EFFICIENT LOCALIZATION SCHEMES IN SENSOR NETWORKS WITH MALICIOUS NODES

Kaiqi Xiong and David Thuente
Department of Computer Science, North Carolina State University, Raleigh, NC 27695-7534, U.S.A.

Keywords: Wireless sensor network (WSN), Sensor localization, Vulnerability, and Security.

Abstract: The accuracy of location information is critical for many applications of wireless sensor networks (WSN), especially those used in hostile environments where malicious adversaries can be present. It is impractical to have a GPS device on each sensor in WSN due to costs. Most of the existing location discovery schemes can only be used in the trusted environment. Recent research has addressed security issues in sensor network localization but, to the best of our knowledge, none has completely solved the secure localization problem. In this paper, we propose novel schemes for secure dynamic localization in sensor networks. The proposed algorithms tolerate up to 50% of beacon nodes being malicious and they have linear computation time with respect to the number of reference nodes. We have conducted simulations to analyze their performance.

1 INTRODUCTION

Sensor networks may become the next wave of information technology. Distributed networks of thousands of collaborative sensors promise long-lived and unattended systems for many monitoring, surveillance and control applications such as health and gas pipe monitoring and data acquisition in battlefield and other hazardous environments. Many applications require knowledge of sensor positions. The location information may save energy and life, e.g., see (Hu and Evans, 2004), (Karp and Kung, 2003), and (Mauve et al., 2001). Secure location discovery for sensor networks is crucial in a hostile environment. Without security, sensor locations may be estimated through compromised nodes. Finding sensor locations is a challenging problem due to sensor constraints such as limited energy, computation, and communication.

Due to computation, power, cost, and storage constraints of sensor networks, GPS will not usually be installed on every sensor node. Furthermore, GPS works only in outdoor unshielded environments (He et al., 2003) and (Wellenhoff et al., 1997). In recent years, many localization schemes (see (Bahl and Padmanabhan, 2000), (Liu et al., 2005a), (Mainnwaring et al., 2002), and (Niculescu and Nath, 2001)) have been proposed for sensor networks without depending on expensive GPS devices. Most of these schemes assume some special nodes, called beacon nodes, have the capability to know their own location either through GPS receivers or manual configuration. Non-beacon sensor nodes can be equipped with relatively cheap measuring devices for signal strength, directionality, or time of arrival, etc. The non-beacon nodes can use these measurements and the locations of two or more beacon nodes to estimate their own locations. In addition, range-free techniques have also been proposed to solve for sensor localization problem (Bulusu et al., 2004) and (He et al., 2003). No range equipment except for beacon nodes is needed in these techniques. For example, a sensor node computes its position using hop-counts received from beacons instead of distances. The hop-count is used as an estimate of sensor's physical location. Then the node finds the average distance per hop through the beacon node's communication. Moreover, Niculescu et al. (Niculescu and Nath, 2001) described a similar scheme but improved the accuracy of the distance estimation by using the average hop count of all the neighbors of a node as a distance estimate. When three location references are received by a sensor node, triangulation is used to estimate its location. If a node receives more than three location references from beacons then the least-square optimization method will be performed to find the location.

Most of the above protocols discussed are vulnerable. Security has played an important role in many sensor networks applications because sensors are often unattended and easily attacked. An unprotected sensor node may localize to a wrong posi-

Xiong K. and Thuente D. (2008).
EFFICIENT LOCALIZATION SCHEMES IN SENSOR NETWORKS WITH MALICIOUS NODES.
In *Proceedings of the International Conference on Security and Cryptography*, pages 190-196
DOI: 10.5220/0001927501900196

tion through compromised nodes with possible severe consequences. Secure localization has attracted considerable attention over the last a few years. In this paper, we propose several methods for secure location discovery in sensor networks.

The remainder of this paper is organized as follows. In section 2 we describe several secure sensor localization methods, including a secure dynamic localization method. Security analysis for the secure dynamic localization method is studied in section 3. Our simulation results are reported in section 4. Related work is discussed in section 5 and conclusions are presented in section 6.

2 SECURE SENSOR LOCALIZATION METHODS

In this section we present several novel approaches for secure localization in sensor networks. We describe two naive methods using the concepts of mean and median values. Then we develop dynamic localization methods to improve the accuracy of location estimates so these methods become feasible in practice.

