Anti-Collusion Position Estimation in Wireless Sensor Networks

Negar Kiyavash
Information Trust Institude
Department of Computer Science
University of Illinois at Urbana-Champaign
Email: kiyavash@uiuc.edu

Farinaz Koushanfar
Department of Electrical and Computer Eng.
Department of Computer Science
Rice University
Email: farinaz@rice.edu

Abstract

Sensor networks are highly susceptible to errors and malicious attacks. A host of nefarious attacks are targeted at preventing nodes from discovering their correct positions. In this work, we present a novel framework for position estimation in presence of malicious attacks on distance measurements of sensor networks. Additionally, we propose a practical randomized algorithm in the framework, which efficiently detects and rejects the corrupted measurements. The algorithm searches for an agreeable solution starting from randomly sampled minimal subsets of data; it subsequently enhances its estimate by augmenting consistent data points to the best random sample. The performance of the proposed algorithm is evaluated and compared to state-of-the-art robust positioning algorithms, both for independent and colluding attackers. While our method performs the same or better compared with the other algorithms on independent attacks, it is significantly more robust against collusion attacks, in terms of both the position estimation error and attack diagnosis and isolation. Moreover, the algorithm has a shorter runtime due to its randomized nature.

1 Introduction

The goal of this paper is to present a theoretical framework for attack-resistant beacon-based position estimation. In light of proliferation of sensor networks both in military and civilian capacity, security breaches are inevitable [1]. Position information of the sensors is required by a variety of applications such as environmental monitoring and target tracking which directly depend on the physical location of the nodes or in protocols that indirectly rely on the position of the participating nodes, e.g., security protocols [2]. Due to the demand-initiated self-organizing structure of sensor networks, sensors are not apriori aware of their locations.

Constraints on size, cost and power consumption of nodes make the position estimation in sensor networks challenging. Current state of the art positioning techniques typically use distance measurements from a special set of reference nodes called beacons to estimate their position. A sensor node requires distances from three or more beacons to compute its coordinate. These computations apply multilateration (triangulation) [3] techniques to estimate the unknown position. The sensor nodes use signal features for instance Received Signal Strength (RSSI) [4] measurements for calculating the distance from the beacons.

The attackers can modify the position information of beacons either by spoofing or by compromising the beacons without limit. In response to these threats, we introduce a theoretical framework and a randomized algorithm in this framework for attack-resistance position estimation. In our view, the position information modifications by the attackers are regarded as *inconsistencies (anomalies)*. The node wishing to find its position performs anomaly diagnosis and isolation by using our proposed algorithm. The final position estimate is a function of the largest anomaly free data set identified by the algorithm.

Our attack-resilient position estimation framework is built upon the random sample consensus paradigm of Fischler and Bolles [5]. Random sample consensus is a paradigm for fitting a model to a data set corrupted by benign or malicious data; it was introduced in computer vision and has been successfully applied for more than 25 years in vision literature. Our main contributions are:

- A novel mathematical framework for attack-resilient position estimation is proposed. A novelty of our approach is that unlike traditional top-down optimization methods, we use a minimal subset of

1-4244-1455-5/07/$25.00 2007 IEEE

the data to form the initial model and then augment it with more data. The new framework *does not impose any restrictions on the whole data set* such as smoothness or prior distributions. Another key advantage of the proposed mathematical framework is that it *does not impose any limitations on the form of the objective function* used for position discovery.

- The paper presents a practical *randomized* algorithm for implementing the mathematical attack-resilient position estimation framework. Randomization had proven to be a powerful algorithmic technique that provides simpler and more efficient algorithms than its deterministic counterpart, both in theory (and more importantly) in practice. Randomized algorithms are widely used throughout computer science from cryptography and learning to networking. In particular, their unpredictable nature makes them inherently less vulnerable to a malicious adversary, see [6] for more details.

- The random sample consensus paradigm is fortified with a binary search and progressive threshold selection. The original paradigm assumes that an estimate of the number of attackers is available. For cases where this estimate is not available, our threshold selection can be also based on the distribution of the errors that we analytically derive. We devise a consistency measure to be used with our algorithm. We also select the number of runs for the randomized algorithm that probabilistically guarantees a high quality solution.

- A thorough comparison of the proposed algorithm with the best known robust positioning algorithms is presented. The comparison is carefully done with respect to the number of outliers, amplitude of errors, as well as altering the input parameters to our algorithm. We have considered both the case where the attackers act independently as well as the case of colluding attackers. We illustrate that in all scenarios, our proposed algorithm has the best performance in presence of collusion attacks.

The remainder of the paper is organized as follows: The related work is presented in Section 2. In Section 3 we describe the attack model and the mathematical framework for position estimation. We explain our proposed algorithm for attack-resistant position discovery in Section 4. Performance of the proposed method with regards to independent and collusion attacks, obtained from computer simulations, are presented in Section 5. In the same section we give a thorough comparison of our algorithm and the state-of-the-art robust position discovery methods.

