LAD: Localization Anomaly Detection for Wireless Sensor Networks

Wenliang Du* and Lei Fang
Department of Electrical Engineering and Computer Science
Syracuse University
Email: {wedu,lefang}@ecs.syr.edu

Peng Ning†
Department of Computer Science
North Carolina State University
Email: pning@ncsu.edu

Abstract

In wireless sensor networks (WSNs), sensors' locations play a critical role in many applications. Having a GPS receiver on every sensor node is costly. In the past, a number of location discovery (localization) schemes have been proposed. Most of these schemes share a common feature: they use some special nodes, called beacon nodes, which are assumed to know their own locations (e.g., through GPS receivers or manual configuration). Other sensors discover their locations based on the reference information provided by these beacon nodes.

Most of the beacon-based localization schemes assume a benign environment, where all beacon nodes are supposed to provide correct reference information. However, when the sensor networks are deployed in a hostile environment, where beacon nodes can be compromised, such an assumption does not hold anymore.

In this paper, we propose a general scheme to detect localization anomalies that are caused by adversaries. Our scheme is independent from the localization schemes. We formulate the problem as an anomaly intrusion detection problem, and we propose a number of ways to detect localization anomalies. We have conducted simulations to evaluate the performance of our scheme, including the false positive rates, the detection rates, and the resilience to node compromises.

1. Introduction

Sensor networks have been proposed for various applications. In many of these applications, nodes need to find their locations. For example, in rescue applications, rescue personnel can perform their tasks only if the location of the hazardous event (reported by sensors) is known. Location information is also important for geographic routing protocols, in which such information (in the form of coordinates) is used to select the next forwarding host among the sender's neighbors [2, 3, 14, 18, 21, 42]. Because of the constraints on sensors, finding locations for sensors is a challenging problem. The location discovery problem is referred to as *localization* problem in the literature.

The Global Positioning System (GPS) [13] solves the problem of localization in outdoor environments for PC-class nodes. However, due to the cost reason, it is highly undesirable to have a GPS receiver on every sensor node. This creates a demand for efficient and cost-effective localization algorithms in sensor networks. In the past several years, a number of localization protocols have been proposed to reduce or completely remove the dependence on GPS in wireless sensor networks [4, 6, 12, 29–32, 35, 36]. Most of these schemes share a common feature: they use some special nodes, called beacon nodes, which are assumed to know their own locations (e.g., through GPS receivers or manual configuration). Other sensors discover their locations based on the *beacons* provided by these beacon nodes.

Sensor networks may be deployed in hostile environments, where sensor nodes can be compromised, beacon nodes can be compromised, communication can be redirected, etc. Most of the proposed localization schemes are designed to work in environments where all the beacon nodes behave correctly; when those nodes can be compromised and act maliciously, sensors using the existing localization schemes might be misled to believe that they are in locations far away from their actual locations. This can cause severe consequence. For example, when sensor networks are used for battle fields surveillance, if sensors are misled by enemies, such that their derived locations are far off, then when sensors report that their regions are safe, this wrong information can cause significant damage. Therefore, it will be of great importance if sensors can discover whether their derived location is correct or not.

* Du's work was supported by Grants ISS-0219560 and CNS-0430252 from the United States National Science Foundation.

† Ning's work was supported by Grants CNS-0430223 and CAREER-0447761 from the United States National Science Foundation.

In this paper we propose a novel scheme to detect malicious attacks in localizations. Our scheme takes advantage of the deployment knowledge that is available in many sensor network applications. For example, let us look at a deployment method that uses an airplane to deploy sensor nodes. The sensors are first pre-arranged in a sequence of smaller groups. These groups are dropped out of the airplane sequentially as the plane flies forward. This is analogous to parachuting troops or dropping cargo in a sequence. The positions where the sensor groups are dropped out of the airplane are referred to as *deployment points*; their coordinates can be easily determined (e.g. using GPS) and stored in sensors' memories prior to sensors' deployment. During the deployment, sensors can land on random locations, the distribution of which usually follows some probability distribution function (pdf) that can be modeled prior to the deployment. Although not all sensor network are deployed like this, deployment knowledge can be modeled more or less for many sensor network deployments.

We show that, equipped with deployment knowledge, sensors can efficiently detect localization anomalies. Our strategy is let sensors verify whether their derived locations are consistent with the deployment knowledge. The level of inconsistency that is above certain threshold is usually an indicator of malicious attacks. We formulate this inconsistency as an anomaly and study this problem in the framework of anomaly intrusion detection by leveraging the methodologies from the intrusion detection field. We call our problem the *Localization Anomaly Detection (LAD)* problem. We study the effectiveness of our proposed scheme in this paper.

In addition to being effective in detection attacks against the localization, a localization anomaly detection scheme must also resist against attacks on the detection scheme itself. As much as adversaries like to attack localization schemes, they will attack the detection scheme if they know such a scheme is deployed. There are a number of attacks the adversaries can launch. We have developed a mathematical framework to model those attacks, and this model is used in our simulation-based evaluation to generate attacks. Our results show that the proposed detection scheme is highly resilient against attacks that can cause large damage.

The rest of the paper is organized as follows: the next section describes the related work. Section 3 presents the modeling of deployment knowledge. Section 4 formally defines the Localization Anomaly Detection problem. Section 5 describes our proposed LAD detection scheme. Section 6 describes the potential attacks on our detection scheme. Section 7 then presents the evaluation results. Finally we conclude and lay out some future work in Section 8.

2. Related Work

2.1. Localization Problems and Schemes

In the past several years, a number of localization protocols have been proposed to reduce or completely remove the dependence on GPS in wireless sensor networks [1, 4–6, 11, 12, 29–33, 35, 36].

Most localization solutions in sensor networks require a few nodes called beacons (which are also called anchors or reference points), which already know their absolute locations via GPS or manual configuration. The density of the anchors depends on the characteristics and probably the budget of the network since GPS is a costly solution. Anchors are typically equipped with high-power transmitters to broadcast their location beacons. The remainders of the nodes then compute their own locations from the knowledge of the known locations and the communication links. Based on the type of knowledge used in localization, localization schemes are divided into two classes: range-based schemes and range-free schemes.

Range-based protocols use absolute point-to-point distance or angle information to calculate the location between neighboring sensors. Common techniques for distance/angle estimation include Time of Arrival (TOA) [13], Time Difference of Arrival (TDOA) [1,11,33], Angle of Arrival (AOA) [31], and Received Signal Strength (RSS) [1]. While producing fine-grained locations, range-based protocols remain cost-ineffective due to the cost of hardware for radio, sound, or video signals, as well as the strict requirements on time synchronization and energy consumption.

Alternatively, coarse-grained range-free protocols are cost-effective because no distance/angle measurement among nodes is involved. In such schemes, errors can be masked by fault tolerance of the network, redundancy computation, and aggregation [12]. A simple algorithm proposed in [4] and [5] computes the location as the centroid of its proximate anchor nodes. It induces low overhead, but high inaccuracy as compared to others. An alternate solution, DV-Hop [32], extends the single-hop broadcast to multiple-hop flooding, so that sensors can find their distance from the anchors in terms of hop counts. Using the information about the average distance per hop, sensors can estimate their distance from the anchors. Amorphous positioning scheme [29] adopts a similar strategy as DV-Hop; the major difference is that Amorphous improves location estimates using offline hop-distance estimations through neighbor information exchange.

Another existing range-free scheme is APIT algorithm [12]. APIT resolves the localization problem by isolating the environment into triangular regions between anchor nodes. A node uses the point-in-triangle test to determine its relative location with triangles formed by

anchors and thus narrows down the area in which it probably resides. APIT defines the center of gravity of the intersection of all triangles that a node resides in as the estimated node location.

Localization can also be achieved without using beacons. A beaconless localization scheme is described in [8]. Instead of using the beacon information, the beaconless scheme uses the deployment knowledge to derive the location.

2.2. Localization in Hostile Environments

Most of the current localization schemes become vulnerable when there are malicious attacks. Although authentication can be used to protect the integrity of the messages sent by beacon nodes, it will not help if beacon nodes themselves are compromised, because a compromised beacon node may provide incorrect location references. If a sensor uses these incorrect references, it may derive a false location.

