

# Detecting Phantom Nodes in Wireless Sensor Networks

Joengmin Hwang, Tian He, Yongdae Kim

Department of Computer Science, University of Minnesota, Minneapolis

{jhwang,tianhe,kyd}@cs.umn.edu

**Abstract**—In an adversarial environment, various kinds of security attacks become possible if malicious nodes could claim fake locations that are different from where they are physically located. In this paper, we propose a secure localization mechanism that detects the existence of these nodes, termed as phantom nodes, without relying on any trusted entities, an approach significantly different from the existing ones. The proposed mechanism enjoys a set of nice features. First, it does not have any central point of attack. All nodes play the role of verifier, by generating *local map*, i.e. a view constructed based on ranging information from its neighbors. Second, this distributed and localized construction results in quite strong results: even when the number of phantom nodes is greater than that of honest nodes, we can filter out most phantom nodes. Our analysis and simulations under realistic noisy settings demonstrate our scheme is effective in the presence of a large number of phantom nodes.

## I. INTRODUCTION

With thousands of tiny devices, Wireless Sensor Networks (WSNs) can support ubiquitous surveillance with a very low profile, and they can be quickly deployed without infrastructure. These features make them attractive for a wide variety of applications such as environmental and habitat monitoring [1], surveillance and tracking for military [2]. For many of these applications, the location of a node plays an important role. It can be used for 1) identifying the location of specific events (for example, tracking the positions of enemy's tanks), 2) location-based routing algorithms (e.g. such as the geographic routing [3] and the geographic hash table [4]), or 3) other location based services such as sensing coverage [5]. These applications run correctly when the localization error is limited to a certain range [6]. However, if malicious nodes (attackers) can distort the coordinate system severely, the performance of these applications could degrade significantly. To address these issues, various methods [7], [8], [9], [10], [11], [12], [13], [14] are proposed. They provide a set of nice mechanisms to detect and filter out compromised nodes and anchors. Most approaches depend on a few trusted entities (nodes or anchors), requiring at least the majority of these entities are not compromised. We argue that since the number of trusted entities in these approaches is relatively small, it would be relatively easy to break. Naturally, we raise the following question: *Is it possible to design a decentralized secure localization algorithm that can detect phantom nodes without requiring any trust entities?* The objective and contribution of this work lie in our answer to this challenging question.

This research was supported, in part, by University of Minnesota McKnight-Land Grant Professorship award, and NSF grant CNS-0626614, CNS-0615063 and CNS-0626609.

In this paper, we are targeting the scenarios where attackers announce phantom nodes, who fake their ranging information, in proximity of legitimate nodes. Especially we focus on the development of the local map for individual nodes. A local map is a visual representation on the locations of neighbors of a node, which can be constructed correctly by verifying all location claims of its legitimate neighbors and filtering out phantom nodes generated by attacks.

Briefly, to find an actual local map without including phantom nodes, we project the neighboring nodes on a virtual plane and identify the inconsistency exhibited by the phantom nodes. Since there are no trusted entities, the process is *speculative* in nature. Interestingly, we demonstrate that this speculative process can filter out the phantom node with a very high probability when the process is repeated multiple times. In addition, since this novel speculative process mandates no agreements among neighboring nodes, it leads to two immediate benefits: First, a node's compromised decision does not propagate to affect other nodes' decisions. Second, much less information exchange is required, leading to less energy consumption. Beside these two benefits, our approach has the following major contributions: First, we contribute two rules to prevent phantom nodes generating consistent ranging claims. Second, our approach recovers a local map agreed by the majority of consistent information. It projects regular nodes at their locations, detect/filter phantom nodes, and identify the source of inconsistent rangings, requiring no trusted anchors or verifiers. We demonstrate that we can successfully filter out phantom nodes even the number of phantom nodes is much larger than the honest nodes. Third, our approach can use any ranging technique, not specially requiring distance bounding technique for location verification [9], [12], [13].