2 Related Work

Position discovery for sensor networks has been the focus of multiple research efforts during the recent years. Some of the most popular of position estimation algorithms which do not use GPS-like infrastructure are presented in [7, 8, 9]. Most of these algorithms make use of a set of special nodes called beacons which have knowledge of their own position as well as their distance from the other nodes in the network.

Lazos and Poovendran have proposed a range independent positioning technique called SeRLoc and a high resolution variant of it, HiRLoc [10] that is robust against wormhole, Sybil and compromise of beacon attacks. This solution only applies to networks that perform range based positioning with directional antennas and does not apply to networks that perform measurements based on signal features such as RSSI, ToA or TDoA.

Li et al. proposed the use of robust outlier detection statistical models to achieve robust position estimation [11]. The authors proposed a probabilistic approximation to the least median of squares (LMS) approach of [12] to circumvent the computationally intensive nature of LMS. We shall call this variant Robust LMS (RLMS). Liu et al. presented a greedy algorithm to filter out the attackers data on the basis of a consistent minimum mean square error (MMSE) criterion between the received measurements from multiple beacons [13].

Our approach removes the anomalies in a shorter runtime than both the greedy algorithm of [13] and LMS with a better accuracy. For independent attackers, the performance of all three algorithms is comparable with our algorithm having a slight edge. However, when the attackers collude to make stronger attacks, the new approach clearly beats both the greedy algorithm of [13] and LMS. In the context of sensor networks, randomized consensus has been applied to distributed object tracking [14] and time-synchronization [15].

3 Problem Statement

Position estimation methods typically use measurements from a set of reference nodes (beacons) to compute the coordinates of a sensor in the network. Because of limited accuracy of distance estimation techniques, e.g. Time Difference of Arrival (TDoA), the measurements of the beacons are often noisy. Robust position estimation algorithms employ a mathematical framework to estimate the coordinates of the sensor of interest from the erroneous measurements while optimizing an overall error criterion. We shall call this form of position estimation *benign position estimation* as the errors (noise)

in measurements are product of inaccuracies of the system rather than deliberate tampering.

However, in presence of malicious attacks, the false measurements from the compromised beacons can mislead the sensor with unknown coordinates. We shall view the malicious measurements injected to the system by attackers *as outliers*. The solution of the benign position estimation is in general largely affected by malicious attacks that corrupt the sample set.

In absence of cryptographic authentication, a malicious attacker can inject misinformation from the same beacon multiple times. Secure broadcast in sensor networks is a whole other topic of research [16]. In this work, we assume that the sensors are equipped with proper cryptographic authentication capability such that no attacker can broadcast multiple times from the same beacon. Even though, use of cryptographic authentication can limit the number of times an attacker broadcasts malicious information, it does not guarantee the integrity of the data. We allow multiple independent or colluding attackers (beacons) that can transmit corrupted position information without any limit.

3.1 Attack Model

In our attack model, the attackers can modify the position measurements of a beacon without any restrictions. This can be achieved when either the attacker modifies the packets sent by the beacon (spoofing) or captures a beacon and sends out packages containing wrong information. For the purpose of this paper, we do not distinguish between the two cases. However, it is assumed that the network is cryptographically protected against protocol attacks such as wormhole [2] and Sybil [17], and the same attacker cannot corrupt more than one measurement sample. More precisely, the measurements from a malicious beacon are entered only once in the overall data set available at the sensor that performs positioning. The largest percentage of outliers that an statistical outlier rejection method can stand is called the breakpoint of the system. When the attackers act independently, good outlier detection mechanisms can even exceed a 50% breakpoint.

We study a more nefarious form of attack posed by coalitions of attackers coordinating their efforts. When attackers collude they shift the break point of the statistical method below 50%, because there is no way the system can detect more consistency between the uncorrupted samples when there are at least as many corrupted ones. In our experimental results of Section 5, we show that our proposed method has the best performance among all the algorithms in presence of colluding attackers.

3.2 Position Estimation in Presence of Attacks

In presence of attackers (*outliers*), the robust positioning methods seek to construct *good estimates* of the unknown. A *good estimates* is the one that is consistent with benign measurements while it differs from the corrupted measurements according to a given criteria. In our proposed framework, this criteria is a consistency metric δ. The metric δ is selected by the user and it is driven by nature of the attack, i.e., the statistical properties of the attack.

The position estimation problem in presence of malicious attackers is formally stated as follows:

Instance. A node s_0 with unknown coordinates (x_0, y_0); a set L of position information tuples $\{(x_n, y_n, d_n)\}$ corresponding to beacon nodes $\{s_n\}$ where (x_n, y_n) are the coordinates of the n-th beacon s_n and d_n is the measured distances from s_n to s_0 for $n = 1, \ldots, N$; a consistency metric $\delta(s_n, s_0)$; a consistency threshold t.

Problem. Find an estimate for the coordinates of s_0 denoted as $\hat{s_0} = (\hat{x}_0, \hat{y}_0)$, such that it is at least δ-consistent with t points in the set L.

