Attack-Resistant Location Estimation in Sensor Networks (*Revised August 2005*)

Donggang Liu
The University of Texas at Arlington
and
Peng Ning
North Carolina State University
and
Wenliang Kevin Du
Syracuse University

Many sensor network applications require sensors' locations to function correctly. Despite the recent advances, location discovery for sensor networks in *hostile environments* has been mostly overlooked. Most of the existing localization protocols for sensor networks are vulnerable in hostile environments. The security of location discovery can certainly be enhanced by authentication. However, the possible node compromises and the fact that location determination uses certain physical features (e.g., received signal strength) of radio signals make authentication not as effective as in traditional security applications. This paper presents two methods to tolerate malicious attacks against beacon-based location discovery in sensor networks. The first method filters out malicious beacon signals on the basis of the "consistency" among multiple beacon signals, while the second method tolerates malicious beacon signals by adopting an iteratively refined voting scheme. Both methods can survive malicious attacks even if the attacks bypass authentication, provided that the benign beacon signals constitute the majority of the beacon signals. This paper also presents the implementation of these techniques on MICA2 motes running TinyOS, and the evaluation through both simulation and field experiments. The experimental results demonstrate that the proposed methods are promising for the current generation of sensor networks.

Categories and Subject Descriptors: C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms: Security, Design, Algorithms

Additional Key Words and Phrases: Sensor Networks, Security, Localization

This work is supported by the National Science Foundation (NSF) under grants CNS-0430223 and CNS-0430252.
A preliminary version of this paper appeared in the *Proceedings of The Fourth International Symposium on Information Processing in Sensor Networks (IPSN '05), pages 99 – 106, April 2005*.

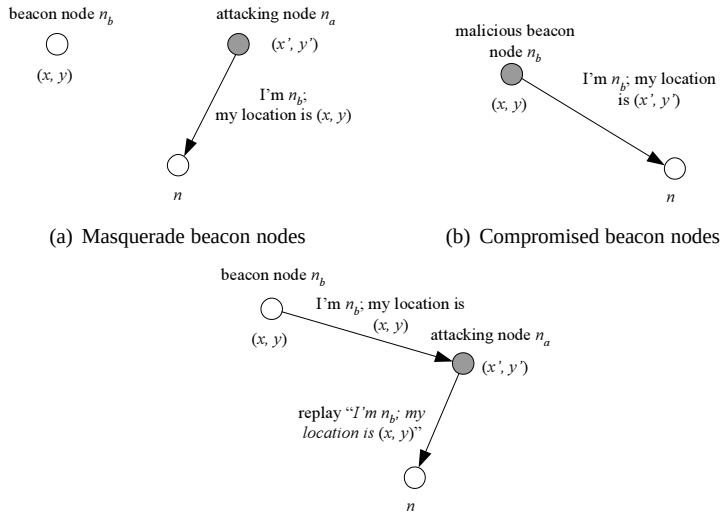
1. INTRODUCTION

Recent technological advances have made it possible to develop distributed sensor networks consisting of a large number of low-cost, low-power, and multi-functional sensor nodes that communicate in short distances through wireless links [Akyildiz et al. 2002]. Such sensor networks are ideal candidates for a wide range of applications such as health monitoring, data acquisition in hazardous environments, and military operations. The desirable features of distributed sensor networks have attracted many researchers to develop protocols and algorithms that can fulfill the requirements of these applications (e.g., [Perrig et al. 2001; Hill et al. 2000; Gay et al. 2003; Niculescu and Nath 2001; Intanagonwiwat et al. 2003; Newsome and Song 2003; Akyildiz et al. 2002]).

Sensors' locations play a critical role in many sensor network applications. Not only do applications such as environment monitoring and target tracking require sensors' location information to fulfill their tasks, but several fundamental techniques developed for wireless sensor networks also require sensor nodes' locations. For example, in geographical routing protocols (e.g., GPSR [Karp and Kung 2000] and GEAR [Yu et al. 2001]), sensor nodes make routing decisions at least partially based on their own and their neighbors' locations. As another example, in some data-centric storage applications such as GHT [Ratnasamy et al. 2002; Shenker et al. 2002], storage and retrieval of sensor data highly depend on sensors' locations. Indeed, many sensor network applications will not work without sensors' location information.

A number of location discovery protocols (e.g., [Savvides et al. 2001; Savvides et al. 2002; Niculescu and Nath 2003a; Nasipuri and Li 2002; Doherty et al. 2001; Bulusu et al. 2000; Niculescu and Nath 2003b; Nagpal et al. 2003; He et al. 2003]) have been proposed for wireless sensor networks in recent years. These protocols share a common feature: They all use some special nodes, called *beacon nodes*, which are assumed to know their own locations (e.g., through GPS receivers or manual configuration). These protocols work in two stages. In the first stage, non-beacon nodes receive radio signals called *beacon signals* from the beacon nodes. The packet carried by a beacon signal, which we call a *beacon packet*, usually includes the location of the beacon node. The non-beacon nodes then estimate certain measurements (e.g., distance between the beacon and the non-beacon nodes) based on features of the beacon signals (e.g., received signal strength indicator (RSSI), time difference of arrival (TDoA)). We refer to such a measurement and the location of the corresponding beacon node collectively as a *location reference*. In the second stage, a sensor node determines its own location when it has enough number of location references from different beacon nodes. A typical approach is to consider the location references as constraints that a sensor node's location must satisfy, and estimate it by finding a mathematical solution that satisfies these constraints with minimum estimation error. Existing approaches either employ *range-based* methods [Savvides et al. 2001; Savvides et al. 2002; Niculescu and Nath 2003a; Nasipuri and Li 2002; Doherty et al. 2001], which use the exact measurements obtained in stage one, or *range-free* ones [Bulusu et al. 2000; Niculescu and Nath 2003b; Nagpal et al. 2003; He et al. 2003; Lazos and Poovendran 2004], which only need the existences of beacon signals in stage one.

Despite the recent advances, location discovery for wireless sensor networks in *hostile environments*, where there may be malicious attacks, has been mostly overlooked. Many existing location discovery protocols become vulnerable in the presence of malicious attacks. As illustrated in Figure 1, an attacker may provide incorrect location reference by



(a) Masquerade beacon nodes (b) Compromised beacon nodes

(c) Replay attack

Fig. 1. Attacks against location discovery schemes

pretending to be valid beacon nodes (Figure 1(a)), compromising beacon nodes (Figure 1(b)), or replaying the beacon packets that he/she intercepted in different locations (Figure 1(c)). In either of the above cases, non-beacon nodes will determine their locations incorrectly. In either of these cases, non-beacon nodes will determine their locations incorrectly.

Without protection, an attacker may easily mislead the location estimation at sensor nodes and subvert the normal operation of sensor networks. The security of location discovery can certainly be enhanced by authentication. Specifically, each beacon packet should be authenticated with a cryptographic key only known to the sender and the intended receivers, and a non-beacon node accepts a beacon signal only when the beacon packet carried by the beacon signal can be authenticated. However, authentication does not guarantee the security of location discovery, either. An attacker may forge beacon packets with keys learned through compromised nodes, or replay beacon signals intercepted in different locations. Indeed, our experiment in Section 3 shows that an attacker can introduce substantial location estimation errors by forging or replaying beacon packets. Thus, it is highly desirable to have additional methods to protect location discovery in sensor networks.

Several techniques has been developed recently to deal with the security problems of location discovery in wireless sensor networks [Sastry et al. 2003; Lazos and Poovendran 2004; Ray et al. 2003; Li et al. 2005; S.Capkun and Hubaux 2005; Lazos et al. 2005]. The location verification technique proposed in [Sastry et al. 2003] can be used to verify the relative distance between a verifying node and a sensor node. However, it does not provide a solution to conduct secure location estimation at non-beacon nodes. A robust location detection is developed in [Ray et al. 2003] using the idea of majority voting. However, it cannot be directly applied in resource constrained sensor networks due to its high computation and storage overheads. Similar to our attack-resistant MMSE techniques, a robust

statistical method is independently discovered in [Li et al. 2005] to achieve robustness through Least Median of Squares.

SeRLoc [Lazos and Poovendran 2004] protects location discovery with the help of sectored antennae at beacon nodes. Similar to the voting-based scheme proposed in this paper, SeRLoc can tolerate malicious attacks by adopting the idea of majority voting. SPINE [S.Capkun and Hubaux 2005] is developed to protect location discovery by using verifiable multilateration. However, the distance bounding techniques required for verifiable multilateration may not be available in sensor networks due to the difficulties to (1) deal with the external attacks in Ultrasound-based distance bounding and (2) achieve nanosecond processing and time measurements in Radio-based distance bounding. ROPE [Lazos et al. 2005] is developed by integrating SerLoc and SPINE. However, it still requires nanosecond processing and time measurements that are not desirable for the current generation of sensor networks.

In this paper, we develop two types of attack-resistant location estimation techniques to tolerate the malicious attacks against range-based location discovery in wireless sensor networks. Our first technique, named *attack-resistant Minimum Mean Square Estimation*, is based on the observation that malicious location references introduced by attacks are intended to mislead a sensor node about its location, and thus are usually inconsistent with the benign ones. To exploit this observation, our method identifies malicious location references by examining the inconsistency among location references (indicated by the mean square error of estimation) and defeats malicious attacks by removing such malicious data. Three variants are developed to identify malicious location references: *the brute-force algorithm*, *the greedy algorithm* and *the enhanced greedy algorithm*. The brute-force algorithm tries every combination of location references to identify the largest set of consistent location references. It introduces high computation overhead at sensor nodes. The greedy algorithm is developed to reduce the computation overhead. It works in rounds and remove the most suspicious location reference in each round. The enhanced greedy algorithm is developed to improve the performance of the greedy algorithm by adopting a more efficient way to identify the most suspicious location reference.

