Robust Statistical Methods for Securing Wireless Localization in Sensor Networks

Zang Li, Wade Trappe, Yanyong Zhang, Badri Nath

***Abstract*— Many sensor applications are being developed that require the location of wireless devices, and localization schemes have been developed to meet this need. However, as location-based services become more prevalent, the localization infrastructure will become the target of malicious attacks. These attacks will not be conventional security threats, but rather threats that adversely affect the ability of localization schemes to provide trustworthy location information. This paper identifies a list of attacks that are unique to localization algorithms. Since these attacks are diverse in nature, and there may be many unforseen attacks that can bypass traditional security countermeasures, it is desirable to alter the underlying localization algorithms to be robust to intentionally corrupted measurements. In this paper, we develop robust statistical methods to make localization attack-tolerant. We examine two broad classes of localization: triangulation and RF-based fingerprinting methods. For triangulation-based localization, we propose an adaptive least squares and least median squares position estimator that has the computational advantages of least squares in the absence of attacks and is capable of switching to a robust mode when being attacked. We introduce robustness to fingerprinting localization through the use of a median-based distance metric. Finally, we evaluate our robust localization schemes under different threat conditions.**

I. Introduction

The infrastructure provided by wireless networks promises to have a significant impact on the way computing is performed. Not only will information be available while we are on the go, but new location-aware computing paradigms along with location-sensitive security policies will emerge. Already, many techniques have emerged to provide the ability to localize a communicating device [1–5].

Enforcement of location-aware security policies (e.g., this laptop should not be taken out of this building, or this file should not be opened outside of a secure room) requires trusted location information. As more of these location-dependent services get deployed, the very mechanisms that provide location information will become the target of misuse and attacks. In particular, the location infrastructure will be subjected to many *localization-specific* threats that cannot be addressed through traditional security services. Therefore, as we move forward with deploying wireless systems that support location services, it is prudent to integrate appropriate mechanisms that protect localization techniques from these new forms of attack.

The purpose of this paper is to examine the problem of secure localization from a viewpoint different from traditional network security services. In addition to identifying different attacks and misuse faced by wireless localization mechanisms, we take the viewpoint that these vulnerabilities can be mitigated by exploiting the redundancy present in typical wireless deployments. Rather than introducing countermeasures for every possible attack, our approach is to provide *localization-specific*, *attack-tolerant* mechanisms that shield the localization infrastructure from threats that bypass traditional security defenses. The idea is to live with bad nodes rather than eliminate all possible bad nodes.

The authors may be reached at Wireless Information Network Laboratory, Rutgers University, 73 Brett Rd., Piscataway, NJ 08854 or by email at {zang, trappe, yyzhang}@winlab.rutgers.edu, and badri@cs.rutgers.edu.

We begin in Section II by presenting an overview of several techniques used in wireless localization, as well as discuss efforts that have been made to provide security to localization. Following the review, we explore localization-specific attacks that can be mounted against wireless localization services in Section III. To address these attacks, we propose the use of robust statistical methods. In Section V and Section VI we focus our discussion on applying robust mechanisms to two broad classes of localization: triangulation and fingerprinting methods. We introduce the notion of coordinated adversarial attacks on the location infrastructure, and present a strategy for launching a coordinated attack on triangulation-based methods. For triangulation-based localization, we propose the use of least median squares (LMS) as an improvement over least squares (LS) for achieving robustness to attacks. We formulate a linearization of the least squares location estimator in order to reduce the computational complexity of LMS. Since LS outperforms LMS in the absence of aggressive attacks, we devise an online algorithm that can adaptively switch between LS and LMS to ensure that our localization algorithm operates in a desirable regime in the presence of varying adversarial threats. For fingerprinting-based location estimation, we show that the use of traditional Euclidean distance metrics is not robust to intentional attacks launched against the base stations involved in localization. We propose a median-based nearest neighbor scheme that employs a median-based distance metric that is robust to location attacks. The use of median does not require additional computational resources, and in the absence of attacks has performance comparable to existing techniques. Finally, we present conclusions in Section VII.

II. Related Work

Broadly speaking, there are two main categories of localization techniques: those that involve range estimation, and those that do not [1]. Range-based localization algorithms involve measuring physical properties that can be used to calculate the distance between a sensor node and an anchor point whose location is known. Time of Arrival (TOA) is an important property that can be used to measure range, and arises in GPS [6]. The Time Difference of Arrival (TdOA) is also widely used, and has been used in MIT's Cricket [2], and appeared in [7, 8]. In addition, APS [3] pointed out that the Angle of Arrival (AOA) can be used to calculate the relative angle between two nodes, which can be further used to calculate the distance. The RSSI value of the received signal, together with the signal propagation model, is also a good indicator of the distance between two nodes [9, 10]. Other properties of arriving signals can also be exploited. One interesting example is to use visual cuing [11], which tries to determine the position and orientation of a mobile robot from visual cues obtained from color cylinders strategically placed in the field of the view.

Range-free localization algorithms do not require the measurement of physical distance-related properties. For example, one can count the number of hops between a sensor node and an anchor point, and further convert the hop counts to physical distances, such as in [12–14]. As another example, a sensor node can estimate its location using the centroid of those anchor nodes that are within its radio range, such as in Centroid [15]. Similarly, APIT [16] employs an area-based estimation scheme to determine a node's location. Compared to range-based localization algorithms, these schemes do not require special hardware, and their accuracies are thus lower as well.

Secure localization has received attention only recently. In [4], the authors listed a few attacks that might affect the correctness of localization algorithms along with a few countermeasures. One technique that may be used to defend against wormhole attacks is to employ packet leashes [17]. SecRLoc [5] employs a sectored antenna, and presented an algorithm that makes use of the property that two sensor nodes that can hear from each other must be within the distance $2r$ assuming r is fixed in order to defend against attacks. A different approach to securing location information was presented in [18], where the concept of location verification was introduced. Compared to these studies, our paper takes a distinct approach that *we should learn how to live with bad guys rather than defeating each type of attack*. In addition, we also identify a more complete list of attacks that are *unique* to localization algorithms.

