Detecting Malicious Beacon Nodes for Secure Location Discovery in Wireless Sensor Networks*

Donggang Liu   Peng Ning
North Carolina State University
{dliu,pning}@ncsu.edu

Wenliang Du
Syracuse University
wedu@ecs.syr.edu

Abstract

Sensors' locations play a critical role in many sensor network applications. A number of techniques have been proposed recently to discover the locations of regular sensors based on a few special nodes called beacon nodes, *which are assumed to know their locations (e.g., through GPS receivers or manual configuration). However, none of these techniques can work properly when there are malicious attacks, especially when some of the beacon nodes are compromised. This paper introduces a suite of techniques to detect and remove compromised beacon nodes that supply misleading location information to the regular sensors, aiming at providing secure location discovery services in wireless sensor networks. These techniques start with a simple but effective method to detect malicious beacon signals. To identify malicious beacon nodes and avoid false detection, this paper also presents several techniques to detect replayed beacon signals. This paper then proposes a method to reason about the suspiciousness of each beacon node at the base station based on the detection results collected from beacon nodes, and then revoke malicious beacon nodes accordingly. Finally, this paper provides detailed analysis and simulation to evaluate the proposed techniques. The results show that our techniques are practical and effective in detecting malicious beacon nodes.*

1   Introduction

Recent technological advances have made it possible to deploy large scale sensor networks consisting of a large number of low-cost, low-power, and multi-functional sensor nodes that communicate in short distances through wireless links [1]. Such networks have a wide range of applications in civilian and military operations such as target tracking and battlefield surveillance. Many researchers have been attracted to develop protocols that can fulfill the requirements of these applications (e.g., [1, 8, 10, 20, 21, 24]).

Sensors' locations play a critical role in many sensor network applications. Not only do applications such as environment monitoring and target tracking require sensors' locations to accomplish their tasks, but several fundamental techniques in wireless sensor networks also require sensors' locations. For example, in geographical routing (e.g., GPSR [15]), sensor nodes make routing decisions at least partially based on their own and their neighbors' locations. However, due to the cost reasons, it is not practical to have a GPS receiver on every sensor node. In the past several years, many location discovery protocols have been proposed to reduce or completely remove the dependence on GPS in wireless sensor networks [2,5,9,18,19,22,23,27,28].

These protocols share a common feature: They all use some special nodes, called *beacon nodes*, which are assumed to know their own locations (e.g., through GPS receivers or manual configuration). These protocols work in two stages. In stage 1, non-beacon nodes receive radio signals called *beacon signals* from the beacon nodes. The packet carried by a beacon signal, which we call the *beacon packet*, usually includes the location of the beacon node. The non-beacon nodes then estimate certain measurements (e.g., distance to the beacon nodes) based on features of the beacon signals. Features that may be used for location determination include Received Signal Strength Indicator (RSSI), Time of Arrival (ToA), Time Difference of Arrival (TDoA), and Angle of Arrival (AoA). We refer to such a measurement and the location of the corresponding beacon node collectively as a *location reference*. In stage 2, when a sensor node has enough number of location references from different beacon nodes, it determines its own location in the network field. A typical approach is to consider the location references as constraints a sensor node's location must satisfy, and estimate it by finding a mathematical solution that satisfy these constraints with minimum estimation error.

Despite the substantial advances in location discovery techniques for sensor networks, location discovery in *hostile environments*, where there may be malicious attacks, has been mostly overlooked. In fact, all existing location

discovery protocols become vulnerable in the presence of malicious attacks. As illustrated in Figure 1, an attacker may provide incorrect location reference by pretending to be valid beacon nodes (Figure 1(a)), compromising beacon nodes (Figure 1(b)), or replaying the beacon packets previously intercepted in other locations (Figure 1(c)). In either of these cases, non-beacon nodes will determine their locations incorrectly. The location verification technique proposed in [26] can verify the relative distance between a verifying node and a beacon node, and the technique proposed in [16] can provide secure location discovery using sectored antennas at beacon nodes. However, neither of them can ensure correct location discovery when beacon nodes are compromised, and nor can they remove the impact of compromised beacon nodes.

This paper introduces a suite of techniques to detect and remove compromised beacon nodes that supply misleading location information to the regular sensors, aiming at providing secure location discovery services in wireless sensor networks. The proposed techniques can be applied to most of existing location discovery schemes. The contribution of this paper are as follows. We first develop an efficient method to detect malicious beacon signals using redundant beacon nodes in the sensing field. The basic idea is to take advantage of the (known) locations of beacon nodes and the constraints that these locations and the measurements (e.g., distance, angle) derived from their beacon signals must satisfy to detect malicious beacon signals. With this method, this paper then proposes a serial of techniques to detect replayed beacon signals to avoid false positives in detecting malicious beacon nodes. This paper also presents a simple method to reason about the suspiciousness of each beacon node and revoke malicious beacon nodes based on the distributed detection results from beacon nodes. Finally, this paper provides detailed analysis and simulation to evaluate the performance of the proposed techniques. The results show that the proposed techniques are practical and effective in detecting and removing malicious beacon nodes.

The rest of this paper is organized as follows. The next section describes the basic detector to detect malicious beacon nodes. Section 3 develops a simple method to reason about the suspiciousness of each beacon node and revoke the high suspicious ones. Section 4 presents our simulation evaluation on the proposed techniques. Section 5 reviews related work, and Section 6 concludes this paper and discusses possible future research directions.

2   A Detector for Malicious Beacon Nodes

In hostile environments, a compromised beacon node or an attacking node that has access to compromised cryptographic keys may send out malicious beacon signals that include incorrect locations, or manipulate the beacon signals so that a receiving node obtains, for example, incorrect distance measurements. Sensor nodes that use such beacon signals for location determination may estimate incorrect locations. In this section, we first describe a simple but effective method to detect malicious beacon signals. With this method, we then develop techniques to filter out replayed beacon signals and thus detect malicious beacon nodes.