One way to find out whether a beacon node is providing correct information or not is to verify the location of the beacon nodes, but no effective solution has been proposed to solve the location verification problem. Sastry et al. made the first attempt towards solving this problem [34]. They proposed a protocol named Echo to verify a node's location claims using both radio frequency and ultrasound. However, the Echo protocol only verifies whether a node is inside a region or not, it does not verify whether a sensor node is at certain specific location.

Our proposed technique differs from the Echo protocol in the following ways. First, the proposed technique is aimed at a broader problem than the Echo protocol. While the Echo protocol is to verify whether a sensor node is inside a region to facilitate location-based access control, our scheme is aimed at detecting *any* location estimation anomaly at *any* sensor node. Second, the Echo protocol relies on the existence of a very fast (e.g., radio frequency) and a relatively slow (e.g., ultrasound) signals to derive distance from time delay; our approach does not need those special signals.

Although the beaconless localization scheme does not use beacons, it does have a similar problem in hostile environments. Because a sensor using this scheme relies on its neighbors to find out the location, if the neighbors are compromised, the estimated location will be incorrect. The work described in [8] assumes that the localization scheme is performed in a benign environment, and it does not show how to deal with the hostile environment.

Recently, Lazos and Poovendran propose a new range-independent localization scheme, SeRLoc, which can tolerate malicious attacks to certain degree [23]. However, when beacon nodes can be compromised, especially those beacon nodes that are close to the victim, SeRLoc will still have hard time deriving the correct locations.

2.3. Intrusion Detection

The proposed approaches fall into the general field of intrusion detection. Intrusion detection has been studied for more than twenty years. Intrusion detection techniques have been traditionally classified into *anomaly detection* and *misuse detection*. Anomaly detection models the normal behaviors of the subjects being monitored and identifies anything that significantly deviates from the normal behaviors as attacks. Many techniques have been proposed for anomaly detection, including statistical approaches (e.g., Haystack [38], NIDES/STAT [17]), machine learning approaches (e.g., TIM [39], IBL [22]), computer immunological approaches [9, 10, 41], and specification based approaches [19, 20, 37, 40]. Misuse detection models the patterns of known attacks or vulnerabilities, and identifies actions that conform to such patterns as attacks. Existing approaches include rule-based methods (e.g., ASAX [28], P-BEST [27]), state transition based methods [7, 16], and data mining approaches [24, 25]. Most of these techniques cannot be directly applied to sensor networks due to the resource constraints on sensor nodes. The technique proposed in this paper are specifically targeted at detecting localization anomalies in sensor networks; it differs from the traditional intrusion detection techniques in that it specifically exploits the semantics of localization to identify the anomalies.

3. Modeling of the Deployment Knowledge

In this section, we present a model for a specific type of deployment. However, the general approach that we use in this paper can be applied to other deployment models. Focusing on a specific deployment model in this paper allows us to evaluate the effectiveness of our detection scheme in a concrete scenario. Evaluation for other deployment models will also be pursued in the future.

We assume that sensor nodes are static once they are deployed. We define the *deployment point* of a sensor as the point location where the sensor is to be deployed. This is not the location where this sensor finally resides. The sensor node can reside at points around this deployment point according to a certain probability distribution. As an example, let us consider the case where sensors are deployed from a helicopter. The deployment point of such a sensor is the location where the sensor is thrown out of the helicopter. We also define the *resident point* of a sensor as the point location where the sensor finally resides.

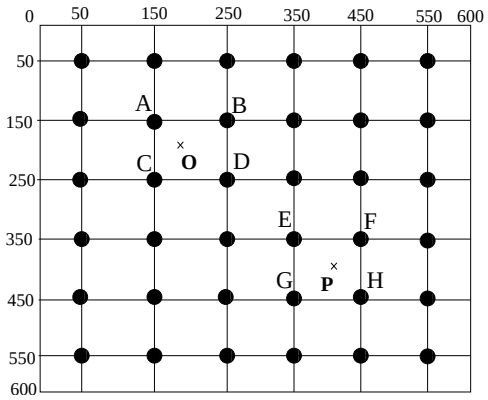

Figure 1. A deployment example (each solid dot represents a deployment point).

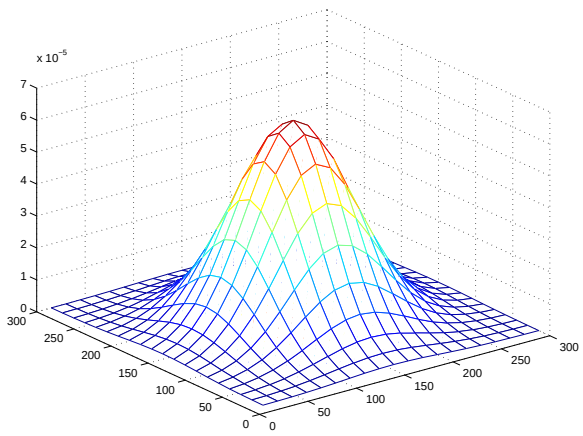

Figure 2. Deployment distribution for one group.

3.1. Group-based Deployment Model

In practice, it is quite common that nodes are deployed in groups, i.e., a group of sensors are deployed at a single deployment point, and the probability distribution functions of the final resident points of all the sensors from the same group are the same.

In this work, we assume such a group-based deployment, and we model the deployment knowledge in the following (we call this model the *group-based deployment model*):

1. N sensor nodes to be deployed are divided into n equal-size groups so that each group, G_i, for $i = 1, \ldots, n$ is deployed from the deployment point with index i. To simplify the notion, we also use G_i to represent the corresponding deployment point, and let (x_i, y_i) represent its coordinates.
2. The deployment points are arranged in a grid (see Figure 1). Note that the scheme we developed for grid-based deployment can be easily extended to other deployment strategies, such as deployments where the deployment points form hexagon shapes, or deployments where the deployment points are random (as long as their locations are given to all sensors).
3. During deployment, the resident point of a node k in group G_i follows a probability distribution function $f_k^i(x, y \mid k \in G_i) = f(x - x_i, y - y_i)$. An example of the pdf $f(x, y)$ is a two-dimensional Gaussian distribution. Figure 2 shows an example of the two-dimensional Gaussian distribution at the deployment point $(150, 150)$.

3.2. Deployment Distribution

In this paper, we model the sensor deployment distribution as a Gaussian distribution (also called Normal distribution). Gaussian distribution is widely studied and used in practice. Although we only employ the Gaussian distribution in this paper, our methodology can also be applied to other distributions.

We assume that the deployment distribution for any node k in group G_i follows a two-dimensional Gaussian distribution, which is centered at the deployment point (x_i, y_i). Namely, the mean of the Gaussian distribution μ equals (x_i, y_i), and the pdf for node k in group G_i is the following [26]:

$$\begin{aligned} f_k^i(x, y \mid k \in G_i) &= \frac{1}{2\pi\sigma^2} e^{-[(x-x_i)^2+(y-y_i)^2]/2\sigma^2} \\ &= f(x - x_i, y - y_i), \end{aligned}$$

where σ is the standard deviation, and $f(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$.

Although the distribution function for each single group is not uniform, we still want the sensor nodes to be evenly deployed throughout the entire region. By choosing a proper distance between the neighboring deployment points with respect to the value of σ in the pdf, the probability of finding a node in each small region can be made approximately equal.

3.3. Computing $g(z)$

We derive a number of formulae that will be used later in our detection scheme. We assume that the probability that a node from group G_i can land at a location ℓ distance from the deployment point of G_i follows a Gaussian distribution. That is:

$$f_R(\ell \mid n_i \in G_i) = \frac{1}{2\pi\sigma^2} e^{-\frac{\ell^2}{2\sigma^2}},$$

where R is the wireless transmission range and σ is the standard deviation of the Gaussian distribution.

Theorem 1 *We define $g(z \mid n_i \in G_i)$ as the probability that the sensor node n_i from group i resides within the neighborhood of a sensor that is z distance from the deployment point of group G_i. Based on geometry knowledge, we can derive the following formula:*

$$\begin{aligned} & g(z \mid n_i \in G_i) \\ = \; & \mathbf{1}\{z < R\} \left[1 - e^{-\frac{(R-z)^2}{2\sigma^2}}\right] \\ + & \int_{|z-R|}^{z+R} f_R(\ell \mid n_i \in G_i) \\ & \cdot 2\ell \cos^{-1}\left(\frac{\ell^2 + z^2 - R^2}{2\ell z}\right) d\ell, \end{aligned} \tag{1}$$

where $\mathbf{1}\{\cdot\}$ is the set indicator function: the value of $\mathbf{1}\{\cdot\}$ is 1 when the evaluated condition is true, and 0 otherwise.