The remainder of the paper is organized as follows: Section II introduces the assumptions and Section III provides an overview. The details of our approach are described in Section IV. We present the experimental results in Section V and conclude in Section VI.

## II. PRELIMINARIES

We assume all legitimate communication channels are established bidirectionally: if Node  $i$  hears Node  $j$ , then Node  $j$  hears Node  $i$ . In asymmetric links, bidirectional links can be easily established through a two-way handshaking. We also assume a reasonable network density (e.g.,  $> 10$  nodes per radio range). For the sake of clarity, we describe the protocol in a two-dimensional plane. However, our approach can be

applied to higher-dimensional spaces as well. The following notations are used throughout the paper.

- $v$ : a node which verifies the locations of its neighbors
- $Nbr(v)$ : the node set consisting of  $v$ 's neighbors and  $v$
- $p_k$ : the location of node  $k$  on virtually computed local plane
- $N$ : the number of neighboring nodes
- $M$ : the number of inter-node distance measurements
- $d_{ij}$ : the **physical** distance between nodes  $i$  and  $j$ .
- $\hat{d}_{ij}$ : the **measured** distance to node  $j$  by  $i$ .
- $\tilde{d}_{ij}$ : the **computed** distance between nodes  $i$  and  $j$
- $D$ : a set of distance measurements, i.e.  $\{\hat{d}_{ij} \mid i, j \in A\}$

### III. OVERVIEW

The main idea of our approach is based on two factors: First, we prevent the phantom nodes from generating consistent ranging (distance) claims<sup>1</sup> to multiple honest nodes. Second, if the phantom nodes generate a set of inconsistent ranging claims, we can detect them by our proposed speculative method.

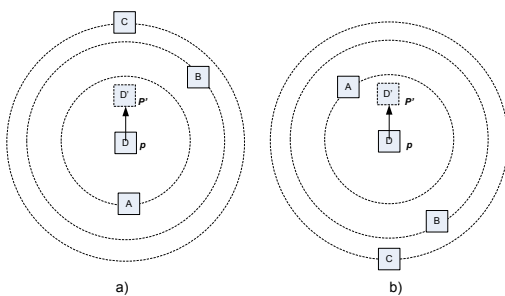


Fig. 1. The difficulty in generating consistent ranging claims

Our design only allows a node to claim about its *distances* to other neighboring nodes, not its own *location*. Therefore, to disrupt the operation of location-dependent applications, (a set of) malicious nodes, whose goal is to create a phantom node, must fake a set of distances to all of its neighboring nodes. If the locations of neighboring nodes are known *a priori*, a set of fake, albeit consistent, ranging distances can be easily created by calculating the distances from a fake location to each of its neighbors' location. Therefore it is important for our design to *hide* the location information during the phase of ranging. Without the location information of the neighboring nodes, it is hard for an attacker to generate a set of consistent ranging values (distances), and hence to fake itself into a different physical location. For example, as shown in Figure 1a, suppose an attacker  $D$  at the location  $p$  obtains three ranging distances in the 2-D space from three honest nodes  $A$ ,  $B$  and  $C$ , it can only conclude that  $A$ ,  $B$  and  $C$  are located at the edges of three concentric circles centered at  $p$ . To claim a different physical location  $p'$  within the 2-D space, the attacker  $D$  needs to fake three different ranging distances that are consistent. Without knowing the precise locations of the neighbors, such consistency is

<sup>1</sup>Here, a set of ranging claims is said to be consistent, when such claims can project a node into a physical location in the 2-D or 3-D space (in case of 3-D localization) and the distances between this physical location and other nodes' locations match the claims.

difficult to achieve. As shown in Figure 1a, to move from the position  $p$  to  $p'$ , the attacker  $D$  needs to claim two shorter ranging distances to Nodes  $B$  and  $C$ , but a longer ranging distance to Node  $A$ ; However in case of Figure 1b, the attacker  $D$  needs to claim the opposite. Since the locations of  $A$ ,  $B$  and  $C$  are unknown, the attacker cannot decide which claim to make. We note that a sensor network normally has a high node density ( $\gg 10$ ), which makes a consistent ranging claim practically impossible without the neighbors' location information.