Our second technique, a *voting-based location estimation* method, quantizes the deployment field into a grid of cells and has each location reference "vote" on the cells in which the node may reside. Moreover, we develop a method that allows iterative refinement of the "voting" results so that it can be executed in resource constrained sensor nodes.

We have implemented the proposed schemes on MICA2 motes [Crossbow Technology Inc.] running TinyOS [Hill et al. 2000], and evaluated the performance through simulation and field experiments. It shows that the proposed schemes can effectively remove the effect of malicious location references when the majority of location references are benign. In addition, the implementation and field experiment also indicates that the proposed schemes are promising for the current generation of sensor networks in terms of the storage overhead and computation overhead.

The rest of the paper is organized as follows. Section 2 discusses some assumptions and the threat model. Sections 3 and 4 present the attack-resistant MMSE location estimation and the voting-based location estimation technique respectively. Section 5 provides the security analysis for the proposed schemes. Sections 6 and 7 present the detailed evaluation through simulation and field experiments. Section 8 discusses related work. Section 9 concludes this paper and points out some future research directions.

2. ASSUMPTIONS AND THREAT MODEL

In this paper, we present two approaches to dealing with malicious attacks against location discovery in wireless sensor networks. The first approach is extended from the minimum mean square estimation (MMSE). It uses the mean square error as an indicator to identify and remove malicious location references. The second one adopts an iteratively refined voting scheme to tolerate malicious location references introduced by attackers.

Our techniques are purely based on a set of location references. The location references may come from beacon nodes that are either single hop or multiple hops away, or from those non-beacon nodes that already estimated their locations. We do not distinguish these location references, though the effect of "error propagation" may affect the performance of our techniques due to the estimation errors at non-beacon nodes. We consider such investigations as possible future work. Since our techniques only utilize the location references from beacon nodes, there is no extra communication overhead involved when compared to the previous localization schemes.

We assume all beacon nodes are uniquely identified. In other words, a non-beacon node can identify the original sender of each beacon packet based on the cryptographic key used to authenticate the packet. This can be easily achieved with a pairwise key establishment scheme [Eschenauer and Gligor 2002; Chan et al. 2003; Du et al. 2003] or a broadcast authentication scheme [Perrig et al. 2001].

We assume each non-beacon node uses at most one location reference derived from the beacon signals sent by each beacon node. As a result, even if a beacon node is compromised, the attacker that has access to the compromised key can only introduce at most one malicious location reference to a given non-beacon node by impersonating the compromised node.

For simplicity, we assume the distances measured from beacon signals (e.g., with RSSI or TDoA [Savvides et al. 2001]) are used for location estimation. (Our techniques can certainly be modified to accommodate other measurements such as angles.) For the sake of presentation, we denote a location reference obtained from a beacon signal as a triple $\langle x, y, \delta \rangle$, where (x, y) is the location of the beacon declared in the beacon packet, and δ is the distance measured from its beacon signal.

We assume an attacker may change any field in a location reference. In other words, it may declare a wrong location in its beacon packets, or carefully manipulate the beacon signals to affect the distance measurement by, for example, adjusting the signal strength when RSSI is used for distance measurement. We also assume multiple malicious beacon nodes may collude together to make the malicious location references appear to be "consistent". Our techniques can still defeat such colluding attacks as long as the majority of location references are benign.

3. ATTACK-RESISTANT MINIMUM MEAN SQUARE ESTIMATION

Intuitively, a location reference introduced by a malicious attack is aimed at misleading a sensor node about its location. Thus, it is usually "different" from benign location references. When there are redundant location references, there must be some "inconsistency" between the malicious location references and the benign ones. (An attacker may still have a location reference consistent with the benign ones after changing both the location and the distance values. However, such a location reference will not generate significantly negative impact on location determination.) To take advantage of this observation, we propose

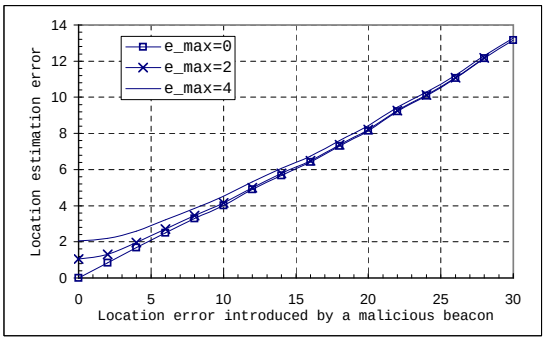


Fig. 2. Location estimation error. Unit of measurement for x and y axes: meter

to use the "inconsistency" among the location references to identify the malicious ones, and discard them before finally estimating the locations at sensor nodes.

In this paper, we assume a sensor node uses a MMSE-based method (e.g., [Savvides et al. 2001; Savvides et al. 2002; Niculescu and Nath 2003a; Nasipuri and Li 2002; Doherty et al. 2001; Niculescu and Nath 2003b]) to estimate its own location. Thus, most current range-based localization methods can be used with this technique. To harness this observation, we first estimate the sensor's location with the MMSE-based method and then assess if the estimated location could be derived from a set of consistent location references. If yes, we accept the estimation result; otherwise, we identify and remove the most "inconsistent" location reference, and repeat the above process. This process may continue until we find a set of consistent location references or it is not possible to find such a set.

3.1 Checking the Consistency of Location References

We use the mean square error ς^2 of the distance measurements based on the estimated location as an indicator of the degree of inconsistency, since all the MMSE-based methods estimate a sensor node's location by (approximately) minimizing this mean square error. Other indicators are possible but need further investigation.

DEFINITION 1. *Given a set of location references* $\mathcal{L} = \{\langle x_1, y_1, \delta_1\rangle, \langle x_2, y_2, \delta_2\rangle, ..., \langle x_m, y_m, \delta_m\rangle\}$ *and a location* $(\tilde{x}_0, \tilde{y}_0)$ *estimated based on* $\mathcal{L}$*, the* mean square error of this location estimation *is*

$$\varsigma^2 = \sum_{i=1}^{m} \frac{(\delta_i - \sqrt{(\tilde{x}_0 - x_i)^2 + (\tilde{y}_0 - y_i)^2})^2}{m}.$$

Intuitively, the more inconsistent a set of location references is, the greater the corresponding mean square error should be. To gain further understanding, we performed an experiment through simulation with the MMSE-based method in [Savvides et al. 2001]. We assume the distance measurement error is uniformly distributed between $-e_{max}$ and e_{max}. We used 9 honest beacon nodes and 1 malicious beacon node evenly deployed in a $30m \times 30m$ field. The node that estimates location is positioned at the center of the field. The malicious beacon node always declares a false location that is x meters away from its real location, where x is a parameter in our experiment.

Figures 2 and 3 show the location estimation error (i.e., the distance between a sensor's real location and the estimated location) and the mean square error ς^2 when x increases.

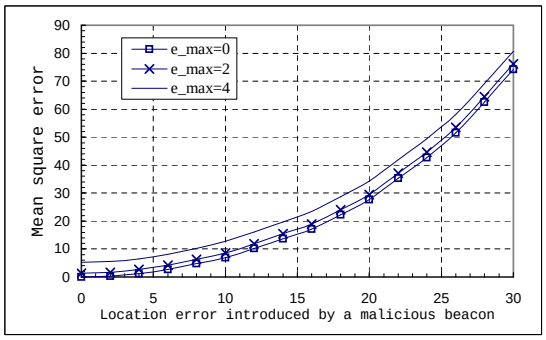


Fig. 3. Mean square error ς^2. Unit of measurement for x-axis: meter

As these figures show, if a malicious beacon node increases the location estimation error by introducing greater errors, it also increases the mean square error ς^2 at the same time. This further demonstrates that the mean square error ς^2 is potentially a good indicator of inconsistent location references.

In this paper, we choose a simple, threshold-based method to determine if a set of location references is consistent. Specifically, a set of location references $\mathcal{L} = \{\langle x_1, y_1, \delta_1\rangle, \langle x_2, y_2, \delta_2\rangle, ..., \langle x_m, y_m, \delta_m\rangle\}$ obtained at a sensor node is *τ-consistent w.r.t. a MMSE-based method* if the method gives an estimated location $(\tilde{x}_0, \tilde{y}_0)$ such that the mean square error of this location estimation

$$\varsigma^2 = \sum_{i=1}^{m} \frac{(\delta_i - \sqrt{(\tilde{x}_0 - x_i)^2 + (\tilde{y}_0 - y_i)^2})^2}{m} \leq \tau^2.$$

3.2 Determining Threshold τ

The determination of threshold τ depends on the measurement error model, which is assumed to be available for us to perform simulation off-line and determine an appropriate τ. The threshold is stored on each sensor node. Usually, the movement of sensor nodes (beacon or non-beacon nodes) does not have significant impact on this threshold, since the measurement error model will not change significantly in most cases. However, when the error model changes frequently and significantly, the performance of our techniques may be affected. In this paper, we assume the measurement error model will not change.

Note that the malicious beacon signals usually increase the variance of estimation. Thus, having a lower bound (e.g., Cramer-Rao bound) is not enough for us to filter malicious beacon signals. In fact, the upper bound or the distribution of the mean square error are more desirable. In this paper, we study the distribution of the mean square error ς^2 when there are no malicious attacks, and use this information to help determine the threshold τ.

Since there is no other error besides the distance measurement error, a benign location reference $\langle x, y, \delta\rangle$ obtained by a sensor node at (x_0, y_0) must satisfy:

$$|\delta - \sqrt{(x - x_0)^2 + (y - y_0)^2}| \leq \epsilon,$$

where ϵ is the maximum distance measurement error.

All the localization techniques are aimed at estimating a location as close to the sensor's real location as possible. Thus, we may assume the estimated location is very close to the real location when there are no attacks. Next, we derive the distribution of the mean

square error ς^2 using the real location as the estimated location, and compare it with the distribution obtained through simulation when there are location estimation errors.