Proof. Due to the page limitation, the proof is omitted from this paper. ■

Therefore, $g_i(\theta)$, the probability that a node from the deployment group G_i can land within the neighborhood of point $\theta = (x, y)$, can be computed in the following (where (x_i, y_i) represents the deployment point of group G_i):

$$g_i(\theta) = g(\sqrt{(x - x_i)^2 + (y - y_i)^2} \mid n_i \in G_i).$$

For simplicity, we use $g(z)$ to represent $g(z \mid n_i \in G_i)$ in the rest of this paper, when it is obvious to see from the context that we are referring to the nodes in group G_i.

The formula for $g(z)$ is quite complicated, and we cannot afford to compute it using Equation (1) in sensor networks. We can solve the performance problem using the table-lookup approach, i.e., we precompute $g(z)$, and store the values in a table. More specifically, we divide the range of z into ω equal-size sub-ranges, and store the $g(z)$ values for these $\omega + 1$ dividing points into a table. When a sensor needs to compute $g(z_0)$, it looks up the table, and finds the sub-range that contains z_0; then it uses the interpolation to compute $g(z_0)$. The computation takes only constant time. Our results also show that to gain satisfactory level of accuracy, ω does not need to be very large.

4. The Localization Anomaly Detection Problem

We introduce a localization anomaly detection phase after the localization phase. In the localization phase, sensors derive their locations. Then in the detection phase, sensors verify whether the derived locations are correct or not. A failure of the verification indicates an anomaly. In this paper, we assume that the localization phase has already ended, and each sensor has already derived a location. This phase can be performed using any localization scheme proposed in the literature. We focus only on the detection phase, namely, to detect whether the derived location is consistent with this node's actual neighborhood information.

In what follows, we use $|L_1 - L_2|$ to represent the distance between two locations L_1 and L_2.

Definition 1 *(Localization Error) Let $L_a = (x_a, y_a)$ represent the actual location of a sensor v. Let $L_e = (x_e, y_e)$ represent the location that the sensor derives via certain localization scheme (note that the localization scheme might be under attacks). We call the distance between L_e and L_a the* localization error.

Most of the sensor networks applications can and should be able to tolerate certain degree of localization errors because, unlike the GPS scheme, all the localization schemes in sensor networks cannot produce very accurate results. We call the error that a sensor network can tolerate the *Maximum Tolerable Error (MTE)*, the value of which is application dependent. We define the anomaly based on *MTE*.

Definition 2 *(Anomaly) An* anomaly *is defined as a phenomenon in which the localization error is greater than the* Maximum Tolerable Error (MTE)*, i.e., $|L_e - L_a| >$* MTE.

To attack a sensor network's localization, attackers need to cause the localization to generate an error that is beyond the network's maximum tolerable error; otherwise, the attacks are not considered as effective. The greater such error is, the more successful the attacks are. An attack causing $|L_e - L_a| = 120$ leads to more severe damage than an attack that causes $|L_e - L_a| = 60$. We quantify the severity of an attack using the error that the attack can achieve, and we use this quantification to further define the anomaly with different degrees of damage.

Definition 3 *(D-Anomaly) An anomaly is called D-*anomaly *if the localization error is greater than D, i.e., $|L_e - L_a| > D$. D is called the* Degree of Damage*. D is chosen by attackers based on their targeted errors.*

If the error $|L_e - L_a|$ is observable, we can easily decide whether a phenomenon is D-anomaly by comparing $|L_e - L_a|$ with D; however, the actual location L_a is not observable, so we need to turn to other observable metrics to find out whether $|L_e - L_a|$ is beyond D or not. The goal of this paper is to develop such a metric A and its corresponding threshold (called the *detection threshold*), such that when A is larger than the detection threshold, we can say that the localization is abnormal. Ideally, a metric A should satisfy the following property: *the metric A is larger than the detection threshold if and only if $|L_e - L_a| > D$.* Unfortunately, such a metric, if exists at all, is difficult to find. In this paper, we turn to heuristic metrics, and we want the above property to be satisfied as much as possible.

Because the above ideal property cannot always be satisfied for the metric A we select, there will be situations when A is larger than the detection threshold, but the distance metric $|L_e - L_a|$ is still below D. In this case, a false alarm will be raised, so we call this situation a *false positive*. Similarly, there will be situations when A is smaller than the detection threshold, but the distance metric $|L_e - L_a|$ is larger than D. In this case, our metric will fail to raise an alarm, and we call this situation a *false negative*. A good metric A should be able to minimize both false positive and false negative rates.

5. Detecting Localization Anomalies Using Deployment Knowledge

Based on deployment knowledge, we illustrate the key idea of our localization anomaly detection scheme using the following example: Assume that the deployment follows the pattern described in Figure 1. Also assume that a sensor v is actually at the location O, but due to the attacks, its derived localization result says P. According to its actual location, v is supposed to observe many neighbors from groups A, B, C, and D. That is, group A is deployed at point A, group B is deployed at point B, etc. On the other hand, v is *not* supposed to see many neighbors from groups E, F, G, or H. However, if P is v's actual location, v should see the opposite. Therefore, the observations at the locations O and P are different. The farther apart O and P are, the more different their observations are. Since v's actual observation at the location O is already known and its expected observation at the location P can be calculated using the deployment knowledge, we can compare these two observations. If they deviate substantially from each other, we can determine that v's localization result P is inconsistent with its actual observation.

We propose three metrics to measure the degree of inconsistency between a node's derived location and its observation. For each metric, we obtain a threshold through training. If the level of inconsistency exceeds such a threshold, we claim that the localization results are inconsistent with the observation, thus an alarm will be raised. We have evaluated the effectiveness of our proposed scheme, including its tolerance to malicious attacks, false positive rates, detection rates, etc. A nice property of our proposed scheme is that even if the anomaly detection thresholds are not optimally selected, our method still has a high detection rate and low false alarm rate for large localization errors. This makes the proposed method an ideal candidate for localization anomaly detection.

5.1. The Detection Scheme

After sensors are deployed, each sensor broadcasts its group id to its neighbors, and each sensor can count the number of neighbors from G_i, for $i = 1, \ldots, n$. Assume that a sensor finds out that it has $o_1, \ldots, o_n$ neighbors from group $G_1, \ldots, G_n$, respectively. We call $\boldsymbol{o} = (o_1, \ldots, o_n)$ the observation of the sensor. Because the observations at different locations can be very different, especially when two locations are far away from each other, we can use the observation to verify whether the localization result is consistent with the observation.

Based on the estimated location $L_e = (x_e, y_e)$ and the deployment knowledge, a sensor can derive the expected observations and the likelihood of its actual observations. If the expected observations are too different from its actual observations, or if the likelihood of the actual observation at L_e is too low, a sensor can claim that L_e is inconsistent with the actual observations, which indicates an anomaly.

We propose three metrics for anomaly detection. The objective of this study is to investigate how effective these metrics are.

5.2. The Difference Metric

Let $L_e = (x_e, y_e)$ represent a sensor node v's estimated location derived using certain localization scheme. Let $\boldsymbol{o} = (o_1, \ldots, o_n)$ represent v's actual observation. This observation might be tainted by adversaries if some neighbors of v are compromised by the adversaries. Assume there are no adversaries, and that v is indeed at the location L_e, then we can compute v's expected observation $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$, where μ_i represents the expected number of neighbors that come from group i. If the expected observation $\boldsymbol{\mu}$ is close to the actual observation $\boldsymbol{o}$, our no-adversary assumption is probably true; however, if $\boldsymbol{\mu}$ and $\boldsymbol{o}$ are very different, the no-adversary assumption might be false.