We assume that two communicating nodes share a unique pairwise key. A number of random key predistribution schemes (e.g., [3, 6, 17]) can be used for this purpose. We assume that a beacon node cannot tell if it is communicating with a beacon or non-beacon node simply from the radio signal or the key used to authenticate the packet. We also assume that communication is two way; that is, if node A can reach node B, then node B can reach node A as well. Moreover, we assume beacon signals are unicasted to non-beacon nodes, and every beacon packet is authenticated (and potentially encrypted) with the pairwise key shared between two communicating nodes. Hence, beacon packets forged by external attackers that do not have the right keys can be easily filtered out.

We assume location estimation is based on the distances measured from beacon signals (through, e.g., RSSI). Nevertheless, our approach can be easily revised to deal with location estimation based on other measurements.

2.1   Detecting Malicious Beacon Signals

The technique to detect malicious beacon signals is the basis of detecting malicious beacon nodes. The basic idea is to take advantage of the (known) locations of beacon nodes and the constraints that these locations and the measurements (e.g., distance, angle) derived from their beacon signals must satisfy to detect malicious beacon signals.

A beacon node can perform detection on the beacon signals it hears from other beacon nodes. For the sake of presentation, we call the node making this detection the *detecting (beacon) node*, and the node being detected the *target (beacon) node*. Note that if a malicious beacon node knows that a detecting beacon node is requesting for its beacon signal, it can send out a normal beacon signal that does not lead to incorrect location estimation, and thus pass the detection mechanism without being noticed. To deal with this problem, the detecting node uses a different node ID, called *detecting ID*, during the detection. This ID should be recognized as a non-beacon node ID. The detecting node also has all keying materials related to this ID so that it can communicate securely with other beacon nodes as a non-beacon node. To increase the probability of detecting a malicious beacon node, we may allocate multiple detecting IDs as well as the related keying materials to each beacon node. With the help of these detecting IDs, it is very difficult for an attacker to distinguish the requests from detecting beacon nodes and those from non-beacon nodes when sensor nodes are densely deployed. If sensor nodes have certain
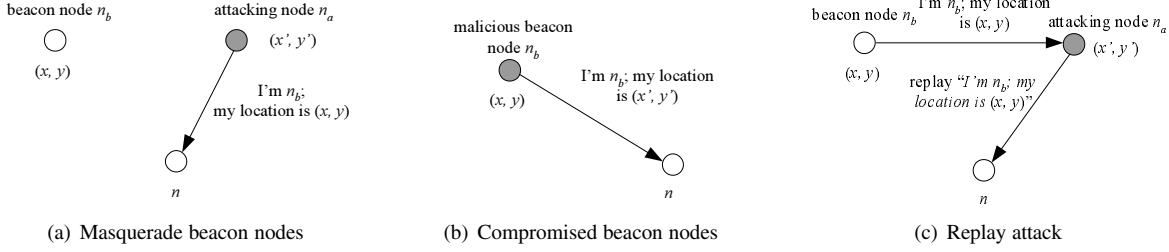
Figure 1. Attacks against location discovery schemes

mobility and/or the detecting node can carefully craft its request message (e.g., adjust the transmission power in RSSI technique), it will become even more difficult for the attacker to determine the source of a request message. For simplicity, we assume that the attacker cannot tell if a request message is from a beacon node or a non-beacon node.

The proposed method works as follows. The detecting node n first sends a request message to the target node n_a as a non-beacon node. Once the target node n_a receives this message, it sends back a beacon packet (beacon signal) that includes its own location (x', y'). The detecting node n then estimates the distance between them from the beacon signal upon receiving it. Since the detecting node n knows its own location, it can also calculate the distance between them based on its own location (x, y) and the target node's location (x', y'). The detecting node n then compares the estimated distance and the calculated one. If the difference between them is larger than the maximum distance error, the detecting node can infer that the received beacon signal must be malicious. Figure 2 illustrates this idea.

$$\left| \sqrt{(x-x')^2 + (y-y')^2} - \text{measured distance} \right|$$

$$> \text{maximum measurement error}$$
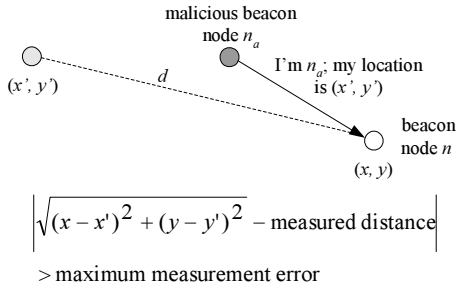
Figure 2. Detect malicious beacon signals

A potential problem in the above method is that even if the calculated distance is consistent with the estimated distance, it is still possible that the beacon signal comes from a compromised beacon node or is replayed by an attacking node. However, a further investigation reveals that this will not generate impact on location estimation. Consider a malicious beacon node that declares a location (x', y'). If the estimated distance from its beacon signal is consistent with the calculated one, it is equivalent to the situation where a benign beacon node located at (x', y') sends a benign beacon signal to the requesting node. In fact, to mislead the location estimation at a non-beacon node, the attacker has to

manipulate its beacon signal and/or beacon packet to make the estimated distance inconsistent with the calculated one. This manipulation will certainly be detected if the requesting node happens to be a detecting node.

2.2 Filtering Replayed Beacon Signals

Suppose a beacon signal from a target node is detected to be malicious, it is still not clear if this node is malicious, since an attacker may replay a previously captured beacon signal. However, if we can determine that a malicious beacon signal indeed comes directly from this target node, this target node must be malicious. Thus, it is necessary to filter out as many replayed beacon signals between benign beacon nodes as possible in the detection.

A beacon signal may be replayed through a *wormhole attack* [13], where an attacker tunnels packets received in one part of the network over a low latency link and replays them in a different part [13]. Wormhole attacks generate big impacts on the security of many protocols (e.g., localization, routing). A number of techniques have been proposed recently to detect such attacks, including geographical leashes [13], temporal leashes [13], and directional antenna [12]. These techniques can be used to filter out beacon signals replayed through a wormhole.

A beacon signal received from a neighbor beacon node may also be replayed by an attacking node. We call such replayed beacon signals *locally replayed beacon signals*. Most of wormhole detectors cannot deal with such attacks, since they can only tell if two nodes are neighbor nodes. It is possible to use temporal leashes [13] to filter out locally replayed beacon signal, since replaying a beacon signal may introduce delay that is detectable with temporal leashes. However, this technique requires a secure and tight time synchronization, and large memory space to store authentication keys. Instead, we study the effectiveness of using round trip time to filter out locally replayed beacon signals, and demonstrate that using round trip time does not require time synchronization method but can detect locally replayed beacon signals effectively.