Briefly, to prevent phantom nodes generating a set of fake, albeit consistent, ranging claims, we should follow two simple design rules: 1) *accepting only ranging claims, not location claims* and 2) *hiding the location information during the ranging phase*. Once the consistent ranging claims by phantom nodes are prevented, we can identify the phantom nodes by detecting the inconsistent ranging claims, which is addressed in the rest of the paper.

### IV. THE DETAILED APPROACH

In this section, we focus on identifying the phantom nodes that generate inconsistent ranging from/to the set of honest nodes. For simplicity, we describe a two-dimensional localization. Formally, we stated the problem as follows:

**Definition:** A set of nodes is *consistent*, if they can be projected on the unique Euclidean plane (in 3-D case, Euclidean space), keeping the measured distances among themselves.

**Problem:** Given a node set  $Nbr(v)$  that consists of a node  $v$  and its neighbors, and a distance set  $D$  that consists of the measured distance, denoted by  $\{\hat{d}_{ij} \mid \hat{d}_{ij} = \hat{d}_{ji}, i, j \in Nbr(v), i \neq j\}$ , find the largest consistent subset of  $Nbr(v)$ .

We divide the algorithm into two main phases: distance measurement phase and filtering phase. In the first phase, each node measures the distances to its neighbors. In the second phase, each node projects its neighboring nodes to a virtual local plane to determine the largest consistent subset of nodes. After the completion of the two phases, each node establishes a local view without phantom nodes. Such a local view is useful in many services such as location-based routing and sensing coverage. Alternatively, any local coordinate system can be reconciled into a unique global coordinate system. The following two sections describe the phases of the approach in more detail.

#### A. Distance Measurement Phase

Each node  $v$  measures the distances to neighbors and disseminate these measurements back to its neighbors. More specifically, each node  $v$  has following distance measurement steps:

- 1) Node  $v$  first measures distance  $\hat{d}_{vi}$  to each neighbor  $i$  through a certain ranging method such as TDOA or TOA. (Note that  $\hat{d}_{vi}$  denotes the distance measurement to the neighbor  $i$  by  $v$ .)
- 2) Node  $v$  then announces the measured distances. The announcement message includes *id* of the node  $v$ , *id* of the node  $i$  and distance measurement to  $i$  by  $v$ ,  $\hat{d}_{vi}$ . Note

that even when  $v$  knows about its location, it should not disclose it in this phase.

- 3) When a neighbor  $i$  announces its measured distance to its neighbor  $j$ ,  $v$  collects  $\hat{d}_{ij}$ . (In other words,  $v$  collects neighbors' announcement on the measured distances to their neighbors.)
- 4) After collection of neighbors' announcements, node  $v$  compares the data collected. For each collected distance, if  $\hat{d}_{ij} = \hat{d}_{ji}$ , it is included in the filtering phase which is described in Section IV-B.

We note that it is possible that an attacker holds the announcements before it collects all the ranging information, and then calculates the relative locations of the honest nodes. Consequently, this attacker could fake a set of consistent range claims. To prevent such type of attack, we require each node announces one distance at a time in a round robin fashion within the neighborhood. This can be achieved by using pairwise ranging techniques [15].