Let (x, y) be the coordinate of node N which wants to determine its position. Assume there are n beacons B_i that know their own positions (x_i, y_i) in the sensor network ($i = 1, 2, \cdots, n$). Denote by d_i the measured distance between (x, y) and (x_i, y_i) which may stem from the different types of measurements such as signal strength, time of arrival or hop count in a single or multi-hop sensor network, see (Bulusu et al., 2004), (Doherty et al., 2001), (He et al., 2003) and (Niculescu and Nath, 2001). The problem of secure sensor localization is to find an accurate location estimation based on references from beacons when there are malicious beacons.

In this section we present two simple localization methods: the mean-based localization method and the median-based localization methods. However, a single malicious reference may result in the average value far from its true coordinate in the former method. Moreover, the latter method can only tolerate up to about 20% malicious beacon reference nodes (see section 3). Hence, we propose secure localization schemes and secure dynamic localization schemes to improve the median-based localization methods.

2.1 The Mean-based Localization Method

In the beacon-based technique, the problem of sensor localization discovery is how to determine the coordinate (x, y) based on the positions of beacon nodes B_i as references. The triangulation process, usually used in this technique, of determining the coordinate is to select three measurement tuples from the collection $\{(x_i, y_i, d_i)\}_{i=1,2,\cdots,n}$, and solve for (x, y) based on the the following equations

$$(x - x_{i_j})^2 + (y - y_{i_j})^2 = d_{i_j}^2 \quad \text{for} \quad j = 1, 2, 3$$

Denote the solutions by $x = x^j$ and $y = y^j$ for $j = 1, 2, \cdots, m$, where m is the total number of combinations consisting of three measurement tuples that can determine the coordinate. Ideally, the tuple reference values $\{(x_i, y_i, d_i)\}_{i=1,2,\cdots,n}$ are not disrupted by a malicious node. Let e_i^j be the estimated difference between d_i and the distance computed by each derived estimation (x^j, y^j) to $\{(x_i, y_i)\}_{i=1,2,\cdots,n}$ for $j = 1, 2, \cdots, m$.

Their differences are caused by the presence of measurement noises. Precisely, let

$$\sigma^x = \left[\frac{1}{m-1}\sum_{j=1}^{m}(x^j - \mu^x)^2\right]^{\frac{1}{2}}, \ \sigma^y = \left[\frac{1}{m-1}\sum_{j=1}^{m}(y^j - \mu^y)^2\right]^{\frac{1}{2}}$$

Then the coordinate (x, y) should follow a two-dimensional uniform (Gaussian) distribution. Its probability distribution function is given by:

$$p(x, y) = \frac{1}{2\pi\sigma^x\sigma^y} e^{-\frac{1}{2}\left[\left(\frac{x-\mu^x}{\sigma^x}\right)^2 + \left(\frac{y-\mu^y}{\sigma^y}\right)^2\right]}$$

where $\sigma^x \neq 0$ and $\sigma^y \neq 0$. For notational simplicity, let $\eta = \eta(x,y)$ be defined by $\eta = \sqrt{(\frac{x-\mu^x}{\sigma^x})^2 + (\frac{y-\mu^y}{\sigma^y})^2}$ and we give the following definition.

Definition 1: Given a predefined value $\gamma > 0$, coordinate $(\tilde{x}, \tilde{y})$ is called a γ-polluted point if $\eta(\tilde{x}, \tilde{y}) \geq \gamma$.

Thus, a mean-based localization method (MALM) to determine coordinate (x, y) is given as follows.

Algorithm 1.

1. Select every three measurement tuples from $\{(x_i, y_i, d_i)\}_{i=1,2,\cdots,n}$ and compute (x^j, y^j) triangulation method. Let S denote a collection of (x^j, y^j) ($j = 1, 2, \cdots, m$).
2. For each (x^j, y^j) and a predefined γ (usually $\gamma > 1$), determine if (x^j, y^j) is a γ-polluted point. If yes, delete it from S. Repeat the step until all elements in S are checked. Denote the remaining set of S by $\hat{S}$.
3. Calculate the average point $(\hat{x}, \hat{y})$ by computing the average x-coordinate and y-coordinate values of all elements in $\hat{S}$. Then $(\hat{x}, \hat{y})$ is an estimation coordinate of (x, y) for sensor N.