2.2.1 Replayed Beacon Signals from Wormholes

We assume that there is a wormhole detector installed on every beacon and non-beacon node. This wormhole detec-

tor can tell whether two communicating nodes are neighbor nodes or not with certain accuracy. The purpose of the following method is to filter out the replayed beacon signals due to the wormhole between two benign beacon nodes that are far away from each other. An observation regarding such replayed beacon signals is that the distance between the location of the detecting node and the location contained in the beacon packet is larger than the communication range of the target node. Thus, we combine the wormhole detector with the location information in the following algorithm.

Once a beacon signal is detected to be a malicious beacon signal, the detecting node begins to verify if it is replayed through a wormhole with the help of the wormhole detector. The detecting node first calculates the distance to the target beacon node based on its own location and the location declared in the beacon packet. If the calculated distance is larger than the radio communication range of the target node and the wormhole detector determines that there is a wormhole attack, the beacon signal is considered as a replayed beacon signal and is ignored by the detecting node. Otherwise, the beacon signal will go through the process to filter locally replayed signals in Section 2.2.2.

Let us briefly study the effectiveness of this method. Since a malicious target node can always manipulate its beacon signals to convince the detecting node that there is a wormhole attack and they are far from each other even if they are neighbor nodes, it is possible that the beacon signal from a malicious target node is removed. Fortunately, non-beacon nodes in the network are also equipped with this wormhole detector. This means that a malicious target node cannot convince all detecting nodes that there are wormhole attacks, and at the same time convince all non-beacon nodes that there are no wormhole attacks so that its beacon signals are not removed by non-beacon nodes. This is because a malicious beacon node does not know if a requesting node is a detecting beacon node.

It is also possible that a replayed beacon signal through a wormhole from a benign target node is not removed. The reason is that the wormhole detector cannot guarantee that it can always detect wormhole attacks.

2.2.2 Locally Replayed Beacon Signals

The method to filter out locally replayed beacon signals is based on the observation that the replay of a beacon signal introduces extra delay. In most cases, this delay is large enough to detect whether there is a locally replayed beacon signal through the round trip time (RTT) between two neighbor nodes. In the following, we first investigate the characteristics of RTT between two neighbor sensor nodes in a typical sensor network, and then use this result to filter out locally replayed signals between benign beacon nodes.

To remove the uncertainty introduced by the MAC layer protocol and the processing delay, we measure the RTT in

the following way. As shown in Figure 3, the sender sends a request message to the receiver, and the receiver responds with a reply message. t_1 is the time of finishing sending the first byte of the request from a sender, t_2 is the time of finishing receiving the first byte of this request at a receiver, t_3 is the time of finishing sending the first byte of the reply from the receiver, and t_4 is the time of finishing receiving the first byte of this reply at the original sender. The sender estimates RTT by computing $RTT = (t_4 - t_1) - (t_3 - t_2)$, where t_4 and t_1 are available at the sender, and $t_3 - t_2$ can be obtained from the receiver by exchanging messages.
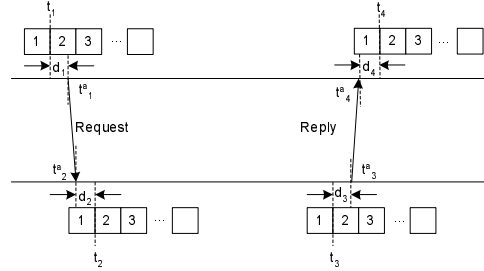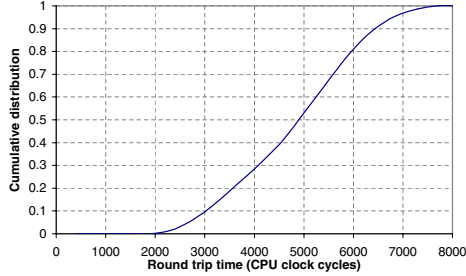

Figure 3. Round trip time

Characteristics of RTT between neighbor nodes: We may perform experiments on actual sensor platform to obtain the characteristics of RTT. To gain further insights and examine our approach, we performed experiments on MICA2 motes [4] running TinyOS [10]. For simplicity, we assume the same type of sensor nodes in the sensor network. Nevertheless, our technique can be easily extended to deal with different types of nodes in the network.

In the experiment, t_1 is measured by recording the time right after the communication module (CC1000) moves the second byte of the request message to the SPDR register, which is used to store the byte being transmitted over the radio channel. In other words, t_1 is the time of finishing shifting the first byte of the request message out of this register. Assume the absolute time of finishing sending the first byte of the request message is t_1^a. We have $t_1 + d_1 = t_1^a$, where d_1 is the delay between shifting the data byte out of the SPDR register and finishing sending this byte over the radio channel. Similarly, we have $t_3 + d_3 = t_3^a$. Similarly, t_2 is measured by recording the time right after the first byte of the request message is ready at the SPDR register. Assume the absolute time of finishing receiving this byte from the radio channel is t_2^a. We have $t_2 = t_2^a + d_2$, where d_2 is the delay between receiving this byte from the radio channel and reading this byte from the SPDR register. Similarly, we have $t_4 = t_4^a + d_4$. Since the radio signal travels at the speed of light, we have $t_4^a - t_1^a - (t_3^a - t_2^a) = \frac{2D}{c}$, where D is the distance between two neighbor nodes and c is the speed of light. Thus, we have $RTT = d_1 + d_2 + d_3 + d_4 + \frac{2D}{c}$.

Note that d_1, d_2, d_3 and d_4 are mainly affected by the underlying radio communication hardware. Since two neighbor nodes are usually close to each other, the value of $\frac{2D}{c}$ in

the above equation is negligible. Hence, the RTT measured by computing $RTT = (t_4 - t_1) - (t_3 - t_2)$ is not affected by the MAC protocol or any processing delay. This means that the distribution of RTT should be within a narrow range. Let F denotes the cumulative distribution function of RTT when there are no replay attacks, x_{min} denotes the maximum value of x such that $F(x) = 0$, and x_{max} denotes the minimum value of x such that $F(x) = 1$.