A measurement (x_n, y_n) is δ-consistent with the estimate $\hat{s}_0$ if and only if $\delta(s_n, \hat{s}_0)$ is within a given confidence interval CI. Note that, we shall call the set of δ-consistent points with the estimate $\hat{s_0} = (\hat{x}_0, \hat{y}_0)$, *consensus set* of $\hat{s}_0$.

The parameter t is the size of the consensus set. In Section 4.3 we present two approaches for determining the threshold t. In our implementation of the position estimation algorithm of Section 4, we choose the metric δ as the Euclidean distance. In general, the metric can be selected based on the nature of the problem at hand. Some other appropriate metrics for position estimation can be L_p norms of error or the median.

4 Attack-Resistant Randomized Position estimation Algorithm

Unlike the previous approaches to attack-resistent position estimation [12, 11, 13] which use as much data as possible to estimate the unknown coordinates, our approach starts by picking a small (but sufficient) subset of the data and subsequently augments it with consistent data. The proposed framework randomly selects the initial subset and employs a randomized algorithm to determine the set of consistent measurements and its pseudocode is formally stated in Algorithm 1.

A minimum of 3 distance measurements are needed for finding the coordinate of a node in two dimensions. In light of the new algorithm's minimalist methodology,

Algorithm 1. Randomized Consistent osition estimation

Input: set L, δ-consistency interval CI, threshold t, maximum number of iterations i_{max}.

1. Initialize i=1;
2. **While** $(i < i_{max})$ {
3. Randomly draw a subset Si of size 3 from L;
4. Use Si to estimate the position $\hat{s}_0$;
5. Calculate K, the number of δ-consistent points with respect to the estimate $\hat{s}_0$ in $L \backslash Si$;
6. **If** $(K > t)$ {
7. Form new estimate $\hat{s}_0$ from K consistent points;
8. Terminate the program;}
9. Increment i; }
10. Terminate program either by announcing failure or output the largest consistent estimate;

the approach first estimates the position of the unknown node $\hat{s}_0$ from some randomly selected subset of 3 nodes, Si (Steps $3-4$). To find this position, we use the MMSE approximation algorithm described by Savvides et al. [8]. Next, the algorithm verifies if this estimate $\hat{s}_0$ is *consistent* with enough data points, or equivalently if the size of the consensus set, (parameter K), is large enough, i.e it is larger than the given threshold t (Steps $5-6$). As mentioned earlier in Section 3.2, the consistency is computed with respect to the measure δ. Two methods for determining the threshold t which determines the size of consensus set, are presented in Section 4.3.

Ideally, one would like to test all possible subsets of size 3, i.e. $\binom{N}{3}$ and choose the one with largest consistency set. However, when the data set L is large, this exhaustive approach is too expensive. Instead, we attempt a total of i_{max} times where, the quantity i_{max} is the predetermined total number of the trials. In Section 4.1 we demonstrate how to choose i_{max} such that the algorithm can find a consensus set with high probability.

Once a consistency set has been identified, the algorithm uses all points in the consensus set to form the final estimate of $\hat{s}_0$ and it terminates (Steps 7-8). In our experiments, we use a MMSE procedure for computing both the initial estimate $\hat{s}_0$ from the subset Si and for the final estimate derived from the consensus set.

If the algorithm performs all the i_{max} iterations and does not find a consensus set of at least size t, it either declares a failure or it outputs the MMSE estimate obtained from the largest consensus set that has been found (Step 10). In the remainder of this Section, we will explain the procedure for selection of the inputs to Algorithm 1.

4.1 Choice of Total Number of Iterations

It is expected that the number of iterations i_{max} of Algorithm 1 depends on the percentage of the outliers. Intuitively, the algorithm must keep picking random subsets of data set L for at least expected number of trials $\mathbb{E}[i]$ to find a good subset of size 3. Let N_a denote the number of the outliers in the data set L and let q be the probability that a randomly drawn data point is consistent with the model. The expected number of trials $\mathbb{E}[i] = \frac{1}{q^3}$.

One way of computing i_{max} is to exceed $\mathbb{E}[i]$ by say two standard deviations. It is shown [5] that the standard deviation of $\mathbb{E}[i]$ is approximately equal to $\mathbb{E}[i]$. Therefore, we can choose $i_{max} \approx 3\mathbb{E}[i]$. Another approach for choosing i_{max} is to ensure that the probability of missing a good subset is below a threshold η. This implies that the total number of iterations i_{max} must satisfy: $(1-q^3)^{i_{max}} = \eta$, or equivalently,

$$i_{max} = \frac{\ln \eta}{\ln(1-q^3)} \tag{1}$$

Let I denote the set of inliers. If $\rho = 1 - \frac{N_a}{N}$ is the percentage of inliers, then $q = \frac{\binom{I}{3}}{\binom{N}{3}} = \prod_{j=0}^{2} \frac{I-j}{N-j}$ and for large data set, $q \approx \rho^3$ and $\mathbb{E}[i] = \rho^{-9}$. Substituting for q in (1), gives the number of iterations i_{max} in terms of the percentage of the outliers in the data set L, $i_{max} = \frac{\ln \eta}{\ln(1-\rho^9)}$.