However, when there are malicious nodes in a sensor network, some of values (x^j, y^j) may be significantly different from the true values because of an

attack such as a wormhole attack. When the number of samples is small, a single incorrect value (x^j, y^j) may significantly change the distribution of (x^j, y^j) $(j = 1, 2, \cdots, m)$. Thus, the MALM method will not work well. This is because a mean-value point may not be in the center of measurement tuples. To improve the estimation, we now propose the following methods based on the concept of a center of gravity.

2.2 The Median-based Localization Methods

When there is a significant point far away from others, a mean-value point is not in the center of estimation points. The median-value point is located in the center of these estimation points in term of a predefined metric and is a random variable (a robust estimator of the center) (Huber, 1981).

Let d^j be the Euclidean distance of (x^j, y^j) from the origin given by $d^j = \sqrt{(x^j)^2 + (y^j)^2}$ $(j = 1, 2, \cdots, m)$. Sort the sequence $\{d^j\}$ $(j = 1, 2, \cdots, m)$ in increasing order. Without loss of generality, assume that the sequence $\{(x^1, y^1), \cdots, (x^m, y^m)\}$ is sorted. A simple way to define the center of the sequence $\{(x^j, y^j)\}_{j=1,2,\cdots,m}$ is to use distance d^j as a measure. The median point of the sequence $\{(x^j, y^j)\}_{j=1,2,\cdots,m}$ is, (x^M, y^M) is a point such that $d^M = \sqrt{(x^M)^2 + (y^M)^2}$ is in the center of sequence $\{d^j\}_{j=1,2,\cdots,m}$. Then $x^M = x^{\frac{m+1}{2}}$ if m is odd; otherwise, $x^M = \frac{x^{\frac{m}{2}} + x^{\frac{m}{2}+1}}{2}$. Similarly, y^M is defined. However, such a definition does not really reflect the center of sequence $\{(x^j, y^j)\}_{j=1,2,\cdots,m}$. Here we let x^M and y^M be the medians of sequences $\{x^j\}$ and $\{y^j\}$ respectively. Then (x^M, y^M) is used as the center of sequence $\{(x^j, y^j)\}$. Another possible definition is to use such a point in $\{(x^j, y^j)\}$ $(j = 1, 2, \cdots, m)$ that it is the closest to (x^M, y^M) in term of an Euclidean distance. Please also refer (Bernholt and Fried, 2003) for a further definition and computation of a median as well. For the estimation points (x^j, y^j), we can shift them by (x^M, y^M), denoted $\check{x}^j = x^j - x^M$ and $\check{y}^j = y^j - y^M$. Then we calculate their means by $\check{\mu}^x = \frac{1}{m}\sum_{j=1}^{m}(x^j - x^M) = \mu^x - x^M$ and $\check{\mu}^y = \frac{1}{m}\sum_{j=1}^{m}(y^j - y^M) = \mu^y - y^M$. Furthermore, we compute their standard deviations by

$$\check{\sigma}^x = \left[\frac{1}{m-1}\sum_{j=1}^{m}(\check{x}^j - \check{\mu}^x)^2\right]^{\frac{1}{2}}, \quad \check{\sigma}^y = \left[\frac{1}{m-1}\sum_{j=1}^{m}(\check{y}^j - \check{\mu}^y)^2\right]^{\frac{1}{2}}$$

It is easy to see that $\check{\sigma}^x = \sigma^x$ and $\check{\sigma}^y = \sigma^y$.

Similar to the previous section, a median-based localization method (MDLM-1) is derived as follows.

Algorithm 2.

1. Use Step 1 in the MALM method to find (x^j, y^j) and then compute $(\check{x}^j, \check{y}^j)$ $(j = 1, 2, \cdots, m)$.
2. For each $(\check{x}^j, \check{x}^j)$ and a predefined γ (usually $\gamma > 1$), determine if $(\check{x}^j, \check{x}^j)$ is a γ-polluted point. If yes, delete it from S. Repeat the step until all elements in S are checked and denote the remaining set of S by $\hat{S}$. At this time, note that η is given by $\eta = \sqrt{(\frac{\check{x}-\check{\mu}^x}{\check{\sigma}^x})^2 + (\frac{\check{y}-\check{\mu}^y}{\check{\sigma}^y})^2}$.
3. Calculate the average point by computing the average values of x-coordinate and y-coordinate of all elements in $\hat{S}$ respectively, denoted by $(\hat{x}, \hat{y})$. Then $(\hat{x}, \hat{y})$ is an estimation coordinate of (x, y) for sensor N.

