M - 2m_k)}. \end{aligned} \tag{6}$$

Likewise, let $p_{1s}(i)$ be the probability that two sensor nodes in the same $G(i)$ share at least one common key. The sensor nodes in the same group have all their EGs in common. They can be connected unless they do not share any key in their EGs, which are $\mathsf{EG}(i, j), G(j) \in NZ(i)$. For each $\mathsf{EG}(i, j)$ that they share, the key graph connecting probability is $p_{1n}(i, j)$. Thus,

$$p_{1s}(i) = 1 - \prod_{G(j)}^{NZ(i)} \left(1 - p_{1n}(i, j)\right). \tag{7}$$

If a single key predistribution scheme is applied to all EGs of $G(i)$, $p_{1s}(i)$ is calculated as

$$\begin{aligned} p_{1s}(i) &= 1 - \left[\frac{\binom{M}{m_k}\binom{M-m_k}{m_k}}{\binom{M}{m_k}^2}\right]^{L_{G(i)}} \\ &= 1 - \left[\frac{\{(M - m_k)!\}^2}{M!\,(M - 2m_k)!}\right]^{L_{G(i)}}, \end{aligned} \tag{8}$$

since the number of shared EGs of the two sensor nodes in the same group is $L_{G(i)}$. Appendix A shows the probabilities, p_{1n} and p_{1s}, on the instantiations of key predistribution schemes in GenDep.

A sensor node can be connected to neighbors directly (1 hop) or through relaying of other neighbor nodes (more than 1 hop). For simplicity and clarity, we consider 2 hops at maximum. Let $N_i(x, y, R)$ be the number of sensors of $G(i)$ within the communication range R from a position (x, y). We assume a sensor node u in $G(i)$ at (x, y). For the sensor node u, we take $n_i' = N_i(x, y, R) - 1$ as the number of reachable neighbor nodes in $G(i)$ (the number of sensor nodes within the communication range except itself) and $n_j' = N_j(x, y, R)$ as the number of reachable neighbor nodes in $G(j)$, where $j \neq i$ and $G(j) \in NZ(i)$.

6.2.1. Key Graph Connectivity within the Same Group. Suppose that a sensor node u connects to another sensor node v in the same group $G(i)$, $u, v \in G(i)$. We represent the probability that two nodes are connected, $P_{u,v}$, as

$$P_{u,v} = P_{u,v}[1 \text{ hop}] + (1 - P_{u,v}[1 \text{ hop}])\, P_{u,v}[2 \text{ hops}], \quad (9)$$

where the probability of the 1-hop connection, $P_{u,v}[1 \text{ hop}]$, is equal to $p_{1s}(i)$. When we calculate the probability of the 2-hop connection, $P_{u,v}[2 \text{ hops}]$, we take account of two cases of relaying: by a node in the same group or another node in a neighbor group. Let us denote by P_i and P_j the probabilities that the sensor nodes, u and v, are connected via a node in $G(i)$ and $G(j) \in NZ(i)$, respectively. The two nodes can be connected in any of these cases. Hence,

$$P_{u,v}[2 \text{ hops}] = 1 - (1 - P_i) \cdot \prod_{G(j)}^{NZ(i)} (1 - P_j). \quad (10)$$

We calculate P_i and P_j by developing the binomial probability distribution (the development of the binomial probability distribution for the key graph connecting probability is inspired by [28]). We compute P_i as follows:

$$P_i = \sum_{k_1=1}^{n_i'-1} \binom{n_i' - 1}{k_1} \{p_{1s}(i)\}^{k_1} \times \{1 - p_{1s}(i)\}^{n_i'-k_1-1} \left\{1 - (1 - p_c p_{1s}(i))^{k_1}\right\}, \quad (11)$$

where the part $\sum_{k_1=1}^{n_i'-1} \binom{n_i'-1}{k_1} \{p_{1s}(i)\}^{k_1} \{1 - p_{1s}(i)\}^{n_i'-k_1-1}$ is the binomial probability distribution that represents that u is connected to k_1 sensor nodes out of $n_i' - 1$ neighbors (the number of reachable neighbor nodes except v) and u is connected to one of k_1 sensor nodes by $p_{1s}(i)$. The term, $1 - (1 - p_c p_{1s}(i))^{k_1}$, represents the probability that at least one of the k_1 sensor nodes is connected to v. Note that p_c is the probability that any two sensor nodes are within the communication range of each other. If the communication range is circle-shaped and of equal radius, then p_c is 0.5865 regardless of R according to [28]. Likewise, we calculate P_j as

$$P_j = \sum_{k_2=1}^{n_j'} \binom{n_j'}{k_2} \{p_{1n}(i, j)\}^{k_2} \times \{1 - p_{1n}(i, j)\}^{n_j'-k_2} \left\{1 - (1 - p_c p_{1n}(i, j))^{k_2}\right\}. \quad (12)$$

Equation (12) represents that u is connected to v via one of k_2 sensor nodes in $G(j) \in NZ(i)$. Since the k_2 sensor nodes belong to $G(j)$, the connecting probability between them and u or v is $p_{1n}(i, j)$. Finally, we get (13) by joining (9), (10), (11), and (12).