III. Attacks Unique to Localization

Different localization methods are built upon the measurement of some basic properties. In Table I, we enumerate several properties that are used by localization algorithms, along with different threats that may be employed against these properties. The threats we identify are specific to localization, and are primarily *non-cryptographic* attacks that are directed against the measurement process. Consequently, these attacks bypass conventional security services.

We note, however, that there are many classical security threats that may be launched against a wireless or sensor network, which can have an adverse affect on the localization process. For example, a Sybil attack can disrupt localization services by allowing a device to claim multiple identities. In order to address the Sybil attack, one may employ entity verification techniques, such as radio resource testing or the verification of key sets, which were presented in [19]. In general, for attacks that are cryptographic in nature, there are extensive efforts to migrate traditional security services, such as authentication, to the sensor platform in order to handle these threats.

Even so, though, it should be realized that it is unlikely that any single technique will remove all possible threat models and, in spite of the security countermeasures that are employed, many adversarial attacks will be able to bypass security layers. To address threats that are *non-cryptographic*, or threats that bypass conventional security countermeasures, we take the viewpoint that statistical robustness needs to be introduced into the wireless localization process.

We now explore several of these threats. We start by looking at methods that employ time of flight. The basic concept behind time of flight methods is that there is a direct relationship between the distance between two points, the propagation speed, and the duration needed for a signal to propagate between these two points. For time of flight methods, an attacker may try to bias the estimation of distance to a larger value by forcing the observed signal to come from a multipath. This may be accomplished by placing a barrier sufficiently close to the transmitter

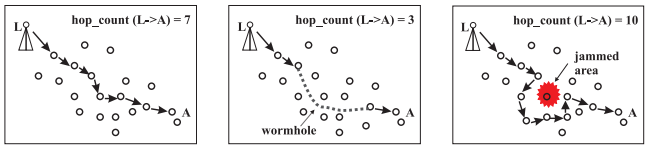


Fig. 1. (Left) Operation of localization using hop count, (Middle) Wormhole attack on hop count methods, and (Right) Jamming attack on hop count methods.

and effectively removing the line-of-sight signal. Another technique that may be used to falsely increase the distance estimate occurs in techniques employing round-trip time of flight. Here, an adversarial target that does not wish to be located by the network receives a transmission and holds it for a short time before retransmitting. An attack that skews the distances to smaller values can be accomplished by exploiting the propagation speed of different media. For example, in CRICKET [2], the combination of an RF signal and an ultrasound signal allows for the estimation of distance since the acoustic signal travels at a slower propagation velocity. An adversary located near the target may therefore hear the RF signal and then transmit an ultrasound signal that would arrive before the original ultrasound signal can reach the receiver [4].

As another example, consider a location system that uses signal strength as the basis for location. Such a system is very closely tied to the underlying physical-layer path loss model that is employed (such as a free space model where signal strength decays in inverse proportion to the square of distance). In order to attack such a system, an adversary could introduce an absorbing barrier between the transmitter and the target, changing the underlying propagation physics. As the signal propagates through the barrier, it is attenuated, and hence the target would observe a significantly lower received signal strength. Consequently, the receiver would conclude that it is further from the transmitter than it actually is.

Hop count based localization schemes [13] usually consist of two phases. In the first phase, per-hop distance is measured. In the second phase, anchor points flood beacons to individual sensor nodes, which count the number of hops between them, and these hop counts are translated into physical distances. As a result, adversaries can initiate attacks as follows: (1) manipulate the hop count measurement, and (2) manipulate the translation from hop count to physical distance. A number of tricks can be played to tweak hop count measurements, ranging from PHY-layer attacks, such as increasing/decreasing transmission power, to network layer attacks that tamper with the routing path. Since PHY-layer attacks have been discussed earlier, we now focus on some possible network layer attacks, namely jamming [20] and wormholes [17]. By jamming a certain area between two nodes, beacons may take a longer route to reach the other end (as shown in Figure 1), which increases the measured hop count. While jamming may not always increase the hop count, for it may not block the shortest path between the two nodes, the other type of attacks, which involve wormhole links, are more harmful because they can often significantly shorten the shortest path and result in a much smaller hop count. Figure 1 illustrates such a scenario: the shortest path between anchor L and node A has 7 hops, while the illustrated wormhole brings the hop count down to 3. Consequently, these attacks can also affect the translation from hop count to physical distance. In addition, if adversaries can manage to physically remove or displace some sensor nodes, even correct hop counts are not useful for obtaining accurate location calculations.

Property	Example Algorithms	Attack Threats
Time of Flight	Cricket	Remove direct path and force radio transmission to employ a multipath; Delay transmission of a response message; Exploit difference in propagation speeds (speedup attack, transmission through a different medium).
Signal Strength	RADAR, SpotON, Nibble	Remove direct path and force radio transmission to employ a multipath; Introduce different microwave or acoustic propagation loss model; Transmit at a different power than specified by protocol; Locally elevate ambient channel noise.
Angle of Arrival	APS	Remove direct path and force radio transmission to employ a multipath; Change the signal arrival angel by using reflective objects, e.g., mirrors; Alter clockwise/counter-clockwise orientation of receiver (up-down attack).
Region Inclusion	APIT, SerLoc	Enlarge neighborhood by wormholes; Manipulate the one-hop distance measurements; Alter neighborhood by jamming along certain directions.
Hop Count	DV-Hop	Shorten the routing path between two nodes through wormholes; Lengthen the routing path between two nodes by jamming; Alter the hop count by manipulating the radio range; Vary per-hop distance by physically removing/displacing nodes.
Neighbor Location	Centroid Method, SerLoc	Shrink radio region (jamming); Enlarge radio region (transmit at higher power, wormhole); Replay; Modify the message; Physically move locators; Change antenna receive pattern.

TABLE I

PROPERTIES EMPLOYED BY DIFFERENT LOCALIZATION ALGORITHMS AND ATTACKS THAT MAY BE LAUNCHED AGAINST THESE PROPERTIES.