Figure 4. Cumulative distribution of round trip time

Figure 4 shows the cumulative distribution of RTT when there are no replay attacks. We use one CPU clock cycle as the basic unit to measure the time. This figure is derived by measuring RTT $100,000$ times. The result shows that $x_{min} = 1,951$ and $x_{max} = 7,506$. Since the transmission time of one bit is about 384 clock cycles, we can detect any replayed signal if the delay introduced by this replay is longer than the transmission time of $\frac{7506-1951}{384} \approx 14.5$ bits.

The detector for locally replayed beacon signals: With RTT's cumulative distribution, we can detect locally replayed signals between benign beacon nodes effectively. The basic idea is to check if there is any significant difference between the observed RTT and the range of RTT derived during our experiments. For example, if the observed RTT at the requesting node is larger than the maximum RTT in Figure 4, it is very likely that the reply signal is replayed. The following local replay detector will be installed on every beacon and non-beacon node.

The requesting node u communicates with a beacon node v following the request-reply protocol shown in Figure 3. As a result, node u can compute $RTT = (t_4 - t_1) - (t_3 - t_2)$. There are two cases: (1) When $RTT \leq x_{max}$, the beacon signal is considered as not locally replayed. If the requesting node is a detecting node, it will report an alert when the beacon signal is detected to be malicious. If the requesting node is a non-beacon node, this beacon signal will be used in its location estimation. (2) When $RTT > x_{max}$, this beacon signal is considered as locally replayed, and will be ignored by the requesting node.

When the target node is a benign beacon node and is a neighbor of the detecting node, but the beacon signal is replayed by a malicious node, the detecting node will report an alert if the delay introduced by the locally replayed signal is less than the transmission time of 14.5 bits data. However,

it is very difficult for the attacker to achieve, since the attacker has to replay the beacon signal to the detecting node when the target node is still sending its beacon signal. This implies that the attacker has to physically shield every signal to the detecting node and replay the intercepted packet at the same time. When the target benign beacon node is not a neighbor node of the detecting node, the detecting node will report an alert if the delay introduced by the undetected wormhole attack is less than the transmission time of 14.5 bits data. Note that this implies this replayed beacon signal has bypassed the wormhole detector.

Note that the purpose of the above method is to filter the replayed beacon signals between benign beacon nodes to avoid false positives. This method becomes trivial when the target node is a malicious beacon node, since it can easily convince a detecting node that the beacon signal is locally replayed and thus prevent the detecting node from reporting an alert. However, the malicious target node cannot convince all detecting nodes that the beacon signals are locally replayed, and at the same time convince all non-beacon nodes that its beacon signals are not locally replayed so that its beacon signals are accepted by non-beacon nodes.

2.3 Analysis

Theoretically, the proposed techniques can be used to provide security for any existing localization scheme based on location references from beacon nodes. However, when TDoA technique is used for measuring distances to beacon nodes, the proposed techniques do not work as effective as in other techniques (e.g., RSSI, ToA, and AoA), since it is usually more difficult to protect ultrasound signals, especially when ultrasound signals cannot carry data packets.

In some cases, a non-beacon node may become a beacon node to supply location references once it discovers its own location. Localization error may accumulate when more and more non-beacon nodes turn into beacon nodes. However, there are still constraints between estimated measurements and calculated measurements; otherwise, it is impossible to estimate locations with required accuracy. With these constraints, we can still apply the proposed detector to catch possible malicious beacon nodes, though the specific solutions need further investigation.

Overheads: Since beacon signals are unicasted from beacon nodes to their neighbor non-beacon nodes, our techniques sacrifice certain amount of communication overhead for security. This trade-off is practical, since location estimation only needs to be done once for each non-beacon node in most cases, and a sensor node (beacon or non-beacon node) usually only needs to communicate with a few other nodes within its communication range. The computation and storage overheads are mainly introduced by key establishment protocol and cryptographic operations.

False positives: Our techniques cannot prevent a ma-

licious detecting node from reporting alerts against other beacon nodes. The techniques are aimed at reducing the probability of a benign beacon node reporting alerts against other benign beacon nodes and increasing the probability of a benign beacon node reporting alerts against malicious beacon nodes.

For simplicity, we assume that when a node A is sending a beacon signal to its neighbor node B during time period T, node B either receives the original signal or receives nothing (in case of collision) at the end of T. Thus, the delay of replaying a signal between two neighbor nodes is at least the transmission time of one entire packet, which is typically much larger than 14.5 bits. This means that our detector can always detect locally replayed beacon signals between two benign neighbor nodes. Hence, the situation where a benign beacon node reports an alert against another benign beacon node only happens when (1) they are not neighbor nodes, (2) the attacker creates a wormhole between them, (3) this wormhole cannot be detected by the detecting node, and (4) the delay introduced by this wormhole is less than the transmission time of 14.5 bits. Assume the detection rate of the wormhole detector is p_d. The probability that a replayed beacon signal through a wormhole from a benign beacon node is not removed can be estimated by $1 - p_d$. Thus, the probability of a benign beacon node reporting an alert on another benign beacon node is at most $1 - p_d$ if there is a wormhole between them, and 0 otherwise.

Detection rate (P_r): The detection rate, which is the probability of a malicious target node being detected by a detecting node, is an important metric to evaluate the performance of our detector. Assume a malicious beacon node u sends normal beacon signals to a fraction p_n of the requesting nodes, convinces a fraction p_w of requesting nodes that its beacon signal are replayed from wormholes, and convince a fraction p_l of requesting nodes that its beacon signals are locally replayed. We also assume the malicious beacon node u behaves in the same way for the same requesting node, which is the best strategy for the node to avoid being detected. Thus, using one detecting ID, a benign detecting node v that hears beacon signals from this malicious node u will detect malicious beacon signals with a probability of $1 - p_n$. If a malicious beacon signal from malicious node u is detected, the probability of going through the process of filtering locally replayed beacon signals is $1 - p_w$. During the process of filtering locally replayed beacon signals, the probability of node v reporting an alert against malicious node u is $1 - p_l$. Hence, the probability of malicious node u being detected by node v can be estimated by $(1-p_n)(1-p_w)(1-p_l)$. When each detecting node has m detecting IDs. The probability P_r of a malicious beacon node being detected by a benign detecting node can be estimated by $P_r = 1 - (1 - (1-p_n)(1-p_w)(1-p_l))^m$.

