Coordinated Sensor Deployment for Improving Secure Communications and Sensing Coverage

by Yinian Mao, Min Wu

TR 2005-98

Coordinated Sensor Deployment for Improving Secure Communications and Sensing Coverage

Yinian Mao and Min Wu
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD, USA
Email: {ymao, minwu}@eng.umd.edu

Abstract— **Sensor network has a great potential in applications such as habitat monitoring, wildlife tracking, building surveillance, and military combat. The design of a sensor network system involves several important issues, including the sensing coverage, node-to-node or node-to-base-station communications, and the security in information gathering and relay by the sensors. In this paper, we show that the system performance on these aspects depends closely on how the sensors are deployed in the field, and on how the sensor locations can be adjusted after the initial deployment. For static sensor deployment, we investigate the hexagon and square lattice topology and analyze their impact on secure connectivity and sensing coverage. For advanced sensing devices that allow for location adjustment after deployment, we have established a new framework for coordinated updates of sensor locations. We propose two new sensor location updating algorithms, the *VFSec* and the *Weighted Centroid* algorithm, to jointly optimize sensing coverage and secure connectivity. Simulation results show that these new algorithms provide superior tradeoff over the existing approaches that do not take security into considerations.**

I. Introduction

Wireless sensor network has shown a great potential in applications such as habitat monitoring, wildlife tracking, building surveillance, as well as battlefield surveillance [1], [2]. As these broad applications make sensor networks a promising technology, the design of sensor networks is also challenging, especially when one design aspect is intricately involved with a number of other aspects [2]. One such example is the the sensor node deployment. In sensor deployment, there are works concerning the efficient sensing coverage issue [3]–[5], or the secure communication problem [6], [7], but not both. Since sensor network systems have inherent complex criteria and objectives, optimizing a single objective may impair the system performance on other aspects. As a first step toward developing the theory and algorithms of more *coordinated* sensor deployment, this paper focuses on jointly considering two important aspects, namely, the sensing coverage and communication security.

The main jobs for most sensor nodes in a sensor network include sensing and communications. Depending on applications, appropriate type of environmental information in the field are first gathered by the individual sensor nodes and processed; and the necessary information is then relayed to and/or collected by other nodes [8], [11]. The physical characteristics of the sensing and communication devices on board of a sensor impose limits on both the sensing range and the communication range. Therefore the placement of sensor nodes will have great impacts on both the sensing coverage [3] and the communication connectivity [9].

Recently, the security of sensor networks has been brought to the attention of the research community [10] [12]. As the sensor nodes rely on wireless transmission for communications, malicious adversaries could intercept the communications, modify the data packets, or inject falsified packets. To ensure secure sensor communications, cryptographic mechanisms can be employed to encrypt the data and produce message authentication code (MAC). As symmetric-key cryptography that employ the same cryptographic key in the sending and receiving ends generally have substantially lower computational complexity than the public-key ones, symmetric-key cryptographic tools is generally preferred in practice because of limited computational resources at each sensor node. Furthermore, resource-constrained sensor networks impose stringent constraints on the key establishment scheme. Conventional key management schemes are either centralized by employing a key distribution server, or contributory by using public-key cryptography, and both often require a non-trivial amount of communications. These conventional schemes are not suitable in the sensor network scenarios [13]. To meet the challenge in designing secure sensor networks, key pre-distribution schemes have been introduced to address the special needs in sensor networks [13], [14].

There are two main scenarios that sensor deployment are modelled and studied, depending on whether the locations of sensor nodes can be adjusted after the initial deployment. The first scenario is static deployment, where the location of the sensors will not change once deployed. When efficient sensing coverage is the sole concern, the existing literature suggests that different deployment topologies lead to substantially different efficiency in sensing coverage [15], [16]. In the mean time, researchers focusing on secure sensor communications have recently shown that, if the key pre-distribution can be adapted according to the sensors' locations, we can substantially improve the probability for establishing secure communication links between sensors as well as the security against compromised nodes [6], [7], [26]. However, there is a very limited amount of analysis on how the topology of sensor locations affects both security and coverage issues. In Section III of this paper, we present analytic model and experimental validations on several practical topologies in terms of both sensing coverage and ability to establish secure communication links. We shall consider both the case when each sensor can be accurately placed at any desired location, and the case when the actual deployment deviates from the desired location. This investigation will provide important guidelines to sensor deployment for a variety of applications.

The second scenario of sensor deployment considers more advanced sensing devices, where the sensors have the capability of adjusting their locations after being deployed in the field. This is particularly useful when the actual deployment deviates from the desired location. The current literature primarily concerns the development of adjustment algorithms to optimize the sensing coverage. As we shall show in Section IV, such adjustment may negatively affect the probability for sensors to establish secure communication links. This motivates us to develop new adjustment algorithms that can jointly optimize the sensing coverage and communication security. We further relate to the first scenario by examining how different topologies for the desired deployment locations affects the overall performance under these security-aware adjustment algorithms.

The rest of the paper is organized as follows. Section II introduces the background and the prior works related to sensor network deployment. In Section III we jointly investigate the sensing coverage and communication security under the static sensor deployment scenario. We then consider the mobile scenario where sensors can adjust their locations after being deployed and propose two new location-adjusting algorithms in Section IV, by jointly considering the sensing coverage and secure communications. Finally, the conclusions are drawn in Section V.

II. Background and Related Works

In this section, we review the background on sensing coverage and secure communications in sensor networks, and briefly survey the related prior works. Throughout the discussion we adopt a simplified mathematical model for sensing coverage. A sensor node located at $\mathbf{x_0}$ has the capability S of sensing for a given location $\mathbf{x}$

$$S(\mathbf{x_0}, \mathbf{x}) = \begin{cases} 1 & \text{if } d(\mathbf{x_0}, \mathbf{x}) \leq R_s; \\ 0 & \text{if } d(\mathbf{x_0}, \mathbf{x}) > R_s. \end{cases} \tag{1}$$

where R_s is referred to as the sensing radius, and the distance metric $d(\cdot, \cdot)$ is usually the Euclidean distance. $S = 1$ indicates the sensor has the capability to sense and $S = 0$ otherwise. Analogous to the sensing capability, we can simplify the existence of a communication link between two sensor nodes n_0 (located at $\mathbf{x_0}$) and n_1 (located at $\mathbf{x_1}$) using the following model

$$T(\mathbf{x_0}, \mathbf{x_1}) = \begin{cases} 1 & \text{if } d(\mathbf{x_0}, \mathbf{x_1}) \leq R_c; \\ 0 & \text{if } d(\mathbf{x_0}, \mathbf{x_1}) > R_c. \end{cases} \tag{2}$$

where R_c is referred to as the communication radius. $T = 1$ indicates the link exists and $T = 0$ otherwise.

A. The Sensing Coverage Problem

1) Efficient Sensing in Static Deployment: Suppose the sensor nodes with sensing radius R_s can be hand placed in the field to the exact location of our choice. We are interested in the optimal way to place the sensors so that: (1) any location in the field can be covered by at least one sensor; and (2) the nodes can perform sensing in an efficient way. To quantify the efficiency of sensing coverage, we define the sensing efficiency ratio ρ, which is the ratio of two areas

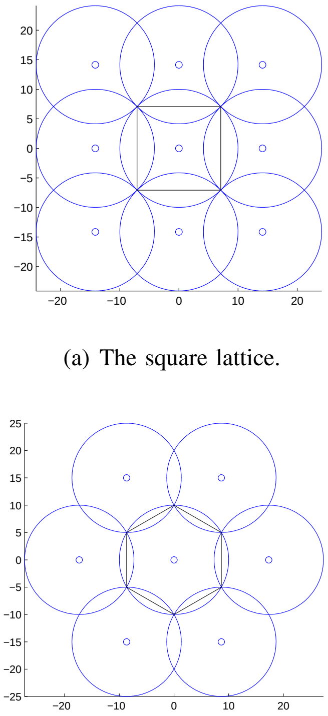
$$\rho = \frac{A_{sep}}{A_{col}}.$$

Here A_{col} is the actual covered area by all the sensor nodes, and A_{sep} is the sum of the area covered by each individual sensor. Apparently, we have $A_{sep} \geq A_{col}$ as the coverage between sensors may overlap, thus $\rho \geq 1$. The closer the efficiency ratio gets to 1, the higher the efficiency. So the problem of optimizing coverage efficiency can be formulated as to minimize the efficiency coefficient ρ subject to the whole area can be fully covered. This problem is traditionally known as the *circle* *covering* problem [18], where a number of equivalent circles (i.e. circles with the same radius) are placed into a field to completely cover the field area. A sensor node is located at the center of a circle, and the radius of the circle corresponds to the sensing radius.