Localization methods that use neighbor locations are built upon the implicit assumption that neighbors are uniformly distributed in space around the wireless device. These localization methods, such as the Centroid method, can be attacked by altering the shape of the received radio region. For example, an attacker can shrink the effective radio region through blocking some neighbors by introducing a strong absorbing barrier around several neighbors. Another approach to shrinking the radio region is for an adversary to employ a set of strategically located jammers. Since these neighbors are not heard by the wireless device, the location estimate will be biased toward the unblocked side.

IV. ROBUST LOCALIZATION: LIVING WITH BAD GUYS

As discussed in the previous section, wireless networks are susceptible to numerous localization-specific attacks. These attacks will be mounted by clever adversaries, and as a result will behave dramatically different from measurement anomalies that arise due to the underlying wireless medium. For example, signal strength measurements may be significantly altered by opening doorways in a hallway, or by the presence of passersby. Although these errors are severe, and can degrade the performance of a localization scheme, they are not intentional, and therefore not likely to provide a persistent bias to any specific localization scheme. However, the attacks mentioned in Section III will be intelligent and coordinated, causing significant bias to the localization results.

Solutions that can combat some of these localization attacks have been proposed, often involving conventional security techniques [4, 5]. However, as noted earlier, it is unlikely that conventional security will be able to remove all threats to wireless localization. We therefore take the viewpoint that instead of coming up with solutions for each attack, it is essential to achieve robustness to unforeseen and non-filterable attacks. Particularly, localization must function properly even in the presence of these attacks.

Our strategy to accomplish this is to take advantage of the redundancy in the deployment of the localization infrastructure to provide stability to contaminated measurements. In particular, we develop statistical tools that may be used to make localization techniques robust to adversarial data. As a byproduct, our techniques will be robust to non-adversarial corruption of measurement data. For the purpose of the discussion, we shall focus our attention on two classes of localization schemes: triangulation, and the method of RF fingerprinting. We have chosen these two methods since they represent a broad survey of the methods used. Our discussion and evaluations will focus on the case where we localize a single device. Localizing multiple nodes involves applying the proposed techniques for each device that is to be localized.

The methods we will propose here make use of the median. Median-based approaches for data aggregation in sensor networks have recently been proposed [21, 22], and use the median as a resilient estimate of the average of aggregated data. On the other hand, localizing a device involves estimating a device's position from physical measurements not directly related to position, such as signal strength. Applying robust techniques to wireless sensor localization is challenging as it involves not only integrating robust statistical methods that estimate position from other types of measurements, but also must consider important issues such as computational overhead.

V. ROBUST METHODS FOR TRIANGULATION

Triangulation methods constitute a large class of localization algorithms that exploit some measurement to estimate distances to anchors, and from these distances an optimization procedure is used to determine the optimal position. The robust methods that we describe can be easily extended to other localization techniques, such as the Centroid method.

Triangulation methods involve gathering a collection of $\{(x, y, d)\}$ values, where d represents an estimated distance from the wireless device to an anchor at (x, y). These distances d may be stem from different types of measurements, such as hop counts in multi-hop networks (as in the case of DV-hop [13]), time of flight (as in the case of CRICKET), or signal strength. For example, in a hop-based scheme like DV-hop, following the flooding of beacons by anchor nodes, hop counts are measured between anchor points and the wireless device, which are then transformed into distance estimates.

In the ideal case, where the distances are not subjected to any measurement noise, these $\{(x, y, d)\}$ values map out a parabolic surface

$$d^2(x, y) = (x - x_0)^2 + (y - y_0)^2, \quad (1)$$

whose minimum value (x_0, y_0) is the wireless device location. Gathering several $\{(x_j, y_j, d_j)\}$ values and solving for (x_0, y_0) is a simple least squares problem that accounts for overdetermination of the system and the presence of measurement noise.

However, such an approach is not suitable in the presence of malicious perturbations to the $\{(x, y, d)\}$ values. For example, if an adversary alters the hop count, perhaps through a wormhole attack or jamming attack, the altered hop count may result in significant deviation of the distance measurement d

from its true value. The use of a single, significantly incorrect $\{(x, y, d)\}$ value will drive the location estimate significantly away from the true location in spite of the presence of other, correct $\{(x, y, d)\}$ values. This exposes the vulnerability of least squares localization method to attacks, and we would like to find a robust alternative, as discussed below, to reduce the impact of attacks on localization.

A. Robust Fitting: Least Median of Squares

The vulnerability of the least squares algorithm to attacks is essentially due to its non-robustness to "outliers". The general formulation for the LS algorithm minimizes the cost function

$$J(\theta) = \sum_{i=1}^{N} [u_i - f(v_i, \theta)]^2, \tag{2}$$

where θ is the parameter to be estimated, u_i corresponds to the i-th measured data sample, v_i corresponds to the absissas for the parameterized surface $f(v_i, \theta)$, $|y_i - f(x_i, \theta)|$ is the residue for the i-th sample, and N is the total number of samples. Due to the summation in the cost function, a single influential outlier may ruin the estimation.

To increase robustness to outliers, a robust cost function is needed. For example, the method of least median of squares, introduced by Rousseeuw and described in detail in [23], is one of the most commonly used robust fitting algorithms. Instead of minimizing the summation of the residue squares, LMS fitting minimizes the median of the residue squares

$$J(\theta) = \mathrm{med}_i [y_i - f(x_i, \theta)]^2. \tag{3}$$

Now a single outlier has little effect on the cost function, and won't bias the estimate significantly. It is known that in absence of noise, LMS tolerates up to 50 percent outliers among N total measurements, and still give the correct estimate [23].

The exact solution for LMS is computationally prohibitive. An efficient and statistically robust alternative [23] is to solve random subsets of $\{(x_i, y_i)\}$ values to get several candidate $\hat{\theta}$. The median of the residue squares for each candidate is then computed, and the one with the least median of residue squares is chosen as a tentative estimate. However, this tentative estimate is obtained from a small subset of samples. It is desirable to include more samples that are not outliers for a better estimation. So, the samples are reweighted based on their residues for the tentative estimate, followed by a reweighted least squares fitting to get the final estimate.