The measurement error of a benign location reference $\langle x_i, y_i, \delta_i \rangle$ can be computed as $e_i = \delta_i - \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2}$, where (x_0, y_0) is the real location of the sensor node. Assuming the measurement errors introduced by different benign location references are independent, we can get the distribution of the mean square error through the following Lemma.

LEMMA 1. *Let $\{e_1, ..., e_m\}$ be a set of independent random variables, and μ_i, σ_i^2 be the mean and the variance of e_i^2, respectively. If the estimated location of a sensor node is its real location, the probability distribution of ς^2 is*

$$\lim_{m\to\infty} F[\varsigma^2 \le \varsigma_0^2] = \Phi(\frac{m\varsigma_0^2 - \mu'}{\sigma'}),$$

where $\mu' = \sum_{i=1}^{m} \mu_i$, $\sigma' = \sqrt{\sum_{i=0}^{m} \sigma_i^2}$, and $\Phi(x)$ is the probability of a standard normal random variable being less than x.

PROOF. Obviously, the mean square error can be computed by $\varsigma^2 = \sum_{i=1}^{m} \frac{e_i^2}{m}$. Thus, the cumulative distribution function can be calculated by

$$F(\varsigma^2 \le \varsigma_0^2) = F(\sum_{i=1}^{m} e_i^2 \le m\varsigma_0^2).$$

Since $\{e_1^2, e_2^2, \cdots, e_m^2\}$ are independent, according to the central limit theorem, we have

$$\lim_{m\to\infty} P(\frac{S_m - \mu'}{\sigma'} \le x) = \Phi(x),$$

where $S_m = \sum_{i=0}^{m}(e_i^2)$. Thus, we have

$$\begin{aligned} \lim_{m\to\infty} F(\varsigma^2 \le \varsigma_0^2) &= \lim_{m\to\infty} F(S_m \le m\varsigma_0^2) \\ &= \lim_{m\to\infty} P(\tfrac{S_m - \mu'}{\sigma'} \le \tfrac{m\varsigma_0^2 - \mu'}{\sigma'}) \\ &= \Phi(\tfrac{m\varsigma_0^2 - \mu'}{\sigma'}) \end{aligned}$$

□

Lemma 1 describes the probability distribution of ς^2 based on a sensor's real location. Though it is different from the probability distribution of ς^2 based on a sensor's estimated location, it can be used to approximate such distribution in most cases.

Let us further assume a simple model for measurement errors, where the measurement error is evenly distributed between $-\epsilon$ and ϵ. Then the mean and the variance for e_i are 0 and $\frac{\epsilon^2}{3}$, respectively, and the mean and the variance for any e_i^2 are $\frac{\epsilon^2}{3}$ and $\frac{4\epsilon^4}{45}$, respectively. Let $c = \frac{\varsigma_0}{\epsilon}$, we have

$$F(\varsigma^2 \le (c \times \epsilon)^2) = \Phi(\frac{\sqrt{5m}(3c^2 - 1)}{2}).$$

Figure 4 shows the probability distribution of ς^2 derived from Lemma 1 and the simulated results using sensors' estimated locations. We can see that when the number of location references m is large (e.g., $m = 9$) the theoretical result derived from Lemma 1 is very close to the simulation results. However, when m is small (e.g., $m = 4$), there

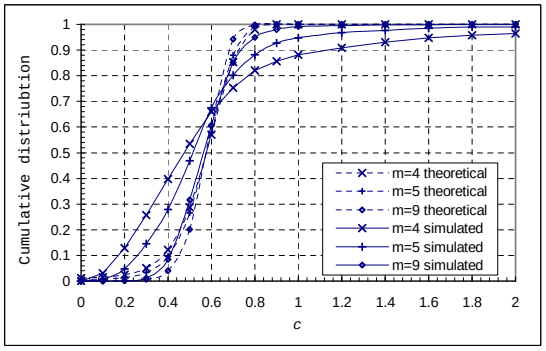


Fig. 4. Cumulative distribution function $F(\varsigma^2 \leq \varsigma_0^2)$. Let $c = \frac{\varsigma_0}{\epsilon}$.

are observable differences between the theoretical results and the simulation. The reasons are twofold. First, our theoretical analysis is based on the central limit theorem, which is only an approximation of the distribution when m is a large number. Second, we used the MMSE-based method proposed in [Savvides et al. 2001] in the simulation, which estimates a node's location by only *approximately* minimizing the mean square error. (Otherwise, the value of ς^2 for benign location references should never exceed ϵ^2.)

Figure 4 gives three hints about the choice of the threshold τ. First, when there are enough number of benign location references, a threshold less than the maximum measurement error is enough. For example, when $m = 9$, $\tau = 0.8\epsilon$ can guarantee the nine benign location references are considered consistent with high probability. Besides, a large threshold may lead to the failure to filter out malicious location references. Second, when m is small (e.g. 4), the cumulative probability becomes flatter and flatter when $c > 0.8$. This means that setting a large threshold τ for small m may not help much to guarantee the consistency test for benign location references; instead, it may give an attacker high chance to survive the detection. Third, the threshold cannot be too small; otherwise, a set of benign location references has high probability to be determined as a non-consistent reference set.

Based on the above observations, we propose to choose the value for τ with a hybrid method. Specifically, when the number of location references is large (e.g., more than 8), we determine the value of τ based on Lemma 1. Specifically, we choose a value of τ corresponding to a high cumulative probability (e.g., 0.9). When the number location references is small, we perform simulation to derive the actual distribution of the mean square error, and then determine the value of τ accordingly. Since there are only a small number of simulations to run, we believe this approach is practical.

3.3 Identifying the Largest Consistent Set

Since the MMSE-based methods can deal with measurement errors better if there are more benign location references, we should keep as many benign location references as possible when the malicious ones are removed. This implies we should get the largest set of consistent location references.

3.3.1 *Brute-force Algorithm.* Given a set $\mathcal{L}$ of n location references and a threshold τ, a simple approach to computing the largest set of τ-consistent location references is to

check all subsets of $\mathcal{L}$ with i location references about τ-consistency, where i starts from n and drops until a subset of $\mathcal{L}$ is found to be τ-consistent or it is not possible to find such a set. Thus, if the largest set of consistent location references consists of m elements, a sensor node has to use the MMSE method at least $1 + \binom{n}{m+1} + \cdots + \binom{n}{n}$ times to find out the right one. If $n = 10$ and $m = 5$, a node needs to perform the MMSE method for at least 387 times. It is certainly not desirable to do such expensive operations on resource constrained sensor nodes.

3.3.2 *Greedy Algorithm.* To reduce the computation on sensor nodes, we may use a greedy algorithm, which is simple but suboptimal. This greedy algorithm works in rounds. It starts with the set of all location references in the first round. In each round, it first verifies if the current set of location references is τ-consistent. If yes, the algorithm outputs the estimated location and stops. Optionally, it may also output the set of location references. Otherwise, it considers all subsets of location references with one fewer location reference, and chooses the subset with the least mean square error as the input to the next round. This algorithm continues until it finds a set of τ-consistent location references or when it is not possible to find such a set (i.e., there are only 3 remaining location references).

The greedy algorithm significantly reduces the computational overhead in sensor nodes. To continue the earlier example, a sensor node only needs to perform MMSE operations for about 50 times (instead of 387 times) using this algorithm. In general, a sensor node needs to use a MMSE-based method for at most $1+n+(n-1)+\cdots+4 = 1+\frac{(n-3)(n+4)}{2}$ times.

However, as we mentioned, the greedy algorithm cannot guarantee that it can always identify the largest consistent set. It is possible that benign location references are removed. In out earlier version of this paper [Liu et al. 2005a], we note that this generates a big impact on the accuracy of location estimation – especially when there are multiple malicious location references. To deal with this problem, we develop an enhanced greedy algorithm in the following. The new algorithm is based on an efficient approach to identifying the most suspicious location reference from a set of location references.

3.3.3 *Enhanced Greedy Algorithm.* In the previous discussion, we only consider the consistency of 3 or more location references. A further investigation also reveals that two benign location references are usually "consistent" with each other in the sense that there exists at least one location in the deployment field on which both location references agree. Hence, when the majority of location references are benign, we can usually find many location references that are consistent with a benign location reference. In addition, when a malicious location reference tries to create a larger location error, the number of location references that are consistent with the malicious one will decrease quickly.

According to the above discussion, for each location reference, we simply count the number of location references that are consistent with this location reference. We call this number the *degree of consistency* and use it to rank the suspiciousness of the location references received at a particular non-beacon node. The smaller the degree is, the more likely that the corresponding location reference is malicious.

The consistency between two location references can be verified as follows. For any location reference $\langle x, y, \delta \rangle$, the non-beacon node derives the area that it may reside based on this location reference. This area can be represented by a ring centered at (x, y), with the inner radius $\max\{\delta - \epsilon, 0\}$ and the outer radius $\delta + \epsilon$, where ϵ is the maximum distance

error. For the sake of presentation, we refer to such a ring a *candidate ring (centered) at location* (x, y). The non-beacon node then check whether the candidate rings of two location references overlap each other. If yes, they are consistent; otherwise, they are not consistent.

The algorithm to check whether the candidate rings of two location references $a = \langle x_a, y_a, \delta_a \rangle$ and $b = \langle x_b, y_b, \delta_b \rangle$ overlap can be done efficiently in the following way. Let d_{ab} denote the distance between (x_a, y_a) and (x_b, y_b). Let $rmax(x)$ and $rmin(x)$ denote the outer radius and the inner radius of the candidate ring of location reference x respectively. We can easily figure out that the candidate rings of location references a and b will not overlap when either of the following three conditions is true: (1) $d_{ab} > rmax(a) + rmax(b)$, (2) $d_{ab} + rmax(a) < rmin(b)$ and (3) $d_{ab} + rmax(b) < rmin(a)$.