Note that, if an estimate of the number of the attackers N_a is available, then (1) gives the maximum number of subsets Algorithm 1 must try before quitting to search for outlier free subsets of size 3. Section 4.4 presents an approach for determining number of iterations i_{max} when no estimate of the number of outliers is available.

4.2 Determining δ-Consistency Interval

Our assumption is that non-malicious (benign) distance measurement errors are i.i.d Gaussian random variables distributed according to $\mathcal{N}(0, \sigma^2)$. The Gaussian model for measurement errors may not capture all practical cases, but it is a good starting point for our theoretical analysis. More general cases of the distance measurement error distributions are considered in [18]. The consistency metric δ calculates the distance between the real position of s_0 (x_0, y_0) with respect to the position reference (x_n, y_n), i.e. $\delta(s_n, s_0) = d_n - \sqrt{(x_0 - x_n)^2 + (y_0 - y_n)^2}$.

The assumption of Normal distribution for the errors is based on s_0's real position (x_0, y_0). We apply the same distribution to approximate the distribution of δ

which measures the error in estimated position coordinates $\hat{s}_0$=($\hat{x}_0, \hat{y}_0$). For example, under the assumption of Normality, the 95% confidence interval (CI) that a given δ'_n is drawn from the Normal distribution $\mathcal{N}(0, \sigma^2)$, is [-1.96 σ, 1.96 σ]. In other words, given δ'_n a realization of the random variable δ_n, the confidence interval CI determines wether δ'_n is drawn from the same distribution as δ_n, with more than 95% probability. We refer to the confidence interval as the δ-consistency interval CI which is an input to the Position Estimation Algorithm 1. Note that, the error variance σ^2 is usually dependent on the distance measurement technique (e.g., RSSI, TDoA) and the environment where the sensors are deployed. Therefore, the error variance can be estimated via a set of offline measurements.

4.3 Consensus with Respect to t

The randomized paradigm presented in Algorithm 1 assumes that t, the estimate of the size of the consensus set is given. If the number of attackers (N_a) are known a priori, one could set the size of the consensus set equal to $N - N_a$. However, in many practical scenarios one would need to estimate t. Our first method for estimating t is to employ the threshold selection strategy proposed by Liu et al. [13]. The major difference is that unlike [13], we do not rely on the approximate distribution of the error metric and we derive its exact distribution.

We use mean square error (MSE) of the distance measurements as our error metric. The selection of MSE metric is driven by its asymptotic optimality in presence the Gaussian noise, and its pervasive usage in position estimation literature in WSNs [13, 8]. MSE also facilitate sound formal analysis and creation of effective algorithms. The position estimation procedure in Algorithm 1. is also minimizing the MSE of $\hat{s}_0$ with respect to the beacons that participate in the position estimation procedure. MSE of the estimated position $\hat{s}_0$ is denoted as Δ^2_{MSE}:

$$\Delta^2_{MSE} = \sum_{n=1}^{N} \frac{(d_n - \sqrt{(\hat{x}_0 - x_n)^2 + (\hat{y}_0 - y_n)^2})^2}{N}$$

To ensure that the final consensus set includes only the consistent points within a statistical error interval, we compute the Q-th quantile of the inverse probability of Δ^2_{MSE} for K points. This quantile corresponds to a τ, such that Prob($\Delta^2_{MSE} \leq \tau$) $= Q$. We derive the distribution of Δ^2_{MSE} to find the value of τ. Note that, since the quantile distribution function is monotonic (non-decreasing), there is a monotonic transformation between t and τ. Under the assumption that the measurement errors are distributed i.i.d Gaussian with $\mathcal{N}(0, \sigma^2)$ for $n = 1, \ldots, K$, the distribution of the mean square error Δ^2_{MSE} is Chi-square (χ^2_K) with K degrees of freedom (DF). The probability distribution of Δ^2_{MSE} is based on s_0's real positions (x_0, y_0); this distribution is used to approximate the cases where we have an estimate $(\hat{x}_0, \hat{y}_0)$ for s_0's position. Note that, under the assumption of Uniform distribution of errors, Liu et al. [13] have previously used the central limit theorem to approximate the distribution of Δ^2_{MSE} as a Gaussian. Our approach is different and it improves their results in at least two ways: First, we believe that the assumption of Gaussian distribution of errors is more realistic than the Uniform distribution; it is much more likely than smaller values of error occur more frequently than the larger ones. Second, as the landmark results by Pearson and Fisher [19] have indicated, the chi-square and F- distribution are the exact models for smaller values of N rather than the Gaussian distribution. Since $\Delta^2_{MSE} \sim \chi^2_K$, then as K tends to infinity, the distribution of Δ^2_{MSE} tends to normality but the tendency is very slow [20]. To determine the threshold, we employ the same method as [13], except that we use χ^2_K instead of the Gaussian distribution.