### B. Filtering Phase

In this section, we propose a novel speculative procedure, which can effectively and efficiently filters out phantom nodes. The filtering procedure is described in Algorithm 1. Initially, the node  $v$  picks up two neighbors  $i$  and  $j$  randomly as pivots. (Note that node  $i$  and  $j$  could be phantom nodes themselves). Using the node  $v$  as the origin, the neighbors  $i$  and  $j$  and three distance information among  $v, i$  and  $j$ , the local coordinate system is constructed. In the node  $v$ 's coordinate system, we use a graph  $G(V, E)$  to construct a consistent subset. The set  $V$  is used to contain the node  $v$  and its neighbors, and the set  $E$  is used to keep the edges between two nodes when the distance information between them maintains consistency. Initially the graph  $G$  is empty. The update process of the graph  $G$  is as follows: The location of the neighbor  $k$  is determined on the local coordinate system  $L$  by trilateration [16] from three nodes  $v, i, j$  with measured distances  $\hat{d}_{kv}, \hat{d}_{ki}$  and  $\hat{d}_{kj}$ . After projecting all the neighbors on  $L$ , the distance between the projected neighbors is compared with the measured distance. For any two nodes  $i$  and  $j$  the distance  $\tilde{d}_{ij} = |p_i - p_j|$  is calculated from the projected location on  $L$ . If  $|\hat{d}_{ij} - \tilde{d}_{ij}| \geq \epsilon$ , the edge between  $i$  and  $j$  is not included in  $E$ . (the threshold value  $\epsilon$  depends on the noise in the ranging measurement. See Section V for more details.) The largest connected set  $V$  that contains node  $v$  is regarded as the largest consistent subset in the speculative plane  $L$ . This filtering procedure is done *iter* times (*iter* is a key parameter discussed later), and the cluster with the largest size is chosen as a final result.

### C. Identifying Consistent Subset

Algorithm 1 obtains a connected cluster in each iteration. In this section, we show that (i) the largest cluster must consist of only legitimate nodes and (ii) we can determine the case where a chosen pivot is, unfortunately, a phantom node.

**Theorem 1:** *When all the pivots chosen are honest nodes, the consistent cluster computed by the proposed solution in Algorithm 1 does not contain phantom nodes.*

### Algorithm 1 Speculative filtering

```

for  $i = 1$  to  $iter$  do
  each node  $v$  picks up two neighbors  $i$  and  $j$  randomly
  create local coordinate system  $L$  using  $v, i, j, \hat{d}_{vi}, \hat{d}_{vj}, \hat{d}_{ij}$ 
  initialize undirected graph  $G(V, E)$ 
  for each neighbor  $k \in Nbr(v)$  do
    calculate the location of  $k, p_k$ , on  $L$  by multilateration of
     $\hat{d}_{kv}, \hat{d}_{ki}$  and  $\hat{d}_{kj}$  from  $v, i, j$ 
  end for
  create node  $v$  in  $V$  with location  $p_v$ 
  for each neighbor  $k \in Nbr(v)$  do
    create node  $k$  in  $V$  with location  $p_k$ 
  end for
  for each pair of nodes  $i, j \in V$  and their ranging  $\hat{d}_{ij}$  do
     $\tilde{d}_{ij} = |p_i - p_j|$ 
    if  $|\hat{d}_{ij} - \tilde{d}_{ij}| < \epsilon$  then
      create edge  $e(i, j)$  in  $E$ 
    end if
  end for
  find the largest connected cluster  $C$  and save it
end for
Among all saved  $C$ , choose the one with the largest size

```

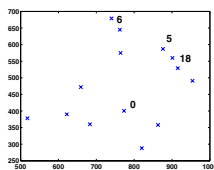


Fig. 2. Real plane, node 0,5,6,18 indicated

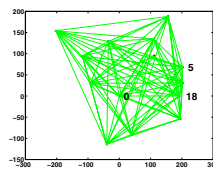


Fig. 3. Computed plane from pivot 0,5,18

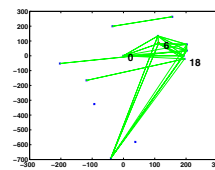


Fig. 4. Computed plane from pivot 0,6,18

**Theorem 2:** *If at least one of pivots is a phantom node, the size of largest cluster is smaller than the one when none of pivots is a phantom node.*

**Case Study:** As an example, Figures 2, 3 and 4 reflect the properties of **Theorem 1 and 2**. Figure 2 plots the real locations of the nodes, among which node 0 is a verifying node, node 6 is a phantom node, node 5 and 18 are not compromised, Figure 3 shows the cluster created when the pivot is not compromised (**Theorem 1**), Figure 4 is the cluster when the phantom pivot (node 6) is used, whose size is much smaller than the size of cluster shown in Figure 3 (**Theorem 2**).