The samples can be reweighted in various ways. A simple thresholding method given by [23] is

$$w_i = \begin{cases} 1, & |\frac{r_i}{s_0}| \le \gamma \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where γ is a predetermined threshold, r_i is the residue of the i-th sample for the least median subset estimate $\hat{\theta}$, and s_0 is the scale estimate given by [23]

$$s_0 = 1.4826(1 + \frac{5}{N-p})\sqrt{\mathrm{med}_i r_i^2(\hat{\theta})}, \tag{5}$$

where p is the dimension of the estimated variable. The term $(1 + \frac{5}{N-p})$ is used to compensate the tendency for a small scale estimate when there are few samples.

Assume we are given a set of N samples, and that we aim to estimate a p-dimensional variable θ from this ensemble. The procedure for implementing the robust LMS algorithm is summarized as follows:

1) Choose an appropriate subset size n, the total number of subsets randomly drawn M, and a threshold γ.
2) Randomly draw M subsets of size n from the data ensemble. Find the estimate $\hat{\theta}_j$ for each subset. Calculate the median of residues r_{ij}^2 for every $\hat{\theta}_j$. Here $i = 1, 2, \cdots, N$ is the index for samples, while $j = 1, 2, \cdots, M$ is the index for the subsets.
3) Define $m = \arg\min_j \mathrm{med}_i\{r_{ij}^2\}$, then $\hat{\theta}_m$ is the subset estimate with the least median of residues, and $\{r_{im}\}$ is the corresponding residues.
4) Calculate $s_0 = 1.4826(1 + \frac{5}{N-p})\sqrt{\mathrm{med}_i r_{im}^2}$.
5) Assign weight w_i to each sample using Equation (4).
6) Do a weighted least squares fitting to all data with weights $\{w_i\}$ to get the final estimate $\hat{\theta}$.

B. Robust Localization with LMS

In the absence of attacks, the device location estimate $(\hat{x}_0, \hat{y}_0)$ can be found by least squares, i.e.

$$(\hat{x}_0, \hat{y}_0) = \arg \min_{(x_0, y_0)} \sum_{i=1}^{N} [\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - d_i]^2. \tag{6}$$

In presence of attacks, however, the adversary produces "outliers" in the measurements. Instead of identifying this misinformation, we would like to live with them and still get a reasonable location estimate (identification of misinformation will come out as a byproduct naturally). To achieve this goal, we use LMS instead of least squares to estimate the location. That is, we can find $(\hat{x}_0, \hat{y}_0)$ such that

$$(\hat{x}_0, \hat{y}_0) = \arg \min_{(x_0, y_0)} \mathrm{med}_i [\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - d_i]^2. \tag{7}$$

Then the above LMS procedure can be used.

However, before using the algorithm, we need to consider two issues: First, how to choose the appropriate n and M for LMS-based localization? Second, how to get an estimate from the samples efficiently? The answers depend on the required performance and the affordable computational complexity. Considering that power is limited for sensor networks, and that the computational complexity of LMS depends on both the parameters and algorithmic implementation, we would like to gain the robustness of LMS with minimal additional computation compared to least squares, while exhibiting only negligible performance degradation. These two issues are now addressed.

1) How to choose the appropriate n and M?

The basic idea of the LMS implementation is that, hopefully, at least one subset among all subsets does not contain any contaminated samples, and the estimate from this good subset will thus fit the inlier (non-corrupted) data well. Since the inlier data are the majority ($> 50\%$) of the data, the median of residues corresponding to this estimate will be smaller than that from the bad subsets.

We now calculate the probability P to get at least one good subset without contamination. Assuming the contamination ratio is ϵ, i.e, ϵN samples are outliers, it is easy to get that

$$P = 1 - (1 - (1 - \epsilon)^n)^M. \tag{8}$$

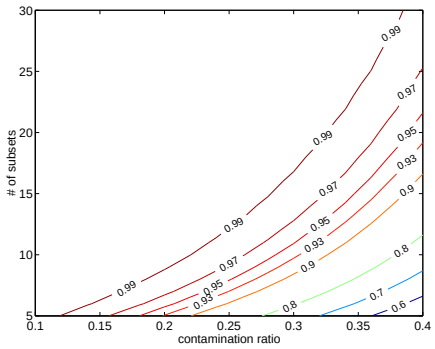


Fig. 2. The contour plot of the equation (8): probability to get at least one good subset over contamination ratio and the number of subsets when $n = 4$.

For a fixed M and ϵ, the larger n, the smaller is P. So the size of a subset n is often chosen such that it's just enough to get an estimate. In our case, although the minimum number of samples needed to decide a location is 3, we have chosen $n = 4$ to reduce the chance that the samples are too close to each other to produce a numerically stable position estimate.

Once n is chosen, we can decide the value of P for a given pair of M and ϵ. A contour plot of P over a grid of M and ϵ is shown in Figure 2. For larger ϵ, a larger M is needed to obtain a satisfactory probability of at least one good subset. Depending on how much contamination the network localization system is required to tolerate and how much computation the system can afford, M can be chosen correspondingly. Because the energy budget of the sensors is limited, and the functionality of the sensor network may be ruined when the contamination ratio is high, we chose $M = 20$ in our simulations, so that the system is resistant up to 30 percent contamination with $P \geq 0.99$.

2) How to get a location estimate from the samples efficiently?

To estimate the device location (x_0, y_0) from the measurements $\{x_i, y_i, d_i\}$, we can use the least squares solution specified by equation (6). This is a nonlinear least squares problem, and usually involves some iterative searching technique, such as gradient descent or Newton method, to get the solution. Moreover, to avoid local minimum, it is necessary to rerun the algorithm using several initial starting points, and as a result the computation is relatively expensive. Considering that sensors have limited power, and LMS finds estimates for M subsets, we may want to have a suboptimal but more computationally efficient algorithm.