The difference between MALM and MDLM-1 methods is that (x^j, y^j) is shifted by its mean value in MALM and its median-value point in MDLM-1. Both methods have the computation time of $\Theta(m)$.

Let e_i^j be the difference between d_j and the estimated distance computed by each estimated coordinate (x^e, y^e) to $\{(x_i, y_i)\}_{i=1,2,\cdots,n}$ for $j = 1, 2, \cdots, m$. Assume that e_i^j follows a normal distribution with mean value 0 and standard deviation ε. (Note that we do not care about the specific distribution of e_i^j. We only need to have the absolute value of e_i^j's offset, denoted by the parameter ε.) Then we derive a different median-based localization method, called MDLM-2.

Algorithm 3.

1. Use Step 1 in MALM to find (x^j, y^j) and their median coordinate (x^M, y^M) $(j = 1, 2, \cdots, m)$.
2. For each $\{(x_i, y_i)\}_{i=1,2,\cdots,n}$, compute
$$e_i = d_i - \sqrt{(x_i - x^M)^2 + (y_i - y^M)^2}$$
Let $\mathcal{D}$ be the set of points $\{(x_i, y_i, d_i)\}$ satisfying $|e_i| \leq \varepsilon$
3. Apply the minimum mean square error (MMSE) method to $\mathcal{D}$ to find an estimation coordinate of (x, y) for sensor N.

MDLA-2 rechecks the accuracy of (x^M, y^M), a prediction by computing e_i^j. But, (x^M, y^M) can be produced by correct location references only if a sensor network has no more than 20% malicious beacons. A study is conducted to verify this in section 3.

2.3 The Secure Dynamic Localization Method

In the previous two sections, we developed three localization methods for securely determining the coordinates of a sensor. The efficiency of these three

methods depends on m. Every three nonlinear tuples $\{(x_i, y_i, d_i)\}_{i=1,2,\cdots,n}$ can be used to derive an estimation coordinate. There are $\binom{n}{3}$ possible choices in selecting 3 from n, that is, $m = \binom{n}{3} = \frac{n(n-1)(n-2)}{6}$. Recall that each of these three previous methods has the computational cost of $\Theta(m)$. For example, when there are 150 beacons, $m = 551300$. Hence, all three methods are computationally burdensome to a sensor with low computational capacity or depletable battery. We will present an algorithm that significantly enhances the efficiency of the MDLM-2 method and also tolerates up to 50% beacon nodes being malicious.

We denote by $\mathcal{A}$ the collection of measurement tuples $\{(x_i, y_i, d_i)\}_{i=1,2,\cdots,n}$. The secure localization method (SELM) is:

Algorithm 4.

1. Choose an integer number r and randomly select k measurement tuples from $\mathcal{A}$. By applying Step 1 in Algorithm 1 to every three of the chosen k measurement tuples, we find its estimated coordinates and their median coordinate. Repeat the above procedure r times and let (x_j^M, y_j^M) be the median coordinate where $j = 1, \cdots, r$, and k should be chosen as $3 \le k << n$.
2. For each (x_j^M, y_j^M), calculate
$$e_{ij} = d_i - \sqrt{(x_i - x_j^M)^2 + (y_i - y_j^M)^2}$$
for $i = 1, 2, \cdots, n$ and $j = 1, 2, \cdots, r$.
3. For a predefined value $\varepsilon > 0$, let $\mathcal{D}_j$ be a set of such points $\{(x_i, y_i, d_i)\}$ satisfying $|e_{ij}| \le \varepsilon$. Let $\mathcal{D}_{\max}$ be one of $\{\mathcal{D}_j\}_{j=1,2\cdots,r}$ that contains the largest number of elements.
4. By applying the MMSE method to $\mathcal{D}_{\max}$, we find an estimation coordinate of (x, y) for sensor N.