Similar to the greedy algorithm, the enhanced algorithm to identify the largest consistent set starts with the set of all location references in the first round. In each round, it verifies whether the current set of location references is τ-consistent. If yes, the algorithm outputs the estimated location and stops. Optionally, it may also output the set of location references. If not, it removes the location reference corresponding to the smallest degree and use the remaining location references as the input to the next round. This algorithm continues until it finds a set of τ-consistent location references or when it is not possible to find such a set (i.e., there are only 3 remaining location references).

The enhanced algorithm not only improves the accuracy of location estimation in the presence of malicious attacks, but also reduces the computation overhead significantly since it can identify the most suspicious location reference efficiently and effectively. To continue the earlier example, a non-beacon node only needs to perform MMSE operations for 5 times. In general, a non-beacon node needs to use a MMSE-based method for at most $n - 3$ times.

4. VOTING-BASED LOCATION ESTIMATION

In this approach, we have each location reference "vote" on the locations at which the node of concern may reside. To facilitate the voting process, we quantize the target field into a grid of cells, and have each sensor node determine how likely it is in each cell based on each location reference. We then select the cell(s) with the highest vote and use the "center" of the cell(s) as the estimated location. To deal with the resource constraints on sensor nodes, we further develop an iterative refinement scheme to reduce the storage overhead, improve the accuracy of estimation, and make the voting scheme efficient on resource constrained sensor nodes.

4.1 The Basic Scheme

After collecting a set of location references, a sensor node should determine the target field. The node does so by first identifying the minimum rectangle that covers all the locations declared in the location references, and then extending this rectangle by R_b, where R_b is the maximum transmission range of a beacon signal. This extended rectangle forms the target field, which contains all possible locations for the sensor node. The sensor node then divides this rectangle into M small squares (cells) with the same side length L, as illustrated in Figure 5. (The node may further extend the target field to have square cells.) The node then keeps a voting state variable for each cell, initially set to 0.

At the beginning of this algorithm, the non-beacon node needs to identify the candidate ring of each location reference. For example, in Figure 5, the ring centered at point A is a

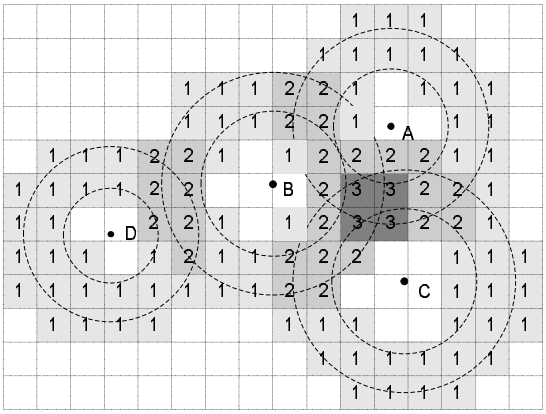


Fig. 5. The voting-based location estimation

candidate ring at A, which is derived from the location reference with the declared location at A.

For each location reference $\langle x, y, \delta \rangle$, the sensor node identifies the cells that overlap with the corresponding candidate ring, and increments the voting variables for these cells by 1. After the node processes all the location references, it chooses the cell(s) with the highest vote, and uses its (their) geometric centroid as the estimated location of the sensor node.

4.2 Overlap of Candidate Rings and Cells

A critical problem in the voting-based approach is to determine if a candidate ring overlaps with a cell. We discuss how to determine this efficiently below.

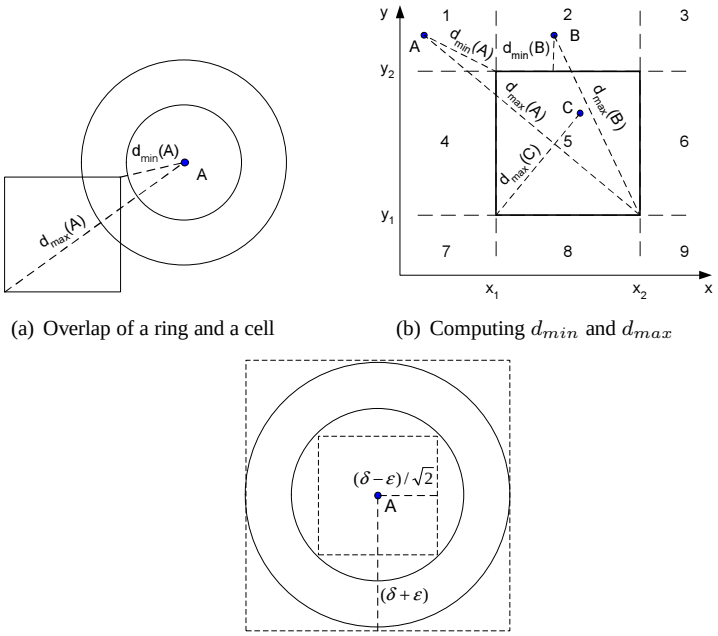
Suppose we need to check if the candidate ring at A overlaps with the cell shown in Figure 6(a). Let $d_{min}(A)$ and $d_{max}(A)$ denote the minimum and maximum distances from a point in the cell to point A, respectively. We can see that the candidate ring does not overlap with the cell only when $d_{min}(A) > r_o$ or $d_{max}(A) < r_i$, where $r_i = \max\{0, \delta - \epsilon\}$ and $r_o = \delta + \epsilon$ are the inner and the outer radius of the candidate ring, respectively.

To compute d_{min} and d_{max}, we divide the target field into 9 regions based on the cell, as shown in Figure 6(b). It is easy to see that given the center of any candidate ring, we can determine the region in which it falls with at most 6 comparisons between the coordinates of the center and those of the corners of the cell. When the center of a candidate ring is in region 1 (e.g., point A in Figure 6(b)), it can be shown that the closest point in the cell to A is the upper left corner, and the farthest point in the cell from A is the lower right corner. Thus, $d_{min}(A)$ and $d_{max}(A)$ can be calculated accordingly. These two distances can be computed similarly when the center of a candidate ring falls into regions 3, 7, and 9.

Consider point B in region 2. Assume the coordinate of point B is (x_B, y_B). We can see that $d_{min}(B) = y_B - y_2$. Computing $d_{max}(B)$ is a little more complex. We first need to check if $x_B - x_1 > x_2 - x_B$. If yes, the farthest point in the cell from B must be the lower left corner of the cell. Otherwise, the farthest point in the cell from B should be the lower right corner of the cell. Thus, we have

$$d_{max}(B) = \sqrt{(\max\{x_B - x_1, x_2 - x_B\})^2 + (y_B - y_1)^2}.$$

These two distances can be computed similarly when the center of a candidate ring falls



(a) Overlap of a ring and a cell

(b) Computing d_{min} and d_{max}

(c) Limiting the examinations of cells

Fig. 6. Determine whether a ring overlaps with a cell

into regions 4, 6, and 8.

Consider a point C in region 5. Obviously, $d_{min}(C) = 0$ since point C itself is in the cell. Assume the coordinate of point C is (x_c, y_c). The farthest point in the cell from C must be one of its corners. Similarly to the above case for point B, we may check which point is farther away from C by checking $x_c - x_1 > x_2 - x_c$ and $y_c - y_1 > y_2 - y_c$. As a result, we get

$$d_{max}(C) = \sqrt{(\max\{x_c - x_1, x_2 - x_c\})^2 + (\max\{t_c - y_1, y_2 - y_c\})^2}.$$

Based on the above discussion, we can determine if a cell and a candidate ring overlap with at most 10 comparisons and a few arithmetic operations. To prove the correctness of the above approach only involves elementary geometry, and thus is omitted.

For a given candidate ring, a sensor node does not have to check all the cells for which it maintains voting states. As shown in Figure 6(c), with simple computation, the node can get the outer bounding box centered at A with side length $2(\delta + \epsilon)$. The node only needs to consider the cells that intersect with or fall inside this box. Moreover, the node can get the inside bounding box with simple computation, which is centered at A with side length $\sqrt{2}(\delta - \epsilon)$, and all the cells that fall into this box need not be checked.

4.3 Iterative Refinement

The number of cells M (or equivalently, the quantization step L) is a critical parameter for the voting-based algorithm. It has several implications to the performance of our approach. First, the larger M is, the more state variables a sensor node has to keep, and thus the more

storage is required. Second, the value of M (or L) determines the precision of location estimation. The larger M is, the smaller each cell will be. As a result, a sensor node can determine its location more precisely based on the overlap of the cells and the candidate rings.

However, due to the resource constraints on sensor nodes, the granularity of the partition is usually limited by the memory available for the voting state variables on the nodes. This puts a hard limit on the accuracy of location estimation. To address this problem, we propose an *iterative refinement* of the above basic algorithm to achieve fine accuracy with reduced storage overhead.

In this version, the number of cells M is chosen according to the memory constraint in a sensor node. After the first round of the algorithm, the node may find one or more cells having the largest vote. To improve the accuracy of location estimation, the sensor node then identifies the smallest rectangle that contains all the cells having the largest vote, and performs the voting process again. For example, in Figure 5, the same algorithm will be performed in a rectangle which exactly includes the 4 cells having 3 votes. Note that in a later iteration of the basic voting-based algorithm, a location reference does not have to be used if it does not contribute to any of the cells with the highest vote in the current iteration.

Due to a smaller rectangle to quantize in a later iteration, the size of cells can be reduced, resulting in a higher precision. Moreover, a malicious location reference will most likely be discarded, since its candidate ring usually does not overlap with those derived from benign location references. For example, in Figure 5, the candidate ring centered at point D will not be used in the second iteration.

The iterative refinement process should terminate when a desired precision is reached or the estimation cannot be refined. The former condition can be tested by checking if the side length L of each cell is less than a predefined threshold S, while the latter condition can be determined by checking whether L remains the same in two consecutive iterations. The algorithm then stops and outputs the estimated location obtained in the last iteration. It is easy to see that the algorithm will fall into either of these two cases, and thus will alway terminate. In practice, we may set the desired precision to 0 in order to get the best precision.