4.4 Dynamic determination of the number of outliers N_a

Another method for determining the threshold t is to perform a search on the number of attackers N_a. Based on the value of N_a, we can compute the maximum number of iterations i_{max}, and the threshold t. We perform a binary search on the value of N_a between 0 and N to determine the best N_a such that the randomized consistent position estimation procedure produces the largest consensus set. Every time one alters the value of N_a in the binary search procedure, it computes a new value for i_{max} via the Equation (1) and a new t. The runtime of the randomized position estimation algorithm will be only increased by a $log(N_a)$ value that is a very small number compared to N and is essentially a constant for all practical implementation purposes.

5 Experimental Evaluations

In all of our simulations, a total of N beacons were uniformly randomly distributed in a $10m \times 10m$ square area. The square is centered at the origin. We assume that the real coordinate (x_0, y_0) of the node s_0 (the target node for position estimation) is at the origin. The distance measurement noise between each beacon and s_0 is Gaussian with variance $\sigma = 0.2$. Note that, even though we use the Gaussian distribution to adaptively compute the consistency threshold in Subsection 4.3, the procedure presented in Algorithm 1 in independent of

the noise distribution and for a certain value of parameter t, the algorithm is applicable to any distribution. Our choice of Gaussian in simulation is only driven by the wide usage of this model.

We study the trade-off between noisy and inconsistent measurement by fixing the noise variance and changing the amplitude of the attacks. Typically, the noise variance depends on the signal feature used in distance estimation, deployment environment and sensor specification. Therefore, once the components and protocols of a network are configured, the value of the σ is expected to remain unchanged. The confidence interval (see Subsection 3.2) of all the runs of our algorithm is chosen to be $CI = 90\%$. We also performed experiments over the CI of the model for different intervals of confidence, at 80%, 85%, 90%, and 95% and decided that CI=90% (0.9) was the best, since it would accept the noisy (but benign) distance estimate, while it is tight enough to diagnose the attackers. Note that, to minimize the randomness effects, each presented result is smoothed over a 1000 runs of the corresponding algorithm.

When the attackers act independently, N_a points are randomly selected among the N beacons in box. The amplitude of the attack (A_a) is the percentage increase (decrease) in the noisy distance; for example, a malicious beacon at distance d_i would report its distance to be $d_i(1 \pm A_a)$. In case of the colluding attackers, again we randomly pick total of N_a beacons from the box and subsequently perturb their measured distances such that they shift the node s_0 from the origin to a false position (x_f, y_f). In both regimes, we corrupt the samples in L with an i.i.d. zero mean Gaussian noise with power σ^2.

5.0.1 Performance of the New Algorithm

We shall first study the performance of the new approach for independent attackers and then compare it with the case where the attackers collude.

(a) Independent Attacks. In our first set of experiments, we alter percentage of attackers $\frac{N_a}{N}\%$, and amplitude of the attack A_a, for a fixed N. We study the percentage of *false negatives (FN)* and the error in the position estimate $|\hat{s}_0 - s_0|$ as a function of $\frac{N_a}{N}\%$ and A_a. A false negative occurs when the algorithm fails to diagnose and isolate an attacker. Given that the algorithm is designed for identifying the attackers, FN is the relevant figure of merit for evaluating the algorithm. The final error in the estimated position with respect to the real position is used as a measure of success for the position estimation.

We have also studied the percentage of *false positives (FPs)*, which was low for all cases. In the case of independent attackers, even with a large number of FPs, as long as the number of inliers is higher than three (the minimum required inliers for position estimation), one can estimate a consistent final position with an error. The final error is lessened by increasing the number of inliers up to seven, but beyond that the estimation error stays constant [18]. Because of space considerations, we do not report FP values here, since we often have more beacons than the minimum of three, and thus, it does not have a significant effect on the estimated position's error.

$\frac{N_a}{N}\%$	$\lvert\hat{s}_0 - s_0\rvert$	$\overline{FN}(\%)$
10%	0.06	0
20%	0.07	1.1
30%	0.07	2.4
40%	0.11	3.4
50%	0.13	3.7

(a)

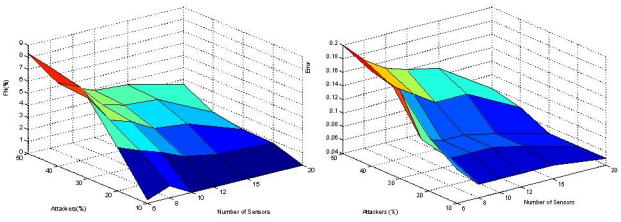
A_a	$\overline{FN}(\%)$
20%	4
30%	2
40%	0.6
50%	0.2
60%	0

(b)

Table 1. Position estimation error ($|s_0 - \hat{s}_0|$), average FN vs. percentage of attackers, $\frac{N_a}{N}\%$(left table). Average FN vs. amplitude of attack A_a (right table).