$$\begin{aligned} P_{u,v} &= p_{1s}(i) + (1 - p_{1s}(i)) \\ &\quad \times \left\{1 - \left[1 - \sum_{k_1=1}^{n_i'-1} \binom{n_i' - 1}{k_1} \{p_{1s}(i)\}^{k_1} \times \{1 - p_{1s}(i)\}^{n_i'-k_1-1} \times \left\{1 - (1 - p_c p_{1s}(i))^{k_1}\right\}\right]\right. \\ &\quad \left. \cdot \prod_{j}^{NZ(i)} \left[1 - \sum_{k_2=1}^{n_j'} \binom{n_j'}{k_2} \{p_{1n}(i, j)\}^{k_2} \times \{1 - p_{1n}(i, j)\}^{n_j'-k_2} \times \left\{1 - (1 - p_c p_{1n}(i, j))^{k_2}\right\}\right]\right\}. \end{aligned} \quad (13)$$

6.2.2. Key Graph Connectivity between Neighbor Groups. P_{u,v^*} is the probability that a sensor node u in $G(i)$ connects to another node v in $G(j)$, that is, a neighbor group of $G(i)$ ($G(j) \in NZ(i)$) with or without help from all neighbor nodes. Obviously, $P_{u,v^*}[1 \text{ hop}] = p_{1n}(i, j)$, but, in $P_{u,v^*}[2 \text{ hops}]$, three cases of 2-hop connections are possible: P_i^*, P_j^*, and P_g^*.

First, P_i^*, the probability of 2-hop connection by relaying of a sensor node in $G(i)$ is

$$P_i^* = \sum_{k_1=1}^{n_i'} \binom{n_i'}{k_1} \{p_{1s}(i)\}^{k_1} \times \{1 - p_{1s}(i)\}^{n_i'-k_1} \left\{1 - (1 - p_c p_{1n}(i, j))^{k_1}\right\}. \quad (14)$$

In (14), note that $p_{1s}(i)$ and $p_{1n}(i, j)$, respectively, represent the probability that node u connects to k_1 neighbor nodes and the probability that k_1 neighbor nodes connect to node v. Second, P_j^* is the probability of 2-hop connection by relaying of a sensor node in $G(j)$ and is given as

$$P_j^* = \sum_{k_2=1}^{n_j'-1} \binom{n_j' - 1}{k_1} \{p_{1n}(i, j)\}^{k_2} \times \{1 - p_{1n}(i, j)\}^{n_j'-k_2-1} \times \left\{1 - (1 - p_c p_{1s}(j))^{k_2}\right\}. \quad (15)$$

Third, P_g^* is the probability of 2-hop connection by the relay of a sensor node in neighbor groups except $G(j)$, which are $NZ(i) \setminus \{G(j)\}$. Hence, P_g^* is

$$P_g^* = \sum_{k_3=1}^{n_g'} \binom{n_g'}{k_3} \{p_{1n}(i,g)\}^{k_3} \times \{1 - p_{1n}(i,g)\}^{n_g'-k_3} \times \left\{1 - (1 - p_c p_{1n}(i,g))^{k_3}\right\}. \tag{16}$$

From (14), (15), and (16), we derive that

$$P_{u,v}[2 \text{ hops}] = 1 - (1 - P_i^*)(1 - P_j^*) \cdot \prod_{g}^{NZ(i)\setminus\{G(j)\}} (1 - P_g^*). \tag{17}$$

Finally,

$$P_{u,v^*} = P_{u,v^*}[1 \text{ hop}] + (1 - P_{u,v^*}[1 \text{ hop}]) P_{u,v^*}[2 \text{ hops}]. \tag{18}$$

Joining (14), (15), (16), and (17), we represent P_{u,v^*} as (19).

$$\begin{aligned} P_{u,v^*} &= p_{1n}(i,j) + (1 - p_{1n}(i,j)) \\ &\cdot \left\{1 - \left[1 - \sum_{k_1=1}^{n_i'} \binom{n_i'}{k_1} \{p_{1s}(i)\}^{k_1} \times \{1 - p_{1s}(i)\}^{n_i'-k_1} \times \left\{1 - (1 - p_c p_{1n}(i,j))^{k_1}\right\}\right] \right. \\ &\cdot \left[1 - \sum_{k_2=1}^{n_j'-1} \binom{n_j'-1}{k_1} \{p_{1n}(i,j)\}^{k_2} \times \{1 - p_{1n}(i,j)\}^{n_j'-k_2-1} \times \left\{1 - (1 - p_c p_{1s}(j))^{k_2}\right\}\right] \\ &\cdot \prod_{g}^{NZ(i)\setminus\{G(j)\}} \left[1 - \sum_{k_3=1}^{n_g'} \binom{n_g'}{k_3} () \{p_{1n}(i,g)\}^{k_3} \times \{1 - p_{1n}(i,g)\}^{n_g'-k_3} \times \left\{1 - (1 - p_c p_{1n}(i,g))^{k_3}\right\}\right] \left.\right\}. \end{aligned} \tag{19}$$

Table 1: Local connectivity (p_{1s} and p_{1n}).

	p_{1s}	p_{1n}
GGD	0.8857	0.01
skRKP-D	0.5338	0.0816
G-skRKP	0.4602	0.1429
G-RP	0.4779	0.15
G-COMB	0.62	0.33 (RKP), 0.08 (RP)