5. SECURITY ANALYSIS

Both proposed techniques can usually remove the effect of the malicious location references from the final location estimation when there are more benign location references than the malicious ones. Theorem 1 shows that when the majority of location references are benign, the location estimation error of the attack-resistant MMSE is bounded if we can successfully identify the largest consistent set. Hence, to defeat the attack-resistant MMSE approach, the attacker has to distribute to a victim node more malicious location references than the benign ones, and control the declared locations and the physical features (e.g., signal strength) of beacon signals so that the malicious location references are considered consistent.

LEMMA 2. *Assume there are m benign location references and n malicious location references in a τ-consistent set. The location estimation error from this set of location references using MMSE is no more than $2R + \sqrt{\frac{m+n}{m}}\tau$, where R is the radio communication range of a sensor node.*

PROOF. Let $O = (x_0, y_0)$ denote the real location of the non-beacon node and $O' = (x'_0, y'_0)$ denote the estimated location of the non-beacon node based on all location references (including the malicious ones). Let $|AB|$ denote the distance between A and B. Thus, the location estimation error can be represented by $|OO'|$. Let $\{L_1, \cdots, L_m\}$ denote the set of benign location references and $\{L_{m+1}, \cdots, L_{m+n}\}$ denote the set of malicious location references.

Consider a particular benign location references $L_i = \langle x_i, y_i, \delta_i \rangle$. Since the communication range of sensor nodes is R, we have $|OL_i| \leq R$. In addition, $e_i = \delta_i - |O'L_i|$ and $\delta_i \leq R$. Thus, we have

$$|OO'| \leq |OL_i| + |L_iO'| \leq R + \delta_i - e_i \leq 2R - e_i.$$

There are two different cases: $e_i \geq 0$ or $e_i < 0$. When $e_i \geq 0$, we have $|OO'| \leq 2R$. When $e_i < 0$, we have $|OO'| - 2R \leq -e_i$. Assume $|OO'| \geq 2R$, we have $e_i^2 \geq (|OO'| - 2R)^2$. Since $\{L_1, \cdots, L_{m+n}\}$ is τ-consistent, we have $\sum_{i=1}^{m+n} e_i^2 \leq (m+n)\tau^2$. Therefore,

$$m(|OO'| - 2R)^2 \leq \sum_{i=1}^{m} e_i^2 \leq \sum_{i=1}^{m+n} e_i^2 \leq (m+n)\tau^2.$$

Hence, we have $(|OO'| - 2R)^2 \leq \frac{(m+n)\tau^2}{m}$. It implies

$$|OO'| \leq 2R + \sqrt{\frac{m+n}{m}}\tau.$$

According to the above analysis, we can conclude that the statement in Lemma 2 is true. □

THEOREM 1. *Assume a non-beacon node receives m benign location references and n malicious loccation references, where $m > n$. The location estimation error at this non-beacon node using the attack-resistant MMSE scheme with the brute-force algorithm is no more than $2R + \sqrt{\frac{m}{m-n}}\tau$ if the threshold τ is set greater than the maximum distance error ϵ, where R is the radio communication range of a sensor node.*

PROOF. It is easy to know that the set of m benign location references is always τ-consistent if $\tau \geq \epsilon$. Thus, there are at least m location references in the largest consistent set. Assume there are k location references in the largest consistent set, where $k \geq m$. According to Lemma 2, we have

$$|OO'| \leq 2R + \sqrt{\frac{k}{k-n}}\tau \leq 2R + \sqrt{\frac{m}{m-n}}\tau.$$

□

Similarly, theorem 2 shows that when the majority of location references are benign, the location estimation error of the voting-based scheme is bounded. Hence, to defeat the voting-based approach, the attacker needs similar efforts so that the cell containing the attacker's choice gets more votes than those containing the sensor's real location.

THEOREM 2. *When the majority of location references at a non-beacon node are benign, the location estimation error at this non-beacon node using the voting-based scheme is no more than $2R + \sqrt{2}L$, where L is the side length of the cell.*

PROOF. Assume the real location of the sensor node is $O = (x_0, y_0)$ and the estimated location of the sensor node using the voting-based scheme is $O' = (x'_0, y'_0)$.

Since the candidate ring of a benign location reference always covers the real location O of the sensor node, the number of votes in the cell that contains O is at least m. Thus, the number of votes in the cell that contains O' is at least m. Since the number of votes coming from the malicious location references is at most n, we know that there is at least one benign location reference whose candidate ring covers the cell that contains O'. Assume one of such benign location referecnes is $L_i = \langle x_i, y_i, \delta_i \rangle$, we have

$$|L_i O'| \leq R + \sqrt{2}L,$$

where L is the side length of a cell. Therefore, we have

$$|OO'| \leq |OL_i| + |L_i O'| \leq 2R + \sqrt{2}L.$$

□

An attacker has two ways to satisfy the above conditions (in order to defeat our techniques). First, the attacker may compromise beacon nodes and then generate malicious beacon signals. Since all beacon packets are authenticated, and a sensor node uses at most one location reference derived from the beacon signals sent by each beacon node, the attacker needs to compromise more beacon nodes than the benign beacon nodes from which a target sensor node may receive beacon signals, besides carefully crafting the forged beacon signals.

Second, the attacker may launch wormhole attacks [Hu et al. 2003] (or replay attacks) to tunnel benign beacon signals from one area to another. In this case, the attacker does not have to compromise any beacon node, though he/she has to coordinate the wormhole attacks. This paper does not provide techniques to address wormhole attacks. However, our methods can still tolerate wormhole attacks to a certain degree as long as the number of malicious location references at a sensor node is less than the number of benign location references. On the other hand, we may also use some of existing wormhole detection methods (e.g., packet leashes [Hu et al. 2003], directional antennae [Hu and Evans 2003b]) to make it more difficult for an attacker to introduce many malicious location references to a sensor node by launching wormhole attacks.

Our techniques certainly have a limit. In an extreme case, if all the beacon nodes are compromised, our techniques will fail. However, the proposed techniques offer a graceful performance degradation as more malicious location references are introduced. In contrast, an attacker may introduce arbitrary location error with a single malicious location reference in the previous schemes. To further improve the security of location discovery, other complementary mechanisms (e.g., detection of malicious beacon nodes) should be used.

6. SIMULATION EVALUATION

This section presents the simulation results for both proposed schemes. The evaluation focuses on the performance under different configurations and the improvement on the accuracy of location estimation in hostile environments.

Three attack scenarios are considered. The first scenario considers a single malicious location reference that declares a wrong location e meters away from the beacon node's

real location. (An attacker may also modify the distance component δ in a location reference, which will generate a similar impact.) In the second scenario, there are multiple non-colluding malicious location references, and each of them independently declares a wrong location that is e meters away from the beacon node's real location. In the third scenario, multiple colluding malicious location references are considered. In this case, the malicious location references declare false locations by coordinating with each other to create a virtual location e meters away from the sensor's real location. Thus, the malicious location references may appear to be consistent to a victim node.

In all simulations, a set of benign beacon nodes and a few malicious beacon nodes are evenly deployed in a $30m \times 30m$ target field. The non-beacon sensor node is located at the center of this target field. We assume the maximum transmission range of beacon signals is $R_b = 22m$, so that the non-beacon node can receive the beacon signal from every beacon node located in the target field. We assume the entire deployment field is much larger than this target field so that an attacker can create a very large location estimation error inside the deployment field. Each malicious beacon node declares a false location according to the three attack scenarios discussed above. We assume a simple distance measurement error model. That is, the distance measurement error is uniformly distributed between $-\epsilon$ and ϵ, where the maximum distance measurement error ϵ is set to $\epsilon = 4$m.

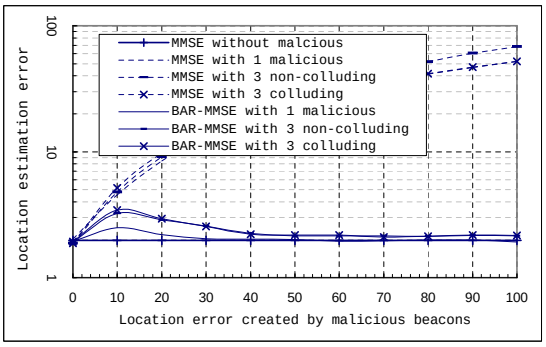
6.1 Evaluation of Attack-Resistant MMSE

In the simulation, we use the MMSE-based method proposed in [Savvides et al. 2001], which we call the *basic MMSE method*, to perform the basic location estimation. Our attack-resistant MMSE method is then implemented on the basis of this method, as discussed in Section 3. We set $\tau = 0.8\epsilon$ according to Figure 4, which guarantees 9 benign location references are considered consistent with probability of 0.999.

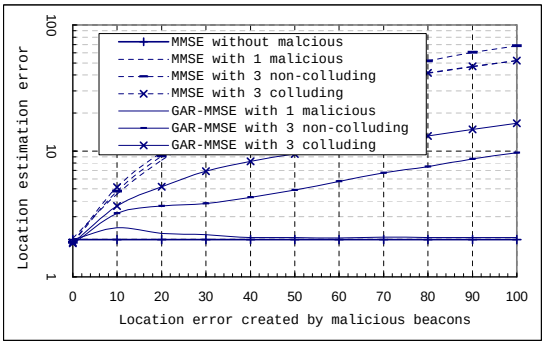
Figure 7(a) shows the performance of the attack-resistant MMSE method when the brute-force algorithm is used to identify the largest consistent set. We can see that our technique can significantly reduce the location estimation error when there are malicious location references. The figure also indicates that when the malicious location references create large location errors, they are always removed from the final location estimation at a non-beacon node.