We denote P as the probability that (1) a requesting non-

beacon node receives a malicious beacon signal from a malicious beacon node, and (2) this malicious beacon signal is not removed by the replay detector. For a requesting non-beacon node w, the probability of hearing malicious beacon signals from node u is $(1 - p_n)$. If w receives a malicious beacon signal, the probability of going through the detection of locally replayed signals is $1 - p_w$. During the detection of locally replayed signals, the probability of this malicious beacon signal not being filtered out is $1 - p_l$. Since the above three events are independent from each other, the probability P can be estimated by $P = (1-p_n)(1-p_w)(1-p_l)$. Thus, we have $P_r = 1-(1-P)^m$.
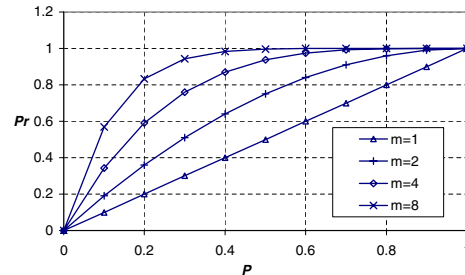

Figure 5. Relationship between P_r and P.

Figure 5 shows the relationship between the detection rate P_r and P. It indicates that an attacker cannot increase P without increasing the probability of being detected. On the other hand, a benign detecting node can always increase m to have higher detection rate P_r.

3 Revoking Malicious Beacon Nodes

With the detector in the previous section, a detecting beacon node may report alerts about other suspicious beacon nodes. In this section, we propose to use the base station to further remove malicious beacon nodes from the network to reduce their impact on the location discovery service. We assume that the base station has mechanisms to revoke malicious beacon nodes when it determines what nodes to remove.

3.1 The Revocation Scheme

We assume each beacon node shares a unique random key with the base station. With this key, a beacon node can report its detecting results securely to the base station.

The basic idea is to evaluate the suspiciousness of each beacon node based on the alerts from detecting nodes. The beacon nodes with high degree of suspiciousness will be considered as being compromised. We measure the *suspiciousness* of a beacon node with the number of alerts against this beacon node. Since malicious beacon nodes may report many alerts against benign beacon nodes, we limit the number of alerts each beacon node can report to mitigate this effect. The detail of the algorithm is described below.

Every alert from a detecting node includes the ID of the detecting node and the ID of the target node. The base sta-

tion maintains an *alert counter* and a *report counter* for each beacon node. The alert counter records the suspiciousness of this beacon node, while the report counter records the number of alerts this node reported and accepted by the base station. Whenever a detecting node determines that a particular beacon node is compromised, it reports an alert to the base station. Once the base station receives the alert, it checks if the report counter of the detecting node has not exceed a threshold τ' and the target node is not revoked. If this is true, it increases both the alert counter of the target node and the report counter of the detecting node by 1; otherwise, the base station ignores this alert. The base station then checks if the alert counter of the target node exceeds another threshold τ. If yes, the target node is considered as a malicious beacon node and revoked from the network.

Note that the alert from a revoked detecting node will still be accepted by the base station if its report counter does not exceed threshold τ' and the target node is not revoked. The purpose is to prevent malicious beacon nodes from reporting a lot of alerts against benign beacon nodes and having these benign beacon nodes revoked before they can report any alert.

3.2 Analysis

For simplicity, we assume beacon nodes and non-beacon nodes in the network are randomly deployed in the field. We assume there are N sensor nodes, N_b beacon nodes, and N_a malicious beacon nodes in the network. Thus, there are $N - N_b$ non-beacon nodes and $N_b - N_a$ benign beacon nodes. We assume malicious beacon nodes do not report alerts against other malicious beacon nodes, since this will increase the probability of a malicious beacon node being detected. We also assume that every alert from beacon nodes can be successfully delivered to the base station using some standard fault tolerant techniques (e.g., retransmission) when there are message losses. When it is necessary to evaluate certain aspect with specific numbers (e.g., in figures), we always assume 10% of sensor nodes are benign beacon nodes ($\frac{N_b - N_a}{N} = 0.1$).

Overheads: The revocation scheme requires beacon nodes to report theirs observations to the base station, which introduces additional communication overhead. However, a beacon node usually only needs to monitor a few number of other beacon nodes that it can communicate with. Thus, only a limited number of alerts need to be delivered to the base station. There are no additional computation overhead and storage overhead introduced by the above algorithm for the beacon nodes in the network. For the base station, it is usually not a problem to run the above algorithm, since the base station is much more resourceful than a beacon node.

Detection rate (P_d): The detection rate studied here is the probability of a malicious beacon node being revoked by the base station. Consider any requesting node u of a par-

(a) m=1

(b) $\tau=4$
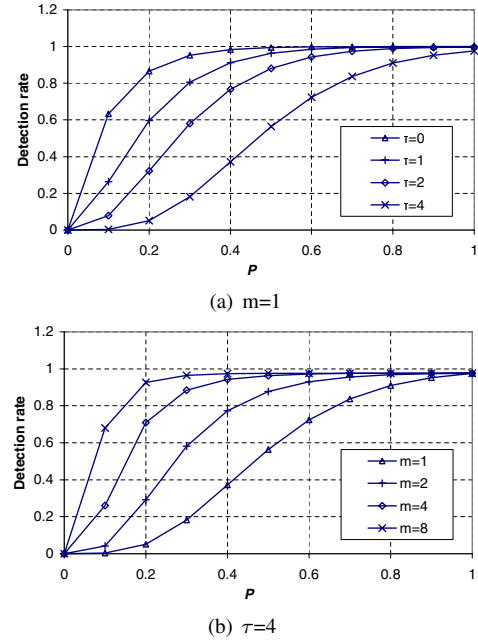
Figure 6. Detection rate v.s. probability of non-beacon nodes being affected. $N_c = 100$.

ticular malicious beacon node v. The probability that u is a benign beacon node can be estimated by $\frac{N_b - N_a}{N}$. If node u is a benign beacon node, the probability of reporting an alert is P_r. Hence, for any requesting node, the probability of the base station having an alert reported against the malicious beacon node v can be estimated by $P_a = \frac{(N_b - N_a) \times P_r}{N}$. Suppose there are N_c requesting nodes for node v. The probability of having exactly i alerts reported can be estimated by $P(i) = \frac{N_c!}{(N_c-i)!i!}P_a^i(1 - P_a)^{N_c-i}$. Assume the threshold τ' is large enough so that an alert from a detecting node will not be ignored by the base station simply because its report counter exceeds τ'. (The method to determine τ' will be discussed later.) The probability of the number of alerts against the malicious beacon node v exceeding τ can be estimated by $P_d = 1 - \sum_{i=0}^{\tau} P(i)$, which is the expected detection rate.