Clearly, the computation times are $\Theta(rk^3)$ for Step 1, $\Theta(rn)$ for Steps 2 and 3, and $\Theta(|\mathcal{D}_{\max}|)$ for Step 4, where $|\mathcal{D}_{\max}|$ is the number of elements in $\mathcal{D}_{\max}$. Thus, the total computation time of Algorithm 4 is C=$\Theta(rk^3)$+$\Theta(rn)$+$\Theta(|\mathcal{D}_{\max}|)$. That is, $C = \max\{\Theta(rk^3), \Theta(rn)\}$ is much less than the computation time, $\Theta(n^3)$, in Algorithms 1-3 respectively since r can be chosen as a small number. In section 3, we prove that for a given k, we can get a correct median coordinate as the estimation of node N's position in r times, when r is chosen as a sufficiently big (but still small) integer. We will show how to choose positive integers k and r to meet predefined performance based on affordable resources in section 3. According to our analysis there, r can be chosen as a relatively small number and the computation time of Algorithm 4 is approximately equal to $\max\{\Theta(k^3), \Theta(n)\}$. Furthermore, since more than 50% of beacon nodes provide correct reference information, $\mathcal{D}_{\max}$ will be generated by a correct median coordinate that is computed based on correct location references, or a correct estimation of node N's coordinate, according to the computation method of the median coordinate.

In Algorithm 4, r is a pre-selected value based on the information provided in the security analysis of section 3. However, due to the limited computation and storage in sensor networks, any extra computation and storage may be a burden and deplete the battery. Moreover, we may be lucky to get a correct median coordinate before finishing r rounds. This suggests the following secure dynamic localization method (SDLM).

Algorithm 5.

1. Randomly select k measurement tuples from $\mathcal{A}$. By applying Step 1 in Algorithm 1 to every three of the chosen k measurement tuples, we find its estimated coordinates and their median coordinate, denoted as $\{(x^M, y^M)\}$, where $3 \le k << n$.
2. For each $(x^M, y^M)\}$, calculate
$$e_i = d_i - \sqrt{(x_i - x^M)^2 + (y_i - y^M)^2} \quad (i = 1, \cdots, n)$$
3. For a predefined value $\varepsilon > 0$, let $\mathcal{D}$ be a set of such points $\{(x_i, y_i, d_i)\}$ satisfying $|e_i| \le \varepsilon$.
4. If $\mathcal{D}$ contains more than 50% of beacon nodes, then apply MMSE to $\mathcal{D}$ to find an estimation coordinate of (x, y) for sensor N, denoted by (x^e, y^e).
5. For a predefined $\delta > 0$, calculate
$$\hat{e}_i = d_i - \sqrt{(x_i^d - x^e)^2 + (y_i^d - y^e)^2}$$
where $(x_i^d, x_i^d) \in \mathcal{D}$ $(i = 1, 2, \cdots, |\mathcal{D}|)$. If $\frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|} |\hat{e}_i| \le \delta$, then select (x^e, y^e) as the estimation of (x, y) for node N's coordinate and then exit. Otherwise, repeat Steps 1-5.

Similarly, the computation times are $\Theta(k^3)$ for Step 1, $\Theta(n)$ for Steps 2 and 3, $\Theta(|\mathcal{D}|)$ for Step 4, and $\Theta(|\mathcal{D}|)$ for Step 5. Thus, the total computational cost of Algorithm 5 is $C = \Theta(k^3) + \Theta(n) + \Theta(|\mathcal{D}|)$ multiplied by the number of repeated times. As mentioned early, the number of repeated times is usually small as studied in section 3. Hence, the computational cost in Algorithm 5 is $\max\{\Theta(k^3), \Theta(n)\}$. Hence, Algorithm 5 also has much less computation time than Algorithms 1-3. Accordingly, the selection of $\delta > 0$ is based on a sensor's performance requirement and available resources. Usually, $\delta > 0$ should be chosen as a value such that it is bigger than $\varepsilon > 0$. This is because using the MMSE method to find a best-fit cannot guarantee that it satisfies $|e_i| \le \varepsilon$ for all

$(x_i^d, x_i^d) \in \mathcal{D}$ $(i = 1, 2, \cdots, |\mathcal{D}|)$. Moreover, from the above analysis we know that if the number of iterations in Algorithm 5 is big enough, such a coordinate (x^e, y^e) can be found.

Algorithms 4 and 5 greatly improve the efficiency of MDLM-1 and MDLM-2. The SDLM method cannot guarantee deriving an optimal coordinate of a sensor. But, its solution is sub-optimal. Moreover, an obvious question is how to choose k. In general, the smaller k, the less computation time. In particular, if k is chosen as a number with $k \leq n^{1/3}$, then the computation time of the SELM and SDLM methods is $\Theta(n)$, i.e., the SELM and SDLM methods are linear. To keep SELM and SDLM being practical, k should be chosen as an integer with $k << n$. Actually, based on the security analysis of section 3, k and r can be chosen as relatively small values.