If the circles are placed in repeated regular patterns, the circle centers form a lattice and the dissecting lines among the centers form a cell pattern. In Fig. 1 we show two possible covering layout using the square lattice and the hexagon lattice, respectively. Each layout leads to a specific efficiency ratio ρ, which is also known as the *covering density* or *covering thickness* in the mathematics literatures [16]. For the simplified sensing model of Eqn. (1), $A_{sep} = n \cdot \pi R_s^2$, where n is the total number of sensors. Kershner [15] has shown that a lower bound for ρ is $2\pi/\sqrt{27} \approx 1.21$, which is achieved when the center of the circles (i.e. the sensor nodes) form a hexagon lattice. In this case, the distance between any two neighboring nodes is $D = \sqrt{3}R_s$. Fig. 1(b) illustrates the geometry of such a placement. Compared to the square lattice placement, which has efficiency ratio $\rho = \pi/2$, the hexagon lattice placement gives much more efficient coverage. Further, sensor placement can be viewed as spatial sampling from signal processing perspective. The literature there also suggests the superiority of hexagonal sampling lattice over the square lattice when the spatial spectrum of a 2-D signal being measured (such as a temperature field) is bandlimited with a circular support.

For the convenience of discussion, we define the following notations. In the square lattice deployment, we denote the distance from a node to its horizontal/vertical neighbor by D_1, and the distance to its diagonal neighbor by D_2. Thus we have $D_2 = \sqrt{2}D_1 \approx 1.41D_1$, and the covering density $\rho = \pi R_s^2/D_1^2$. In the hexagon lattice, we denote the distance from a node to its six neighbors by D_3. Further, if we require that the two lattice have the same node density, we have $D_3 = \sqrt{2/\sqrt{3}}D_1 \approx 1.07D_1$. Throughout this paper, we will use the normalized distance with respect to D_1 as the distance metric, and study the impact of deployment topology on the performance of sensing coverage and secure communications.

2) Sensor Location Adjustment Algorithms: In recent years, the advances in micro-electromechanical systems (MEMS) have made it possible for tiny sensor devices to walk as microrobots [22]. The locomotion capabilities of sensor nodes have made it possible to improve the sensing coverage after the initial sensor deployment.



(a) The square lattice.

(b) The hexagon lattice.

Fig. 1. Two possible lattice deployment. Under full coverage requirement, the hexagon lattice has the lowest node density.

Consequently, a number of prior works have studied how to adjust the location of sensor nodes to maximize the total coverage in a given area [3]–[5], [17], [21]. The total sensing coverage, η, is the percentage of the area covered within the sensing range with respect to the total field area. We can see that $\eta \leq 1$ and a larger η represent a better coverage. Most existing algorithms for sensor location adjustment uses an iterative framework. In each iteration, sensors (or a cluster head) obtain their current locations and the relative locations to their neighbors. Based on these information, each node will compute a new location using the location updating algorithm. The general strategy is to spread out the sensor nodes as evenly as possible. For example, the virtual force algorithm (VFA) proposed in [5] compares the distance between a sensor and its neighbor nodes with a threshold distance. An attractive (resp. repulsive) virtual force is applied to the sensor node if the distance is greater (resp. smaller) than the threshold. The Minmax algorithm proposed in [4] employs the Voronoi cell concept and move the sensors to the center of the minimum-radius circum-circle of its Voronoi. Further, for calculating the sensing coverage, the authors of [3] and [17] have proposed polynomial-time algorithms to calculate the worst case and average case coverage.

From secure communication point of view, however, the location adjustment intended only to maximize sensing coverage may reduce the secure communication connectivity. This is because the secure links established before location adjustment may no longer exist after location adjustment and some sensor nodes can be moved to un-preferred locations in terms of establishing secure communications using pre-distributed key. In Section IV, we will present a detailed example to illustrate the limitation of the existing adjustment methods and discuss how to balance the tradeoff between the sensing coverage and the node connectivity using secure links.

B. Key Pre-distribution for Sensor Networks

As reviewed earlier, one of the critical issues for secure sensor communication is to establish a cryptographic key between two sensors. To accommodate stringent resource constraints in sensor network systems, an increasingly popular approach is to preload each sensor with a set of keys from a large collection of keys. This entire collection of keys is referred to as the *key pool* and the set of the keys loaded by each sensor is referred to as the *key ring*. Once the sensors are deployed in the field, neighboring sensors will follow certain protocol to discover whether they share some secret keys. If so, they will use these shared secret keys to establish a secure communication link. There are two requirements for establishing a secure communication link between two sensor nodes: (1) two nodes should be within communication range; and (2) two nodes should share at least one secret key. The first work on random key pre-distribution [13] was proposed by considering the sensor nodes are randomly deployed into the field. As such, the connectivity between nodes using secure links can be modelled as a random graph, and the number of secure links per node can be obtained using a probabilistic model. This will lead to node connectivity using the random graph theory [13]. Later, Du *et al.* and Liu *et al.* proposed to incorporate sensor location knowledge into key pre-distribution [6] [7]. The deployment model in these works considers the sensors being deployed at the center of evenly partitioned square cells. Each cell will have its own key pool, and only neighboring cells will have overlap between their key pools. The sensors in each cell will randomly pre-load keys from the key pool of its own cell. Since the key pool of each cell is much smaller compared to the key pool for the entire sensor node collection, neighboring sensors will have a higher chance to share keys. Most recently, Zhou *et al.* identify the improved circular symmetry of the hexagon cell than the square one to reflect the common shape of sensors' communication range, and propose to use hexagon lattice in location-based key pre-distribution [26].

In location-based key pre-distribution, each sensor has a designed deployment location for establishing secure communication link. In practice, these designed locations may not be the same as the locations determined according to the sensing performance. This motivates us to study the impact of practical sensor deployment on establishing secure communications.

III. Static Sensor Deployment with Lattice Structure

In this section, we jointly examine the sensing coverage and communication security under the static sensor deployment scenario. Given that a very limited amount of study has been done in the literature on how the sensor topology affects both security and coverage issues, we focus on analyzing the impact of deployment topology on the performance of sensing coverage and efficiency as well as to the ability of establishing secure communications. We will consider two main deployment topologies, namely, the square lattice and the hexagon lattice.

A. Fundamental Relations Between Deployment Lattices

As the first example to illustrate the impact of sensor deployment topology on the establishment of secure communications links, we consider the simple case of sensors being placed exactly at the desired location. We deploy sensors under a square lattice and a hexagon lattice, respectively, and employ the basic key pre-distribution scheme [13], where each node has the same probability to share a secret key with any other node. We denote the key sharing probability by P_{share}, and use the same node densities in the two lattice deployment, which is the number of nodes per unit area. As the communication radius R_c increases from 0 to $1.6D_1$, each sensor can gradually have more reachable neighbors, and this in turn will affect the number of secure links per node. Because the distance between a node and its eight neighboring nodes in a square lattice is not circularly symmetric [26], the number of neighbor nodes that can be reached is a two-step function of the communication range. That is, as the communication range increases, four vertical and horizontal neighbors of the center nodes (also known as the *4-way connection* [27]) will be reached first, before the other four neighbors on the diagonal directions being reached. This can be seen from Fig. 2, where we show the relation of the expected number of secure links per

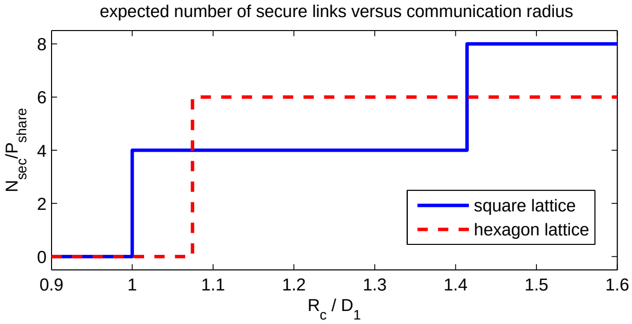


Fig. 2. Expected number of secure links versus communication radius using the basic key pre-distribution scheme.

node, normalized by the key sharing probability, with respect to the normalized communication radius. The result for a hexagonal lattice, on the other hand, is a one-step function, owing to the circular symmetry between a center node and its six neighbors. Under the same node density, Fig. 2 shows that hexagon lattice achieves a better topology when the communication radius R_c is between 1.07 and 1.41 times of D_1; and outside this range, square lattice achieves a better connectivity. Later in this paper, we shall see several more examples reflecting this fundamental relations between the square and hexagon deployment lattice.