### D. Localized Adversarial Effect of a Phantom Plane

In Section IV-C, due to the speculative nature of our approach, we assume that with a high probability, a large phantom plane can not be created consistently among non-collusive phantom nodes. In this section, we study the impact if this assumption does not hold, i.e., phantom nodes are able to launch collusion attack. As shown in the Figure 5, node  $v$  is located at the intersection of a phantom plane and a real plane. If the number of phantom nodes on the phantom plane is larger than the number of node on the real plane, node  $v$  will be deceived. However, we note that any honest node that is not located at the intersection, can not perceive the existence of the phantom plane  $P'$  from its own local view. It can only

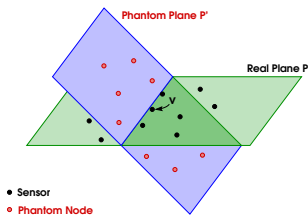


Fig. 5. Real plane vs. phantom plane

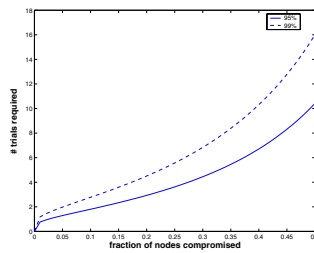


Fig. 6. The number of trials required to ensure the correct pivoting

perceive the large real plane  $P$ . This observation indicates that even if a large phantom plane is created through collusion attack, it can only compromise the views of a limited number of nodes located at the intersection line. Suppose that there are  $N$  honest nodes in the real plane  $P$  and none of three nodes are collinear. To compromise the views of all these nodes, more than  $\frac{N^2}{2}$  phantom nodes are needed, which is much larger than the number of honest nodes  $N$ . This gives us the insight why our scheme works well in the presence of a large number of phantom nodes.

#### E. Number of Trials Needed for a Successful Speculation

Our algorithm is speculative in nature. Obviously, it is unacceptable such speculation takes a large number of trials. From Algorithm 1, we know that  $iter$  controls the number of trials. In this section, we show the expected number of trials  $iter$  in Algorithm 1 is small even with a large percentage of phantom nodes.

To identify the largest consistent subset, our speculative algorithm cannot stop before the node  $v$  successfully selects two honest pivots. If  $q$  is the probability that a random neighbor is a phantom node, the probability that at least one of pivot is a phantom pivot is  $1 - (1 - q)^2$ . During  $iter$  trials, the probability that at least one of trial succeeds in selecting two honest pivots is:

$$P[X \geq 1] = 1 - (1 - (1 - q)^2)^{iter}$$

The number of trials required to ensure the successful filtering with probability 95% and 99% is shown in Figure 6. Even when 50% nodes are phantom nodes, the number of trials needed is only 16 to achieve 99% success ratio. We also note that the number of trials only affects the computation overhead. No extra communication is needed when the number of trials increases.

#### F. Cost Analysis

The cost for our protocol consists of the communication cost during exchange of distance information, and the computation cost in filtering phase. In our proposed solution, if a node has  $N - 1$  neighbors, it generates ranging information with each neighbor. Therefore,  $N - 1$  number of ranging information are generated per node to exchange. Each neighbor announces  $N - 1$  distance information in a message and collects  $(N - 1)^2$  messages from neighbors. The computation cost in filtering

phase is  $iter \times trilateration\ projection\ cost$  per node. For example, with the node density of 10, to achieve 99% success rate, we need 16 iterations with 10 trilateration per iteration, a computation that finishes within several milliseconds in Mica notes.