Figure 7(b) shows the performance of the attack-resistant MMSE method when the greedy algorithm is used to identify the largest consistent set. From the figure, we can see that the technique can significantly reduce the location estimation error when there is only one malicious location reference. However, the performance degrades quickly when there are multiple malicious location references. This is because multiple malicious location references, especially when they collude together, make the filtering of malicious location references more difficult. It is very possible that benign location references being identified as malicious and being removed. Hence, we know that the greedy algorithm cannot effectively identify the largest consistent set when there are more than one malicious location references.

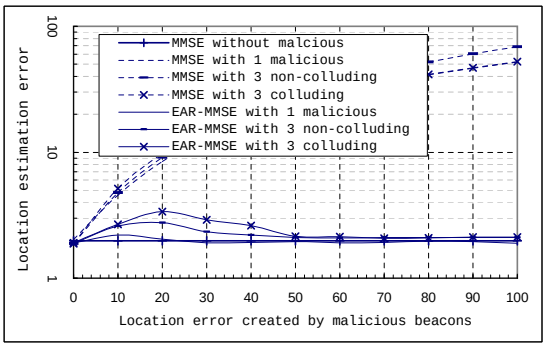
Figure 7(c) shows the performance of the attack resistant MMSE method when the enhanced greedy algorithm is used to identify the largest consistent set. We can see that that the attack-resistant MMSE with the enhanced greedy algorithm reduces the location estimation error significantly compared with the one with greedy algorithm. It indicates that the enhanced greedy algorithm can identify the largest consistent set more effectively than the greedy algorithm. Hence, in the following evaluation, we always assume that the



(a) Performance of attack-resistant MMSE using the brute-force algorithm.



(b) Performance of attack-resistant MMSE using the greedy algorithm.



(c) Performance of attack-resistant MMSE using the enhanced greedy algorithm

Fig. 7. $\tau = 0.8\epsilon$. Unit of measurement for x and y axes: meter

enhanced greedy algorithm is used in the attack-resistant MMSE method.

From 7(c), we note that the malicious location references can generate big impact when the location errors created by them are around 20m. Therefore, we assume that malicious location references introduce 20m location errors and evaluate the effect of threshold τ on the performance of the attack-resistant MMSE method. Figure 8 indicates that the

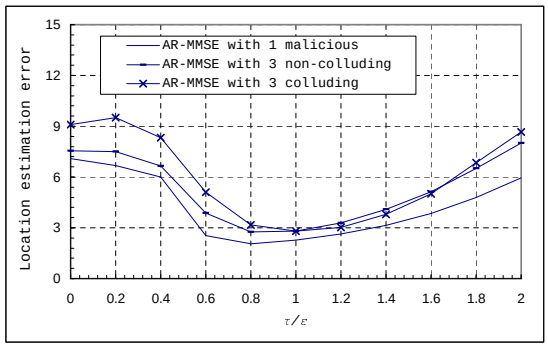


Fig. 8. Performance of Attack-resistant MMSE under different threshold τ. Assume each malicious location reference introduces 20m location error.

threshold τ cannot be set too large or too small since it will either fail to remove malicious location references or remove a large number of benign location references. This result is consistent with our analysis in Section 3.2.

6.2 Evaluation of Voting-Based Scheme

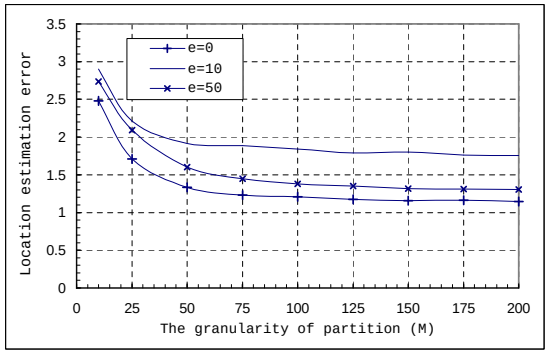


Fig. 9. Performance for different M (e: error introduced by a malicious location reference)

We first study the impact of parameter M on the voting-based method. Figure 9 shows the performance of the voting-based scheme with different values of M when there is only one malicious location reference. We can see that the location estimation error initially decreases when M increases, but then does not decrease much when M is greater than 100. Moreover, the parameter M also has implications in computational cost. Since the voting-based method is finally reduced to checking whether a candidate ring derived from a location reference overlaps with the cells in the grid, we use the number of cells being examined as an indicator of the computational cost. Figure 10 shows the computational costs of the voting-based method for different values of M when there is one malicious location reference. As this figure shows, the computational cost increases almost linearly with the value of M. When there are no or more malicious location references, the computational cost will increase similarly as M increases. Based on these results, we set $M = 100$, which

implies 100 Bytes memory for the voting variables, in the later simulations to trade-off the accuracy with the storage and computation overhead.

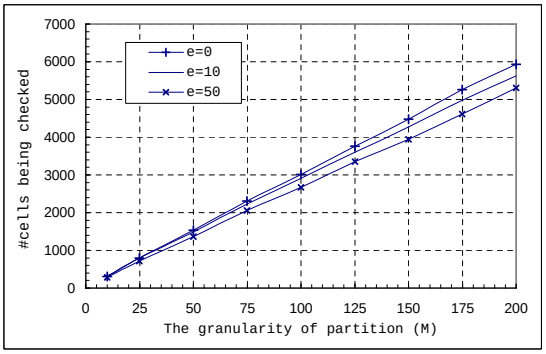


Fig. 10. Computational cost for different M (e: error introduced by a malicious location reference)

Next, we study the performance of the voting-based scheme under malicious attacks. In the simulation, we also set $S = 0$ to get the minimum location estimation error achievable by this method. Figure 11 compares the accuracy of the basic MMSE method and our voting-based scheme under different types of attacks. We can clearly see that the accuracy of location estimation is improved significantly in our scheme. In addition, unlike the attack-resistant MMSE scheme, the voting-based scheme can tolerate multiple (colluding or non-colluding) malicious location references more effectively.

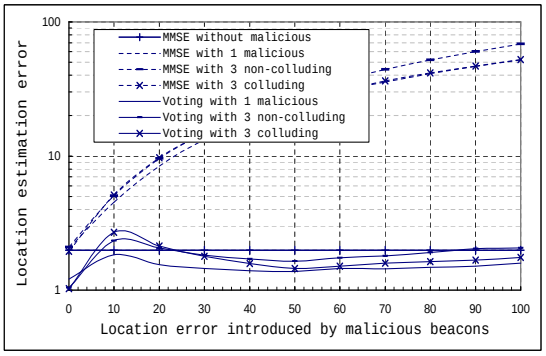


Fig. 11. Performance of the voting-based scheme ($M = 100, S = 0$). Unit of measurement for x and y axes: meter

Note that the curves for the voting-based scheme in Figure 11 have a bump when the location error introduced by malicious location references is around 10m. This is because the malicious location references are not significantly different from the benign location references around this point, and our scheme cannot completely shield the effect of malicious location references. Nevertheless, the attacker will not be able to introduce large location estimation errors by simply creating large location errors. As a result, the location estimation errors are always bounded even if there are malicious attacks. In addition, we

also note that the performance of voting-based scheme under attacks is usually better than the performance of MMSE scheme without attacks. This is because we used the MMSE-based method in [Savvides et al. 2001] in the simulation, which estimates a node's location by only *approximately* minimizing the mean square error.

Now let us compare the attack-resistant MMSE and the voting-based methods. Based on the earlier results, we choose threshold $\tau = 0.8\epsilon$ for the attack-resistant MMSE, and set $M = 100$ and $S = 0$ for the voting-based scheme. Figure 12 shows that the voting-based scheme performs slightly better than the attack-resistant MMSE scheme in terms of the location estimation accuracy when there are malicious location references.

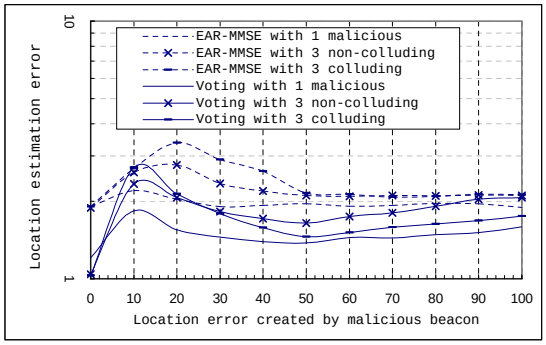


Fig. 12. Comparison between the attack-resistant MMSE and the voting-based scheme. Unit of measurement: meter

7. IMPLEMENTATION AND FIELD EXPERIMENTS

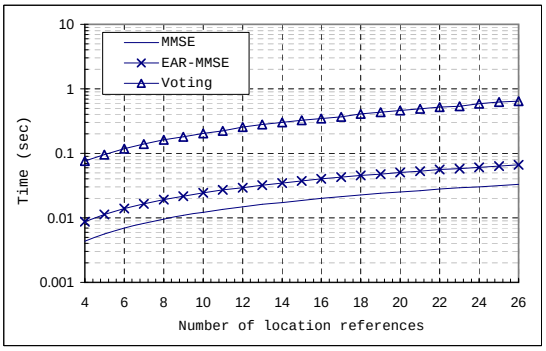
We have implemented both schemes on TinyOS [Hill et al. 2000], an operating system for networked sensors. These implementations are targeted at MICA2 motes [Crossbow Technology Inc.] running TinyOS. The attack-resistant MMSE is implemented based on the basic MMSE method proposed in [Savvides et al. 2001]. However, our implementation of the basic MMSE method is simplified by only using the location coordinates (without the ultrasound propagation speed, which is not necessary in our study).

