LocJam: A Novel Jamming-based Approach to Secure Localization in Wireless Networks

Arash Boustani, Navid Alamatsaz, Murtuza Jadliwala and Vinod Namboodiri
Wichita State University, Wichita, Kansas 67260, USA
Email: {axboustani,nxalamatsaz,murtuza.jadliwala,vinod.namboodiri}@wichita.edu

***Abstract*—Location discovery is an essential service in modern wireless consumer devices such as smartphones and mobile PCs. Existing anchor or base station-based positioning systems work well in non-hostile scenarios, but their accuracy suffers in the presence of cheating anchors. Securing location discovery in these positioning systems is an important, and still open, research problem. Earlier research efforts in this direction have mainly focused on either efficient detection and elimination of cheating anchors or on localization in the presence of cheating anchors. Proposals on localization in the presence of cheating anchors fail to perform well in the presence of a large number of cheating anchors, whereas, the issue of elimination of cheating anchors (once detected) has not been clearly addressed in techniques that focus on detection and elimination of cheating anchors. In this paper, we present a novel and deterministic strategy for securing anchor-based location discovery. Our technique employs a novel *CDMA-based jamming* strategy to eliminate (the effect of) cheating anchors during localization. We validate the performance of our proposal under various adversarial strengths and operating scenarios by means of extensive simulations.**

I. Introduction

Location discovery is one of the most important services on modern consumer wireless devices. This service enables a wireless device in a wireless Wide Area Network (WWAN) or in a wireless Local Area Network (WLAN) to determine its own location with respect to some local or global coordinate system. Range or distance-based localization techniques, where a device computes its location by estimating distances to other devices, can be further classified as (a) *anchor-based* or (b) *anchor-free* systems [1]. Anchor-based systems [2]–[9] employ special anchors (or base stations) that are strategically placed in the network and know their own position. A popular example of an anchor-based positioning system, available on most modern consumer wireless devices, is the Global Positioning System (GPS). In anchor-based systems, the mobile target device first estimates its distance to a set of anchors (within its radio range) by using well-known techniques such as *Received Signal Strength Indicator (RSSI)* [10], *Time of Arrival (ToA)* [11], and *Time Difference of Arrival (TDoA)* [12]. The target device then applies a constraint satisfaction or optimization procedure, such as trilateration or multilateration, to compute its location (Figure 1(a)). Anchor-free schemes do not involve specifically marked anchor nodes.

Although anchor-based schemes generally perform well, these techniques operate under the assumption that anchor nodes behave honestly during the localization process. This assumption is not valid in hostile wireless environments where

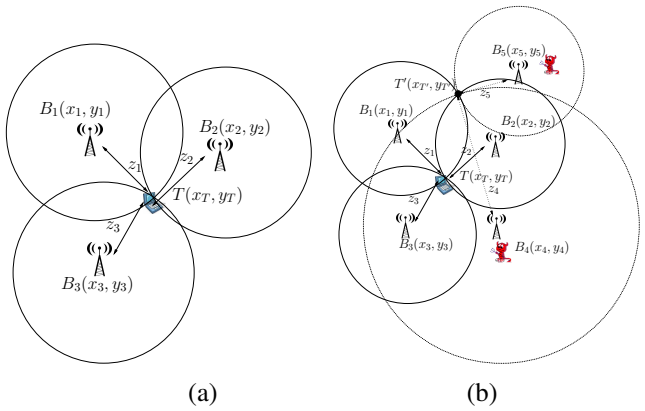


Fig. 1. Distance-based localization (a) Trilateration (b) Cheating anchors

anchors could cheat by manipulating the distance estimation process, thus affecting the accuracy of the location estimated by the target node (Figure 1(b)). As a matter of fact, attacks [13], [14] on popular WLAN-based positioning systems such as Skyhook (used in Apple iPod and iPhone) and satellite-based systems such as GPS have been demonstrated. Numerous proposals for overcoming the problem of cheating in range-based localization protocols exist in the literature [15]–[25]. These proposals have primarily followed one of the following two approaches. The first approach is to localize in the presence of cheating anchor nodes and securely verify that the estimated location is within some maximum error bound. The second approach relies on efficiently detecting and eliminating measurements emanating from cheating anchors. Localization schemes following the first approach often tolerate only a fixed number of malicious anchors, given a set of honest anchors, and could result in relatively larger localization errors. Schemes following the second approach suffer due to the non-triviality of the detection and elimination process of malicious anchors in a distributed network setting.