Because the probability that a sensor from group i becomes v's neighbor is $g_i(L_e)$, v is expected to see $m \cdot g_i(L_e)$ neighbors from group i, where m is the total number of sensors in group i. Therefore, μ_i can be computed using the following equation:

$$\begin{aligned} \mu_i &= m \cdot g_i(L_e) \\ &= m \cdot g(\sqrt{(x_e - x_i)^2 + (y_e - y_i)^2}). \end{aligned} \quad (2)$$

We use the difference between the expected observation $\boldsymbol{\mu}$ and the actual observation $\boldsymbol{o}$ to measure how close these two observations are. The difference DM is defined as the following:

$$DM = \sum_{i=1}^{n} \mid o_i - \mu_i \mid .$$

We use DM as one of our anomaly indicators. We call this metric the *Difference Metric* (or the *Diff Metric* in short). When the value of DM is greater than a threshold value (as we will explain later, such a threshold is derived via training), we say that the location L_e is abnormal.

5.3. The Add-all Metric

We use an example to illustrate the motivations underlying this metric. We will use Figure 1 again. Assume that a node v's actual location is O, but due to the attacks on the localization phase, v finds out that its estimated location is at P. Therefore, we have two sets of observations: one is node v's actual observation $\boldsymbol{o}$ at O, the other is the expected observation $\boldsymbol{\mu}$ at P. The observation $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$ can be computed using Equation (2).

We define the *union* of the observations $\boldsymbol{o}$ and $\boldsymbol{\mu}$: let $\boldsymbol{t} = (t_1, \ldots, t_n)$ be the union of the observation $\boldsymbol{\mu}$ and $\boldsymbol{o}$, where t_i is defined as the following:

$$t_i = max\{o_i, u_i\}, \text{ for } i = 1, \ldots, n. \quad (3)$$

There is an important fact about the observations $\boldsymbol{o}$ and $\boldsymbol{\mu}$: From Figure 1, we can see that in the observation $\boldsymbol{o}$, the number of neighbors from groups A, B, C, and D is high, but the number of neighbors from groups E, F, G, and H is low. In the observation $\boldsymbol{\mu}$, the situation is the opposite. If we *union* these two observations together, in the resultant observation $\boldsymbol{t}$, the the number of neighbors from A, B, C, D, E, F, G, and H will be high. Therefore, compared to $\boldsymbol{o}$ and $\boldsymbol{\mu}$, the total number of neighbors in $\boldsymbol{t}$ is higher (we use $|\boldsymbol{t}|$ to represent the total number of neighbors in $\boldsymbol{t}$); and the more different O and P are, the higher the value of $|\boldsymbol{t}|$ is. On the other hand, if O and P are close to each other, the observations $\boldsymbol{o}$ and $\boldsymbol{\mu}$ should be similar; thus $|\boldsymbol{t}|$ will not gain much from the union of $\boldsymbol{o}$ and $\boldsymbol{\mu}$.

With the above discussion, we propose to use the total number of neighbors in the union observation as another anomaly indicator. Therefore, we define the following metric:

$$AM = \sum_{i=1}^{n} max\{o_i, u_i\}.$$

We call the metric AM the *Add-all Metric*. We compare the result of this metric with a threshold derived from the training to decide whether the estimated location L_e is abnormal.

5.4. The Probability Metric

When a sensor node sees o_i sensors from group i, based on its estimated location, the sensor can calculate how likely it can have o_i neighbors from group i. If the probability is too small, it indicates a potential anomaly. Therefore, we can use this probability as another anomaly indicator. Since there are n deployment groups, we can compute the probability value for each group, and choose the smallest probability value. We then compare this smallest probability value with a threshold (also explained later in Section 7). If the probability value is smaller than the threshold, we say that there is an anomaly with the localization result. We call such a metric the *Probability Metric*.

Assume a sensor v has estimated its location $L_e = (x_e, y_e)$ using a localization scheme. Given the number (m) of nodes deployed in each group and the pdf function of the deployment, we can compute the probability that exactly o_i nodes from group G_i (for each $i = 1, \ldots, n$) can be observed by v. Let X_i represent the number of v's neighbors that come from group G_i. The probability that the node v at the location L_e has exactly o_i neighbors from group i can be computed using the following formula:

$$\begin{aligned} PM &= Pr(X_i = o_i \mid L_e) \\ &= \binom{m}{o_i} (g_i(L_e))^{o_i} (1 - g_i(L_e))^{(m-o_i)}. \end{aligned}$$

If any of the $Pr(X_i = o_i \mid L_e)$, for $i = 1, \ldots, n$, is less than a threshold, our detection will raise an alarm to indicate that the location is abnormal. Obviously, the choice of the threshold is important, if the threshold is too large, we will have a lot of false positives; if the threshold is too small, the false negative rate might be too large.

5.5. Obtaining the Thresholds Using Training

Ideally, to obtain the training data, we need to conduct an actual sensor network deployment. After the deployment, each sensor obtains the list of its neighbors, and then uses a selected localization scheme to estimate its own location. However, collecting the training data in this way is beyond our capability; we turn to simulation. We describe our data collection and training process in the following:

1. *Collecting data from simulation*: Based on our model of deployment knowledge, we generate a number of sensor networks. We randomly select N sensor nodes from these sensor networks. For each sensor, we collect the following data:
   (a) We obtain the node's observation $\boldsymbol{o}$ and its actual location (x_a, y_a).
   (b) We use a localization scheme to compute the location (x_e, y_e) for the node.
2. *Training*: we compute the proposed metrics for all the data in the training data sets, using (x_a, y_a), (x_e, y_e), and the observations $\boldsymbol{o}$. The metric results form a sample distribution.

3. *Deriving the Detection Threshold*: we use τ percentile to decide a threshold from the metrics computation results. Namely, the τ percent of the training results should be within this selected threshold. The value of τ is a configurable parameter and is application dependent. The value of $(1-\tau)$ is the false positive rate.[1]

For example, for the *Diff* metric, if through the simulation, we find that $\tau = 99.99\%$ of the metric results is within 30 in the non-compromised network, we will use 30 as the detection threshold.

Obtaining thresholds for anomaly detection is in general a challenging task, because it is usually difficult to observe all possible "normal" behaviors during the training process. However, we are targeting at a specific localization application in sensor networks, in which the only inputs are the estimated locations and the observed neighbor information. Thus, it is likely to observe most (if not all) of the normal behaviors during the training process.

Since the anomaly detection thresholds are obtained through the simulated deployments, the quality of the simulations has a potential impact on the quality of anomaly detection. However, as we will show through our experiments in Section 7, our anomaly detection method has a nice property, i.e., our method has a high detection rate and low false positive rate for large localization errors introduced by attacks, even if the anomaly detection thresholds are not optimally selected. In other words, the detection performance of the LAD scheme is not sensitive to the quality of the detection thresholds for high-impact localization anomalies. This property makes the LAD scheme an ideal candidate for localization anomaly detection.

6. Attacks on Our Detection Scheme

Just like the localization phase, which might be conducted in a hostile environment, the detection phase is conducted in the same environment. This means, if adversaries have already attacked the localization phase, very likely they will attack the detection phase to prevent their attacks on the localization from being detected. Therefore, because a sensor's detection is based on the information provided by its neighbors, we must consider the situations where a subset of this sensor's neighbors are compromised. A compromised neighbor can send out false information or refuse to send out correct information. A good detection scheme should be able to achieve decent detection rate and low false positive rate even when a non-trivial portion of the neighbors are compromised.

1 This false positive rate is for the training data set only, but the actual false positive rate should be close to this value if our deployment knowledge is modeled correctly.

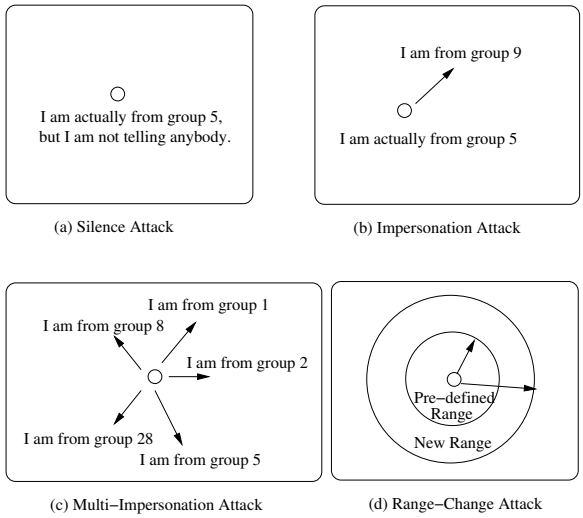

Figure 3. Various Attacking Scenarios.