In addition to the ability to establish secure links between nodes, resilience against node compromise is another important security aspect to be examined. As sensor nodes may be deployed in adversarial environment, a deployed node could be captured by adversaries. Using pre-loaded keys in the captured node, an adversary can potentially eavesdrop secure links among sensor nodes that are not compromised. In measuring such a potential threat, the probability of link compromise due to node compromise is an important security metric considered by previous key pre-distribution works [6], [13], [14], [25]. To allow fair comparison on different deployment topologies, we should require each topology to have the same link compromise probability when the same number of nodes are compromised, and then compare the connectivity of secure communication. We have constructed a probabilistic model to compute and approximate the link compromise probability in the location-based key pre-distribution scheme [6] using lattice deployment. It can be shown that if the compromised nodes are statistically uniformly distributed among all nodes and each node carries the same number of keys, then the link compromise probability is approximately the same for the location-based scheme using the square lattice, the hexagon lattice, and the basic scheme using random deployment, up to the first-order Taylor expansion [13]. The detailed derivation can be found in the Appendix. With this finding, we can construct a fair comparison between deployment topologies. As we shall see later, for fixed-size key ring, the group-based scheme using structured/lattice deployment usually can achieve a better connectivity than the random deployment.

B. Secure Connectivity Under Perturbed Deployment Lattice

While Fig. 2 depicts the trend for the secure communication connectivity using square and hexagon lattices in the ideal situation, sensors may not be deployed with high accuracy at the designed lattice locations in practice. Such inaccuracy may be caused by measurement error (if sensors are deployed by human), or by wind speed (if sensors are deployed by vehicle or airborne methods). Suppose a sensor node is designed to be deployed at location (x_0, y_0) in the field. The actual deployment location (x, y) can be modelled as

$$x = x_0 + r_x; \quad y = y_0 + r_y.$$

Here the deviation terms r_x and r_y are zero-mean random variables. One can model these deviation terms as Gaussian distributed [6] or uniformly distributed [25] random variables with variance σ^2 as the deviation parameter.

Taking the deployment deviation into consideration, we investigate the impact of deployment topology on the connectivity of secure communication. Here we choose the location-based key pre-distribution in [6] and the Gaussian deployment deviation model and compare the square lattice deployment used in [6] with the hexagon lattice deployment. In the hexagon lattice deployment, each node is surrounded by six neighbor nodes. By using location-based key pre-distribution, the key pool for any given node, referred to as the center node, has $1/6$ overlap with each of its six neighbors' key pools. Thus the the center node will have equal probability to share keys with each neighbor node. We denote the probability that the center node can still be a neighbor with one of its neighbor node under Gaussian deployment deviation by $\Pr(neighbor)$, and the probability that the two nodes can share a key by $\Pr(share)$. As the deployment deviation is independent of key distribution, the probability that a designed neighbor in the hexagon lattice can establish a secure link with the center node is $\Pr(hexlink) = \Pr(neighbor)\Pr(share)$. Because of the geometrical symmetry, the expected number of secure links for the center node is

$$\mathrm{E}(N_{sec}^{hex}) = 6\,\Pr(hexlink).$$

Similarly, we can compute expected number of secure links per node in the square lattice deployment. In the square lattice, we refer to the horizontal/vertical neighbors of a node as type-A neighbors and the diagonal neighbors as type-B neighbors. Denote the probability that a node can establish a secure link with one of its type-A neighbors as $\Pr(sqlinkA)$, and that with type-B neighbors as $\Pr(sqlinkB)$. The expected number of secure links per node is

$$\mathrm{E}(N_{sec}^{sq}) = 4\,\Pr(sqlinkA) + 4\,\Pr(sqlinkB).$$

In Fig. 3 we show the expected number of secure links per node under Gaussian deployment deviation. Each node carries 100 keys and each key pool contains 1200 keys. The neighbor probability and key sharing probability can be computed using the model in [24]. In Fig. 3, in both square and hexagon lattice deployment, the expected number of secure links increases with the normalized communication radius R_c/D_1. The hexagon lattice achieves a slightly higher connectivity over the range of 0.9 and 1.7 in the normalized communication radius, exhibiting a similar trend as in Fig. 2. This suggests that the communication radius, deployment topology, deviation parameter, and the number of preloaded keys per sensor all play a role in establishing secure links. It is also worth noticing that while the numerical gain in connectivity by hexagon lattice over the above mentioned communication range is small, its practical impact is non-trivial. Within this communication range, the average node degree increases from 0.5 to around 4, and the sensors gradually change from isolated nodes to connected components, where the boundary value for the average links per node is around 2. This phenomenon will be illustrated later in Section IV. To achieve the same connectivity, the square lattice would require a larger communication range. As the power consumption of wireless communications is related to the communication range by a power law (from the $2nd$ to the 4-th power, depending on the propagation environment [19]), a lower requirement on communication range with lower power consumption while maintaining communication connectivity is more desirable in many sensor network designs. This makes hexagon deployment lattice attractive for power-limited applications.

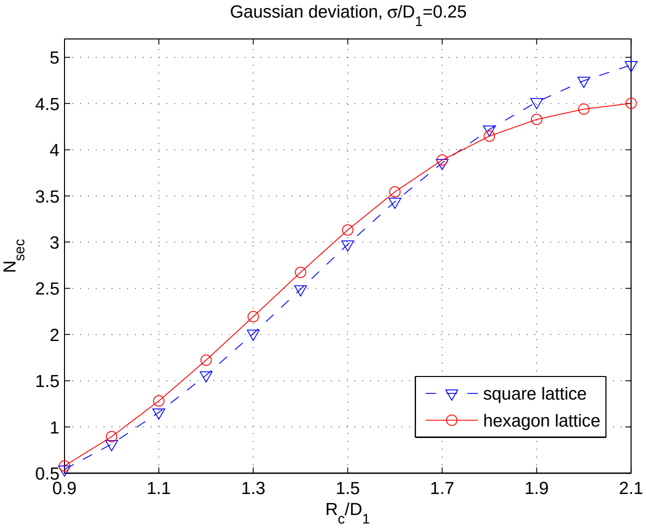


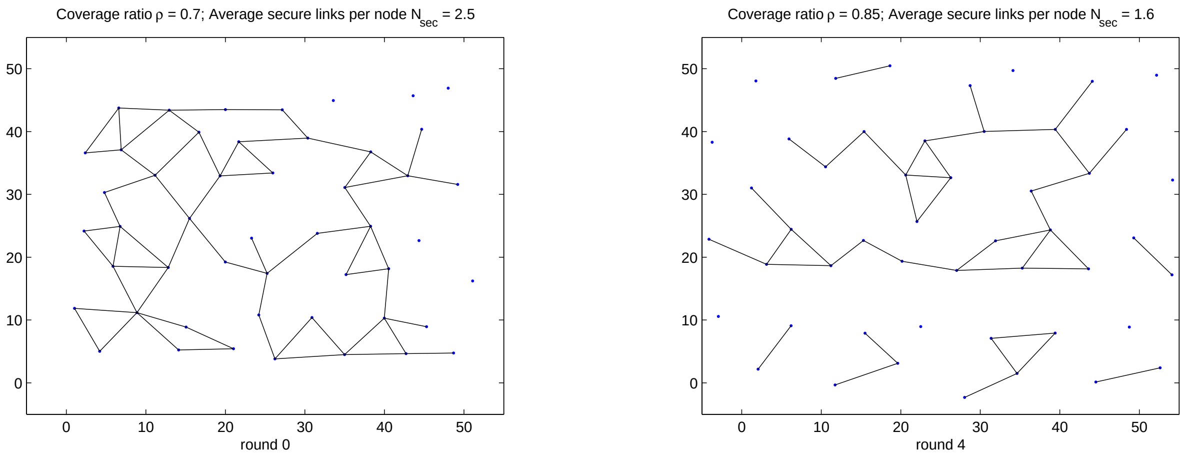
Fig. 3. Expected number of secure links per node versus communication radius. Shown here are the analytical values under Gaussian deployment deviations.