We are motivated by the fact that radio signal jamming has traditionally been always considered as an adversarial tool used to disrupt network protocols. We would like to pursue a reverse ideology here and use jamming to protect network protocols such as location discovery. In this paper, we propose a fresh approach to overcome the problem of cheating in anchor-based localization systems. Our proposal comprises of two strategies: (i) an asynchronous "request-confusion" strategy to anonymize localization requests from

the target mobile device, which enables efficient detection of cheating anchors, (ii) a Direct-Sequence Spread Spectrum (DSSS) or Code Division Multiple Access (CDMA)-based jamming strategy to eliminate measurements from cheating anchors during the location determination stage. Spread spectrum communications using Orthogonal Chip Sequences (OCS) has several advantages in wireless communications, including higher number of simultaneous transmissions and lower interference [26]. Our jamming strategy takes advantage of these properties to not only eliminate information from cheating anchors, but to also simultaneously enable honest anchors to correctly transmit valid location information to the target device. Other CDMA-based localization approaches in the literature work only in a non-hostile setting [27], [28]. To the best of our knowledge, this is the first proposal that considers a CDMA-based communication and jamming strategy to secure anchor-based localization in wireless networks.

II. Background and Related Work

A. Detection and Elimination of Malicious Anchors

The first approach for securing distance-based localization is to detect cheating anchors and eliminate them from consideration. Liu et al. [16] propose a Minimum Mean Square Estimation (MMSE) technique for eliminating malicious anchor data. Sastry et al. [25] propose a location verification protocol to securely verify location claims by computing the relative distance between the prover and the verifier node using the time of propagation of ultrasound signals. Čapkun et al. [19] outline various attacks on node localization and propose mechanisms such as authenticated distance estimation, authenticated distance bounding, verifiable trilateration and verifiable time difference of arrival, in order to detect cheating anchors. Pires et al. [18] propose an approach to detect those message transmissions whose signal strength is incompatible with its originator's position. Liu et al. [17] employ special detector anchors to detect malicious anchors.

B. Secure Localization in the presence of Malicious Anchors

The second approach is to design techniques that are robust against cheating by malicious anchors. Priyantha et al. [4] eliminate the dependence on anchors by using communication hops to estimate the network's global layout, and then apply force-based relaxation to optimize this layout. Li et al. [15] utilize Adaptive Least Squares and Least Median Squares methods to make anchor-based localization attack-tolerant. Doherty et al. [29] employ convex optimization on a set of connectivity constraints to secure range-based localization. Liu et al. [16] propose an intelligent voting-based scheme for resisting cheating anchors during localization. In another approach, Yi et al. [30] and Ji et al. [31] apply data analysis techniques such as *Multi-Dimensional Scaling (MDS)* to connectivity and distance information in order to infer target locations. Fang et al. [32] use Maximum Likelihood Estimation (MLE) to estimate the most probable node location, given a set of neighborhood observations. Lazos et al. [33] propose a robust location computation and verification approach that does not require centralized management and is robust against jamming by malicious anchors. Misra et al. [20] propose a convex optimization-based scheme to secure distance-based localization. Jadliwala et al. [24] outline a class of algorithms that bound the localization error under cheating.

C. Localization using Coding Theory

Concepts from coding theory have also been used to secure distributed range-based localization. Ray et al. [34] use Identifying Codes (ID-Codes), whereas, Yedavalli et al. [35] use Error Correcting Codes (ECC) for robust localization in wireless sensor networks. Cao et al. [28] outline a CDMA based technique for mobile location discovery. The authors showed that the use of OCS for localization helps to cancel the interference at the mobile target caused by simultaneous transmission of the anchors. However, they do not address the problem of secure localization.