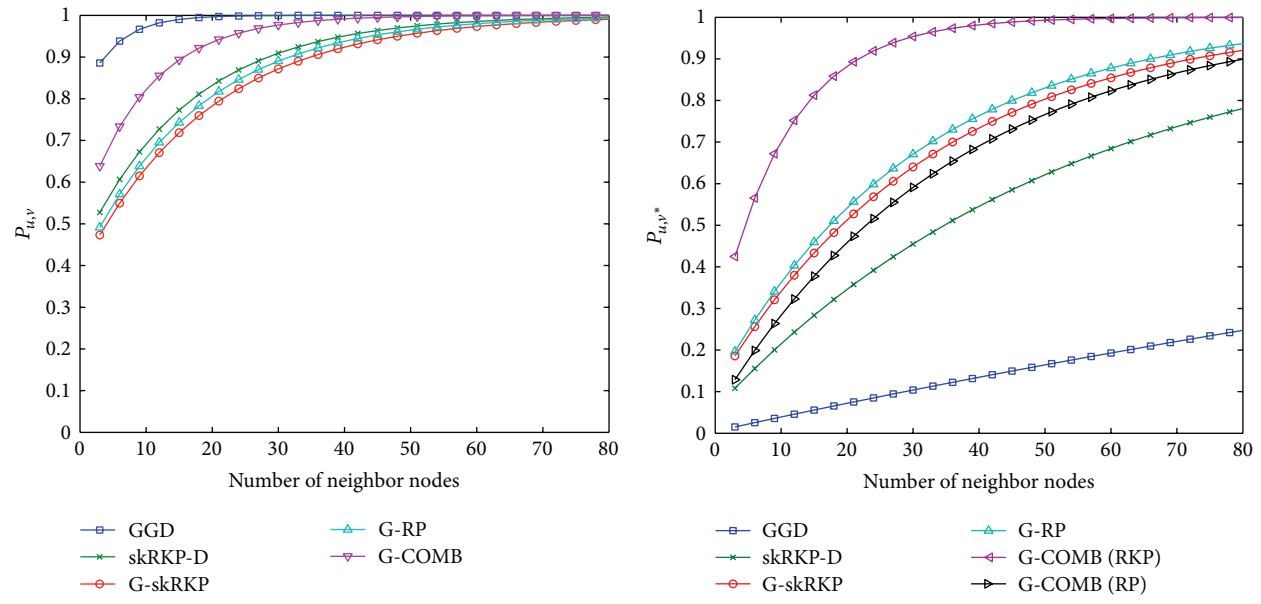
6.2.3. Numerical Results in Key Graph Connectivity. Figure 6 depicts $P_{u,v}$ and P_{u,v^*} when n_i' and n_j' are the same. Table 1 shows the parameters, p_{1s} and p_{1n}, computed by the equations as referred to Appendix A. As illustrated in Figure 6, the connectivity of GGD is best at the same group but worst between the neighbor groups at the same time. The reason is that GGD employs different methods for the same and the neighbor groups. The connectivities at the same group are similar in G-skRKP,G-RP, and skRKP-D. But the connectivities between the neighbor groups are better than skRKP-D in G-skRKP and G-RP as illustrated in Figure 6(b). When we assume the number of neighbor nodes is 21, in Figure 6(a), the $P_{u,v}$ of G-RP is 22% and 3% lower than GGD and skRKP-D, respectively, but, in Figure 6(b), P_{u,v^*} of G-RP is 640.7% and 55.7% higher than the two schemes. In other words, GenDep has slightly lower connectivity within the same group but far higher connectivity between the neighbor groups. In both cases of Figure 6, G-COMB achieves high connectivity. Furthermore, it is noteworthy that RP is a good choice as a key predistribution scheme because G-RP shows better performance than skRKP-D and G-skRKP. In addition, RP does not require complex computation in the key establishment and has better network resilience [22].

6.2.4. Discussion on Key Graph Connectivity according to Deployment Distribution. We model the deployment method for a sensor group as probabilistic distribution of sensor nodes. In this section, we take into consideration two kinds of probabilistic distributions: Gaussian and uniform distributions.

Gaussian Distribution. First, in the Gaussian distribution, sensor nodes are deployed in the 2-dimension (2D) Gaussian distribution from their deployment point. We denote by $g(z,r)$ the cumulative density function (CDF) of the 2D Gaussian distribution within a circle whose center is located at the distance of z from the deployment point and its radius is r. More formally,

$$g(z,r) = 1\{z < r\}\left[1 - e^{-(r-z)^2/2\sigma^2}\right] + \frac{1}{2\pi\sigma^2}\int_{|z-r|}^{z+r} 2l\cos^{-1}\left(\frac{l^2+z^2-r^2}{2lz}\right) e^{-l^2/2\sigma^2} dl, \tag{20}$$

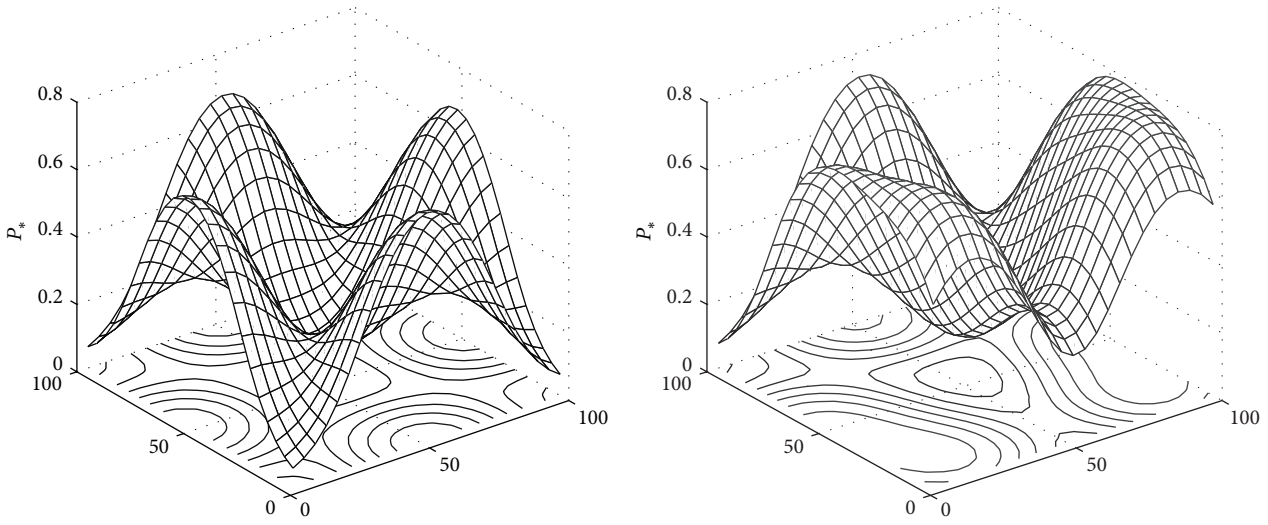
where σ is the standard deviation of the Gaussian distribution. Please refer to [29] about derivation of $g(z,r)$. Let us