Figure 6(a) and Figure 6(b) illustrate the effect of m, τ and P on the detection rate, assuming $N_c = 100$. We can see that the detection rate increases quickly when a malicious beacon node behaves maliciously more often (a larger P). In addition, the detection rate decreases with a larger threshold τ, since we need more alerts to revoke a malicious beacon node. Finally, the detection rate also increases with more detecting IDs at each beacon node, since each detecting node has more chances to detect a malicious beacon node and report an alert.

Figure 7 shows the effect of N_c on the detection rate, assuming $m = 8$ and $\tau = 2$. We can see that the detection rate increases when more requesting nodes contact a ma-

licious beacon node. This is because the more requesting nodes contact a malicious beacon node, the more alerts are reported.
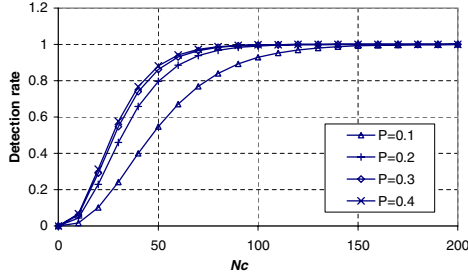

Figure 7. Detection rate. $m = 8$ **and** $\tau = 2$.

Average number of affected non-beacon nodes (N'): An important target of attacks is to mislead the location estimate at as many non-beacon nodes as possible. Thus, it is necessary to study the average number of non-beacon nodes that are really affected by malicious beacon nodes. We assume that a malicious beacon signal will not be used in the location estimation if the corresponding beacon node is revoked. This can be achieved by using some standard fault tolerance techniques (e.g., retransmission) so that the revocation message from the base station can reach most of sensor nodes.

After all detected malicious beacon nodes are revoked, the probability of a non-beacon requesting node accepting the malicious beacon signal from a malicious beacon node can be estimated by $P' = P \times (1 - P_d)$. Thus, the average number of non-beacon nodes that have been really affected can be estimated by $N' = \frac{P' \times N_c \times (N - N_b)}{N}$. Since τ and m are system parameters, the attacker may adjust P to maximize P' and thus N'. (Note that the attacker is able to control P.) Figure 8 shows that in practice, there are only a few non-beacon nodes accepting the malicious beacon signals. It also shows that N' as well as P' increases with a larger τ, and decreases with a larger m. This is because a malicious beacon node has a higher chance to be detected with a larger m, and a higher chance not to be revoked with a larger τ.

Figure 9 shows the relationship between N' and P when the attacker can always choose P to maximize N'. We can see that N' increases dramatically at the beginning. However, when N_c reaches a certain point (about 20), N' begins to drop quickly and finally remains at certain level. This is because after the number of request nodes reaches a certain point, a malicious beacon node has a higher chance of being revoked from the network if it is contacted by more requesting nodes. We also note that N' decreases when threshold τ decreases. This is because the probability of a malicious beacon node being revoked increases with a smaller τ.

Number of false positives (N_f): Assume there are wormhole attacks between N_w pairs of benign beacon nodes in the network. For any wormhole created between
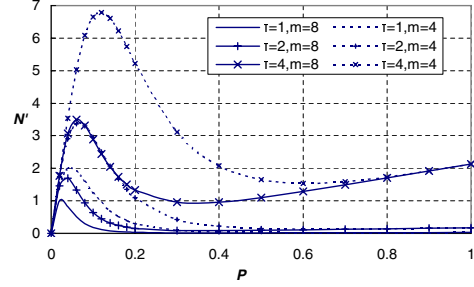

Figure 8. Average number of affected non-beacon nodes after all detected malicious beacon nodes are revoked from the network. $m = 8$ **and** $N_c = 100$**.**
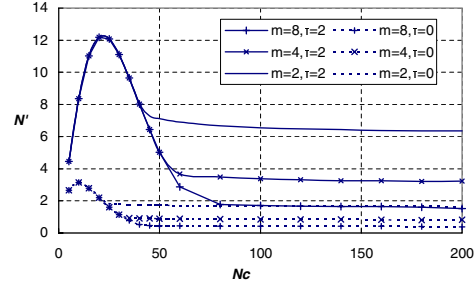

Figure 9. Average number of affected non-beacon nodes when P **is chosen in such a way that** P' **is maximized.**

two benign beacon nodes, the probability of one of them reporting an alert against the other is $1 - p_d$, where p_d is the wormhole detection rate. Thus, on average, there are $2(1 - p_d)N_w$ alerts reported between benign beacon nodes. We consider the worst case where each beacon node reports $\tau' + 1$ alerts. Thus, the total number of alerts against benign beacon nodes can be estimated by $2(1 - p_d)N_w + N_a(\tau' + 1)$, and the average number of benign beacon nodes revoked by the base station (the number of false positives) is at most $N_f = \frac{2(1 - p_d)N_w + N_a(\tau' + 1)}{\tau + 1}$.

According to the above equation, we note that the number of false positives depends on N_w, N_a, and the two thresholds. Thus, to reduce the number of false positives, we have to decrease τ' and/or increase τ. However, decreasing τ' implies less number of alerts against a malicious node, while increasing τ imples more alerts needed to revoke a malicious node. Both of these two options will decrease the probability of malicious beacon nodes being detected. In practice, we have to make trade-offs between the number of false positives and the detection rate. The next part of the analysis will show a possible way to deal with this problem.