IV. Security-Aware Sensor Location Adjustment

The static deployment strategy described in the previous section considers the sensor deployment as a one-time task. Once the sensors are deployed in the field, their locations are fixed and cannot be further adjusted. In recent years, a number of works on practical sensor deployment have considered movement-adjusted sensor deployment for improving sensing coverage [4], [5], [21]. In this section, we investigate the impact of location adjustment in sensor deployment on secure communications. We propose two new location updating algorithms for sensor deployment that jointly consider sensing coverage and secure communications.

A. Improving Secure Connectivity Using the Virtual Force Framework

1) Effect on Secure Connectivity by the Existing Approach: When secure communications is required for sensor nodes, the existing location adjustment algorithms may negatively affect the establishment of secure communication links. As an example, we examine the establishment of secure communication links when the sensors are moved by the Virtual Force [5] location updating algorithm. The Virtual Force algorithm adjusts the sensor locations based on the relative distance from a sensor to its neighbors compared to a pre-determined threshold d_{th}. Suppose a node n_i has a neighbor node n_j and their distance is d_{ij}. The virtual force applied by



(a) Initial deployment locations

(b) Locations after 4 VFA iterations

Fig. 4. Impact of location adjustment to the establishment of secure links using VFA

n_j to n_i is

$$\overrightarrow{F}_{ij} = \begin{cases} w_A(d_{ij} - d_{th}) \cdot \overrightarrow{v}_{ij} & \text{if} \quad d_{ij} \geq d_{th}; \\ (w_R/d_{ij}) \cdot \overrightarrow{v}_{ji} & d_{ij} < d_{th}. \end{cases} \tag{3}$$

Here $\overrightarrow{v}_{ij}$ is the unit-length pointing from the location of n_i to that of n_j. The total virtual force $\overrightarrow{F}_i$ on n_i is the aggregated virtual force from all of its neighbors, i.e.,

$$\overrightarrow{F}_i = \sum_{j=1}^{Nb} \overrightarrow{F}_{ij}. \tag{4}$$

After computing the virtual force for each node, the node n_i is moved to the direction specified by the aggregated virtual force $\overrightarrow{F}_i$ with a step size equal to its magnitude $|\overrightarrow{F}_i|$.

Fig. 4 shows an example of location updating using the VFA and its impact on secure communications. Initially, 49 sensors are deployed into a 60 × 60 area with a hexagon lattice pattern. A uniform distributed deployment deviation is applied to the initial locations, with the deployment variance $\sigma^2 = 4/3$. This initial deployment is shown in Fig. 4(a) with the established secure links marked as lines connecting the sensor nodes. In this example, the sensing radius is 5 and the communication radius is 9. The sensing coverage achieved by the initial deployment is $\eta = 0.7$ and the average number of secure links per node is $N_{sec} = 2.5$. Next, we apply the VFA to update the sensor locations. After four iterations, the sensing coverage has been improved to $\eta = 0.85$, while N_{sec} has been reduced significantly to $N_{sec} = 1.6$, implying most of the nodes are no longer connected. This is illustrated in Fig. 4(b). At the initial deployment, most nodes form a connected component using the secure links; after four iterations, about half of the nodes are no longer connected with the largest connected group, which reduces the capability of secure communications between the sensor nodes. Our study shows that such a phenomenon is common in sensor location update using virtual force type of schemes. To maintain a comparable sensing coverage while improving secure connectivity, we propose a modified sensor location updating algorithm based on the virtual force framework. We call the modified algorithm *VFSec*, indicating that secure communications is one of the main factors in updating the sensor locations.

2) VFSec Algorithm: As we have seen, there is a tradeoff between the sensing coverage and secure connectivity. For balancing this tradeoff, we define an additional performance metric as

$$\Gamma = w_1\eta + w_2 N_{sec}. \tag{5}$$

The weights w_1 and w_2 are chosen such that $w_1\eta$ and $w_2 N_{sec}$ are approximately in the same value range, so as to achieve a desired tradeoff. Since the sensing coverage is always within $[0, 1]$, and the average number of secure links per node is around 3 in most of our experiments, we choose $w_1 = 1$ and $w_2 = 1/3$ in our experiments.

Our algorithm uses the combined performance metric Γ to measure the optimality of sensor locations, which balances the tradeoff between coverage and secure communications. During each iteration of location adjustment, the algorithm tries to keep the distance between those nodes that can establish secure links closer. To achieve this, we add a new term of virtual force, $\overrightarrow{F}^{sec}$, to the total virtual force. The virtual force $\overrightarrow{F}^{sec}_{ij}$ applied to a node n_i by a neighbor node n_j is as follows

$$\overrightarrow{F}^{sec}_{ij} = \begin{cases} w_s(d_{ij} - D_{sec}) \cdot \overrightarrow{v}_{ij} & \text{if } \ D_{sec} < d_{ij} < R_c \\ & (n_i,\ n_j) \text{ share key;} \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

In computing $\overrightarrow{F}^{sec}_{ij}$, D_{sec} is a threshold distance smaller than the communication radius R_c, d_{ij} is the distance between node n_i and n_j, w_s is the weight assigned to the added virtual force, and $\overrightarrow{v}_{ij}$ is the unit-length vector pointing from the location of n_i to that of n_j. The total virtual force for secure link applied on n_i is $\overrightarrow{F}^{sec}_{i} = \sum_j \overrightarrow{F}^{sec}_{ij}$. This force is added to distance-based forces $\overrightarrow{F}_{ij}$ in Eqn.(4) to compute the total virtual force $\overrightarrow{F}_i$. To update the sensor location, the sensor node n_i is moved along the direction of $\overrightarrow{F}_i$ by a distance equal to the magnitude of vector $|\overrightarrow{F}_i|$. The complete algorithm is described in Algorithm 1.

Algorithm 1 VFSec algorithm

```
Input: sensor locations $\{(x_i, y_i)\}_{i=1}^{n}$ and key index set $\{K_i\}_{i=1}^{n}$
Output: new locations $(x_1^{opt}, y_1^{opt})$, ... $(x_n^{opt}, y_n^{opt})$
/* Initialization */
Compute $\Gamma^{opt}$ using Eqn.(5) with $(\{(x_i, y_i)\}_{i=1}^{n}, \{K_i\}_{i=1}^{n})$
$(x_i^{opt}, y_i^{opt}) \longleftarrow (x_i, y_i)$ for $1 \leq i \leq n$
/* Iteration */
for $i = 1$ to MAX-ITERATION do
  for each sensor node $n_i$ do
    Calculate $\overrightarrow{F}_{ij}$ using the formulation in [5]
    Calculate $\overrightarrow{F}^{sec}_{ij}$ using (6)
    $\overrightarrow{F}_i \longleftarrow \sum \overrightarrow{F}_{ij} + \sum \overrightarrow{F}^{sec}_{ij}$
  end for
  /*Update sensor locations*/
  $(x_1', y_1') \longleftarrow (x_i + F_{ix}, y_i + F_{iy})$ for $1 \leq i \leq n$
  Compute $\Gamma$ using Eqn.(5) with $(\{(x_i', y_i')\}_{i=1}^{n}, \{K_i\}_{i=1}^{n})$
  if $\Gamma > \Gamma^{opt}$ then
    $(x_i^{opt}, y_i^{opt}) \longleftarrow (x_i', y_i')$ for $1 \leq i \leq n$
  end if
end for
```

3) Simulation Results and Discussions: To study the performance of VFSec, we have performed three experiments and compared the VFA and VFSec in terms of sensing coverage and secure link establishment. Throughout these experiments, we set the communication radius R_c as twice the sensing radius R_s. This is to ensure that even when the sensing range is very small and two neighboring sensors are barely disjointly placed (i.e. the distance between two neighboring sensor nodes is $2R_s$), it is still possible to establish a communication link between the two sensors. In all the experiments, both the VFA and VFSec algorithms are run for seven iterations.