Recall that equation (6) is equivalent to solving the following

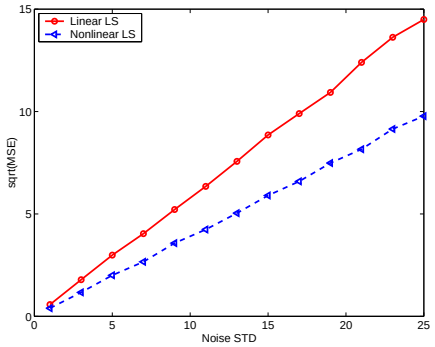


Fig. 3. The comparison between linear LS, and nonlinear LS starting from the linear estimate.

equations when $N \geq 2$:

$$\begin{aligned}(x_1 - x_0)^2 + (y_1 - y_0)^2 &= d_1^2 \\ (x_2 - x_0)^2 + (y_2 - y_0)^2 &= d_2^2 \\ &\vdots \\ (x_N - x_0)^2 + (y_N - y_0)^2 &= d_N^2\end{aligned} \tag{9}$$

Averaging all the left parts and right parts respectively, we get

$$\frac{1}{N}\sum_{i=1}^{N}[(x_i - x_0)^2 + (y_i - y_0)^2] = \frac{1}{N}\sum_{i=1}^{N} d_i^2. \tag{10}$$

Subtracting each side of the equation above from equation (9), we linearize to get the new equations

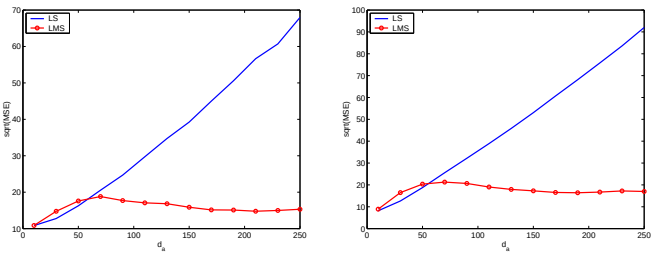
$$\begin{aligned}&(x_1 - \frac{1}{N}\sum_{i=1}^{N} x_i)x_0 + (y_1 - \frac{1}{N}\sum_{i=1}^{N} y_i)y_0 = \\ &\frac{1}{2}[(x_1^2 - \frac{1}{N}\sum_{i=1}^{N} x_i^2) + (y_1^2 - \frac{1}{N}\sum_{i=1}^{N} y_i^2) - (d_1^2 - \frac{1}{N}\sum_{i=1}^{N} d_i^2)] \\ &\vdots \\ &(x_N - \frac{1}{N}\sum_{i=1}^{N} x_i)x_0 + (y_N - \frac{1}{N}\sum_{i=1}^{N} y_i)y_0 = \\ &\frac{1}{2}[(x_N^2 - \frac{1}{N}\sum_{i=1}^{N} x_i^2) + (y_N^2 - \frac{1}{N}\sum_{i=1}^{N} y_i^2) - (d_N^2 - \frac{1}{N}\sum_{i=1}^{N} d_i^2)],\end{aligned} \tag{11}$$

which can be easily solved using linear least squares.

Due to the subtraction, the optimum solution of the linear equations (11) is not exactly the same as the optimum solution of the nonlinear equations (9), or equivalently equation (6). However, it can save computation and also serve as the starting point for the nonlinear LS problem. We noticed that there is a non-negligible probability of falling into a local minimum of the error surface when a random initial value is used with Matlab's *fminsearch* function to find the solution to equation (6). We observed that initiating the nonlinear LS from the linear estimate does not get trapped in a local minimum. In other words, the linear estimate is close to the global minimum of the error surface. A comparison of the performance of the linear LS technique, and the nonlinear LS searching starting from the linear estimate is presented in Figure 3. Nonlinear searching from the linear estimate performs better than the linear method at the price of a higher computational complexity. Here, we only used 30 samples, and that the performance difference between the linear and nonlinear methods should decrease as the number of samples increases.

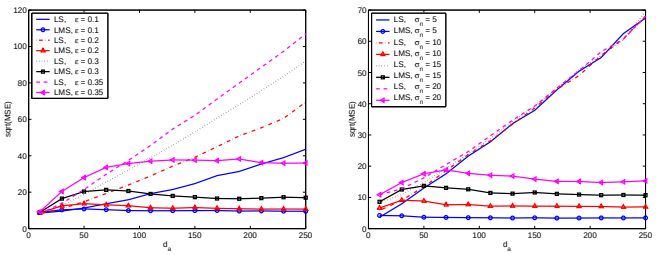
C. Simulation

To test the performance of localization using LMS, we need to build a threat model first. In this work, we assume that the adversary successfully gains the ability to arbitrarily modify the distance measurements for a fraction ϵ of the total anchor nodes. The contamination ratio ϵ should be less than 50 percent, the highest contamination ratio LMS can tolerate. The goal of the adversary is to drive the location estimate as far away from the true location as possible. Rather than randomly perturbing the measurements of these contaminated devices, the adversary should *coordinate* his corruption of the measurements so that



(a) $\epsilon = 0.2, \sigma_n = 20$ (b) $\epsilon = 0.3, \sigma_n = 15$

Fig. 4. The performance comparison between LS and LMS for localization in presence of attack.



(a) (b)

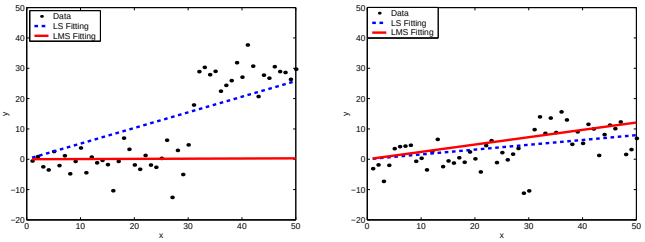
Fig. 5. (a) The impact of ϵ on the performance of LS and LMS algorithms at $\sigma_n = 15$. (b) The impact of σ_n on the performance of LS and LMS algorithms at $\epsilon = 0.2$.

they will push the localization toward the same wrong direction. The adversary will thus tamper measurements so they lie on the parabolic surface $d_a^2(x, y)$ with a minimum at (x_a, y_a). As a result the localization estimate will be pushed toward (x_a, y_a) from the true position (x_0, y_0) in the absence of robust countermeasures. The larger distance between (x_a, y_a) and (x_0, y_0), the larger the estimate deviates from (x_0, y_0). So we use the distance $d_a = \sqrt{(x_a - x_0)^2 + (y_a - y_0)^2}$ as a measurement of the strength of the attack.