In Table 1, we show two sub-tables. The number of nodes is fixed to 15 here. Sub-table (a) is presenting the position estimation error, and average percentage of FN versus various percentage of attackers ($\frac{N_a}{N}\%$), when the amplitude of the attack is fixed at $A_a = 30\%$. Sub-table (b), shows FN when the malicious beacon i increases (decreases) its distance d_i by (A_a) percent. In this experiment, $N = 15$, and $N_a = 5$ (i.e., $\frac{N_a}{N}$=33%).
The results of Sub-table (a) shows that both the amplitude of the estimation error and FN increase as the number of the attackers grow. Note that, when the amplitude of the attacks increases, the performance of the algorithm improves. This is due to the fact that larger outliers are easier to distinguish from noise. For instance, for outliers that are 60% larger or smaller than the benign measurements, the algorithm diagnoses and isolates all the attackers.

Next, we study the performance of the new algorithm vs. number of beacons N and percentage of attackers $\frac{N_a}{N}\%$. The results are presented in Figure 1, where Plot (a) illustrates the percentage of FNs on the z-axis and Plot (b) presents the absolute error in estimated position. As we can see in Plot (a), the percentage of FNs is almost exponentially growing with increasing the percentage of attackers for a small number of nodes. This is because the number of benign beacons reaches the minimum of three, and there are no redundancies in the system to check the validity of an estimate of three nodes in presence of noise. A similar effect can be traced in the absolute error value depicted in Plot (b), where the errors are increasingly growing for low number of nodes



(a) FN (b) Estimation error

Figure 1. (a) Percentage of FNs (z-axis), and (b) Position estimation error (z-axis) vs. percentage of independent attackers and number of nodes (y-axis).

and high percentage of attackers. As can be seen on both plots, increasing the number of beacons beyond a certain number (around 10), ensures that there is a low percentage of FNs as well as a low absolute position estimation error for up to about 25% attacking beacons.

Figure 2 depicts the boxplot of the error in the estimated position ($|s_0 - \hat{s}_0|$) for various amplitudes of independent attacks A_a, $N = 15$, and $N_a = 5$ (33%). Each box in the boxplot presents the 25% percentile and the 75% of the error distribution. The line within the box shows the position of the median. As we can see here, the bound on error becomes tighter as we increase A_a, while the median is decreasing and then stabilizing around $A_a = 40\%$. Again, for both large or small amplitude of outliers, the algorithm succeeds in estimating the coordinate of the unknown node up to an error margin due to the noisy measurements.

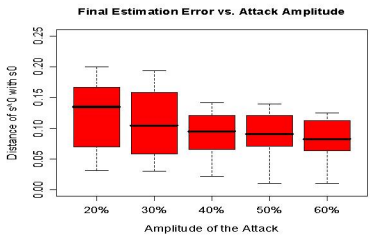
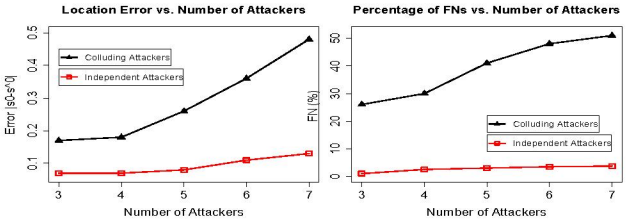


Figure 2. Final estimation error vs. attack amplitude, A_a.

We have also evaluated the performance of the new algorithm in cases where the number of attackers N_a is not available. The binary search procedure described in Subsection 4.4 is used in this case. The binary search procedure does not have a big effect on the false negatives and the percentage of FNs essentially remains the same. However, the binary search often incurs a lot of FPs, since it tends to remove the benign measurements with a large noise. The position estimation error is slightly improved, since only the more consistent measurements are used in determining the final position.

(b) Collusion Attacks. The attack resilient paradigm of



(a) Estimation error (b)FN

Figure 3. (a) Position estimation error vs. number of attackers, N_a, and (b) FN vs. number of attackers, N_a.

Section 4 relies on the fact that the benign measurements, even the noisy ones, are consistent with the other measurements. When the attackers act separately, the outliers are statistically independent. Therefore, as long as they do not overwhelmingly (more than 70%) dominate the measurement set, Algorithm 1 succeeds in finding a good estimate. However, when the attackers collude, we expect that the system can not tolerate more than 50% malicious beacons. Our simulation result indeed prove this.

In the remainder of the experiments in this section, unless otherwise stated, the number of nodes is N=15, the amplitude of independent attacks is $A_a = 30\%$ and the displacement of colluding attacks is $(x_f, y_f) = (1, 0)$. Recall that (x_0, y_0)=(0,0) and the area is 10×10.

Figure 3 (a) compares the performance of our algorithm in terms of position error estimate $|\hat{s}_0 - s_0|$ for colluding and independent attackers. Figure 3 (b) depicts performance of the algorithm in terms of FN, in presence of colluding and independent attacks. It shows that when almost half of the attackers (seven) collude, the algorithm misses 50% of the colluders, while for the same number of independent attackers, only a small fraction of attackers are missed.