Once adversaries have compromised a sensor node, they can launch a number of attacks in order to corrupt detection. Their goal is to change the victim's observation, so they can affect the victim's detection results. There are four types of attacks that an adversary can launch against localization (we use $\boldsymbol{o} = (o_1, \ldots, o_n)$ to represent the victim's observation if none of its neighbors is compromised):

- *Silence attack:* A compromised sensor can keep silent. If this sensor comes from group i, the victim's observation o_i on group i can be decreased by one.
- *Impersonation attack:* Instead of reporting its actual group membership i, a compromised sensor can claim that it comes from another group, e.g. group j. Therefore, o_j can be increase by one.
- *Multi-Impersonation attack:* If no pair-wise authentication mechanism is used, a compromised sensor can send out many messages, each of which can appear to come from any group. Therefore, o_k (for $k = 1, \ldots, n$) can be *increased* by a arbitrary number.
- *Range-Change attack:* In this attack, the adversaries cause the range of a compromised node to be changed. When the compromised node's range increases, a victim far away from this compromised node will now consider this node as a neighbor. If this compromised node comes from group i, the victim's observation o_i can be increased by one. The range-change attack can be achieved via three different ways: (1) The compromised sensor can change its transmission power. (2) The compromised sensor can use wormhole attacks [15]. In the wormhole attack, an attacker records packets at one location in the network, tunnels them to another location, and retransmits them there

into the network. This essentially achieves the range-change attack. (3) The range-change attack can also be achieved if adversaries can physically move a compromised node that is outside of the victim's neighborhood into the victim's neighborhood.

The above attacks can be combined to form a variety of new attacks. Regardless of how they are combined, two key observations can be made for any compromised node (assume that this compromised node comes from group i). First, the number of observations on groups other than group i can only be increased. This is because a compromised node cannot stop the non-compromised nodes from broadcasting their group memberships. However, a compromised node can broadcast multiple messages claiming it comes from other groups, thus increasing the observations on those groups. Second, the number of observation on group i can be decreased by only one for each compromised neighbors from group i; this is achieved via the silence attack.

We have generalized the various attacks into two classes, and have defined them in a unified framework. In our definitions, we let x represent the total number of compromised nodes in a sensor's neighborhood. Let $\boldsymbol{a} = (a_1, \ldots, a_n)$ be what a node can observe when none of its neighbors is compromised. Let $\boldsymbol{o} = (o_1, \ldots, o_n)$ be what a node has actually observed when up to x of its neighbors are compromised.

6.1. Dec-Bounded Attacks

When all the attacks described in Figure 3 are possible, adversaries can make each observation o_i arbitrarily larger than a_i, i.e. $o_i - a_i$ can be as large as possible[2]. However, adversaries cannot make o_i arbitrarily smaller than a_i, because the adversaries cannot prevent the non-compromised neighbors from broadcasting their membership. This means that $a_i - o_i$ has an upper bound. We call this type of attack the *Decrease-Bounded (Dec-Bounded)* attack. It is formally defined in the following:

Definition 4 *(Dec-Bounded Attack) We say that an attack is a* Dec-Bounded *attack if any observation $\boldsymbol{o}$ resulting from this attack satisfies the following condition:*

$$\sum_{\forall i, a_i > o_i} (a_i - o_i) \quad \leq \quad x.$$

6.2. Dec-Only Attacks

If we use authentication mechanisms along with the wormhole detection mechanism [15], and also assume that attackers cannot physically move sensors, we can limit the attacks to the silence attack only Therefore, the attackers can only decrease the victim's observations via the silence attack, and they cannot increase the observations. We call this type of attack the *Decrease-Only* attack (in short *Dec-Only*). It is formally defined in the following:

Definition 5 *(Dec-Only Attack) We say that an attack is a* Dec-Only *attack if any observation $\boldsymbol{o}$ resulting from this attack satisfies the following condition:*

$$o_i \leq a_i, \text{ for } i = 1, \ldots, n,$$
$$\sum_{i=1}^{n} (a_i - o_i) \leq x.$$

The *Dec-Only* attacks are less powerful than the *Dec-Bounded* because of the assumptions and constraints made on the attackers' behaviors; some of the assumptions might not be realistic in sensor network applications. Therefore, the *Dec-Only* attacks only have theoretic value; in our evaluation, we mainly focus on the powerful *Dec-Bounded* attacks.

6.3. Impacts of the Attacks

An adversary may launch the above attacks to decrease the detection rate of the LAD scheme, so that it has a greater chance to convince sensor nodes to take false locations without noticing the localization anomalies. Similarly, an adversary may also increase the false alarm rate by launching the above attacks (without compromising the localization scheme). As a result, a sensor node may raise an alarm and stop using the estimated location even if there is no localization anomaly. This will certainly lead to a denial of service (DOS) attack against the localization service. Obviously, the more sensor nodes the adversary can compromise and the less constraints on the adversary's capabilities, the more the adversary can decrease the detection rate or increase the false alarm rate. We will investigate how well LAD can tolerate these attacks through simulations in the next section.

Because of the above attacks, it is possible that an adversary will simply launch DOS attacks against LAD rather than compromising the localization scheme. We make two observations about this threat. First, by launching such DOS attacks, an adversary can only render the localization service unusable, but cannot convince a sensor node to accept a false location. In other words, if the LAD scheme maintains an acceptable detection rate, it will narrow the adversary's choices down to DOS attacks. Second, such attacks are only sensible when they are less expensive than attacking the localization scheme itself. Otherwise, an adversary can already achieve the DOS attacks at the localization phase; launching DOS attacks during the detection phase cannot achieve extra benefit.

2 Of course, o_i must be bounded by the total number of nodes coming from Group i.

Based on the above discussion, it is necessary to compare the difficulty to compromise LAD and the difficulty to compromise a localization scheme. Indeed, as discussed earlier, all the current localization schemes are vulnerable, since they were not developed to handle malicious attacks. Almost all of the range-based localization schemes and some range-free schemes (e.g., [6, 30–32, 35, 36]) eventually reduce localization to a Minimum Mean Square Estimation (MMSE) problem. Though this is effective to reduce the impact of measurement errors that occur during localization, an adversary can still introduce arbitrarily large location errors by compromising a single anchor node and having the compromised anchor node declaring a false location.

Some other range-free localization schemes (e.g., APIT [12]) are more resistant to malicious attacks. However, the correctness of location estimation is still dependent on the anchor nodes, whose quantity is usually limited due to cost reasons. These beacon nodes will become obvious targets. An adversary can either compromise selected anchor nodes, or simply replay beacons intercepted in other areas (without compromising those anchor nodes) to launch attacks. Thus, we believe attacking these localization schemes is not substantially more difficult than attacking LAD.

It is possible to enhance the security of existing localization schemes or develop new attack-resistant ones. However well these schemes work, they will still be vulnerable if the adversaries are able to compromise sensor nodes. Thus, having LAD as a second line of defense will still be valuable.

7. Evaluation

This section provides a detailed quantitative analysis evaluating the performance of our LAD scheme.

7.1. Experiments Setup and Methodologies

In our experiments, the deployment area is a square plane of 1000 meters by 1000 meters. The plane is divided into 10×10 grids. Each grid is $100m \times 100m$. The center of each grid is the deployment point. Figure 1 shows the deployment points. We still use m to represent the number of nodes in each group and R to represent the transmission range. We set the parameter σ of the Gaussian distribution to 50 in all of the experiments.

To simulate attacks with the degree of damage D, we use the following procedure:

1. We randomly pick a node v at the location L_a, and get the actual (non-tainted) observation $\boldsymbol{a}$ at the location.
2. We simulate an attack against the localization of node v by letting v's estimated location be a random location L_e, where $|L_e - L_a| = D$ (which simulates the D-anomaly attack). We compute the expected observation $\boldsymbol{\mu}$ at the location L_e.
3. To simulate the *Dec-Bounded* or *Dec-Only* attacks against the LAD detection scheme, we generate a new observation $\boldsymbol{o}$ from $\boldsymbol{a}$. The relationship of $\boldsymbol{o}$ and $\boldsymbol{a}$ should comply with the constraints defined for those attacks. The generated (tainted) observation $\boldsymbol{o}$ becomes node v's actual observation.