(a) Connectivity within the same group

(b) Connectivity between neighbor groups

Figure 6: Key graph connectivity.



(a) Rectangular zone

(b) Pentagonal zone

Figure 7: Key graph connectivity with Gaussian sensor deployment ($n_z = 100, \sigma = 25, r = 10$, G-skRKP).

denote the position of the deployment point, $[x, y]_i$, by using x_i and y_i. The number of nodes in $G(i)$ at the point of (x, y) within r is

$$N_i(x, y, r) = \left\lfloor n_z \cdot g\left(\sqrt{(x - x_i)^2 + (y - y_i)^2}, r\right)\right\rfloor. \quad (21)$$

Figure 7 depicts the probability of the key graph connectivity to nodes in any neighbor group, $P_* = 1 - \prod_k^{NZ}(1 - P_{u,v^*})$, from $(0, 0)$ to $(100, 100)$ where the deployment point is located at $(50, 50)$. Since the distribution of neighbor nodes determines the probability of the key graph connectivity, Figure 7 shows high connectivity along each direction to the deployment points of neighbor groups. Additionally, note that the shape of the zone in Figure 7(a) is a regular polygon, that is, a square, but that of Figure 7(b) is the irregular pentagon, which is the same shape to the zone G in Figure 4(b).

Uniform Distribution. Second, in the uniform distribution, the coverage area and density determine the number of neighbor nodes regardless of the deployment point. Let ρ_i be

the sensor node density of $G(i)$, which can be calculated by $\rho_i = n_z/|\text{zone}_i|$, where zone_i is the zone of $G(i)$. Hence, the number of sensor nodes within r in $G(i)$ is

$$N_i(x, y, r) = \left\lfloor \rho_i \cdot \left| \text{zone}_i \cap \left\{ (x, y) \mid \sqrt{(x - x')^2 + (y - y')^2} < r \right\} \right| \right\rfloor, \quad (22)$$

where $|\text{zone}_i \cap \{(x, y) \mid \sqrt{(x - x')^2 + (y - y')^2} < r\}|$ is the area of the intersected region between zone_i and the communication range of the node located at (x', y'). The intersected area can be calculated geometrically using the sensor coverage analysis such as [5].

6.3. Simulation. We perform simulation to validate our analysis result in more realistic situations. Our simulation follows the system setup of Sections 6.1 and 6.2.4 ($n_z = 100, \sigma = 25, R = 10$ m). In the simulation, we place nine sensor groups as shown in Figure 8. Each group is located from a given group distance. We ran simulation with varying distances between groups to check the number of connections to confirm the trends of key graph connectivity of our analysis in Section 6.2.3. In the simulation, we implemented GGD and G-skRKP, since the relative performance of G-skRKP to GGD is a good indicator for the trend of the analysis result.

Figure 9 shows the result of the simulation. We change the distance between groups from 20 m to 50 m and count the total number of connections established within the two-hop key graph after deployment. Since sensor nodes are deployed randomly in Gaussian distribution, we ran simulation 30 times to get the average of the results. As expected, we can observe that the connectivity within the same group does not change by varying the distance in Figure 9(a). GGD has 18.8% more connections among sensor node within the same group than G-skRKP. As shown in Figure 9(b), however, G-skRKP always outperforms GGD by 338% in the number of connections among neighbor groups. Therefore, both results show the same trend as the result of the numerical analysis in Section 6.2.3 and G-skRKP can provide higher connectivity between groups.

6.4. Security Analysis. The group structures in GenDep prevent the general deployment model from incurring additional security concerns. The security conditions for the key predistribution schemes, such as the λ-secure property, can also be applied easily.

Figure 10 depicts the network resilience against node capture attacks for the same size of key sets, $m_k = 30$, and the local connectivity, $p_1 = 0.14$. We refer to [3, 19] and Appendix B for the equations of the network resilience. In Figure 10, skRKP shows the strong resistance to the node capture attacks when n_z is only restricted under 100, that is, satisfying the λ-secure property. However, RP can achieve the perfect resilience against the node capture attacks. Note that the network resilience of qRKP is worse than RKP due to the small size of m_k and key pool. Therefore, we exclude the qRKP scheme in the performance comparison. In case