Thresholds τ **and** τ': Thresholds τ and τ' are two critical parameters. Threshold τ can be configured according to similar constraints in Figure 9. Intuitively, we may derive the relationship between N' and N_c as shown in Figure 9

given expected values of N, N_b, N_a, p_d and m. We can then choose a set of τ that make the maximum number of affected non-beacon nodes remain under a given value.

For each of the selected thresholds τ, we configure threshold τ' in the following way so that most of the alerts from benign beacon nodes will not be ignored by the base station simply because their report counters exceed τ'.

We assume malicious nodes are also randomly deployed in the network. Consider a particular benign beacon node u. The probability of a particular malicious beacon node v being contacted by node u can be estimated by $\frac{N_c}{N}$. Since the probability of node u reporting an alert against node v is P_r, and the probability of node v having not been revoked can be approximately estimated by $1 - P_d$, the probability of the report counter of node u being increased by 1 for node v can be estimated by $P_1 = \frac{P_r \times N_c \times (1 - P_d)}{N}$ if this report counter has not exceeded τ' yet. In addition, the probability of a particular wormhole being created for node u can be estimated by $\frac{2}{N_b - N_a}$, and the probability of node u reporting an alert due to this wormhole can be estimated by $1 - p_d$. Since the probability of the node at the other side of the wormhole is revoked can be approximately estimated by $\frac{N_f}{N_b - N_a}$, the probability of the report counter of node u being increased by 1 due to the wormhole attack can be estimated by $P_2 = \frac{2(1 - p_d)(N_b - N_a - N_f)}{(N_b - N_a)^2}$ if this report counter has not exceeded τ' yet. Hence, the probability that the report counter of node u is i ($i \leq \tau'$) can be estimated by

$$P'(i) = \sum_{j+k=i} \frac{N_a! N_w! P_1^j (1 - P_1)^{N_a - j} P_2^k (1 - P_2)^{N_w - k}}{(N_a - j)! j! (N_w - k)! k!}.$$

Therefore, the probability of the report counter of a benign node exceeding τ' can be estimated by $P_o = 1 - \sum_{i=0}^{\tau'} P'(i)$. Figure 10 plots this probability when $\tau = 2$, assuming $N = 10,000, N_b = 1100, N_a = 100, N_w = 100,$ $p_d = 0.9, m = 8,$ and $P = 0.1$. We can see that the probability of the report counter of a benign beacon node exceeding 2 is close to zero. Thus, we can chose $\tau' = 2$ and have a pair of candidate thresholds ($\tau = 2, \tau' = 2$). We also note that malicious beacon nodes cannot increase this probability by simply having more requesting nodes contact it, since this will increase the chance of being revoked.

After the above analysis, we can find a proper threshold τ' for each selected τ. We then choose a pair of thresholds τ and τ' that satisfy the constraints on the number of false positives N_f or simply choose a pair of thresholds that lead to the minimum N_f given certain p_d, N_w and N_a.

4 Simulation Evaluation

We have implemented the proposed techniques on TinyOS [10], an operation system for networked sensors. In this section, we present the simulation results
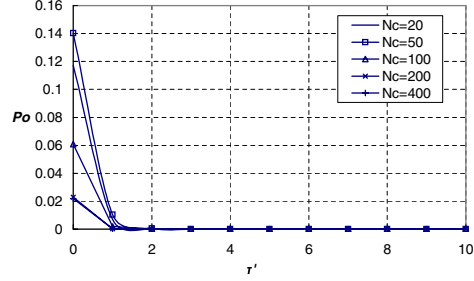

Figure 10. Probability of the report counter of a benign beacon node exceeding τ'. Assume $N = 10,000$**,** $N_b = 1100$**,** $N_a = 100$**,** $N_w = 100$**,** $p_d = 0.9$**,** $\tau = 2$**,** $m = 8$**, and** $P = 0.1$**.**

obtained through the TinyOS simulator Nido, with a focus on the detection rate and the false positive rate (i.e., $\frac{\#incorrect\ revoked\ beacons}{\#total\ benign\ beacons}$) of the proposed schemes.

We assume 1,000 sensor nodes ($N = 1000$) randomly deployed in a sensing field of 1000×1000 square feet. Among these sensor nodes, there are 100 beacon nodes ($N_b = 100$) with 10 compromised beacon nodes ($N_a = 10$). Figure 11 shows the randomly generated deployment used in our simulation, where each blank circle ($\circ$) represents a benign beacon node and each solid circle ($\bullet$) represents a malicious beacon node. We assume the maximum communication range of a beacon or non-beacon node is 150 feet, and a malicious beacon node only contacts the nodes within its communication range.

In the simulation there is a wormhole between location A (100,200) and location B (800,700), which forwards every message received at one side immediately to the other side. We assume malicious beacon nodes collude together to report alerts against benign beacon nodes. Thus, they can always make the base station revoke about $\frac{N_a \times (\tau'+1)}{\tau+1}$ benign beacon nodes by simply reporting alerts. We always assume $m = 8$ and $p_d = 0.9$. We also assume that there is a technique (e.g., RSSI) used to estimate the distance to the beacon node that has the maximum error of 10 feet.
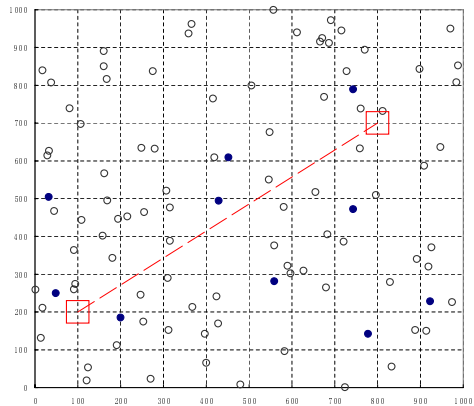

Figure 11. Deployment of beacon nodes in a sensing field.