There are many ways to generate $\boldsymbol{o}$ from $\boldsymbol{a}$. We use a greedy procedure to minimize the selected detection metric. The underlying motivation is that if an attacker can reduce the detection metric result to below the detection threshold, its attacks on the localization will escape from being detected. The greedy procedure depends on both the metrics and the types of attacks. There are 2×3 combinations of attacks and detection metrics; the procedure for each combination is different. We will only describe how to simulate the *Dec-Bounded* attack to minimize the *Diff* metric; the simulation of the other combinations can be achieved similarly. The idea of the procedure is to make o_i as close to μ_i as possible, so the *Diff* metric can be minimized. There are two cases in our procedure (we let X be the number of compromised nodes within the sensor's neighborhood):

1. If $\mu_i > a_i$, attackers can immediately increase a_i by $(\mu_i - a_i)$, thus getting $o_i = \mu_i$, because in the *Dec-Bounded* attack, attackers can arbitrarily increase the observations of any deployment group.
2. If $\mu_i < a_i$, attackers have to decrease a_i to get closer to μ_i. They can achieve this only via the silence attacks. However, each time the attackers decrease a_i by one, it must consume a compromised node. For each $a_i > \mu_i$, we let $o_i = a_i$, and then we repeat decreasing both o_i and X by one, until either $o_i = \mu_i$ or X reaches zero. $X = 0$ means there is no more compromised node to consume, so we cannot decrease the observation anymore.

Once the tainted observation $\boldsymbol{o}$ is simulated, we can use our proposed detection metrics and the derived (from training) detection thresholds to conduct the anomaly detection. Note that the purpose of our experiments is to evaluate the anomaly detection method. Thus, we will use different thresholds to evaluate the detection rate and the false positive rate.

7.2. Selecting a Localization Scheme

Our proposed LAD scheme is a general detection scheme that is independent of localization schemes. It assumes that the estimated location is already obtained using any of the localization schemes; it then detects whether

the estimated location is consistent with its observations.

However, the performance of the LAD scheme does depend on the specific localization schemes. For different schemes, the detection threshold derived from training will be different; thus the false positive and the detection rate will be different. Therefore, in this paper, to evaluate the performance of the LAD scheme, we must combine it with a specific localization scheme. We choose to study LAD for the beaconless localization scheme [8] in this paper. The methodology for studying the LAD scheme for other localization schemes is similar, and will be pursued in our future work.

7.3. Parameters

Seven different parameters are involved in the LAD scheme, including:

- m: the density of the network.
- M: the detection metrics.
- T: the type of attacks.
- D: the degree of damage of an attack.
- x: the percentage of the compromised nodes.
- FP: the false alarm rate.
- DR: the detection rate.

To understand the effectiveness of our scheme, we have selected five interesting parameter combinations to study. We present the results in the rest of this section. We give each figure (or each group of figures) a name using the parameters involved in that figure(s). The name follows the format y-x-a, where the parameters y and x represent the y-axis and the x-axis, respectively; we will plot several curves on the same figure, each using different values for the parameter a. For example, a could be the type of attacks T, the degree of damage D, etc. Sometimes, we use y-x-a-b to represent a group of y-x-a figures, with each figure using a different value for the parameter b.

7.4. ROC Curves for Different Metrics (DR-FP-M-D)

The goal of this experiment is to understand the performance of the LAD scheme for three different metrics, including the *Diff* Metric, the *Add-All* Metric, and the *Probability* Metric. In intrusion detection, the Receive Operating Characteristic (ROC) curve is usually used to measure the performance of a detection method. The ROC curve is a plot of intrusion detection accuracy against the false positive rate. It can be obtained by varying the detection threshold. In this experiment, we want to plot the ROC curves for different detection metrics M and different degrees of damage D.

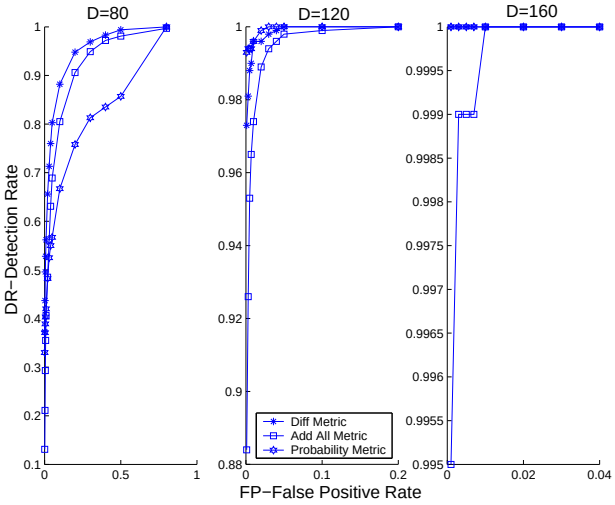

Figure 4. Receive Operating Characteristic (ROC) curves for different detection metrics and different degrees of damage D: DR-FP-M-D ($x = 10\%$, $m = 300$, T="*Dec-Bounded* Attack"). Note: the scales for both x-axis and y-axis in different figures are different.

We fix the values for the other parameters as the followings: the percentage of the compromised nodes x is set to 10%, the network density m is set to 300 nodes per deployment group (there are 10×10 groups in our studies), and the type of the attacks is set to the *Dec-Bounded* attacks, the most powerful attacks against the localization among the two attacks defined in our framework. The results are depicted in Figure 4 (Note that we have used different scales for both x-axis and y-axis in different figures for better presentation effects):

The figure shows that the LAD scheme is more effective for attacks with higher degree of damage. For example, when adversaries launch attacks with $D = 120$ (i.e., an successful attack must cause the localization error to exceed 120), the *Diff* metric can achieve almost 100% detection rate with below 5% false positive rate; when attacks have $D = 160$, the *Diff* metric can achieve 100% detection rate without raising any false alarms. These results show that the attackers' damage to the localization schemes is limited. If they want to cause a damage with a large D, the anomaly will almost be certain to be detected.

From the figures, we can also see that in general, the *Diff* metric performs the best among the three metrics. Therefore, we will only use this metric in the rest of our studies.

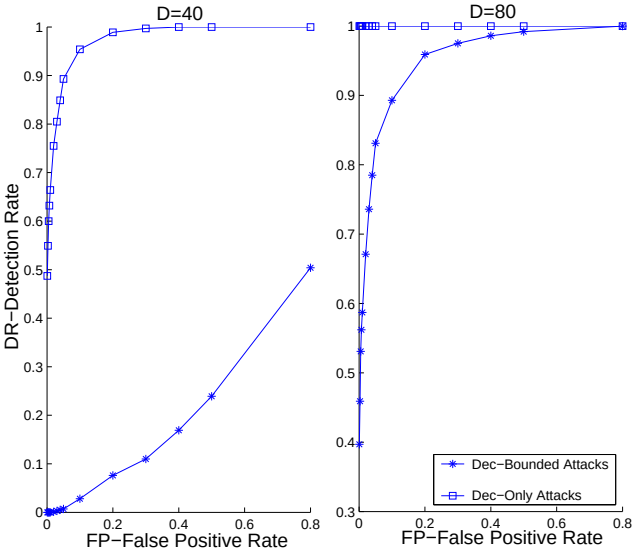

Figure 5. Receive Operating Characteristic (ROC) curves for different attacks with $D = 40$ and $D = 80$: DR-FP-T-D ($x = 10\%$, $m = 300$, M="*Diff* Metric")

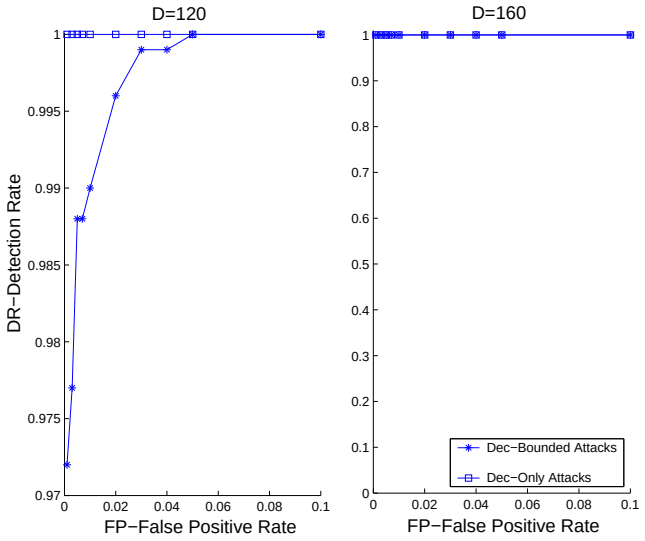

Figure 6. Receive Operating Characteristic (ROC) curves for different attacks with $D = 120$ and $D = 160$: DR-FP-T-D ($x = 10\%$, $m = 300$, M="*Diff* Metric")

7.5. ROC Curves for Different Attacks (DR-FP-T-D)

The goal of this experiment is to understand the effectiveness of the anomaly detection under the *Dec-Bounded* attacks and the *Dec-Only* attacks. Similar to the previous figures, we let $x = 10\%$ and $m = 300$; we only use the *Diff* metric.