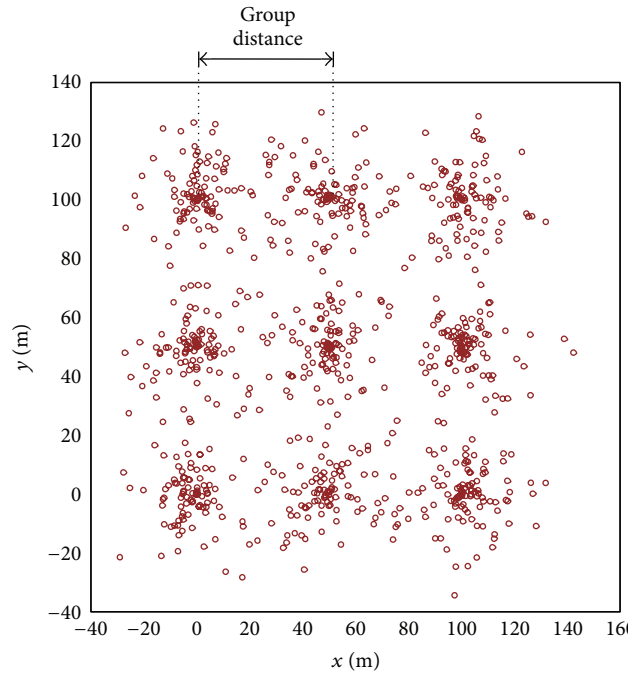


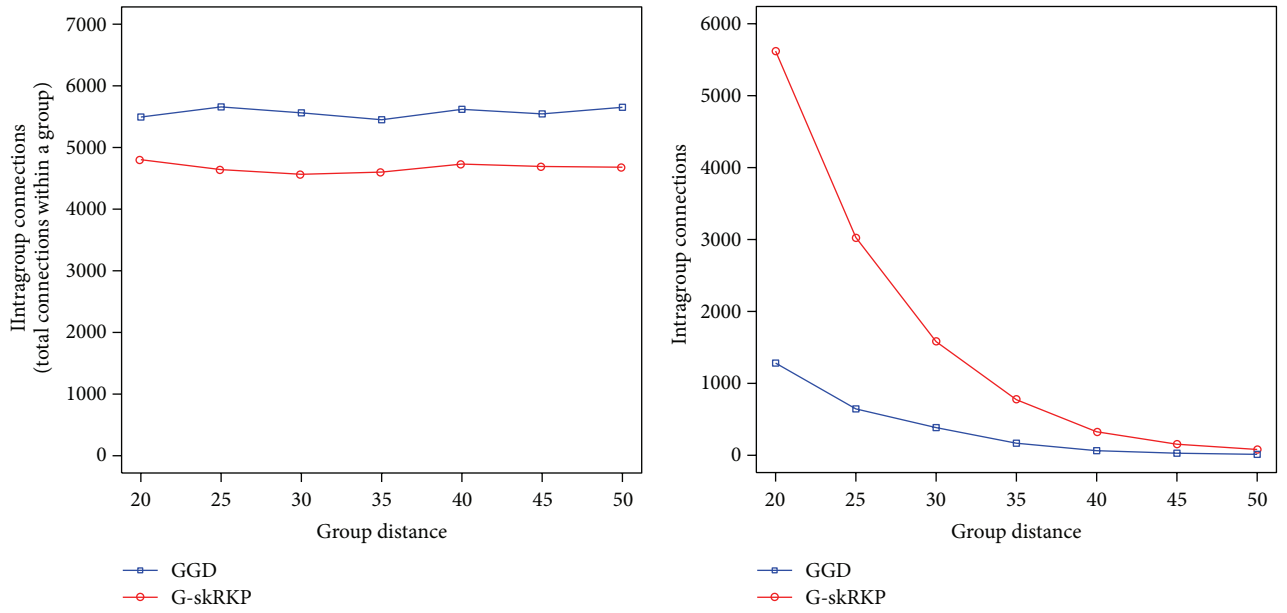
Figure 8: Simulation setting. We place nine sensor groups and deploy sensor nodes in Gaussian distribution. (Please note that we use $\sigma = 10$ in this figure to show sensor groups more distinguishably but $\sigma = 25$ in the simulation to follow the settings of Sections 6.1 and 6.2.4.)

that nodes store a relatively smaller number of keys for an EG, the RKP scheme is better to be selected for security and connectivity than the qRKP scheme.

6.5. Storage Overhead. The size of the key storage required to preinstall key sets in a sensor node is dependent on the key predistribution schemes. Let $m_{k,l}$ be the size of the key storage for the lth EG of a sensor node. The total storage for a node can be computed as $m = \sum_{l=1}^{L_{G(i)}} m_{k,l}$. The security conditions of the key predistribution scheme may restrict the lower bound of m_k. In that sense, for the Blom-based schemes in GenDep, the security parameter λ is calculated as $2n_z\tau_E/\omega$. That is, the selected key space in GenDep, τ_E, should be half of τ that other Blom-based schemes use in order to keep the same value of n_z. Thus, m_k for an EG with skRKP is

$$m_k = \left(\left\lceil \frac{2n_z\tau_E}{\omega} \right\rceil + 1 \right) \tau_E. \quad (23)$$

In case that skRKP is assigned to all EGs, for example, G-skRKP, $m = L_{G(i)} \cdot m_k$. Table 2 shows the key storage overhead on various configurations for this case. Based on the configurations of [3, 5] ($\omega = 7, \tau = 2, \tau_E = 1$, 64-bit key), the required key storage for GenDep is under a few kilobytes for both cases of $L_{G(i)} = 4$ (rectangular zone) and $L_{G(i)} = 6$ (hexagonal zone). Even in the cases of hexagonal zone, 128-bit key, and $\tau_E = 2$, the key storage is under 12 kbytes. Since MICAz sensor nodes have a 128 kbyte program memory and a 512 kbyte secondary memory [30], the required key storage is small enough to be installed to the memory of sensor nodes.