In our simulation, in addition to the underlying sensor network, we had a localization infrastructure with $N = 30$ anchor nodes that were randomly deployed in a $500 \times 500m^2$ region. We assume that the sensor to be localized gets a set of $\{x_i, y_i, d_i\}$ observations by either DV-hop or another distance measurement scheme. In other words, the d_j may come from multihop measurements. The measurement noise obeys a Gaussian distribution with mean 0 and variance σ_n^2. The adversary tampers $N\epsilon$ measurements such that they all "vote" for (x_a, y_a). LS and LMS localization algorithms are applied to the data to obtain the estimates $(\hat{x}_0, \hat{y}_0)$. For computational simplicity, we use linear least squares to get location estimates, realizing that a nonlinear least squares approach will improve the performance a little, but won't change the other features of the algorithms. The distance between the estimate and the true location is the corresponding estimation error.

For each contamination ratio ϵ and measurement noise level σ_n, we observed the change of the square root of mean square error (MSE) with the distance $d_a = \sqrt{(x_a - x_0)^2 + (y_a - y_0)^2}$. As an example, the performances at two different pairs of σ_n and ϵ are presented in Figure 4, where the value at each point is the average over 2000 trials. As expected, the estimation error of ordinary LS increases as d_a increases due to the non-robustness of the least squares to outliers. The estimation error of LMS increases first until it reaches the maximum at some critical value of d_a. After this critical value, the error decreases slightly and then stabilizes. In other words, *if LMS is used in localization, it's useless or even harmful for the adversary to attempt to conduct too powerful of an attack.*

The performance of the LS and LMS algorithms are affected by both the contamination ratio and the noise level. Figure 5 (a) illustrates the degradation of the performance as ϵ increases at a fixed $\sigma_n = 15$, while Figure 5(b) illustrates the impact of measurement noise σ_n on the performance at a fixed $\epsilon = 0.2$. Not surprisingly, the higher the contamination ratio, the larger the measurement noise, the larger is the estimation error. Also, since we chose n and M so the system would be robust up to 30 percent contamination, 35 percent contamination results in severe performance degradation as shown in Figure 5(a). More computations might improve the performance at high contamination ratio, but as noted earlier, due to the limitation of the power in sensor network, we trade the performance for reduced complexity.



(a) (b)

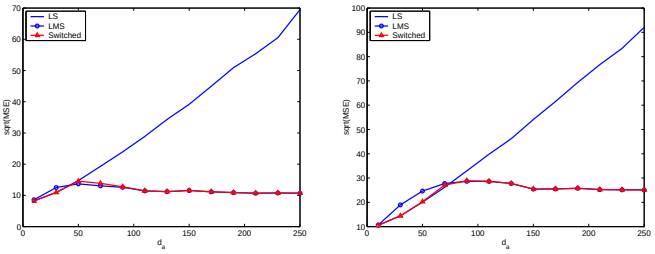
Fig. 6. LMS performs worse than LS when the inlier and outlier data are too close.

We also noticed from Figure 4 and Figure 5 (b) that at low attacking strength, the performance of LS is actually better than LMS. In order to elucidate the reason for this behavior, let us look the simpler problem of fitting a line through data. In Figure 6, we present the line-fitting scenario using an artificial data set with 40 percent contamination. We generated 50 samples, among which 20 samples with $x = 31, \cdots, 50$ are the contaminated outliers. When the outlier data are well separated from the inlier data, LMS can detect this and fit the inlier data only, which gives a better fitting than LS. However, when the outlier data are close to the inlier data, it's hard for LMS to tell the difference, so it may fit part of the inlier data and part of the outlier data, thus giving a worse estimate than LS.

Therefore, when the attack strength is low, LS performs better than LMS. Further, in this case, LS also has a lower computational cost. Since power consumption is an important concern for sensor networks, we do not want to use LMS when not necessary. We have developed an algorithm, discussed below, where we may switch between LS and LMS estimation and achieve the performance advantages of each.

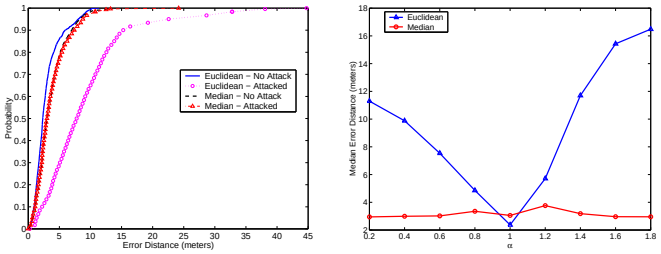
D. An Efficient Switched LS-LMS Localization Scheme

We use the observation that when outliers exist, the variance of the data will be larger than that when no outlier exists. Moreover, the farther outliers are from the inliers, the larger the variance. This suggests that the variance of the data can be used to indicate the distance between inliers and outliers. Therefore, we can do a LS estimate over the data first, and use the residues to estimate the data variance $\hat{\sigma}_n$ from the residuals r_i, i.e. $\hat{\sigma}_n = \sqrt{\frac{\sum_{i=1}^{N} r_i^2}{N-2}}$. Then the ratio $\frac{\hat{\sigma}_n}{\sigma_n}$ represents the variance expansion due to possible outliers. If the normal measurement noise level σ_n is known, which is a reasonable assump-



(a) $\epsilon = 0.2, \sigma_n = 15$ (b) $\epsilon = 0.3, \sigma_n = 20$

Fig. 7. The performance of the switching algorithm comparing to LS and LMS algorithms.