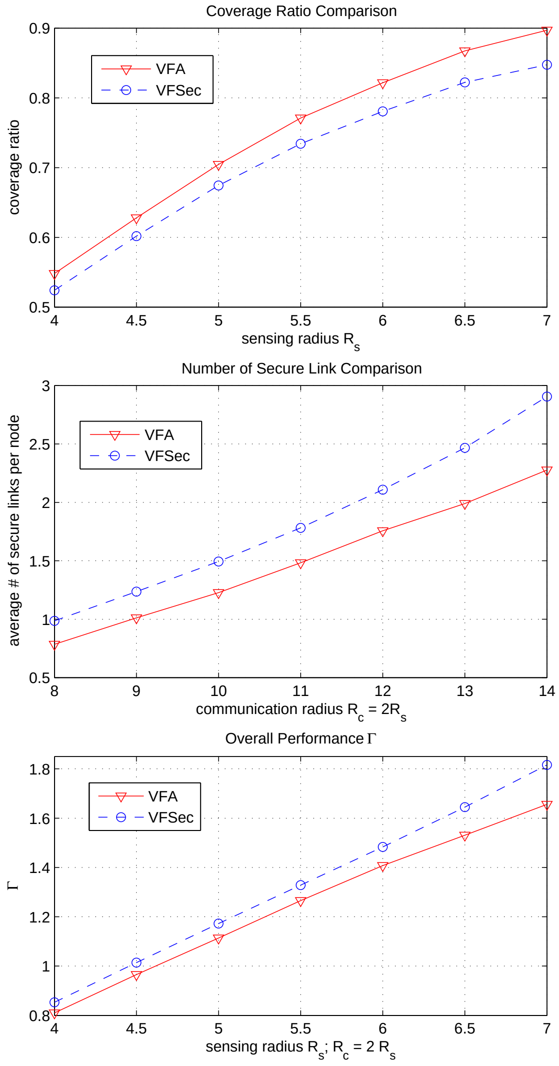


Fig. 5. Comparison of VFA and VFSec with uniform random initial deployment.

In the first experiment, we place 36 sensors nodes uniformly in a 50 × 50 area. Using VFA and VFSec, the locations of the sensors are adjusted. The sensing coverage and the number of secure links per node are recorded. We repeat such experiment 400 times and computed the average coverage and the number of secure links per node under different sensing and communication radius. From the results shown in Fig. 5 we can

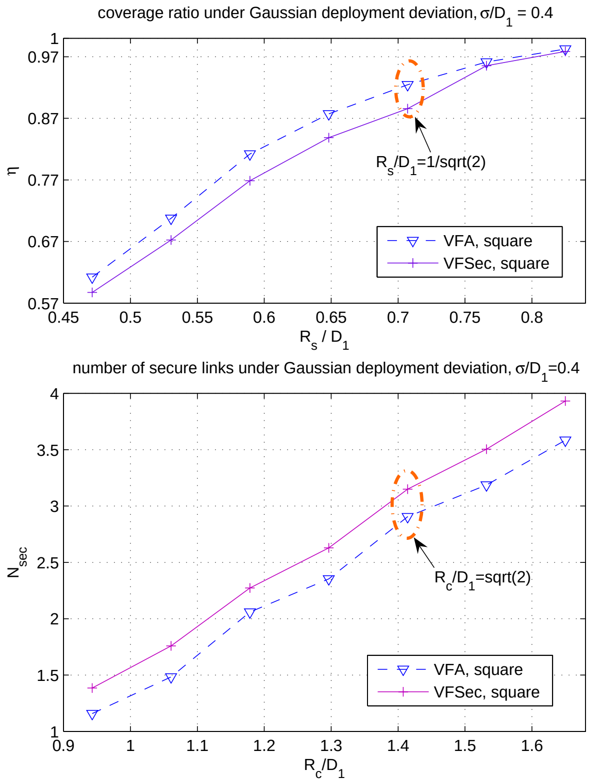


Fig. 6. Comparison of VFA and VFSec using square deployment lattice under Gaussian deployment deviation.

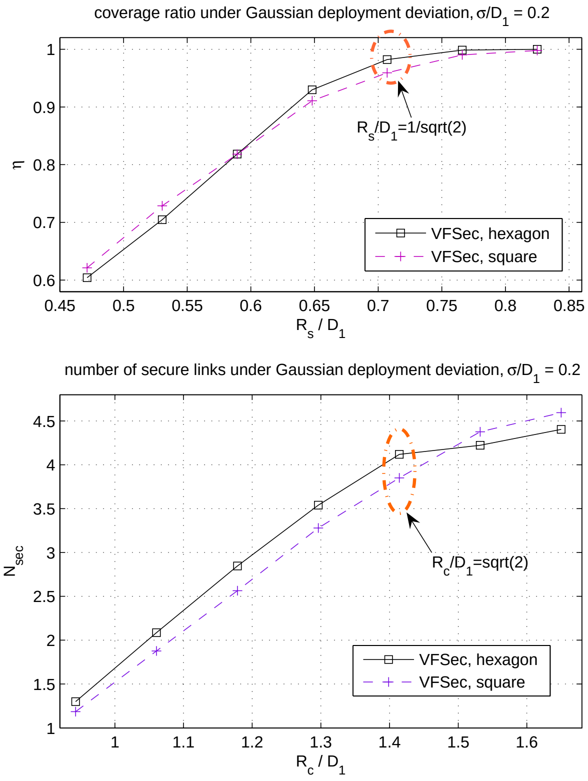


Fig. 7. Comparison of deployment lattice using VFSec under Gaussian deployment deviation.

see that, the proposed VFSec algorithm can improve the average number of secure links by approximately 15-20%, with a small reduction in the sensing coverage by approximately 2-5%. In addition, the VFSec consistently achieves a better performance in terms of the overall performance metric Γ in Eqn.(5).

In the second experiment, we compare the VFSec and VFA using square lattice deployment under Gaussian deployment deviation. We fix the node density and the deviation parameter $\sigma = 0.4D_1$. From Fig. 6, we can see that VFSec improves the average number of secure links per node, with a small compromise in the sensing coverage. In this experiment, we have excluded the boundary nodes of the square deployment area in computing the sensing coverage and number of secure links. Thus the results can be viewed as if the performance is evaluated in an infinitely large area. In spite of the difference in accounting the performance, the results in Fig. 6 shows the same trend as in Fig. 5.

In the third experiment, we compare the square and hexagon lattice deployment using the corresponding location-based key pre-distribution. We use the proposed VFSec algorithm for location updating and the results are obtained for different communication and sensing radius under small deployment deviation ($\sigma/D_1 = 0.2$). Fig. 7 shows that the hexagon and square lattices achieve comparable sensing coverage. In terms of the average number of secure links per node, the hexagon lattice achieves a better performance when the normalized communication radius R_c/D_1 is in the approximate range of $[1, 1.5]$; outside this range, the square lattice performs better. Such a result again shows that there is no all-time winner in terms of deployment lattice, as is shown in the step function connectivity graph in Fig. 2 for the ideal hexagon and square lattices. When designing secure sensor networks, the deployment lattice as well as system parameters, such as the communication radius, should be taken into consideration.

4) Implementation Issues: Similar to the VFA algorithm, the VFSec can be performed by the cluster head of the sensor nodes. As the cluster head is equipped with better computing resource, it would save the computing power of each individual sensors. Further, as indicated in Algorithm 1, the iterations for updating sensor locations can be performed by the cluster head and only the final results obtained are sent back to the sensor nodes.

Therefore the actual location adjustment is performed only once by each sensor. To run the VFSec algorithm, the cluster head needs to collect the sensors' key index sets and their current locations. The key indices are ID's assigned to secret keys, which is used in the shared key discovery phase in key pre-distribution schemes [13] [14].

The advantage of using a cluster head to implement the location updating algorithm is that the power consumption of individual sensors can be reduced. When the cluster head is not available, the algorithm can be performed by the individual sensors only based on its neighborhood information. When the VFSec is performed by individual sensors, Algorithm 1 must be adjusted to better suit the distributed implementation. The sensors need to perform movement adjustment after each iteration. At the same time, computing and comparing the global performance metric Γ^{opt} as in Algorithm 1 would not be feasible; and the number of iterations must be limited to reduce the power consumption in sensor movement.

B. Sensor Location Adjustment Based on Vector Quantization

One limitation of the VFSec algorithm is that in order to achieve a better secure connectivity, the sensing coverage is somewhat sacrificed. The reason is that the virtual force based approach simplifies the problem in a two-dimensional area to a set of vectors. In this part, we propose a new approach for updating sensor locations that can explore more freedom in the two-dimensional space to jointly optimize sensing coverage and secure communications.

The problem of covering a region using distributed sensor nodes is analogous to the vector quantization problem in signal compression [23]. In the sensing coverage problem, each node can sense its nearby region with certain accuracy. The goal is to maximize the total coverage given a limited number of sensors. In the quantization problem, each point in the k-dimensional space is associated with a representative point in the codebook. The goal is to use a limited number of points to represent all points in the region with minimum error. In two-dimensional space, if the input signal is statistically uniformly distributed, the minimum-error quantization lattice and the most efficient covering lattice are the same hexagon lattice [16] as we have seen in Fig. 1(b). This has motivated us to employ insights in the vector quantization literature to explore solutions for sensor deployment.