(a) Number of *intra*group connections (connectivity within the same group)

(b) Number of *inter*group connections (connectivity between neighbor groups)

Figure 9: Key graph connectivity in the simulation.

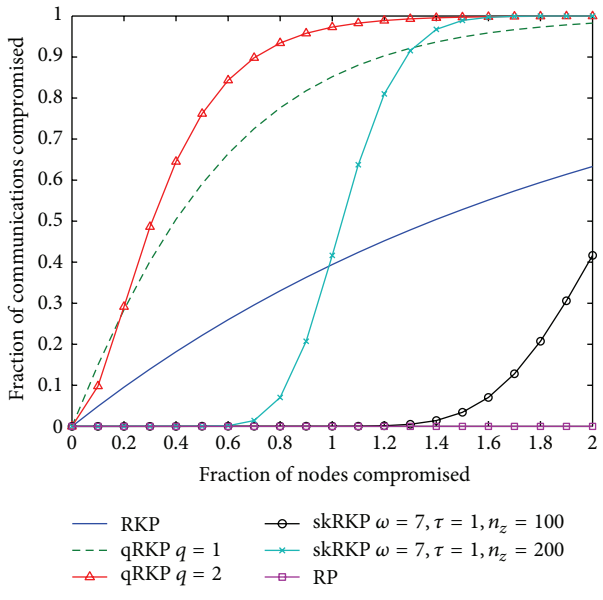


Figure 10: Network resilience of the key predistribution schemes ($m_k = 30$, $p_1 = 0.14$).

Let m_G, m_D, $m_{\texttt{G-skRKP}}$, $m_{\texttt{G-COMP}}$, and $m_{\texttt{G-RP}}$ be the required storages to store keys for GGD, skRKP-D, G-skRKP, G-COMB, and G-RP, respectively. In skRKP-D, each key pool is shared among neighbor zones in proportion to a or b. In other words, due to the λ-secure property, we cannot deploy more than $n + (4a + 4b)n$ nodes. Hence, $\lambda = \lceil n\tau_G/\omega \rceil$ for GGD and $\lambda = \lceil 2n\tau_D/\omega \rceil$ for skRKP-D using the condition $4a + 4b = 1$ in [2]. From these conditions, the required key storage for the schemes is

$$m_G = \left(\left\lceil \frac{n_z \tau_G}{\omega} \right\rceil + 1 \right) \tau_G + \gamma\alpha,$$

$$m_D = \left(\left\lceil \frac{2n_z \tau_D}{\omega} \right\rceil + 1 \right) \tau_D, \tag{24}$$

$$m_{\texttt{G-skRKP}} = 4 \left(\left\lceil \frac{2n_z \tau_E}{\omega} \right\rceil + 1 \right) \tau_E.$$

From (24), the parameters of the system model in Section 6.1 are determined to achieve the same level of key storage, $m_D = m_{\texttt{G-skRKP}} = 120$ and $m_G = 128$. Since G-RP and G-COMB do not require the security conditions on m, we set the $m_{\texttt{G-COMP}} = m_{\texttt{G-RP}} = 120$.

7. Conclusion

In this paper, we proposed GenDep, the novel framework for location-aware key management in WSNs regarding the general deployment model. Due to the notion of *generalized* location awareness, GenDep can apparently provide a practical advantage. The WSN can be deployed without losing its security in the real world where nonuniform geographical features exist. In GenDep, the group placement phase is operated to find optimal deployment positions of sensor groups. In the following key management phase, a whole deployment area is partitioned into a number of different smaller zones based on the deployment points of sensor groups. Subsequently, we build a new group structure to cover all sensor nodes deployed in neighboring zones. Using the group structures, we can render the benefit of location

Table 2: Required key storage.

L	n_z	ω	τ_E	Key size (bits)	Storage (bytes)	L	n_z	ω	τ_E	Key size	Storage
4	50	7	1	64	512	6	50	7	1	64	768
4	100	7	1	64	960	6	100	7	1	64	1440
4	200	7	1	64	1888	6	200	7	1	64	2832
4	50	7	1	128	1024	6	50	7	1	128	1536
4	100	7	1	128	1920	6	100	7	1	128	2880
4	200	7	1	128	3776	6	200	7	1	128	5664
4	50	7	2	64	1920	6	50	7	2	64	2880
4	100	7	2	64	3776	6	100	7	2	64	5664
4	200	7	2	64	7424	6	200	7	2	64	11136

awareness to the existing key predistribution schemes. We proved the superior performance of key predistribution schemes when applied to GenDep, compared to their raw deployment. We also showed that it is possible to achieve the same level of security even in the irregular deployment area due to our generalized location awareness.

The generalized location awareness approach is promising for various practical applications, not limited to WSNs. For WSNs, GenDep enables their deployment to unforeseen circumstances. Thus GenDep is also applicable to three-dimensional (3D) sensor deployment, such as sensors in intelligent buildings. Secure key management for a group of any scattered devices can adopt GenDep as well, for example, RF readers in department stores and remote consoles in cafes and restaurants. We believe it would be promising to scrutinize more applications in the future study.

Appendices

A. Local Connectivity

As described in Section 6.1, `G-skRKP`, `G-RP`, and `G-COMB` represent the proposed frameworks with `skRKP`, `RP`, and the combination of `RP` and `RKP` as the key predistribution schemes, respectively. Additionally, `skRKP-D` and `GGD` are proposed in [2, 5], respectively. Then, the equations of $p_{1s}(i)$ and $p_{1n}(i, j)$ for each scheme are like the following.