(a) (b)

Fig. 8. (a) The CDF of the error distance for the NNSS method in Euclidean distance and in median distance, with and without an attack (one reading is modified to $\alpha \cdot ss_i$, where $\alpha = 0.6$). (b) Median of the error distance vs. the attacking strength α (one reading is modified to $\alpha \cdot ss_i$).

tion in practice, we can compare the $\frac{\hat{\sigma}_n}{\sigma_n}$ to some threshold T. If $\frac{\hat{\sigma}_n}{\sigma_n} > T$, we choose to apply the LMS algorithm; otherwise, we just use the LS estimate we have calculated. We refer to this as the switched algorithm. In our simulation, we found that $T = 1.5$ gives quite good results over all tested ϵ and σ_n pairs. Two examples with different ϵ and σ_n are shown in Figure 7. After the switching strategy is deployed, the performance curves (the triangles in Figure 7) are very close to the lower envelop of the performance of LS and LMS algorithms.

VI. Robust Methods for RF-Based Fingerprinting

A different approach to localization is based upon radio-frequency fingerprinting. One of the first implementations was the RADAR system [9, 24]. The system was shown to have good performance in an office building. In this section, we will show how robustness can be applied to such a RF-based system to obtain attack-tolerant localization.

In RADAR, multiple base stations are deployed to provide overlapping coverage of an area, such as a floor in an office building. During set up, a mobile host with known position broadcasts beacons periodically. The signal strengths at each base station are measured and stored. Each record has the format of $\{x, y, ss_1, \cdots, ss_N\}$, where (x, y) is the mobile position, and ss_i is the received signal strength in dBm at the i-th base station. N, the total number of base stations, should be at least 3 to provide good localization performance. To reduce the noise effect, each ss_i is usually the average of multiple measurements collected over a time period. The collection of all measurements forms a radio map that consists of the featured signal strengths, or fingerprints, at each sampled position.

Following setup, a mobile may be localized by broadcasting beacons and using the signal strengths measured at each base station. To localize the mobile user, we search the radio map collected in the setup phase, and find the fingerprint that best matches the signal strengths observed. That is, the central base station compares the observed signal energy $\{ss'_1, \cdots, ss'_N\}$ with the recorded $\{x, y, ss_1, \cdots, ss_N\}$, and pick the location (x, y) that minimizes the Euclidean distance $\sqrt{\sum_{i=1}^{N}(ss_i - ss'_i)^2}$ as the location estimate of the mobile user. This technique is called *nearest neighbor in signal space (NNSS)*. A slight variant of the technique involves finding the k nearest neighbors in signal space, and averaging their coordinates to get the location estimate. It was shown in [9] that averaging 2 to 4 nearest neighbors improves the location accuracy significantly.

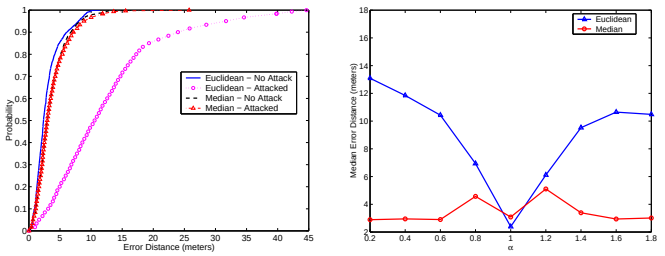
The location estimation method described above is not robust to possible attacks. If the reading of signal strength at one base station is corrupted, the estimate can be dramatically different from the true location. Such an attack can be easily launched by inserting an absorbing barrier between the mobile host and the base station. Sudden change of local environment, such as turning on a microwave near one base station, can also cause incorrect signal strength readings. To obtain reasonable location estimates, in spite of attacks or sudden environmental changes, we propose to deploy more base stations and use a robust estimation method to utilize the redundancy introduced. In particular, instead of minimizing the Euclidean distance $\sqrt{\sum_{i=1}^{N}(ss_i - ss'_i)^2}$ to find nearest neighbors in signal space, we can minimize the median of the distances in all dimensions, i.e. minimize $med_{i=1}^{N}(ss_i - ss'_i)^2$ to get the "nearest" neighbor. In this way, a corrupted estimate won't bias the neighbor searching significantly.

We tested the proposed method through simulations. As pointed out in [9], the radio map can be generated either by empirical measurements, or by signal propagation modeling. Although the modeling method is less accurate than the empirical method, it still captures the data fairly well and provides good localization. In [9] a *wall attenuation factor* model was used to fit the collected empirical data and, after compensating for attenuation due to intervening walls, it was found that the signal strength varies with the distance in a trend similar to the generic exponential path loss [25]. In our simulation, we use the model, which we adopted from [9],

$$P(d)[dBm] = P(d_0)[dBm] - 10\gamma log(\frac{d}{d_0}), \qquad (12)$$

to generate signal strength data. We used the parameter $d_0 = 1m$, $P(d_0) = 58.48$ and $\gamma = 1.523$, which were obtained in [9] when fitting the model with the empirical data. We emphasize that the trends shown in our results are not affected by the selection of the parameters. We also added random zero-mean Gaussian noise with variance 1dBm so that the received signal strengths at a distance have a similar amount of variation as was observed in [9].

The rectangular area we simulated was similar to the region used in [9], and had a size $45m \times 25m$, which is a reasonable size for a large indoor environment. Instead of three base stations, we employed six to provide redundancy for robust localization. We collected samples on a grid of 50 regularly spaced positions in order to form the radio map. During localization, a mobile sends beacons, and the signal strengths at the base stations are recorded. The nearest neighbors in signal space in terms of Euclidean distance and median distance are each found. The coordinates of the four nearest neighbors are aver-



(a) (b)

Fig. 9. (a) The CDF of the error distance for the NNSS method in Euclidean distance and in median distance, with and without an attack (two readings are modified to $\alpha \cdot ss_i$, where $\alpha = 0.6$). (b) Median of the error distance vs. the attacking strength α (two readings are modified to $\alpha \cdot ss_i$).

aged to get the final location estimate of the mobile user.