We have plotted a number of ROC curves for different types of attacks and different degrees of damage D. The results are depicted in Figures 5 and 6.

From the figure, we can see that the *Dec-Bounded* attack is the most powerful attack, namely, it is the most difficult one to detect, especially when D is small. For instance, when $D = 40$, the detection rates for the *Dec-Only* attack are high with small false alarm rates, but the detection rate for the *Dec-Bounded* attack is still very low.

However, with the increase of D, the detection rate under different attacks becomes less and less different. For example, when $D = 120$ and the false positive is below 2%, the detection rate for the *Dec-Bounded* attacks is already over 99.5%, close to the detection rates (100%) achieved by the *Dec-Only* attacks. This useful observation tells us that to detect the attacks with large degree of damage, we do not need to use the expensive authentication and wormhole detection mechanisms to prevent the powerful *Dec-Bounded* attacks. Although these mechanisms can achieve significant benefits for small D, when D is high, the benefits become not significant enough to merit the cost.

In the rest of our experiments, we will only focus on the *Dec-Bounded* attack.

7.6. Detection Rate vs. Degree of Damage (DR-D-x)

The goal of this experiment is to study how the degree of damage D affects the detection rate. To this end, we let $m = 300$, false positive rate $FP = 0.01$, and T be the *Dec-Bounded* attack; we use the *Diff* metric only. We plot curves for $x = 10\%$, $x = 20\%$, and $x = 30\%$. The results are depicted in Figure 7.

The figure clearly shows that when the degree of damage D is low, the detection rate is very low. This indicates that our scheme is not very effective for the attacks that cause small degree of damage. This result is caused by the localization schemes, because most of the localization schemes (except GPS-based) cannot achieve very high accuracy; when the D is too small, it is difficult to distinguish whether the anomaly is caused by attackers or by localization errors.

However, when the degree of damage D becomes large, the results show that the LAD detection scheme can more clearly distinguish anomalies from normal. Therefore, the detection rate is approaching 100% with the increase of the D, when the false positive rate is limited to 1%. This property of our LAD scheme indicates that if attackers want

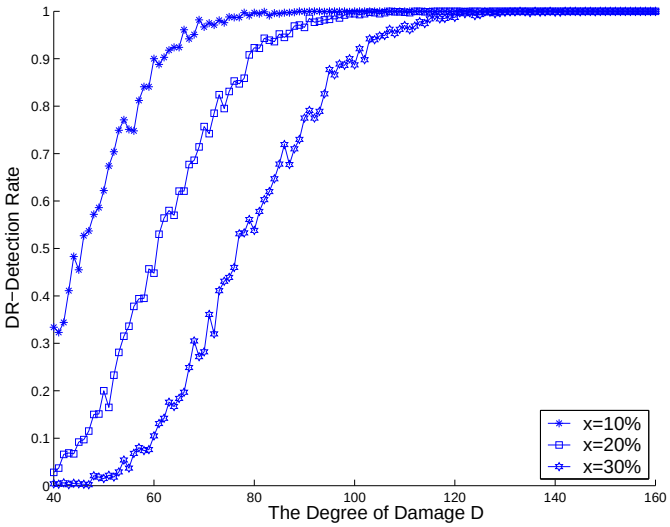

Figure 7. Detection Rate vs. Degree of Damage: DR-D-x ($FP = 0.01$, $m = 300$, M="*Diff* Metric", T="*Dec-Bounded* Attack").

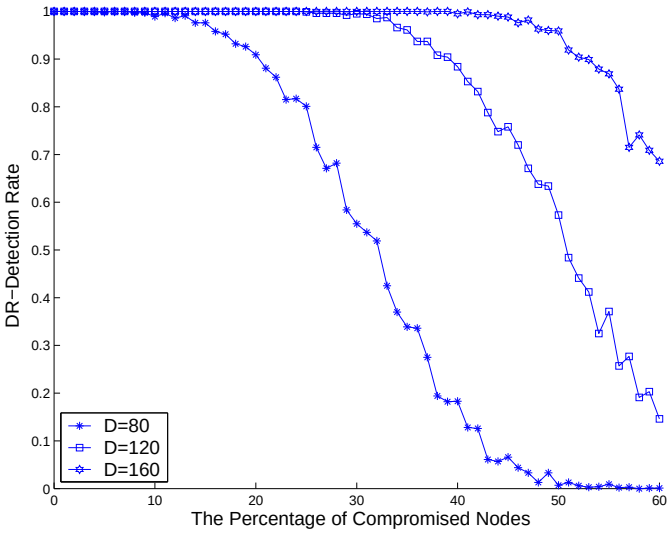

Figure 8. Detection Rate vs. the Percentage of Compromised Nodes: DR-x-D ($FP = 0.01$, $m = 300$, M="*Diff* Metric", T="*Dec-Bounded* Attack").

to cause more severe damage, they will be detected with higher probability. In other words, a successful attack's damage is always limited to a small distance, which does little harm to most of the sensor network applications.

7.7. Detection Rate vs. Node Compromise Ratio (DR-x-D)

The goal of this experiment is to study how much of node compromise can be tolerated by our LAD scheme. To this end, we let $m = 300$, false positive rate $FP = 0.01$, and T be the *Dec-Bounded* attack, and we use the *Diff* metric. We plot curves for $D = 80$, $D = 120$, and $D = 160$. The results are depicted in Figure 8.

The figure shows that the higher the degree of damage D the higher the tolerance. For instance, when $D = 160$, the LAD detection scheme can tolerate up to 50% of the node compromises without significant drop in its detection rate. On the other hand, when $D = 80$, the detection rate drops rapidly if the node compromise percentage reaches 15%.

7.8. Detection Rate vs. Network Density (DR-m-x-D)

Network density m plays an important role in the beaconless localization scheme. The localization becomes more and more accurate when the network density m increases. An interesting question is whether the density affects the LAD scheme. We have conducted an experiment to answer this question.

In our experiment, we let the false positive rate $FP = 0.01$, T be the *Dec-Bounded* attack, and we use the *Diff* metric. The results are depicted in Figure 9.

The figure shows that when the network density m increases, the detection rate increases. However, such an increase is not caused by the LAD detection scheme, but by the beaconless localization scheme. As we know, when m increases, the localization becomes more accurate; in other words, the distance between the estimated location L_e and a sensor's actual location L_a decreases. Such a decrease makes it easier to separate a normal behavior from anomaly. Therefore, when m increases, the detection threshold can be made smaller while still being able to maintain the same level of the false positive rate. The consequence of the smaller detection threshold is that it is easier to catch anomalies.