3 SECURITY PERFORMANCE ANALYSIS

Algorithms 4-5 are designed based on the same technique: randomly select k references from n beacons. Hence, we now only discuss the security performance analysis of Algorithms 4-5; that is, we seek an r in Algorithm 4, or the number of repeated times in Algorithm 5 (for simplicity, it is also denoted as r), required to obtain at least a correct median coordinate with a given probability so that the location of a sensor can be closely estimated.

Recall that n represents the number of beacon nodes in a sensor network that can provide location references for node N. Let q be the number of malicious nodes among these beacon nodes in the network. In the first round of the SELM method, we randomly select k measurement tuples in $\mathcal{A}$ from n beacon nodes, and then estimate the coordinate of node N by every three tuples chosen from the k nodes. The total number of estimated coordinates is $\binom{k}{3}$.

Let's first study how by chance we can get over 50% coordinates that are not determine by any single malicious nodes in the chosen k beacons. Denote by b the malicious nodes in the chosen k beacons. Then, the probability that a coordinate is not determined by malicious beacons is $p_b = \frac{\binom{k-b}{3}}{\binom{k}{3}} = \frac{(k-b)(k-b-1)(k-b-2)}{k(k-1)(k-2)}$. By using both analytical and simulation methods, we have found that in order to get $p_b \geq 50\%$, we need to approximately have $b \leq \lfloor \frac{k}{5} \rfloor$, that is, no more than 20% of the chosen k beacons are malicious, where $\lfloor \frac{k}{5} \rfloor$ is a floor value of $\frac{k}{5}$. (Note that the analysis also indicates that Algorithms 1 and 2 can only tolerate up to about 20% beacon nodes being malicious.) In the first round, the probability for selecting exactly t measurement tuples from q malicious nodes is $p(t) = \frac{\binom{q}{t}\binom{n-q}{k-t}}{\binom{n}{k}}$. As is known, we can determine the coordinate of node N correctly if less than half of these k nodes are malicious. Consequently, the probability that we can determine coordinates for node N is

$$p = \sum_{t=0}^{\lfloor \frac{k}{5} \rfloor} p(t) = \sum_{t=0}^{\lfloor \frac{k}{5} \rfloor} \frac{\binom{q}{t}\binom{n-q}{k-t}}{\binom{n}{k}}$$

Note the identity $\sum_{t=0}^{k} \binom{q}{t}\binom{n-q}{k-t} = \binom{n}{k}$.

We want the probability that we randomly select k reference tuples from n beacon nodes and repeat the selection r times. Then, the probability that we have at least one chance to get a median coordinate as a correct estimation of node N's coordinate is $P = 1 - (1-p)^r$. Table 1 shows the number of times (r) that we need to repeatedly choose (k) location references so that we have 99% chance to get at least one correct median coordinate in r trials, when k=5 and 10. Subsequently, we can find $\mathcal{D}_{\max}$ in Algorithm 4 and the estimated coordinate of sensor node N when $\varepsilon > 0$ is properly chosen. Surprisingly, when p is fixed, the selection of k and r does not depend on the number of beacon nodes, n, based on our simulation. Also, the chance that we can get a correct median does not increase as k increases. Conversely, the bigger k, the bigger r. This means that the more computation and storage cost is required as k increases. Hence, according to our experiment, $k = 5$ is a good selection.

Table 1: The number of repeated times so that 99% chance to obtain at least one correct median coordinate.

	Percentage of Malicious Location References		
	30%	40%	50%
k=5	r=6	r=11	r=21
k=10	r=12	r=25	r=84

Due to the above analysis, k and r can be chosen as very small integers compared to n. Hence, Algorithms 4-5 usually have linear computation time with respect to the number of beacon nodes n. However, Algorithms 1-3 have cubic computation time in n.

4 NUMERICAL SIMULATION

In this section we shall demonstrate the simulation results of our proposed method. Due to the page limit we only present the simulation results of Algorithm 5 in section 2.3. We shall show how the SDLM method performs in terms of localization error and efficiency.