1) The Weighted Centroid Algorithm: Several prior works have proposed location updating algorithms that are similar to the two-dimensional vector quantization solution [3] [4]. In particular, the MinMax algorithm proposed in [4] computes the Voronoi cell V for each sensor node n, and move the sensor to the minmax location $\mathbf{x}_{minmax}$ so that the maximum distance from the new location to any point in the cell V is minimized, *i.e.*,

$$\mathbf{x}_{minmax} = \arg\min_{\mathbf{x}}\{\max_{\forall \mathbf{y}\in V} d(\mathbf{x},\mathbf{y})\}.$$

It has been shown in [4] that the minmax location is the center of the minimum-radius circum-circle of the Voronoi cell associated with each node.

Inspired by these works, we propose a new approach for updating sensor locations based on the Lloyd-Max quantization algorithm [23]. We consider that the sensor has a communication range R_c and can know the locations of its neighbors and its own location [3]. Furthermore, the proposed approach will allow the sensors to take secure communication as a factor in updating locations.

Our proposed algorithm aims at minimizing the weighted average distance of a sensor node to the points in its Voronoi cell. We choose a weighted square distance as the distance metric. Suppose in the two-dimensional space, there are N points uniformly distributed at locations $\{(x_i, y_i)\}_{i=1}^{N}$ inside the Voronoi cell formed of a sensor node located at (x_0, y_0) and its neighbors. Each point is associated with a weight w_i. Then the weighted square distance D_w is

$$D_w = \frac{1}{N}\sum_{i=1}^{N} w_i[(x_0 - x_i)^2 + (y_0 - y_i)^2]. \qquad (7)$$

From the classic vector quantization results [23], we know that given the set of points $\{(x_i, y_i)\}_{i=1}^{N}$ and the weight $\{w_i\}_{i=1}^{N}$, the optimal value for (x_0, y_0) that minimizes the weighted distance D_w is

$$x_0^{opt} = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i}; \quad y_0^{opt} = \frac{\sum_{i=1}^{N} w_i y_i}{\sum_{i=1}^{N} w_i}. \qquad (8)$$

For the dual problem, we know directly from the definition of Voronoi cell that, for any point p located inside a voronoi cell V_i of sensor n_i, the node n_i is closer to the point p than any other node outside the Voronoi cell V_i. Thus we have naturally obtained an iterative algorithm for location updating.

The proposed algorithm works as follows. In each iteration, each sensor n_i discovers its neighbors and generates its Voronoi cell V_i according to the neighbor locations. Next, the sensor node generates a set of uniformly distributed grid points $\{(x_i, y_i)\}_{i=1}^{N}$ inside V_i and assign weight to each point. Then the node will compute its new location (x_0', y_0') that can minimize the weighted square distance D_w according to Eqn.(8). The simplest weight assignment is to assign equal weight of one to all points. When different weights are assigned to the sampling grid points, the solution (x_0^{opt}, y_0^{opt}) is the *centroid* of the Voronoi cell with respect to weight assignment $\{w_i\}$. Therefore we refer to the algorithm as the weighted centroid (WTC) algorithm and describe it in Algorithm 2.

Algorithm 2 The Weighted Centroid Algorithm

```
Input: sensor location $(x_0, y_0)$, neighbor locations $\{(x_i, y_i)\}_{i=1}^{Nb}$
Output: movement vector $\vec{v}$
Compute Voronoi cell $V$
Generate uniform grid points $\{(x_i, y_i)\}_{i=1}^{N} \in V$
Assign weight $\{w_i\}_{i=1}^{N}$ using Alg. 3
Compute updated location $(x_0', y_0')$ using (8)
Compute the movement vector $\vec{v} \longleftarrow [(x_0', y_0') - (x_0, y_0)]$
/* adjustment for stability */
if $|\vec{v}| > R_s/2$ then
    $\vec{v} \longleftarrow R_s\vec{v}/(2|\vec{v}|)$
end if
```

Algorithm 3 The Weight Assignment Procedure

```
Input: neighbor locations $\{\mathbf{x_i}\}_{i=1}^{Nb}$, sampling points $\{\mathbf{p_i}\}_{i=1}^{N}$, $R_c$, and $w_{sec}$
Output: weight vector $\{w_i\}_{i=1}^{N}$
$w_i \longleftarrow 1$ for $1 \le i \le N$
for $i = 1$ to $Nb$ do
    for $j = 1$ to $N$ do
        if Sec($n_i$, $n_c$) and $0.7R_c \le d(\mathbf{x_i}, \mathbf{p_j}) \le 0.95R_c$ then
            $w_i \longleftarrow w_i + w_{sec}$
        end if
    end for
end for
```

To jointly consider secure communication and sensing coverage, we propose the following weight assignment procedure. For each sensor node n_i, after the Voronoi cell has been formed and the grid points are generated, the base weight for each grid point inside the cell is 1. If the node n_i already has a secure link with a neighbor node n_j, each grid point that falls into the ring area centered at n_j, and between the radius $0.7R_c$ and $0.95R_c$ will be assigned an extra weight of $w_{sec} = 0.5$. An algorithmic description of the weight assignment procedure is presented in Algorithm 3. In Algorithm 3, Nb refers to the number of neighbors of a center node n_c; and the function Sec(n_i, n_j) is an indicator function, which returns $true$ if node n_i and n_j has a secure communication link and $false$ otherwise.

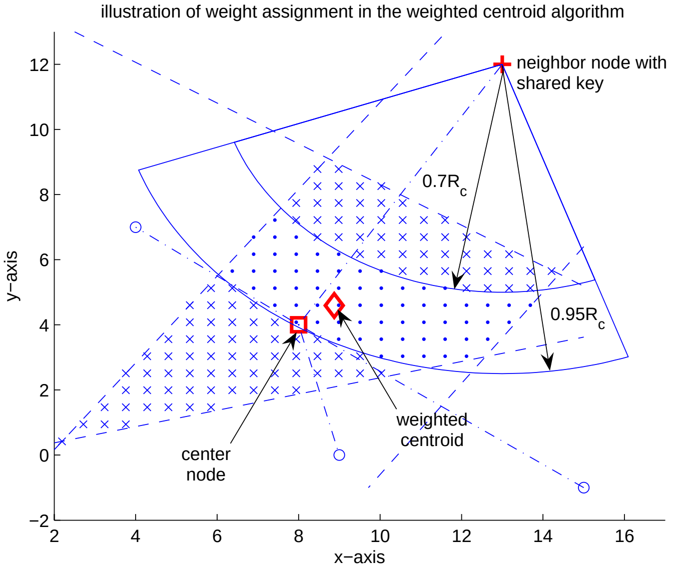


Fig. 8. Illustration of the weight assignment in the weighted centroid algorithm.

In Fig. 8 we illustrate the weight assignment procedure. In this figure, the center node is shown as a square, its neighbor nodes are shown as circles, and the node that already has a secure link with the center node is shown as a plus sign. The Voronoi cell is shown as the shaded area. The grid points are shown either as cross or as dots, where a dot indicates that grid point is inside the ring area between radius $0.7R_c$ and $0.95R_c$ of its secure communication neighbor. The weighted centroid is shown as a diamond in Fig. 8. We can see that the updated location is within the center of the ring area, at the same time tends to cover more areas in the Voronoi cell.