5.0.2 Comparison with Other Attack-Resistent Algorithms

We compare the performance of the new algorithm with three state-of-the-art position estimation methods: LMS algorithm of Rousseeuw and Leroy [12], RLMS algorithm of [11] and Greedy MMSE algorithm of [13]. Herein, we refer to the new randomized position estimation algorithm as RANLD. The first method, LMS, solves the position estimation problem by minimizing an least median of squares (LMS) error metric over all the nodes. It is indeed an application of the general paradigm of least median of squares optimization to sensor network position estimation. The second method, RLMS, is a probabilistic approximation of first method [11]. The third method, Greedy MMSE, is a greedy algorithm that minimizes the MSE error metric subject

to a consistency threshold τ'. Instead of exhaustively searching all combinations of the variables for the best estimate, the authors propose using a stepwise backward deletion algorithm, a greedy algorithm that at each stage, deletes the largest outlier. This method works well when there are only a few independent attackers, but fails as the number of attackers increase or the attackers collude.

We analyze the performance of the various methods over a range of different parameters, including number of nodes, number of attackers, amplitude of attacks, and displacement of colluding attacks. Given that the goal of attack-resilient position estimation is to find the coordinates of an unknown node while diagnosing and removing the attackers, the position estimation error and FN are the most relevant criteria for comparison purposes, both for independent and collusion attacks.

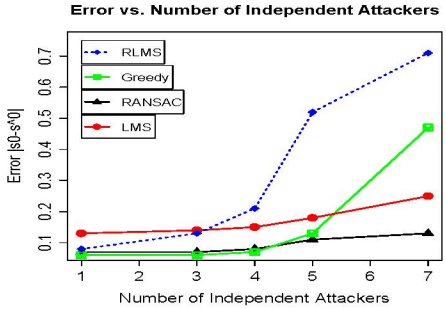


Figure 4. Position estimation error $|\hat{s}_0 - s_0|$ vs. number of N_a independent attackers.

(a) Independent Attacks. Figure 4 depicts the position estimation error ($|\hat{s}_0 - s_0|$) versus the number of independent attackers. As long as the percentage of the attackers does not exceed 33%, i.e. $N_a \leq 5$ for N=15, the performance of the Greedy MMSE is comparable to RANLD. For higher percentage of attackers, the Greedy MMSE algorithm is not as robust as RANLD. The LMS position estimation algorithm almost always has a higher error than the new method, but it has a high break point and does not break for the number of outliers less than 50%. The performance of RLMS is good for small number of attackers or in the absence of coalition attacks, but deteriorates rapidly with increasing N_a. We believe that we might not have a good implementation of this algorithm, due to the fact that the authors do not specify how they select the critical value γ [11]. We tried γ =2.5 as was suggested by [21] but the results were not satisfactory. Also, we tried to optimize for γ experimentally, but we could not find aγ that worked well across all experiments. Thus, we excluded RLMS from the analysis of collusion attacks.

Table 2, illustrates the percentage of FNs for various number of attackers (N_a). The first column shows the number of attackers. The next three columns illustrate

Num of Attackers	Independent Attacks			Colluding Attacks		
	Greedy	LMS	RANLD	Greedy	LMS	RANLD
3	1.3	1.0	1.1	32.6	26.4	27.5
4	2.6	1.9	2.2	36.0	29.0	30.4
5	2.8	2.8	2.5	42.3	43.8	41.8
6	3.6	5.8	3.4	55.7	72.0	46.8
7	4.0	7.6	3.6	61.9	77.2	52.3

Table 2. FNs (in percentage) of the three methods vs. number of attackers, for both independent and colluding attacks.

the percentage of FNs for the three algorithms, Greedy MMSE, LMS, and RANLD, under the assumption of independent attacks. The last three columns illustrate the percentage of FNs for the three algorithms under the assumption of colluding attacks. For illustration purposes, the columns illustrating RANLD algorithm are shaded in grey. As can be seen from the table, for independent attackers, the percentage of FNs is comparable for Greedy MMSE and RANLD, where RANLD incurs slightly less number of FNs. The FNs performance of LMS increasingly deteriorates with increasing number of independent attackers.

(b) Collusion Attacks. As can be seen from the last three columns of Table 2, the percentage of FNs for colluding attacks rapidly increases for all three algorithms, with RANLD having the smallest increase rate, when compared to LMS and the Greedy MMSE. While the percentage of FNs of the LMS algorithm is comparable to RANLD for smaller percentage of attackers, this value rapidly increases for more than 6 (40%) attackers. The Greedy MMSE has a worse FNs performance compared with LMS for smaller number of attackers, but the percentage of FNs of Greedy MMSE grows slower than that of LMS.