### V. EXPERIMENTAL RESULTS

In this section, we provide simulation results for our proposed scheme. We test under a wide range of node densities by generating 5-50 neighbors on the random locations within a certain node's range. The node and all of its neighbors participate in the phantom node detection in a decentralized manner. We describe simulation results with consideration of ranging measurement error. We also provide the simulation result when the sybil node is assumed.

#### A. Case Study

We first illustrate the speculative filter through a case study. In this experiment, we speculatively choose a pair of pivots in each trial (10 trials in total), and record the number of legitimate nodes and the number of phantom nodes identified in each trial as shown in Figure 9. In the trial 1, 3 and 8, the consistent subsets consist of phantom nodes, but it is not selected as a final result because the size is small compared to other trials. The trials 2, 4, 5, 6, 7, 9 and 10 are selected as the final largest consistent subset. In those trials, most of phantom nodes (20 phantom nodes are tried by two attackers) are filtered. A few phantom nodes is included in this example, but interestingly, these phantom nodes included are projected to the actual attacker's locations. Figure 7 shows the real locations of legitimate nodes (indicated by filled circles) and fake locations of phantom nodes (indicated by empty circles). The Figure 8 is the projected location from collected data. The legitimate nodes are projected to their real locations while most of phantom nodes are either filtered or projected to their actual attacker's locations.

#### B. Performance

The effectiveness of the proposed scheme is evaluated by the false positive rate, and by the false negative rate, which are calculated by:

$$\text{false positive rate (fp)} = \frac{\text{number of honest nodes failed verification}}{\text{total number of honest nodes}}$$

$$\text{false negative rate (fn)} = \frac{\text{number of phantom nodes authenticated}}{\text{total number of phantom nodes generated}}$$

The false negative rate includes the case when the phantom node is projected to the attacker's actual location.

#### C. Localization Error

The ranging measurement error is directly related to localization error and localization error affects the performance of phantom node detection. Figure 10 shows the performance according to various threshold values for the ranging measurement error 1%, 3%, 5%. The threshold value  $\epsilon$  is the maximum acceptable difference between the distance estimate in distance measurement phase and the distance between projected nodes

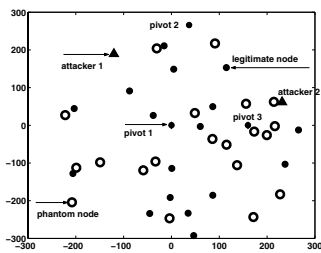


Fig. 7. Real locations of honest nodes and the locations attacker intended to fake

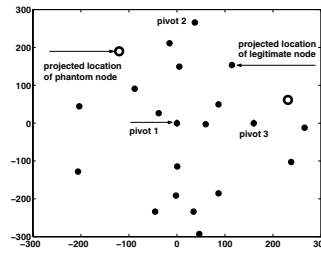


Fig. 8. Projected location of honest nodes and phantom nodes on virtual plane

on the virtual plane. For a given threshold value  $\epsilon$ , the false positive rate trades off with the false negative rate. As shown in the figure, as the ranging measurement error increases, the false negative and false positive rate increase. While the false negative rate decreases sharply in the early stage in the graph, the false positive rate increases gradually but does not have sharp increase in the later stage. Also, we note that the false negative rate varies in range of 0.05 – 0.25, but the phantom nodes not filtered are projected to the actual attacker positions. This indicates that our solution can not only detect the phantom node but also possibly identify the real locations of the attackers.

#### D. Number of Phantom Nodes

We provide simulation results when the number of phantom nodes the attacker can generate is large (more than the honest nodes). As shown in Figure 11, when the number of phantom nodes increases, the fraction of honest nodes excluded increases slightly. The interesting result is shown in the Figure 12: when the total number of phantom nodes increases, the percentage of these nodes that can avoid detection reduces. Most of phantom nodes are filtered even if the number of phantom nodes increases. This is mainly because the phantom plane created by phantom nodes can only deceive the nodes that are located at the intersect line of the phantom plane and real plane as shown in Figure 5. To deceive other honest nodes, the attackers need to create a different phantom plane that intersects with the real plan at the location of individual honest node. The localized view of individual nodes enables us to filter out phantom nodes, even when the number of phantom nodes is larger than the honest nodes.