The choice of the ring area to be within $[0.7R_c, 0.95R_c]$ is due to the joint consideration of sensing and communications. When the center node is far away from its neighbor, the ring-based weighting tend to pull the center node towards its neighbor. When the center node is too close to its neighbor,

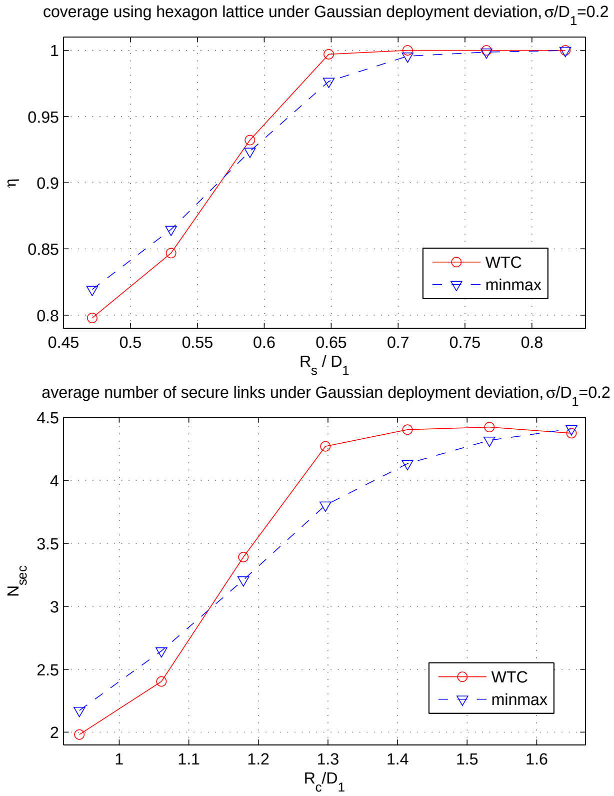


Fig. 9. Comparison of the weighted centroid and minmax algorithm, small Gaussian deployment deviation, hexagon lattice deployment and location-based key pre-distribution.

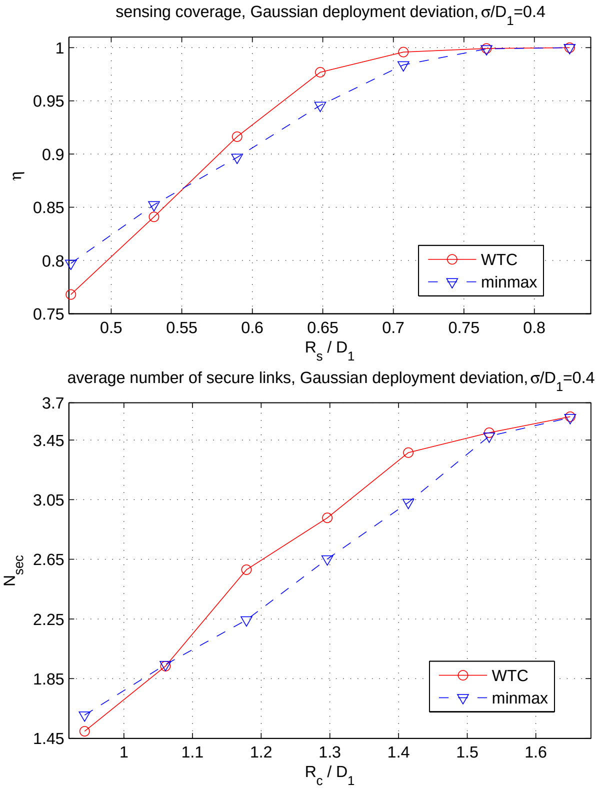


Fig. 10. Comparison of the weighted centroid and minmax algorithm, large Gaussian deployment deviation, hexagon lattice deployment and location-based key pre-distribution.

the ring-based weighting tend to push the center node away from its neighbor. Thus this weight assignment maintains the communication connectivity between the center node and its neighbors, at the same time avoids too much wasteful overlaps between their sensing regions.

2) Simulation Results and Discussions: We study the performance of the weighted centroid (WTC) algorithm using several experiments. We compare it with the performance of the MinMax algorithm proposed in [4], which is known as one of the best schemes in sensor location updating for improving sensing coverage.

In Fig. 9 and Fig. 10 we compare the sensing coverage ratio and the average number of secure links per node achieved by the proposed WTC algorithm and by MinMax, with respect to the normalized sensing/communication radius. We set communication radius $R_c = 2R_s$. The initial deployment uses hexagon lattice and the key pre-distribution uses location-based scheme. Each node has preloaded 100 keys. Both algorithms are run locally by each sensor for four iterations. In these experiments we have excluded all boundary nodes, which allows the results to be interpreted as the expected performance in an infinitely large deployment field. The comparison results can be summarized as follows: (1) when the sensing/communication radius is small, MinMax out perform WTC in both sensing coverage and the average number of secure links; (2) as the sensing/communication radius becomes moderately large, WTC outperforms MinMax in both performance categories; (3) when the sensing/communication radius becomes large enough, the performances of the two schemes will converge.

These results can be interpreted from resource allocation and optimization perspectives [28]. The minmax criterion employed by the MinMax algorithm emphasizes *fairness*, *i.e.*, even when a point in Voronoi cell is very far away from the current sensor location, the location adjusting algorithm tries to cover that point. In contrast, the criterion employed by WTC emphasizes *efficiency*. It tries to minimize the weighted square distance from the sensor node to all points in its Voronoi cell, which is a more greedy philosophy compared to the minmax criterion. In sensor networks, the sensing and communication range are valuable resources to be allocated to the deployment field. The results shown in Fig. 9 and Fig. 10 indicate that, with a resources-scarce situation (relative to the resource needed for a full coverage/connectivity), the MinMax is a better criterion, with a moderate enough amount of resources, WTC outperforms MinMax. To quantify the demarcation for resource-scarce and resource-abundant situations, we note that in the ideal hexagon lattice, the normalized sensing radius needs to be at least $R_s/D_1 = \sqrt{2/\sqrt{27}} \approx 0.62$ to achieve full coverage, and the normalized communication radius needs to be at least $R_c/D_1 = \sqrt{2/\sqrt{3}} \approx 1.07$ to achieve full connectivity with the neighbors, which is a pre-requisite for establishing secure links. As shown in Fig. 9 and Fig. 10, usually the proposed WTC outperforms the MinMax algorithm when the normalized sensing and communication radius are beyond their respective thresholds of 0.62 and 1.07. In practical situations, since the resource budgets are known prior to the design of sensor networks, dynamically determining which criterion to use will best serve the purpose of improving sensing coverage and establishing secure links.

In a separate experiment, we simulated the WTC and MinMax algorithms under random deployment with uniform distribution over the entire field. We place a total of 49 sensors into a 60×60 area and use the basic key pre-distribution scheme for establishing secure links. In this experiment, as it is not possible to exclude the boundary nodes in calculating the average node degree, the average number of secure links drops significantly when compared to the previous lattice-based experiments. In spite of the change in accounting the performance, the simulation results presented in Fig. 11 shows the same trend as Fig. 9 and Fig. 10 in that, the WTC algorithm achieves better performances in both performance categories in the resource-abundant situations, and performs worse than the MinMax in the resource-scarce situations.

3) Implementation Issues: In the WTC algorithm, the grid points are chosen to discretize the computation of the centroid instead of using a continuous integration over the Voronoi cell. As power consumption is a major concern in sensor networks, the grid points can be chosen as sparse or dense according to the power budget, thus trading off computation accuracy with energy. If the only goal of location adjustment is to maximize the coverage, the same weight can be assigned to all w_i's, *i.e.*, $w_i = 1$ for all points.

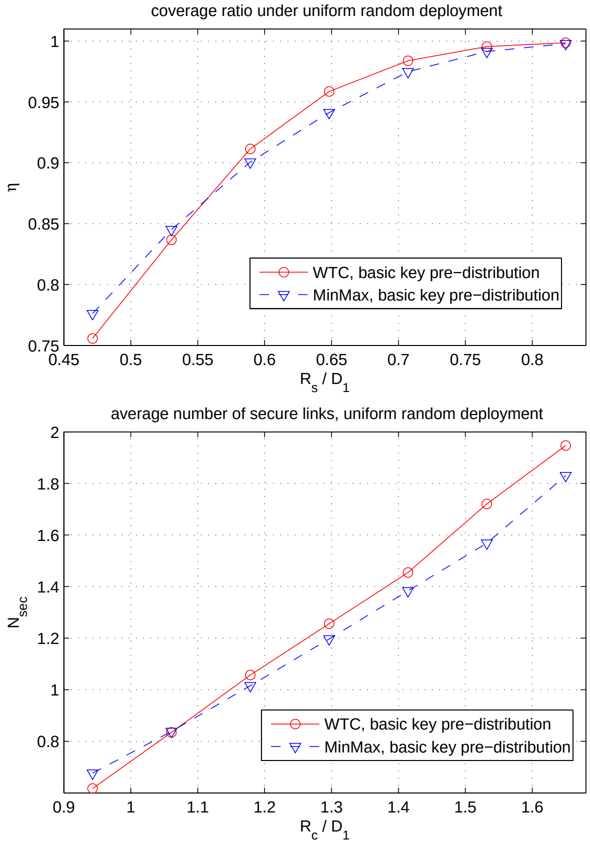


Fig. 11. Comparison of the weighted centroid and minmax algorithm, uniform random deployment with basic key pre-distribution.