In the simulation, we assume that all beacon nodes including malicious nodes are evenly deployed in an $200 \times 200m^2$ square field. Assume that a non-beacon node can receive the signal from each beacon node in this field, but a certain percentage of beacon nodes declare their wrong location information due to attacks. We implement SDLM in Java program over a Linux environment. We assume the origin point in the coordinate system as the true location of a sensor that we want to find. A set of 500 beacon references are first randomly created. Each point contains a tuple of (x, y, d). d is the distance from the reference point to the origin. These references may be malicious points. Value d may be incorrect because of incorrect values x and y, where d is calculated by $d = \sqrt{x^2 + y^2}$. We assume a simple measurement error model, i.e., each sensor cannot be further away from its true location by more than 4 meters. That is, ε in the SDLM method is chosen as 4 meters. This is used to eliminate malicious references. In each run, we randomly choose 10 references from the 500 beacon references, i.e., $k = 10$. The number of runs is 20.

We measured the localization error as the distance from the estimated location to the true location which is the origin. It is shown in Figure 1 that the distance error increases as the percentage of malicious nodes around the sensor node's location to be estimated increases. We can see that the error is below 3 meters even when 50% of reference nodes are malicious. In most applications, that distance error may be acceptable. For example, such applications include finding a missing child in a forest or identifying the location of natural disaster. In applications which have a high demand on location information such as routing protocols, our algorithm can be still used if the number of malicious nodes is less than 10% percent of all reference nodes in the non-polluted range.

Furthermore, we study the efficiency of the method compared to the percentage of malicious nodes. In the simulation, we use $k = 10$ and define the efficiency as the number of runs required to find the accurate location. This test is performed by a brute force method. From Figure 2, we see that a sensor can find its location in fewer steps than the number of steps based on our analytic model, presented in Table 1 when the percentage of malicious nodes is 30%, 40% and 50%.

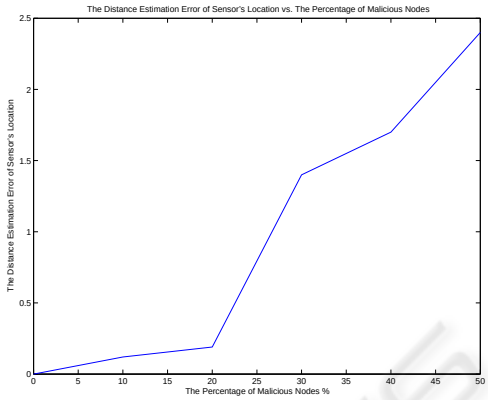

Figure 1: The Distance Estimation Error of Sensor's Location. The Unit of y Axis is Meter.

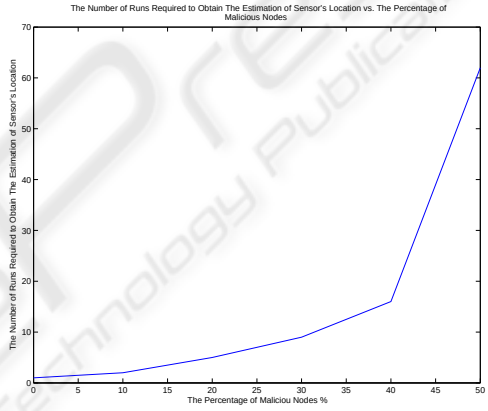

Figure 2: The Number of Runs Required to Obtain The Estimation of Sensor's Location.

5 RELATED WORK

Many studies have been conducted on secure location discovery for wireless sensor networks in the last a few years, for example, (Bahl and Padmanabhan, 2000), (Liu et al., 2005a), (Mainnwaring et al., 2002), and (Niculescu and Nath, 2001). In this section, we summarize related work.

Time of arrival (TOA) technology is commonly used as a means of obtaining range information via signal propagation time (He et al., 2003). It is used in GPS for the most basic localization system (Wellenhoff et al., 1997). However, GPS is expensive for sensor networks. The time difference of arrival (TDOA) technique for range estimation between two communication nodes has been widely proposed as a necessary ingredient in sensor localization. Many infrastructure-based systems have used TDOA as a range estimating tool, for example, see (Bahl and Padmanabhan, 2000), (Doherty et al., 2001), (Priyantha et al., 2000) and (Want et al., 1992). Doherty, et al. in (Doherty et al., 2001) formulated the localization problem as a convex optimization problem and then solved it using the convex optimization approach. In (Bahl and Padmanabhan, 2000), received signal strength indicator (RSSI) was used to translate

signal strength into distance estimates.