Scheme	ROM (bytes)	RAM (bytes)
MMSE	2034	286
EAR-MMSE	3738	434
Voting-Based	4488	174

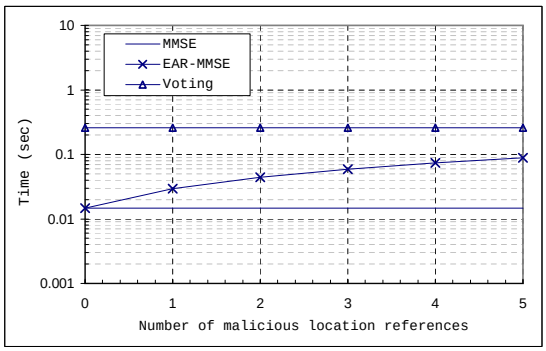
Table I. Code size (assume 12 location references; $M = 100$)

Table I gives the code size (ROM and RAM) for these implementations on MICA2 platform. Table I is obtained by assuming at most 12 location references. More location references will increase the RAM size of the program, but the increased RAM is only required to save the additional location references.

Figure 13 shows the average execution time of the basic MMSE, the attack-resistant MMSE, and the voting-based schemes on real MICA2 motes. These data are collected



(a) A single malicious location reference



(b) A total of 12 location references

Fig. 13. Average execution time on MICA2 motes ($\epsilon = 4$m, $\tau = 0.8\epsilon$, $M = 100$ and $S = 0$)

by counting the numbers of CPU clock cycles spent on location estimation. The location references used in the experiment are generated from the simulation in Section 6. We can see that the basic MMSE method has the least execution time. The attack-resistant MMSE scheme has less computational cost than the voting-based scheme. The number of malicious location references does not affect the computational overheads of the basic MMSE method and the voting-based method but does affect the computational overhead of the attack-resistant MMSE method. From Table I and Figure 13, we conclude that our proposed techniques are practical for the current generation of sensor networks in terms of the storage and the computation overheads, especially when the locations of sensor nodes do not change frequently.

To further study the feasibility of our techniques, we performed an outdoor field experiment. In this experiment, eight MICA2 motes were deployed in a 10×10 target field, where each unit of distance is 4 feet, as shown in Figure 14. The sensor node with ID 0 is configured as a non-beacon node, which is located at the center of the field. All the other sensor nodes are configured as beacon nodes.

We considered three attack scenarios in this experiment. In the first scenario, beacon node 1 is configured as a malicious beacon node that always declares a location e feet away from its real location in the direction away from the non-beacon node. In the second

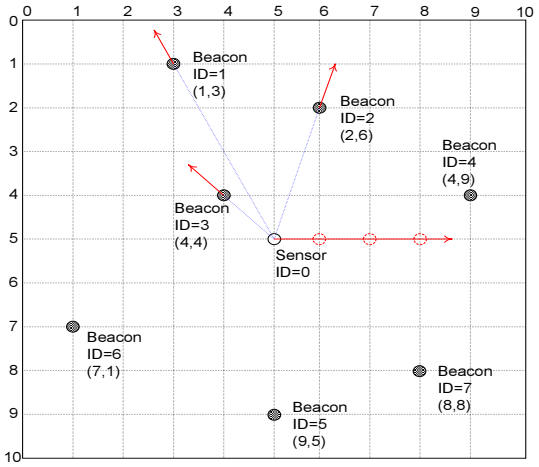


Fig. 14. Target area of field experiment.

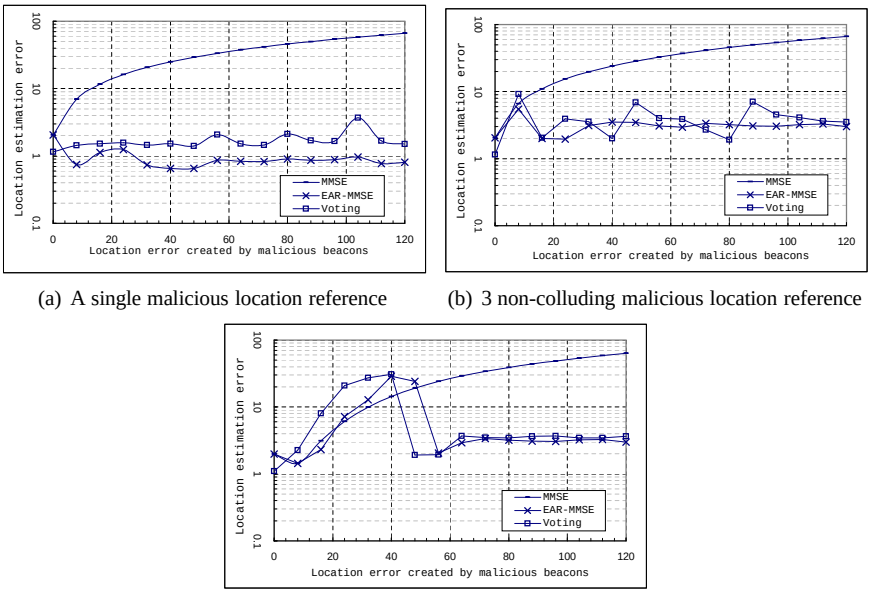
scenario, beacon nodes 1, 2 and 3 are configured as malicious beacon nodes. Each of these three nodes declares a location e feet away from its real location in the directions away from the non-beacon node. In the third scenario, three malicious beacon nodes 1, 2, and 3 work together to create a virtual location. Each of these three nodes declares a false location by increasing its horizontal coordinate by e feet. This actually creates a virtual location in the horizontal axis e feet away from the non-beacon node's real location. This is illustrated in Figure 14 by the horizontal arrow starting from the non-beacon node.

To measure the distance (δ) between sensor nodes, we use a simple RSSI based technique. Note that the Active Message protocol in TinyOS provides a reading in the *strength* field for the MICA2 platform. This value is returned in every received packet, and can be used to compute the signal strength. Thus, we performed an experiment before the actual field experiment to estimate the relationship between the values of this field and the distance between two nodes. For each given distance, we computed the average of this values on 20 observations. We then built a table that contains distances and the corresponding average readings. During the field experiments, when a sensor node receives 20 packets from a beacon node, it computes the average of the *strength* values, and estimates the distance with interpolation according to this table. For example, if the average reading v falls in between two adjacent points (v_i, d_i) and (v_{i+1}, d_{i+1}) in the table, the sensor computes the distance

$$d = d_i + \frac{(v - v_i) \times (d_{i+1} - d_i)}{v_{i+1} - v_i}.$$

We set ϵ to 4 feet, which is the maximum distance measurement error observed in the experiment.

Figure 15 shows the performance of the proposed methods and the basic MMSE method in the field experiment. For the first two attack scenarios, we can see that the proposed methods can tolerate malicious location references quite effectively. The performance in the third scenario is worse than the first two cases. The reason is that the non-beacon nodes has only 4 benign location references, but 3 colluding location references. However, we still see that the location estimation error drops when the location errors introduced by



(a) A single malicious location reference

(b) 3 non-colluding malicious location reference

(c) 3 colluding malicious location reference

Fig. 15. Results of the field experiment. Assume $M = 100$ and $S = 0$ for voting-based scheme; $\tau = 0.8\epsilon = 3.2$ feet for attack-resistant MMSE. Unit of measurement for x and y axes: feet

the malicious attacks are above certain thresholds. Overall, the location estimation errors caused by malicious attacks are bounded when the proposed techniques are used, while the errors can be arbitrarily large when the basic MMSE method is used.

The field experiment further shows that our methods are efficient and effective in tolerating malicious attacks. It also indicates that our methods are promising for the current generation of sensor networks.

8. RELATED WORK

Many range-based localization schemes have been proposed for sensor networks [Savvides et al. 2001; Savvides et al. 2002; Niculescu and Nath 2003a; Nasipuri and Li 2002; Doherty et al. 2001]. Savvides et al. developed AHLoS protocol based on Time Difference of Arrive [Savvides et al. 2001], which was extended in [Savvides et al. 2002]. Doherty et al. presented a localization scheme based on connectivity constraints and relative signal angles between neighbors [Doherty et al. 2001]. Angle of Arrival is used to develop localization scheme in [Niculescu and Nath 2003a] and [Nasipuri and Li 2002]. Range-free schemes are proposed to provide localization services for the applications with less precision requirements [Bulusu et al. 2000; Niculescu and Nath 2003b; Nagpal et al. 2003; He et al. 2003]. Bulusu, Heidemann and Estrin proposed to estimate a sensor's location as the centroid of all locations in the received beacon signals [Bulusu et al. 2000]. Niculescu and Nath proposed to use the minimum hop count and the average hop size to estimate the distance between nodes and then determine sensor nodes' locations accordingly [Niculescu

and Nath 2003b]. None of these schemes will work properly when there are malicious attacks.

The location verification technique proposed in [Sastry et al. 2003] can verify the relative distance between a verifying node and a sensor node. It does not provide a solution to conduct secure location estimation at non-beacon nodes. In this paper, we provide efficient ways to estimate locations of sensor nodes securely. The location verification technique is complementary to our techniques since it can be used to enhance the security of distance measurement between two nodes.

A robust location detection is developed in [Ray et al. 2003]. However, it cannot be directly applied in sensor networks due to its high computation and storage overheads. A voting-based Cooperative Location Sensing (CLS) was proposed in [Fretzagias and Papadopouli 2004]. However, CLS is designed for powerful nodes (e.g., PDAs), while our scheme further uses iterative refinement to improve the performance with small storage overhead. Therefore, our technique can be implemented and executed efficiently on resource constrained sensor nodes.

Similar to our attack-resistant location estimation techniques, the following two techniques are independently discovered to tolerate malicious attacks against location discovery in wireless sensor networks. A robust statistical methods that is similar to the attacker-resistant MMSE scheme is discovered in [Li et al. 2005] to achieve robustness through Least Median of Squares. A secure range-independent localization scheme (SeRLoc) that is similar to our voting-based scheme is discovered in [Lazos and Poovendran 2004] to protect location discovery with the help of sectored antennae at beacon nodes. Compared to these two studies, we provide more alternative ways to tolerate malicious attacks and also include the real implementation and field experiments in this paper.