(1) `G-skRKP`:

$$\begin{aligned} p_{1s}(i) &= 1 - \left[\frac{\binom{\omega}{\tau_E}\binom{\omega-\tau_E}{\tau_E}}{\binom{\omega}{\tau_E}^2} \right]^{L_{G(i)}} \\ &= 1 - \left[\frac{\{(\omega-\tau_E)!\}^2}{\omega!\,(\omega-2\tau_E)!} \right]^{L_{G(i)}}, \\ p_{1n}(i,j) &= 1 - \frac{\binom{\omega}{\tau_E}\binom{\omega-\tau_E}{\tau_E}}{\binom{\omega}{\tau_E}^2} \\ &= 1 - \frac{\{(\omega-\tau_E)!\}^2}{\omega!\cdot(\omega-2\tau_E)!}. \end{aligned} \tag{A.1}$$

(2) `G-RP`:

$$\begin{aligned} p_{1s}(i) &= 1 - \left(1 - \frac{m_k}{2n_z}\right)^{L_{G(i)}}, \\ p_{1n}(i,j) &= \frac{m_k}{2n_z}. \end{aligned} \tag{A.2}$$

(3) `G-COMB`:

$$\begin{aligned} p_{1s}(i) &= 1 - \left(1 - \frac{m_k}{2n_z}\right)^{L_{G(i),\mathrm{RP}}} \\ &\quad \times \left(\frac{\binom{M}{m_k}\binom{M-m_k}{m_k}}{\binom{M}{m_k}^2} \right)^{L_{G(i),\mathrm{RKP}}}, \\ p_{1n,\mathrm{RP}}(i,j) &= \frac{m_k}{2n_z} \\ p_{1n,\mathrm{RKP}}(i,j) &= 1 - \frac{\binom{P}{m_k}\binom{P-m_k}{m_k}}{\binom{P}{m_k}^2}, \end{aligned} \tag{A.3}$$

where $L_{G(i),\mathrm{RP}}$ and $L_{G(i),\mathrm{RKP}}$ are the number of EGs that select `RKP` and `RP`, respectively.

(4) `skRKP-D`:

$$\begin{aligned} p_{1s}(i) &= p_{1n}(i,j) \\ &= 1 - \frac{\sum_{k=0}^{\min(\tau_D,\xi(i,j))} \binom{\xi(i,j)}{k}\binom{\omega-\xi(i,j)}{\tau_D-k}\binom{\omega-k}{\tau_D}}{\binom{\omega}{\tau_D}^2}, \end{aligned} \tag{A.4}$$

where $\xi(i, j)$ is ω for $p_{1s}(i)$ and $a \cdot \omega$ for $p_{1n}(i, j)$.

(5) `GGD`:

$$\begin{aligned} p_{1s}(i) &= 1 - \frac{\binom{\omega}{\tau_G}\binom{\omega-\tau_G}{\tau_G}}{\binom{\omega}{\tau_G}^2} \\ &= 1 - \frac{\{(\omega-\tau_G)!\}^2}{\omega!\cdot(\omega-2\tau_G)!}, \\ p_{1n}(i,j) &= \frac{\alpha_{\mathtt{GGD}}}{n_z}. \end{aligned} \tag{A.5}$$

B. Network Resilience

We define f_x as the fraction of communication compromised when x nodes are captured and the following equations are f_x for the RKP, qRKP, and skRKP schemes.

(1) RKP:

$$f_x = 1 - \left(1 - \frac{m}{|S|}\right)^x. \tag{B.1}$$

(2) qRKP (q-composite):

$$f_x = \sum_{i=q}^{m} \left(1 - \left(1 - \frac{m}{|S|}\right)^x\right)^i \frac{p(i)}{p}, \tag{B.2}$$

where $p(i) = \binom{|S|}{i}\binom{|S|-i}{2(m-i)}\binom{2(m-i)}{m-i}/\binom{|S|}{m}^2$ and $p = p(q) + p(q+1) + \cdots + p(m)$.

(3) skRKP:

$$f_x = \sum_{j=\lambda+1}^{x} \binom{x}{j}\left(\frac{\tau}{\omega}\right)^j \left(1 - \frac{\tau}{\omega}\right)^{x-j}. \tag{B.3}$$