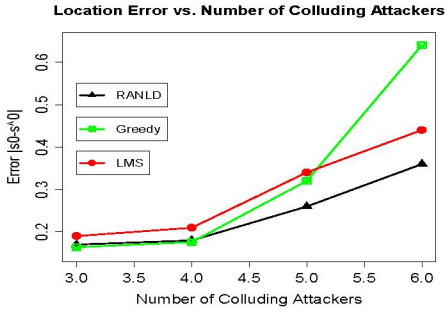


Figure 5. Position error estimate $|\hat{s}_0 - s_0|$ vs. number of N_a colluding attackers.

Figure 5, compares the position estimation error for the three algorithms: RANLD, Greedy MMSE and LMS versus number of colluding attacker N_a. Comparing Figure 4, where the attackers act independently, with Figure 5, we can see that all algorithms suffer from col-

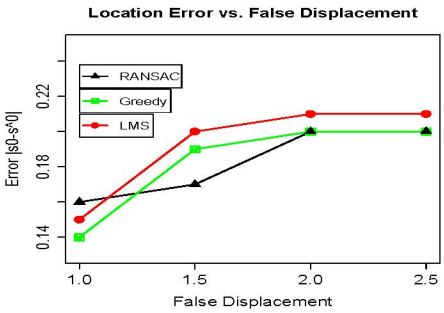


Figure 6. Position error estimate $|\hat{s}_0 - s_0|$ vs. collusion attack displacements.

lusion. For both independent and colluding attackers, the Greedy MMSE demonstrates less tolerance against an increase in the number of the attackers, while LMS and RANLD are more robust. Note that, RANLD has a lower error in estimating the final position. This might be due to the fact that RANLD uses MMSE estimation which is a more efficient estimate than median in its final position estimation.

Finally, Figure 6 plots the position estimation error for different values of false displacements that are injected to the system by colluders. In this experiment, the number of colluders was set to four. All three algorithms seem to perform more or less similarly with respect to changes in magnitude of the displacement.

References

[1] K.S. Killourhy, R.A. Maxion, and K.M.C.Tan, "A defense-centric taxonomy based on attack manifestations," in *Dependable Systems and Networks (DSN)*, 2004, pp. 102–111.

[2] Y. Hu, A. Perrig, and D. Johnson, "Packet leashes: A defense against wormhole attacks in wireless networks," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, 2003, pp. 1976–1986.

[3] G. Stuber and J. C. jr., *The Mobile Communications Handbook, J.D. Gibson and E.M. Gibson ed.*, ch. 24, *Radiolocation Techniques*, CRC Press, 1999.

[4] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, 2000, pp. 775–784.

[5] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[6] R. Motwani and P. Raghavan, *Randomized Algorithms*. New York, NY: Cambridge University Press, 1995.

[7] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2000, pp. 32–43.

[8] A. Savvides, C. Han, and M. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of AC International Conference on Mobile Computing and Networking (MobiCom)*, 2001, pp. 166–179.

[9] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AoA," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, 2003, pp. 1734 – 1743.

[10] L. Lazos and R. Poovendran, "Hirloc: High resolution localization for wireless sensornetworks," *IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Network Security*, vol. 24, no. 2, pp. 233– 246, April 2006.

[11] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks." in *Proceedings of The International Symposium on Information Processing in Sensor Networks (IPSN)*, 2005, pp. 91–98.

[12] P. Rousseeuw and A. Leroy, *Robust Regression & Outlier Detection*. New York, NY: John Wiley & Sons, 1987.

[13] D. Liu, P. Ning, and W. Du, "Attack-resistant location estimation in sensor networks," in *Proceedings of International Symposium on Information Processing in Sensor Networks (IPSN)*, 2005, pp. 99– 106.

[14] T. Roosta, M. Meingast, and S. Sastry, "Distributed reputation system for tracking applications in sensor networks," in *International Workshop on Advances in Sensor Networks (IWASN)*, 2006.

[15] M. Manzo, T. Roosta, and S. Sastry, "Time synchronization attacks in sensor networks," in *Proceedings of the ACM workshop on Security of ad hoc and sensor networks (SASN)*, 2005, pp. 107–116.

[16] A. Perrig, R. Szewczyk, V. Wen, D. Cullar, and J. Tygar, "Spins: Security protocols for sensor networks," in *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2001, pp. 189–199.

[17] J. Douceur, "The sybil attack," in *Proceedings of the International Workshop on Peer-To-Peer Systems Workshop (IPTPS)*, 2002, pp. 251–260.

[18] J. Feng, L. Girod, and M. Potkonjak, "Consistency-based on-line localization in sensor networks," in *Proceedings of International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2006, pp. 529–545.

[19] M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*. New York, NY: John Wiley and Sons, 2000.

[20] E. Wilson and M. Hilferty, "The distribution of chi-square," *in Proceedings of National Academy Science, USA*, vol. 17, pp. 684–686, 1931.

[21] P. Meer, D. Mintz, A. Rosenfeld, and D. Yoon-Kim, "Robust regression methods for computer vision: a review," *International Journal of Computer Vision*, vol. 6, no. 1, pp. 59–70, 1991.