D. Discussion and Motivation

In order to detect and eliminate malicious anchors, approaches outlined in II-A consider inconsistencies in the network measurements caused by cheating. One shortcoming of most of these approaches is that the process of elimination of malicious anchors, once detected, is not clear. Others [16] propose simple collaborative passive approaches for detection, for example, voting to blacklist malicious anchors. But these approaches can be easily circumvented, for example, cheating anchors could regularly change identifiers to avoid collaborative passive detection. These detection mechanisms also require a large number of honest anchors, as well as, the ability to coordinate and communicate detection and verification information with other honest anchors. Approaches discussed in II-B attempt to improve the robustness of distance-based localization by minimizing the effect of inconsistent and erroneous location or distance data. Some shortcomings of these solutions include complexity, higher localization errors and/or requirement of specialized hardware.

We overcome these shortcomings by following an *active* approach for detecting and eliminating cheating anchors during distance-based localization. We employ *signal jamming* as a security tool, as opposed to its typical utility as an adversarial tool. Also, our approach does not require consensus building among honest anchors for eliminating cheating anchors. Moreover, a single honest anchor can successfully eliminate the cheating effect of another or a group of malicious anchors, and at the same time provide accurate location data to the target device. Also, as our approach actively eliminates malicious ranging data, the target node does not have to verify and eliminate these, thus improving the overall performance of the localization process.

III. Network Configuration

A. Network and Communication Model

The network consists of a *mobile* wireless device MT, also referred to as the *target node*, moving in a deterministic fashion over a fixed application area. MT wants to

compute its own location by using distance estimates to a set of *neighboring* (in its radio range) anchor nodes who are stationary and know their own position. Moreover, let us assume that a fixed number (specifically, n) of *stationary* anchor nodes that know their own location are uniformly distributed over the application area. Let these nodes be denoted as $B_1, \ldots, B_n$. For simplicity, we assume that the position of the target node MT and the anchor nodes can be expressed in the two-dimensional coordinate system as a vector (x, y), where, $x, y \in \mathbb{R}$. Each of the anchor nodes and the MT possess an omni-directional radio transceiver. We assume that all anchors are (time) synchronized with each other. Synchronization among the static anchors is feasible and is required for multilateration using TDoA.

All communications in the network takes place over two separate channels. The first channel is a *CSMA-based control channel* for sending and receiving certain control messages (among anchors) and the second channel is a *CDMA-based data channel*. Communication over this data channel is used by the MT to estimate distances to the corresponding anchors. The data channel uses a *DSSS* or *Direct Sequence CDMA (DS CDMA)* scheme where the information signal (or data bits) is multiplied (or encoded) with an Orthogonal Chip Sequence or OCS (also called, code) that is known at the receiver. The receiver then uses this same chip or code to correctly recover or decode the information bits. DS CDMA communications can be synchronous or asynchronous; the MT-to-anchor communications are *asynchronous*. Various asynchronous chip sequences or OCSs, e.g., Golay codes, Walsh Hadamard codes and Gold codes, exist in the literature [36]. Here, we use Golay codes due to their good auto/cross correlation or fault-tolerance properties [26]. The recursive generator matrix for Golay codes is shown in Eqn. 1.

$$C_L = \begin{bmatrix} C_{\frac{L}{2}} & \bar{C}_{\frac{L}{2}} \\ C_{\frac{L}{2}} & -\bar{C}_{\frac{L}{2}} \end{bmatrix} \tag{1}$$

$$C_L = [\, A_L \; B_L \,], \bar{C}_L = [\, A_L \; -B_L \,] \text{ and } C_1 = \bar{C}_1 = [\{0, 1\}]$$

Here, $L = 2^M$ is the total number of available OCSs (and also the number of bits in each OCS) and $M \geq 1$ is the number of recursions.

The set of all OCSs of a particular length can be partitioned into disjoint subsets called *flocks*. Specifically, a flock consists of a set of chip sequences with partially similar bit patterns or chips whose hamming distances are within some fixed threshold [26]. Anchors in the network are tessellated or divided into groups (details in III-B). Each anchor group is assigned to a fixed flock at network initialization. We assume that time is divided into periods of random intervals, denoted by a *random variable* ψ. During each period, each group of anchors randomly chooses a different *subset* of OCSs from its assigned flock for use in that period.