Notations

B:	Set of deployment points: $\{[x, y]_i \mid 1 \geq i \geq N_D\}$
C:	Tunable radio transmission range parameter
D:	Maximum distance where a sensor detects an event
E:	Set of extended groups: $\{\mathsf{EG}(i, j) \mid G(j) \in NZ(i) \vee G(i) \in NZ(j)\}$
$G(i)$:	ith sensor group, $1 \geq i \geq N_D$
ID_u:	Identifier of a sensor
$\mathrm{ID}_{\mathsf{EG}}$:	Identifier of an EG
K:	Deployment plan: $\{([x, y]_i, \sigma_i) \mid 1 \geq i \geq N_D\}$
$L_{G(i)}$:	Number of overlaid EGs for $G(i)$
M:	Key pool size
N:	Number of sensor nodes
N_D:	Number of sensor groups
$N_i(x, y, r)$:	Number of sensors of $G(i)$ within a circle of radius r at $\langle x, y\rangle$
N_{EG}:	Number of EGs
$\widetilde{N}_i$:	Set of well-connected neighbor groups
n_z:	Number of sensors in a group
Q:	Set of sensor groups: $\{G(i) \mid 1 \geq i \geq N_D\}$
R:	Communication range of a sensor (in radius)
T_V:	Threshold of $U(x, y)$ for optimal placement in pVFA
$U(x, y)$:	Desired sensor density at (x, y)
$V(x, y)$:	Density gap between $U(x, y)$ and sensors to be placed at (x, y)
(x, y):	Reference point
$[x, y]_i$:	Deployment point of $G(i)$
$\mathsf{EG}(i, j)$:	Extended group of $G(i)$ and $G(j)$
$\vec{F}^A_{G(i)}(x, y)$:	Attractive force on $G(i)$ from (x, y)
$\vec{F}^A_{G(i)}$:	Gradient (total attractive force) of $G(i)$
α:	Physical characteristics of sensors
δ_d:	Unit for moving distance in the pVFA iteration
δ_s:	Unit for incremental std in the pVFA iteration
ω and τ:	Parameters for λ-secure property [3]
$b(x, y)$:	Number of available sensors to detect an event at the same time at (x, y)
m:	Total number of keys in a sensor
m_k:	Number of keys in a key set per EG
p_1^*:	Expected probability that two nodes of different group share keys
$p_d(x, y)$:	Detection probability that a sensor detects an event at (x, y)
w_i and w_j:	Number of sensors from $G(i)$ and $G(j)$ in their intersection region
$\mathscr{K}_{\mathsf{EG}(i,j)}$:	Key pool for $\mathsf{EG}(i, j)$
RKP:	Random key predistribution scheme [1]
qRKP:	q-composite RKP [19]
RP:	Random-pairwise key scheme [19]
skRKP:	Structure key-pool RKP [3]
GGD:	Grid-group key predistribution scheme [5].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (nos. 2012R1A1A1044693 and 2012R1A1B3000965).

References

[1] 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 (CCS '02)*, pp. 41–47, ACM Press, Washington, DC, USA, November 2002.

[2] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A key predistribution scheme for sensor networks using deployment knowledge," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 1, pp. 62–77, 2006.

[3] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 2, pp. 228–258, 2005.

[4] R. di Pietro, L. V. Mancini, and A. Mei, "Random key-assignment for secure wireless sensor networks," in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03)*, pp. 62–71, ACM Press, Fairfax, Va, USA, October 2003.

[5] 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)*, pp. 29–42, ACM Press, Washington, DC, USA, October 2004.

[6] D. Liu and P. Ning, "Location-based pairwise key establishments for static sensor networks," in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03)*, pp. 72–82, ACM Press, Fairfax, Va, USA, October 2003.

[7] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 247–260, 2006.

[8] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, vol. 3, pp. 1380–1387, Anchorage, Alaska, USA, April 2001.

[9] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 1, pp. 61–91, 2004.

[10] Y. Zou and K. Chakrabarty, "A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks," *IEEE Transactions on Computers*, vol. 54, no. 8, pp. 978–991, 2005.

[11] G. Takahara, K. Xu, and H. Hassanein, "How resilient is grid-based WSN coverage to deployment errors?" in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 2872–2877, Kowloon, China, March 2007.

[12] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network," *IEEE Transactions on Mobile Computing*, vol. 7, no. 2, pp. 262–274, 2008.

[13] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: a survey," *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 303–334, 2008.

[14] T. Lambrou and C. Panayiotou, *Collaborative area monitoring using wireless sensor networks with stationary and mobile nodes [Ph.D. dissertation]*, University of Cyprus.

[15] T. Lambrou and C. Panayiotou, "Collaborative path planning for event search and exploration in mixed sensor networks," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1424–1437, 2013.

[16] T. Das and S. Roy, "Coordination based motion control in mobile wireless sensor network," in *Proceedings of the International Conference on Electronic Systems, Signal Processing and Computing Technologies (ICESC '14)*, pp. 231–236, Nagpur, India, January 2014.

[17] H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 163–174, 2014.

[18] S. He, X. Gong, J. Zhang, J. Chen, and Y. Sun, "Barrier coverage in wireless sensor networks: from lined-based to curve-based deployment," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '13)*, pp. 470–474, Turin, Italy, April 2013.

[19] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the IEEE Symposium on Security And Privacy (SP '03)*, pp. 197–213, Berkeley, Calif, USA, May 2003.

[20] R. Blom, "An optimal class of symmetric key generation systems," in *Advances in Cryptology: Proceedings of EUROCRYPT 84. A Workshop on the Theory and Application of Cryptographic Techniques*, pp. 335–338, Springer, New York, NY, USA, 1985.

[21] A. Fanian, M. Berenjkoub, H. Saidi, and T. A. Gulliver, "A hybrid key establishment protocol for large scale wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '10)*, pp. 1–6, Sydney, Australia, April 2010.

[22] T. Kwon, J. Lee, and J. Song, "Location-based pairwise key predistribution for wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 11, pp. 5436–5442, 2009.

[23] J. Lee, T. Kwon, and J. Song, "Group connectivity model for industrial wireless sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 5, pp. 1835–1844, 2010.

[24] Y. Zou and K. Chakrabarty, "Uncertainty-aware and coverage-oriented deployment for sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 788–798, 2004.

[25] V. Seshadri, *The Inverse Gaussian Distribution: Statistical Theory and Applications*, Springer, 1999.

[26] F. Xue and P. R. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, no. 2, pp. 169–181, 2004.

[27] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.

[28] D. Huang, M. Mehta, A. van de Liefvoort, and D. Medhi, "Modeling pairwise key establishment for random key predistribution in large-scale sensor networks," *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1204–1215, 2007.

[29] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Proceedings of the 23th AnnualJoint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 1, pp. 586–597, March 2004.

[30] "MEMSIC wireless moudules (formerly, micaz was a product of crossbow technology, inc.)," http://www.memsic.com/.