To simulate the attack, we randomly choose one reading ss_i and modify it to $\alpha \cdot ss_i$, where α indicates the attacking strength. $\alpha = 1$ means no attack. Figure 8 (a) shows the cumulative distribution function (CDF) of the error distance for the NNSS method in Euclidean distance and in median distance, with and without an attack. In presence of an attack with $\alpha = 0.6$, which is very easy to launch from a practical point of view, the Euclidean-based NNSS method shows significantly larger error than when there is no attack, while for the median-based NSSS approach there is little change (the curves with and without attack almost completely overlap in Figure 8 (a)). Although its performance is slightly worse than Euclidean-NNSS in the absence of attacks, median-NNSS is much more robust to possible attacks. In Figure 8 (b), we plot the 50th percentile value of the error distance for a series of α from 0.2 to 1.8. NNSS in median distance shows good performance across all α's.

With six base stations, the system can tolerate attacks on up to two readings. For simplicity, we assume the adversary randomly selects two readings and modifies them to $\alpha \cdot ss_i$. We note that such an approach is not a coordinated attack, and there may be better attack strategies able to produce larger localization error. Figure 9 (a) shows the CDF of the error distance at $\alpha = 0.6$, and Figure 9 (b) shows the change of median error distance with α. Again, the median-NNSS exhibits better resistance to attacks. We observed the same phenomenon as that in the triangulation method: it is better for the adversary to not be too greedy when attacking the localization scheme. Finally, we note that the computational requirements for Euclidean-NNSS and median-NNSS are comparable. The fact that there is only marginal performance improvement for Euclidean-NNSS when there are no attacks suggests that a switched algorithm is not critical for fingerprinting-based localization.

VII. Conclusions

As wireless networks are increasingly deployed for location-based services, these networks are becoming more vulnerable to misuses and attacks that can lead to false location calculation. Towards the goal of securing localization, this paper has made two main contributions. It first enumerates a list of novel attacks that are unique to wireless localization algorithms. Further, this paper proposes the idea of tolerating attacks, instead of eliminating them, by exploiting redundancies at various levels within wireless networks. We explored robust statistical methods to make localization attack-tolerant. We examined two broad classes of localization: triangulation and RF-based fingerprinting methods. For triangulation-based localization, we examined the use of a least median squares estimator for estimating position. We provided analysis for selecting system parameters. We then proposed an adaptive least squares and least median squares position estimator that has the computational advantages of least squares in the absence of attacks and switches to a robust mode when being attacked. For fingerprinting-based localization, we introduced robustness through the use of a median-based distance metric.

References

[1] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Comput. Networks*, vol. 43, no. 4, pp. 499–518, 2003.

[2] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The CRICKET location-support system," in *Proceedings of the 6th annual international conference on Mobile computing and networking (Mobicom 2000)*, 2000, pp. 32–43.

[3] D. Nicelescu and B. Nath, "Ad hoc positioning (APS) using AOA," in *Proceedings of IEEE Infocom 2003*, 2003, pp. 1734 – 1743.

[4] S. Capkun and J.P. Hubaux, "Secure positioning in sensor networks," Technical report EPFL/IC/200444, May 2004.

[5] L. Lazos and R. Poovendran, "SeRLoc: Secure range-independent localization for wireless sensor networks," in *Proceedings of the 2004 ACM Workshop on Wireless Security*, 2004, pp. 21–30.

[6] B. H. Wellenhoff, H. Lichtenegger, and J. Collins, *Global Positions System: Theory and Practice, Fourth Edition*, Springer Verlag, 1997.

[7] A. Harter, A. Hopper, P. Steggles, A. Ward, and P.Webster, "The anatomy of a context-aware application," in *Proceedings of the MOBICOM 99*, 1999.

[8] A. Savvides, C. C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the MOBICOM 01*, 2001.

[9] P. Bahl and V.N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of IEEE Infocom 2000*, 2000, pp. 775–784.

[10] J. Hightower, G. Boriello, and R. Want, "SpotON: An indoor 3D Location Sensing Technology Based on RF Signal Strength," Tech. Rep. Technical Report 2000-02-02, University of Washington, February 2000.

[11] R. Volpe, T. Litwin, and L. Matthies, "Mobile robot localization by remote viewing of a colored cylinder," in *Proceedings of IEEE/RSJ International Conference on Robots and Systems (IROS)*, 1995.

[12] C. Savarese, K. Langendoen, and J. Rabaey, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *Proceedings of USENIX Technical Annual Conference*, 2002.

[13] D. Nicelescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1-4, pp. 267–280, 2003.

[14] A. Savvides, H. Park, and M. Srivastava, "The bits and flops of the n-hop multilateration primitive for node localization problems," in *Proceedings of First ACM International Workshop on Wireless Sensor Networks and Application (WSNA)*, 2002, pp. 112–121.

[15] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, 2000.

[16] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of MOBICOM 03*, 2003, pp. 81 – 95.

[17] Y.C. Hu, A. Perrig, and D. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in *Proceedings of IEEE Infocom 2003*, 2003, pp. 1976–1986.

[18] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *Proceedings of the 2003 ACM workshop on Wireless security*, 2003, pp. 1–10.

[19] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis and defenses," in *Third International Symposium on Information Processing in Sensor Networks*, 2004, pp. 259–268.

[20] W. Xu, T. Wood, W. Trappe, and Y. Zhang, "Channel surfing and spatial retreats: defenses against wireless denial of service," in *Proceedings of the 2004 ACM workshop on Wireless security*, 2004, pp. 80–89.

[21] B. Przydatek, D. Song, and A. Perrig, "SIA: secure information aggregation in sensor networks," in *SenSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, 2003, pp. 255–265.

[22] D. Wagner, "Resilient aggregation in sensor networks," in *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, 2004, pp. 78–87.

[23] P. Rousseeuw and A. Leroy, "Robust regression and outlier detection," Wiley-Interscience, September 2003.

[24] P. Bahl, V.N. Padmanabhan, and A. Balachandran, "Enhancements to the RADAR User Location and Tracking System," Tech. Rep. Technical Report MSR-TR-2000-12, Microsoft Research, February 2000.

[25] A. Goldsmith, *Wireless Communications*, Cambridge University Press, to appear 2005.