In addition, range-free techniques have also been proposed to solve for sensor localization problem (see (Bulusu et al., 2004), (He et al., 2003), and (Niculescu and Nath, 2001)). The centroid of all locations in the received beacon signals has been proposed for sensor's location discovery in (Bulusu et al., 2004). In (Niculescu and Nath, 2001) DV-hop was used as an alternative solution. A sensor node computes its position using hop-counts received from beacons, instead of distances. Then, the node finds the average distance per hop through beacon nodes' communication.

The range-based localization schemes have been enhanced to address security concerns for sensor networks (e.g., (Liu et al., 2005a) and (Liu et al., 2005b)). Both an attack-assistant MMSE-based location estimation and a voting-based location estimation have been proposed to deal with attacks in location discovery in (Liu et al., 2005a). In the first method, the key point is to find a consistency set. That is usually not an easy task. There is the same difficulty seeking the highest vote area as in the latter method. Furthermore, in (Liu et al., 2005b) Liu et al. provided a method to reason about the suspiciousness of each beacon node at the base station based on the detection information from beacon nodes. In (Fretzagias and Papadopouli, 2004), Fretzagias et al. proposed another voting-based scheme, called the Cooperative Location Sensing (LCS).

Our median-based method is inspired by the centroid technique (Bulusu et al., 2004) and the MMSE method. As indicated, a mean value does not reflect the center of location references. Instead, a median is used to filter out outliers. In this paper we propose new median-based schemes for dealing with malicious references. In Algorithms 4-5 we can easily filter out malicious references and then estimate the location of a sensor node by using the MMSE method.

6 CONCLUSIONS

In this paper we proposed a suite of secure localization methods, including the secure dynamic localization method (Algorithm 5), for sensor networks. A median-based technique instead of a mean-based technique was used to represent the center of location references so that malicious reference information could be filtered out easily. Our security performance analysis has shown that the proposed secure localization methods can tolerate up to 50%malicious beacon nodes, and they usually have linear computation time. This is the best we can achieve. We further conducted simulations to demonstrate the applicability and accuracy of these algorithms. Preliminary validation tests showed that Algorithms 4-5 have a good accuracy against other algorithms. Detailed validation results are not provided due to the page limit.

REFERENCES

Bahl, P. and Padmanabhan, V. (2000). An in-building RF-based user location and tracking system. In *Proceedings of the IEEE INFOCOM '00*.

Bernholt, T. and Fried, R. (2003). Computing the update of the repeated median regression line in linear time. *Information Processing Letters*, (88):111–117.

Bulusu, N., Heidemann, J., and Estrin, D. (2004). GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34.

Doherty, L., Pister, K., and Ghaoui, L. (2001). Convex optimization methods for sensor node position estimation. In *Proceedings of INFOCOM '01*.

Fretzagias, C. and Papadopouli, M. (2004). Cooperative location-sensing for wireless networks. In *Proceedings of IEEE PerCom '04*.

He, T., Huang, C., Blum, B. M., Stankovic, J., and Abdelzaher, T. (2003). Range-free localization schemes for large scale sensor networks. In *MobiCom '03*.

Hu, L. and Evans, D. (2004). Localization for mobile sensor networks. In *MobiCom '04*.

Huber, P. J. (1981). *Robust statistics*. Addison-Wesley Publishing Company, New York.

Karp, B. and Kung, H. (2003). Greedy perimeter stateless routing. In *MobiCom'03*.

Liu, D., Ning, P., and Du, W. (2005a). Attack-resistant location estimation in sensor networks. In *Proceedings of 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 IPSN '05*.

Mainnwaring, A., Polastre, J., Szewczyk, R., Culler, D., and Anderson, J. (2002). Wireless sensor network for habitat monitoring. In *Proceedings of ACM WSNA'02*.

Mauve, M., Widmer, J., and hartenstein, H. (2001). A survey on position-based routing in mobile Ad-Hoc networks. *IEEE Network Magazine*.

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

Priyantha, N., Chakraborty, A., and Balakrishnan, H. (2000). The cricket location-support system. In *Proceedings of MOBICOM '00*.

Want, R., Hopper, A., Falcao, V., and Gibbons, J. (1992). The active badge location systems. *ACM Transactions on Information Systems*.

Wellenhoff, H., Lichtenegger, H., and Collins, J. (1997). *Global Positions System: Theory and Practice, Fourth Edition*. Springer Verlag.