Figure 12 shows the detection rate when $\tau' = 2$ and $\tau = 2$. The result conforms to the theoretical analysis. We can clearly see the increase in the detection rate when a malicious beacon node tries to increase P to affect more non-beacon nodes. Figure 13 shows the average number N' of requesting non-beacon nodes accepting the malicious beacon signals from a malicious beacon node. We note that the simulation result has observable but small difference from the theoretical analysis. The simulation result and the theoretical result are in general close to each other.
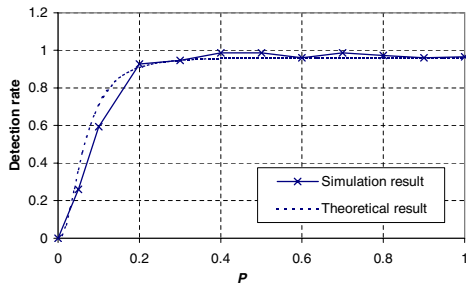

Figure 12. Detection rate v.s. P. Assume $\tau' = 2$ and $\tau = 2$.
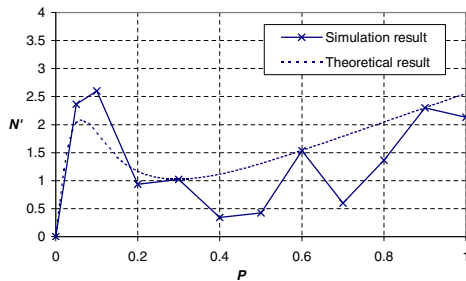
Figure 13. Average number of requesting non-beacon node accepting the malicious beacon signals from a malicious beacon node. Assume $\tau' = 2$ and $\tau = 2$

Based on our earlier analysis, both the detection rate and the false positive rate are affected by τ and τ' given certain p_d, N_w and N_a. Figure 14 shows the ROC (Receiver Operating Characteristic) curves for the proposed techniques under different choice of τ and τ', assuming P is configured in such a way that N' is maximized. (The various points in the figure are obtained by using different values of τ.) It includes the performance when there are either 5 ($N_a = 5$) or 10 ($N_a = 10$) compromised beacon nodes. We can see that our technique can detect most of malicious beacon nodes with small false positive rate (e.g., 5%) when there are a small number of compromised beacon nodes. However, when the number of compromised beacon nodes increases, the performance decreases accordingly. For example, when there are 10 malicious beacon nodes, the false positive rate will reach 20% in order to detect most of malicious beacon nodes. Nevertheless, the figure still shows that our techniques are practical and effective in detecting

malicious beacon nodes. In addition, this figure also gives a way to set τ and τ' to meet the security requirement of different applications.
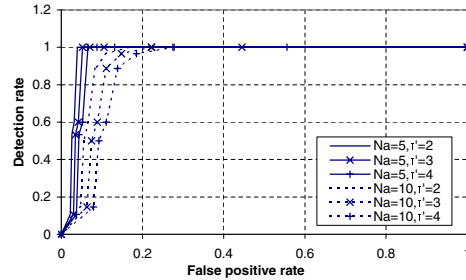
Figure 14. ROC curves. Assume P is chosen to maximize N'.

5 Related Work

Savvides et al. developed AHLoS localization protocol based on Time Difference of Arrive [27] and further extended it in [28]. Doherty et al. presented a localization scheme using connectivity-induced constraints and relative signal angles between neighbor nodes [5]. Angle of Arrive is used to develop localization scheme in [22] and [19]. Bulusu, Heidemann and Estrin proposed a coarse-grained localization scheme by centering the locations contained in the received beacon signals [2]. Niculescu and Nath proposed to use the minimum hop count and the average hop size to estimate the distance between nodes and then determine sensor nodes' locations accordingly [23]. None of these schemes will work properly when there are malicious attacks against location discovery. A secure range-free localization technique was recently developed in [16]. However, it cannot detect and remove compromised beacon nodes. The techniques proposed in this paper can protect localization scheme by detecting compromised beacon nodes.

Security in sensor networks has attracted a lot of attention recently. To provide practical key management for sensor networks, many key pre-distribution techniques have been developed [3, 6, 7, 17]. μTESLA was proposed to enable broadcast authentication in sensor networks [24]. Security of sensor data has been studied in [11, 25]. A thorough analysis of attacks against routing protocols in sensor networks and possible counter measures were given in [14]. The research in this paper addresses another fundamental security problem that has not drawn enough attention.

6 Conclusion and Future Work

This paper presented a practical method to detect malicious beacon nodes in order to protect location discovery services in sensor networks. We developed a simple but effective method to detect malicious beacon signals, and then investigated techniques to detect replayed beacon signals to avoid false positives. We then developed a method

for the base station to reason about the suspiciousness of beacon nodes and revoke the malicious beacon nodes accordingly. The analysis and simulation indicate that these techniques are practical and effective in detecting malicious beacon nodes.

Several future directions are worth investigating. First, we will look for other effective ways to reduce the false alert rate. Second, it is particularly interesting to investigate distributed algorithms to revoke malicious beacon nodes without using the base station.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, 2002.

[2] 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.

[3] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Research in Security and Privacy*, pages 197–213, 2003.

[4] Crossbow Technology Inc. Wireless sensor networks. http://www.xbow.com/Products/Wireless_Sensor_Networks.htm. Accessed in February 2004.

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

[6] W. Du, J. Deng, Y. S. Han, and P. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*, pages 42–51, October 2003.

[7] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 41–47, November 2002.

[8] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler. The nesC language: A holistic approach to networked embedded systems. In *Proceedings of Programming Language Design and Implementation (PLDI 2003)*, June 2003.

[9] 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 ACM MobiCom 2003*, 2003.

[10] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. S. J. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000.

[11] L. Hu and D. Evans. Secure aggregation for wireless networks. In *Workshop on Security and Assurance in Ad Hoc Networks*, January 2003.

[12] L. Hu and D. Evans. Using directional antennas to prevent wormhole attacks. In *Proceedings of the 11th Network and Distributed System Security Symposium*, pages 131–141, February 2003.

[13] Y. Hu, A. Perrig, and D. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In *Proceedings of INFOCOM 2003*, April 2003.

[14] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *Proceedings of 1st IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

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

[16] 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.

[17] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. In *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*, pages 52–61, October 2003.

[18] 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.

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

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

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

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

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

[24] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar. SPINS: Security protocols for sensor networks. In *Proceedings of Seventh Annual International Conference on Mobile Computing and Networks*, July 2001.

[25] B. Przydatek, D. Song, and A. Perrig. SIA: Secure information aggregation in sensor networks. In *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys '03)*, Nov 2003.

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

[27] 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.

[28] 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.