SPINE [S.Capkun and Hubaux 2005] is developed to protect location discovery by using verifiable multilateration. However, the distance bounding techniques required for verifiable multilateration may not be available due to the difficulties to (1) deal with the external replay attacks in Ultrasound-based distance bounding and (2) achieve nanosecond processing and time measurements in Radio-based distance bounding. ROPE [Lazos et al. 2005] is developed by integrating SerLoc and SPINE. However, it still requires nanosecond processing and time measurements that are not desirable for the current generation of sensor networks. Compared with these two studies, we provide techniques to tolerate malicious attacks without the above constraints. Moreover, our proposed techniques can be easily combined with most of existing localization techniques.

To further enhance the security of location discovery, a practical technique is developed to detect malicious beacon nodes that are providing malicious beacon signals [Du et al. 2005; Liu et al. 2005b]. This detection technique can be easily combined with our techniques. We consider it complementary to the techniques in this paper.

In addition to secure location discovery, location privacy becomes a more and more interesting topic recently. Several techniques are developed recently to protect the location privacy in sensor networks [Ozturk et al. 2004; Kamat et al. 2005].

Security in sensor networks has attracted a lot of attention in the past several years. To provide practical key management, researchers have developed key pre-distribution techniques [Eschenauer and Gligor 2002; Chan et al. 2003; Du et al. 2003]. To enable broadcast authentication, a protocol named μTESLA has been explored to adapt to resource constrained sensor networks [Perrig et al. 2001]. Security of sensor data has been stud-

ied in [Przydatek et al. 2003; Hu and Evans 2003a]. Attacks against routing protocols in sensor networks and possible counter measures were investigated in [Karlof and Wagner 2003]. The research in this paper addresses another fundamental security problem that has not drawn enough attention.

9. CONCLUSION AND FUTURE WORK

In this paper, we proposed an attack-resistant MMSE-based location estimation and a voting-based location estimation technique to deal with attacks in localization schemes. We have implemented the proposed techniques on MICA2 motes [Crossbow Technology Inc.] running TinyOS [Hill et al. 2000], and evaluated them through both simulation and field experiments. Our experiences indicate that the proposed techniques are promising solutions for securing location discovery in wireless sensor networks.

Our future research is two-fold. First, we will study how to combine the proposed techniques with other protection mechanisms such as wormhole detection. Second, our simulations and experiments in this paper are conducted in small scales. It is very interesting to study the performance in a large scale.

Acknowledgment

The authors would like to thank the anonymous reviewers and Dr. Adrian Perrig for their valuable comments on the earlier version of this paper.

REFERENCES

AKYILDIZ, I., SU, W., SANKARASUBRAMANIAM, Y., AND CAYIRCI, E. 2002. Wireless sensor networks: A survey. *Computer Networks 38*, 4, 393–422.

BULUSU, N., HEIDEMANN, J., AND ESTRIN, D. 2000. GPS-less low cost outdoor localization for very small devices. In *IEEE Personal Communications Magazine*. 28–34.

CHAN, H., PERRIG, A., AND SONG, D. 2003. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Research in Security and Privacy*. 197–213.

CROSSBOW TECHNOLOGY INC. Wireless sensor networks. `http://www.xbow.com/Products/Wireless_Sensor_Networks.htm`. Accessed in May 2005.

DOHERTY, L., PISTER, K. S., AND GHAOUI, L. E. 2001. Convex optimization methods for sensor node position estimation. In *Proceedings of INFOCOM'01*.

DU, W., DENG, J., HAN, Y. S., AND VARSHNEY, P. 2003. A pairwise key pre-distribution scheme for wireless sensor networks. In *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*. 42–51.

DU, W., FANG, L., AND NING, P. 2005. Lad: Localization anomaly detection for wireless sensor networks. In *Proceedings of the 19th IEEE International Parallel & Distributed Processing Symposium (IPDPS '05)*.

ESCHENAUER, L. AND GLIGOR, V. D. 2002. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*. 41–47.

FRETZAGIAS, C. AND PAPADOPOULI, M. 2004. Cooperative location-sensing for wireless networks. In *Second IEEE International conference on Pervasive Computing and Communications*.

GAY, D., LEVIS, P., VON BEHREN, R., WELSH, M., BREWER, E., AND CULLER, D. 2003. The nesC language: A holistic approach to networked embedded systems. In *Proceedings of Programming Language Design and Implementation (PLDI 2003)*.

HE, T., HUANG, C., BLUM, B. M., STANKOVIC, J. A., AND ABDELZAHER, T. F. 2003. Range-free localization schemes in large scale sensor networks. In *Proceedings of ACM MobiCom 2003*.

HILL, J., SZEWCZYK, R., WOO, A., HOLLAR, S., CULLER, D., AND PISTER, K. S. J. 2000. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*. 93–104.

HU, L. AND EVANS, D. 2003a. Secure aggregation for wireless networks. In *Workshop on Security and Assurance in Ad Hoc Networks*.

HU, L. AND EVANS, D. 2003b. Using directional antennas to prevent wormhole attacks. In *Proceedings of the 11th Network and Distributed System Security Symposium*. 131–141.

HU, Y., PERRIG, A., AND JOHNSON, D. 2003. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In *Proceedings of INFOCOM 2003*.

INTANAGONWIWAT, C., GOVINDAN, R., AND ESTRIN, D. 2003. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the sixth annual international conference on Mobile computing and networking (Mobicom '00)*. 56–67.

KAMAT, P., ZHANG, Y., TRAPPE, W., AND OZTURK, C. 2005. Enhancing source-location privacy in sensor network routing. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS '05)*. 599–608.

KARLOF, C. AND WAGNER, D. 2003. Secure routing in wireless sensor networks: Attacks and countermeasures. In *Proceedings of 1st IEEE International Workshop on Sensor Network Protocols and Applications*.

KARP, B. AND KUNG, H. T. 2000. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of ACM MobiCom 2000*.

LAZOS, L., CAPKUN, S., AND POOVENDRAN, R. 2005. Rope: Robust position estimation in wireless sensor networks. In *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks (IPSN '05)*.

LAZOS, L. AND POOVENDRAN, R. 2004. Serloc: Secure range-independent localization for wireless sensor networks. In *ACM workshop on Wireless security (ACM WiSe 2004)*. Philadelphia, PA.

LI, Z., TRAPPE, W., ZHANG, Y., AND NATH, B. 2005. Robust statistical methods for securing wireless localization in sensor networks. In *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks (IPSN '05)*.

LIU, D., NING, P., AND DU, W. 2005a. Attack-resistant location estimation in wireless sensor networks. In *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks (IPSN '05)*.

LIU, D., NING, P., AND DU, W. 2005b. Detecting malicious beacon nodes for secure location discovery in wireless sensor networks. In *Proceedings of the 25th International Conference on Distributed Computing Systems (ICDCS '05)*.

NAGPAL, R., SHROBE, H., AND BACHRACH, J. 2003. Organizing a global coordinate system from local information on an ad hoc sensor network. In *IPSN'03*.

NASIPURI, A. AND LI, K. 2002. A directionality based location discovery scheme for wireless sensor networks. In *Proceedings of ACM WSNA'02*.

NEWSOME, J. AND SONG, D. 2003. GEM: graph embedding for routing and data-centric storage in sensor networks without geographic information. In *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys '03)*. 76–88.

NICULESCU, D. AND NATH, B. 2001. Ad hoc positioning system (APS). In *Proceedings of IEEE GLOBECOM '01*.

NICULESCU, D. AND NATH, B. 2003a. Ad hoc positioning system (APS) using AoA. In *Proceedings of IEEE INFOCOM 2003*. 1734–1743.

NICULESCU, D. AND NATH, B. 2003b. DV based positioning in ad hoc networks. In *Journal of Telecommunication Systems*.

OZTURK, C., ZHANG, Y., AND TRAPPE, W. 2004. Source-location privacy in energy-constrained sensor network routing. In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks (SASN '04)*. 88–93.

PERRIG, A., SZEWCZYK, R., WEN, V., CULLER, D., AND TYGAR, D. 2001. SPINS: Security protocols for sensor networks. In *Proceedings of Seventh Annual International Conference on Mobile Computing and Networks*.

PRZYDATEK, B., SONG, D., AND PERRIG, A. 2003. SIA: Secure information aggregation in sensor networks. In *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys '03)*.

RATNASAMY, S., KARP, B., YIN, L., YU, F., ESTRIN, D., GOVINDAN, R., AND SHENKER, S. 2002. GHT: A geographic hash table for data-centric storage. In *Proceedings of 1st ACM International Workshop on Wireless Sensor Networks and Applications*.

RAY, S., UNGRANGSI, R., PELLEGRINI, F. D., TRACHTENBERG, A., AND STAROBINSKI, D. 2003. Robust location detection in emergency sensor networks. In *Proceedings of IEEE INFOCOM 2003*.

SASTRY, N., SHANKAR, U., AND WAGNER, D. 2003. Secure verification of location claims. In *ACM Workshop on Wireless Security*.

SAVVIDES, A., HAN, C., AND SRIVASTAVA, M. 2001. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of ACM MobiCom '01*. 166–179.

SAVVIDES, A., PARK, H., AND SRIVASTAVA, M. 2002. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of ACM WSNA '02*.

S.CAPKUN AND HUBAUX, J. 2005. Secure positioning of wireless devices with application to sensor networks. In *Proceedings of IEEE InfoCom'05 (to appear)*.

SHENKER, S., RATNASAMY, S., KARP, B., GOVINDAN, R., AND ESTRIN, D. 2002. Data-centric storage in sensornets. In *Proceedings of the First ACM Workshop on Hot Topics in Networks*.

YU, Y., GOVINDAN, R., AND ESTRIN, D. 2001. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Tech. Rep. UCLA/CSD-TR-01-0023, UCLA, Department of Computer Science. May.