## VI. CONCLUSION

Our secure localization system speculatively projects the neighboring nodes into a local map with the largest consistent subset of ranging claims. This approach authenticates the locations of honest nodes and detects the existence of the phantom nodes without relying on trusted agents. It is devised especially to be efficient when used in distributed way. Our localized construction results in quite strong results: even when the number of phantom nodes is greater than that of honest nodes, we could filter out most of the phantom nodes. In addition, our analysis and simulation indicate our scheme detect phantom nodes efficiently with small overhead.

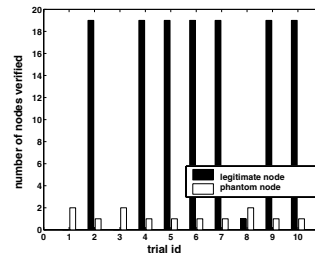


Fig. 9. Distribution of number of nodes verified (10 trials)

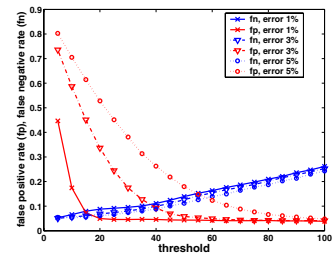


Fig. 10. Performance according to various ranging measurement error

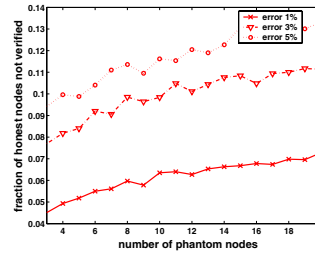


Fig. 11. False positive rate

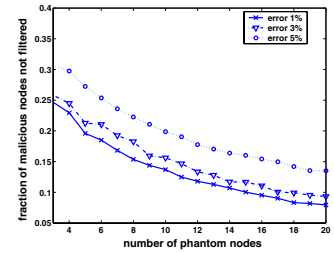


Fig. 12. False negative rate

## REFERENCES

- [1] R. Szewczyk, A. Mainwaring, J. Anderson, and D. Culler, "An analysis of a large scale habit monitoring application," in *SenSys'04*, 2004.
- [2] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, and et al., "An energy-efficient surveillance system using wireless sensor networks," in *MobiSys'04*, June 2004.
- [3] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Mobicom*, 2000.
- [4] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. G. ovindan, and S. Shenker, "GHT: A geographic hash table for data-centric storage in sensor networks," in *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002.
- [5] Z. Abrams, A. Goel, and S. Plotkin, "Set K-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks," in *IEEE IPSN*, 2004.
- [6] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-free localization schemes in large scale sensor networks," in *Mobicom*, 2003.
- [7] S. Capkun, M. Srivastava, and M. Cagalj, "Securing localization with hidden and mobile base stations," in *INFOCOM*, 2006.
- [8] L. Lazos and R. Poovendran, "SeRLoc: Secure range-independent localization for wireless sensor networks," in *ACM WiSe*, 2004.
- [9] L. Lazos, R. Poovendran, and S. Capkun, "ROPE: Robust position estimation in wireless sensor networks," in *IPSN*, 2005.
- [10] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *IPSN*, 2005.
- [11] D. Liu, P. Ning, and W. Du, "Attack-resistant location estimation in sensor networks," in *IPSN*, 2005.
- [12] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *ACM Wise*, 2003.
- [13] S. Capkun and J. P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *INFOCOM*, 2005.
- [14] A. Vora and M. Nesterenko, "Secure location verification using radio broadcast," in *International Conference on Principles of Distributed Systems*, 2004.
- [15] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Mobicom*, 2000.
- [16] A. Savvides, C. Han, and M. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Mobicom*, 2001.