8. Conclusion and Future Work

We propose an anomaly detection scheme named LAD for wireless sensor networks to detect anomalies in localization. The LAD scheme takes advantage of the deployment knowledge and the group membership of its neighbors, and uses such knowledge to find out whether the estimated location L_e is consistent with its observations. If they are inconsistent, LAD will report an anomaly. We have studied various properties of the LAD scheme. Our simulation results show that LAD can effectively detect localization anomalies, even if a significant portion of the neighbors is com-

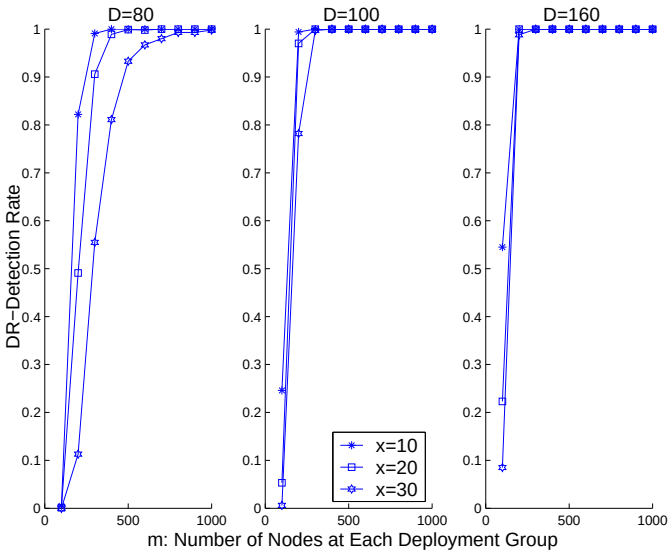

Figure 9. Detection Rate vs. Network Density: DR-m-x-D ($FP = 0.01$, M="*Diff* Metric", T="*Dec-Bounded* Attack").

promised. Our simulation also shows that the more harmful an anomaly is, the higher the detection rate is, and at the same time the lower the false positive rate is, i.e., with the LAD scheme, it is difficult for adversaries to cause a large localization error without being detected.

In our future work, we will study another factor that can affect the effectiveness of the LAD scheme. This factor is the accuracy of the deployment knowledge model. If this model cannot accurately model the actual deployment, there will be extra errors (both on false positive and detection rate) in the anomaly detection. We will study the properties of this kind of errors in our future work.

Our work only achieves the first step toward secure localization in sensor networks. Our ultimate goal is not only to detect the anomalies, but also to correct the errors caused by the anomalies.

References

[1] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of the IEEE INFOCOM*, pages 775–784, March 2000.

[2] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *Proceedings of the 3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 48–55, 1999.

[3] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *ACM Wireless Networks*, 7(6):609–616, 2001.

[4] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. In *IEEE Personal Communications Magazine*, pages 28–34, October 2000.

[5] N. Bulusu, J. Heidemann, and D. Estrin. Density adaptive algorithms for beacon placement. In *Proceedings IEEE ICDCS*, Phoenix, AZ, April 2001.

[6] L. Doherty, K. S. Pister, and L. E. Ghaoui. Convex optimization methods for sensor node position estimation. In *Proceedings of INFOCOM'01*, 2001.

[7] S.T. Eckmann, G. Vigna, and R.A. Kemmerer. STATL: An Attack Language for State-based Intrusion Detection. *Journal of Computer Security*, 10(1/2):71–104, 2002.

[8] Lei Fang, Wenliang Du, and Peng Ning. A beacon-less location discovery scheme for wireless sensor networks. In *Proceedings of IEEE INFOCOM*, Miami, FL, USA, March 13-17 2005.

[9] S. Forrest, S. A. Hofmeyr, and T. A. Longstaff. A sense of self for unix processes. In *Proceedings of 1996 IEEE Symposium on Security and Privacy*, pages 120–128, Oakland, CA, May 1996.

[10] S. Forrest, A.S Perelson, L. Allen, and R. Cherukuri. Self-nonself discrimination in a computer. In *Proceedings of 1994 IEEE Symposium on Security and Privacy*, pages 202–212, Oakland, CA, May 1994.

[11] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *Proceedings of MOBICOM'99*, Seattle, Washington, 1999.

[12] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher. Range-free localization schemes in large scale sensor networks. In *Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, 2003.

[13] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice*. Springer Verlag, 4th ed., 1997.

[14] X. Hong, K. Xu, and M. Gerla. Scalable routing protocols for mobile ad hoc networks. *IEEE Network magazine*, (4), 2002.

[15] Y-C. Hu, A. Perrig, and D. B. Johnson. Packet leashes: A defense against wormhole attacks in wireless networks. In *Proceedings of IEEE Infocomm 2003*, April 2003.

[16] K. Ilgun, R. A. Kemmerer, and P. A. Porras. State transition analysis: A rule-based intrusion detection approach. *IEEE Transaction on Software Engineering*, 21(3):181–199, 1995.

[17] H.S. Javits and A. Valdes. The NIDES statistical component: Description and justification. Technical report, SRI International, Computer Science Laboratory, 1993.

[18] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of ACM MobiCom 2000*, 2000.

[19] C. Ko. Logic induction of valid behavior specifications for intrusion detection. In *Proceedings of 2000 IEEE Symposium of Security and Privacy*, pages 142–153, Oakland, CA, May 2000.

[20] C. Ko, M. Ruschitzka, and K. Levitt. Execution monitoring of security-critical programs in distributed systems: A specification-based approach. In *Proceedings of 1997 IEEE Symposium on Security and Privacy*, pages 175–187, Oakland, CA, May 1997.

[21] Y. B. Ko and N.H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. In *Proceedings ACM/IEEE MOBICOM 98)*, pages 66–75, October 1998.

[22] T. Lane and C. E. Brodley. Temporal sequence learning and data reduction for anomaly detection. In *Proceedings of 5th Conference on Computer & Communications Security*, pages 150–158, 1998.

[23] L. Lazos and R. Poovendran. Serloc: Secure range-independent localization for wireless sensor networks. In *ACM workshop on Wireless security (ACM WiSe 2004)*, Philadelphia, PA, October 1 2004.

[24] W. Lee, S. J. Stolfo, and K. W. Mok. A data mining framework for building intrusion detection models. In *Proceedings 1999 IEEE Symposium on Security and Privacy*, pages 120–132, Oakland, CA, May 1999.

[25] W. Lee and S.J. Stolfo. A framework for constructing features and models for intrusion detection systems. *ACM Transactions on Information and System Security*, 3(4):227–261, Nov 2000.

[26] A. Leon-Garcia. *Probability and Random Processes for Electrical Engineering*. Reading, MA: Addison-Wesley Publishing Company, Inc., second edition, 1994.

[27] U. Lindqvist and P. A. Porras. Detecting computer and network misuse through the production-based expert system toolset (P-BEST). In *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, pages 146–161, Oakland, CA, May 1999.

[28] A. Mounji, B.L. Charlier, D. Zampuniéris, and N. Habra. Distributed audit trail analysis. In *Proceedings of the ISOC '95 Symposium on Network and Distributed System Security*, pages 102–112, 1995.

[29] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *IPSN'03*, 2003.

[30] A. Nasipuri and K. Li. A directionality based location discovery scheme for wireless sensor networks. In *Proceedings of ACM WSNA'02*, September 2002.

[31] D. Niculescu and B. Nath. Ad hoc positioning system (APS) using AoA. In *Proceedings of IEEE INFOCOM 2003*, pages 1734–1743, April 2003.

[32] D. Niculescu and B. Nath. DV based positioning in ad hoc networks. In *Journal of Telecommunication Systems*, 2003.

[33] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proceedings of MOBICOM*, Seattle, Washington'00, August 2000.

[34] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *ACM Workshop on Wireless Security (WiSe 2003)*, 2003.

[35] A. Savvides, C. Han, and M. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of ACM MobiCom '01*, pages 166–179, July 2001.

[36] A. Savvides, H. Park, and M. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of ACM WSNA '02*, September 2002.

[37] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou. Specification-based anomaly detection: A new approach for detecting network intrusions. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 03)*, pages 265–274, November 2002.

[38] S. E. Smaha. Haystack: An intrusion detection system. In *Proceedings of Fourth Aerospace Computer Security Applications Conference*, December 1988.

[39] H. S. Teng, K. Chen, and S.C-Y Lu. Adaptive real-time anomaly detection using inductively generated sequential patterns. In *Proceedings of 1990 IEEE Symposium on Security and Privacy*, pages 278–284, Oakland, CA, May 1990.

[40] D. Wagner and D. Dean. Intrusion detection via static analysis. In *Proceedings of 2001 IEEE Symposium on Security and Privacy*, pages 156–168, Oakland, CA, May 2001.

[41] C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: Alternative data models. In *Proceedings of 1999 IEEE Symposium on Security and Privacy*, pages 133–145, Oakland, CA, May 1999.

[42] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of ACM MobiCom 2000*, Rome, Italy, July 2001.