A group of anchors assigned to the same flock is also referred to as a *Grid Cell (GC)*. As the number of available OCSs is limited, flocks are reused throughout the network. A collection of neighboring GCs in which the flocks are not reused forms a *cluster*. Each anchor will use the OCS uniquely assigned to it in the time period ψ in order to transmit data to the MT on the CDMA data channel. Thus, it is possible for multiple anchors to use the same OCS for data transmission in different parts (non-adjacent GCs) of the network, henceforth referred to as *Code Reuse Factor (CRF)*. A CRF of $\frac{1}{r}$ indicates that a total of r adjacent GCs in a cluster use different flocks of OCS. This is possible if the OCS generation scheme generates OCSs that fall into r categories, each assigned to a different GC in a cluster of r adjacent GCs.

The OCSs used by an anchor group will be changed at the end of the time period ψ, for example, by a group-head, as discussed later. As the OCSs assigned to anchors in adjacent GCs at any time instant are unique, these OCSs are not only useful in the physical layer for CDMA data transmission, but can also act as unique higher layer *identifiers (ID)* or *pseudonyms* for anchors during that time period. Morover, as the OCS assigned to each anchor changes from one time period to next, anchors remain anonymous across time periods.

We assume that honest anchors are pre-configured with appropriate message authentication (e.g., HMAC) and symmetric encryption mechanisms (e.g., AES) for secure communications amongst each other. All honest anchor nodes in the same GC send and receive data, to each other and not the MT, signed using a group signature [37], [38]. Hence, each anchor is able to authenticate the source of any incoming message as being from the same GC or not. We also assume that during each time period, the table of valid OCSs (for that time duration) is exchanged among anchors in a distributed fashion. This can also be accomplished by a group-head that is selected in each time period for each GC by using an appropriate group-head election algorithm (e.g., voting or token-based). For a particular GC, let us denote the elected group-head during a time period ψ_k as p_{ψ_k}. Group-head selection could also be rotated within the GC for security and energy-efficiency. From all possible OCSs in the flock, the group head randomly chooses a portion of valid OCSs for the GC (as a function of the number of anchors in that GC) and broadcasts the list of valid OCSs (and their mapping to specific anchors) to all other anchors in the GC. Let $F_g(\psi_k)$ denote the subset of the OCS flock used by a GC g during time period ψ_k. The OCS advertisements are encrypted using AES encryption.

Let's focus on the anchor communications over the CDMA data channel. The main concept of CDMA is to spread an information signal with bandwidth δ_s over a larger bandwidth δ, where $\delta \gg \delta_s$ and $\frac{\delta}{\delta_s}$ is the processing gain. This is achieved by encoding each data symbol (or bit) using an OCS of length L. The OCS $O_i(t)$ assigned to any anchor B_i at any time instant t can be represented as:

$$O_i(t) = \sum_{j=0}^{L-1} O_{(j,i)} p(t - jT_c) \tag{2}$$

In Eqn. 2, $p(t)$ is a rectangular pulse which is equal to 1 for $0 \leq t < T_c$ and zero otherwise. T_c is the chip duration of the

OCS and $O_{(j,i)}$ is the j^{th} bit (or chip) of the OCS assigned to the anchor B_i. The signal generated after encoding a data symbol of anchor B_i with the corresponding OCS is given by

$$x_i(t) = f_i \sum_{j=0}^{L-1} O_{(j,i)} p(t - jT_c), 0 \leq t < T_f \quad (3)$$

where, f_i is the data symbol of anchor B_i that needs to be encoded and $T_f = LT_c$ is the duration of the encoded data symbol or data frame. The inner product of the sent data with the OCS is done bit-synchronously. Then, the overall transmitted signal $x(t)$ of all n anchors can be given by:

$$x(t) = \sum_{i=1}^{n} x_i(t) \quad (4)$$

The received signal at the receiver (both MT and anchors) will be decoded using the OCSs available in the receiver's OCS table. In order to corrupt the data or signal encoded with a particular OCS in this overall signal, an information signal (or data) of all 1's encoded with that same OCS should be added to the overall signal. It can be shown that, given this, it will be impossible to decode the data encoded with that OCS (signals encoded with other OCSs could still be decoded). Such a signal is referred by us as the *jamming signal* for that OCS.