Unlike the VFSec algorithm, the WTC is more suitable to be performed by individual sensors. This is because computing both the Voronoi cell and the weighted centroid can be done locally. The simulation results have shown that the locally computed location updates using WTC and MinMax can outperform the location updates run by a cluster head using the VFA and VFSec. However, performing Voronoi-cell based scheme generally requires more computation than the schemes based on virtual force, which is due to the added dimensionality in computing the updated locations.

V. Conclusions

In this paper, we have investigated the impact of sensor deployment on the performance of sensing coverage and secure connectivity. For static sensor deployment, we have investigated the hexagon and square lattice topology and compared them with the random deployment. We show that the two lattice topology exhibits range-dependent performance and there is no all-time winner in the context of secure connectivity. For designing secure sensor networks, the system parameters, such as sensing and communication range, should be jointly considered with the deployment topology.

When sensor locations can be adjusted after the initial deployment, to jointly optimize sensing covering and secure connectivity, we have proposed two sensor location updating algorithms, the VFSec and the WTC algorithm. The simulation results show that the WTC algorithm outperforms the existing algorithms in both performance categories under moderate to abundant node density, while VFSec achieves a superior tradeoff in both performance categories than the existing virtual force based algorithms.

REFERENCES

[1] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh and D. Rubenstein, "Energy efficient computing for wildlife tracking: design rradeoffs and early experiences with ZebraNet", in *Proc. of ACM ASPLOS'02*, San Jose, CA, USA, Oct. 2002.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey", *Computer Networks*, vol. 38, 2002, pp 393–422.

[3] S. Megerian, F. Koushanfar, M. Potkonjak and M. B. Srivastava, "Coverage problems in wireless Ad-Hoc sensor networks", in *Proc. of the 2001 IEEE/ACM INFOCOM*, Apr. 2001.

[4] G. Wang, G. Cao and T. L. Porta, "Movement-assisted sensor deployment", in *Proc. of the 2004 IEEE/ACM INFOCOM*, Hongkong, China, Mar. 2004.

[5] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces", in *Proc. of the 2003 IEEE/ACM INFOCOM*, pp. 1293-1303, Apr. 2003.

[6] 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 *Proc. of the 2004 IEEE INFOCOM*, March 2004.

[7] D. Liu and P. Ning, "Location-Based Pairwise Key Establishment for Static Sensor Networks", in *Proc. of the ACM Workshop on Security of Ad Hoc and Sensor Networks*, Oct. 2003.

[8] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann and F. Silva, "Directed diffusion for wireless sensor networking", *IEEE/ACM Tran. on Networking*, vol. 11, no. 1, Feb. 2003, pp 2–16.

[9] K. Lieska, E. Laitinen and J. Lahteenmaki, "Radio coverage optimization with genetic algorithms", *IEEE Inter. Symp. on Personal, Indoor and Mobile Radio Communications*, vol. 1, pp 318 – 322, Sept. 1998.

[10] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges", in *Proceedings of IEEE*, vol. 91, no. 8, pp 1247–1256, Aug. 2003.

[11] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks", in *Proceedings of ACM Mobicom'99*, Seattle, USA, 1999.

[12] H. Chan and A. Perrig, "Security and Privacy in Sensor Networks", *IEEE Computer Magazine*, vol. 36, issue 10, Oct. 2003.

[13] 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*. 2002, pp. 41–47, ACM Press.

[14] H. Chan, A. Perrig and D. Song, "Random key predistribution schemes for sensor networks", in *Proc. of the 2003 IEEE Symposium on Security and Privacy*, May 2003.

[15] R. Kershner, "The number of circles covering a set", *American Journal of Mathematics*, vol. 60, pp 665–671, 1939.

[16] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 3rd Ed., Springer, 1991.

[17] S. Megerian, F. Koushanfar, G. Qu and M. Potkonjak, "Exposure in wireless Ad-Hoc sensor networks", in *Proc. of the 2001 ACM MobiCom*, July 2001.

[18] R. Williams, *The Geometrical Foundations of Natural Structure: a Book of Design*, Dover Publications, 1979.

[19] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, 1996.

[20] G. Wang, G. Cao, T. L. Porta and W. Zhang, "Sensor relocation in mobile sensor networks", in *Proc. of the IEEE/ACM INFOCOM*, Miami, Florida, USA, Mar. 2005.

[21] A. Howard, M. J. Matarić and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem", in *Proc. of DARS'02*, Fukuoka, Japan, June, 2002.

[22] D. Pescovitz, "Robugs: smart dust has legs", http://www.coe.berkeley.edu/labnotes/0903/pister.html

[23] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Springer, 1991.

[24] W. Du, J. Deng, Y. S. Han and P. K. Varshney, "A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks", in *Proc. of the 2003 ACM CCS*, Oct. 2003.

[25] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks", in *Proc. of the 2003 ACM CCS*, Oct. 2003.

[26] Y. Zhou, Y. Zhang and Y. Fang, "LLK: a link-layer key establishment scheme for wireless sensor networks", *IEEE Wireless Communications and Networking Conference (WCNC 2005)*, Mar. 2005, New Orleans, LA, USA.

[27] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1988.

[28] G. Song and Y. Li, "Cross-layer optimization for OFDM wireless networks-part I: theoretical framework", *IEEE Transactions on Wireless Communications*, vol. 4, issue 2, Mar. 2005, pp. 614–624.

APPENDIX

In this appendix, we analyze the link compromise probability in the location-based key pre-distribution scheme in [6] and compare it with that of the basic scheme [13]. This supports the results presented in Section III-A.

Let us denote the size of the key pool in the basic scheme by P, and there are x nodes compromised. For a given link e in the basic scheme, it has been shown in [13] that the probability that the link e is compromised is

$$P_{basic} = \Pr(e|x) = 1 - (1 - \frac{m}{P})^x.$$

Next, we consider in the group-based scheme [6], the size of the group key pool is S and the number of sensor groups is N. We require the total number of distinct keys in the group based scheme to be the same as in the basic scheme. As each distinct key appears in exactly two group key pools, we have $P = (NS)/2$. In the group-based scheme, suppose the given link e uses key K_e. This key K_e is in the key pools of exactly two groups, denoted by G_1 and G_2. Only when a compromised node $n^{(c)}$ is from one of the two groups, the link e can potentially be compromised by $n^{(c)}$. When the compromised nodes are i.i.d. uniformly distributed among all groups, denoting the probability that a compromised node $n^{(c)}$ falls into group G_1 or G_2 by p, we have

$$p = \Pr(n^{(c)} \in \{G_1 \cup G_2\}) = 2/N.$$

In the group-based scheme, the probability that link e is compromised given x nodes are compromised is

$$\begin{aligned} P_{group} &= \Pr(e|x) \\ &= \sum_{k=0}^{x} \Pr(e|(k \text{ out of } x) \in \{G_1 \cup G_2\}) \cdot \\ &\quad \Pr((k \text{ out of } x) \in \{G_1 \cup G_2\}) \\ &= \sum_{k=0}^{x} [1 - (1 - \frac{m}{S})^k] \binom{x}{k} p^k (1-p)^{x-k}. \end{aligned}$$

For function $f(\epsilon) = (1-\epsilon)^a$ with $\epsilon \approx 0$, the first-order approximation at $\epsilon = 0$ using Taylor expansion is $f(\epsilon) \approx 1 - a\epsilon$. As the key pool size P and the group key pool size S are much larger than the key ring size m, both $\frac{m}{P}$ and $\frac{m}{S}$ are close to zero, we can apply first-order approximation to both $(1-\frac{m}{S})^k \approx 1 - \frac{m}{S}k$ and $(1-\frac{m}{P})^x \approx 1 - \frac{m}{P}x$. Thus we arrive at

$$\begin{aligned} P_{basic} &\approx 1 - (1 - x \cdot \frac{m}{P}) \approx x \cdot \frac{m}{P} \\ P_{group} &\approx \sum_{k=0}^{x} \frac{m}{S} k \binom{x}{k} p^k (1-p)^{x-k} \\ &= mpx/S \end{aligned} \tag{9}$$

Since we have $S = 2P/N$ and $p = 2/N$, substituting these into Eqn. (9), we obtain

$$P_{group} \approx x \cdot \frac{m}{P} \approx P_{basic}.$$

This shows that the link compromise probability of the basic scheme and the group-based scheme are approximately the same with the fixed key ring size m.