B. Network Tessellation

Network tessellation or anchor grouping is required for efficiently assigning OCS flocks to specific anchor groups. There are many centralized and distributed algorithms in the literature for tessellating distributed wireless networks [39]–[41]. A abstract tessellation approach similar to Voronoi diagrams can be used and is shown in Figure 2. After node placement, we begin from a randomly selected and centrally-located initial anchor node. This anchor sends an invitation message with specific fixed signal strength to all neighboring nodes on the control channel. Anchor nodes within the signal range become members of that particular GC. Nodes at which the received signal strength is less than a given threshold, can attempt to create new GCs and continue to recursively tessellate the entire network. After this, each independent GC is assigned a unique GC number.

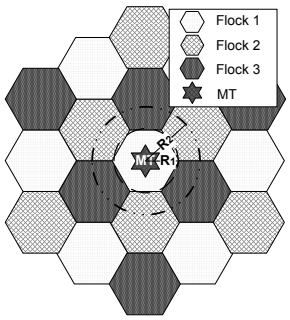


Fig. 2. Network Tessellation

After the tessellation phase, the initial anchor can begin clustering the GCs. This is done by assigning all anchors who received the initial invitation (including the initial anchor), and thus formed a GC, to a flock, say flock 1. All other nodes that received the initial signal with RSS lower than some threshold value, and therefore formed another GC, would be assigned to another flock, say flock 2 or flock 3. Two neighboring GCs can never be in the same flock (or assigned the same flock number). After clustering and flock assignment, all OCSs from all flocks should be added to an OCS table maintained by the MT. We assume that the network is tessellated once at the beginning when the anchors are configured. It can be repeated every time the distribution of the anchors changes significantly.

C. Adversary Model and Localization Attacks

We assume that, amongst a total of n anchors in the network, a maximum of a ($a < n$) anchors are malicious or cheating. The set of all malicious anchors is denoted by $\mathcal{A}$. All anchors that are not malicious are assumed to be honest, i.e., they execute the proposed localization protocol correctly. We assume that in each GC there are at least three honest anchors. We will see later that, although only a single honest anchor is sufficient for jamming all malicious anchors within a GC, at least three honest anchors are required for a successful multilateration operation.

Although many types of attacks in RF-based positioning systems are possible [19], one of the attacks that we focus in this work is the *distance manipulation attack*. In this attack, anchor nodes cheat by manipulating the distance between themselves and the target node. This could be achieved by either manipulating the distance estimation process (e.g., manipulating signal strength in RSSI-based approaches or delaying transmissions in TDoA-based approaches) or by providing an incorrect self-location to the target. In addition to acting independently, a malicious anchor can also collude with other malicious anchors. In order to effectively communicate with the MT on the CDMA-based data channel, all anchors (including malicious anchors) need to transmit localization messages by encoding them using one of the OCS known to the MT. Coordinating with each other helps the malicious anchors in selecting *different OCSs* for data transmission, thus avoiding interference and data corruption at the target node. Malicious data transmitted using an incorrect OCS will be directly discarded at the target, and thus not included in the location calculation process. It is also reasonable to assume that the malicious anchors who joined after network initialization do not possess the secret group keys and other cryptographic materials shared only by the honest anchors. Thus, although the malicious anchors may be able to generate all possible OCSs (of a particular length) using the publicly known Golay algorithm, they are not able to receive and maintain the list of OCSs valid during a particular time period because the OCS updates are encrypted with a group key known only to honest anchors. We assume that there is no trust between the MT and the set of honest anchors, and so even the MT does not know these details. From the point of

view of the MT, it will use all localization messages encoded with any OCS (present in its table of known network OCSs) for determining its own location. In order to successfully cheat, the malicious anchors (not belonging to the GC) must first *guess* a set of all unused OCSs from a valid flock (otherwise the communication would be rejected by the MT) and then broadcast malicious localization data using these OCSs in order to disorient the target node MT.

Other types of attacks, where the malicious anchors attempt to manipulate the distance between the honest anchors and the target by replaying (or modifying) responses from honest anchors, are also possible. In our work, we consider these attacks as well. Finally, readers should note that we currently do not address Denial-of-Service (DoS) type attacks executed by the malicious anchors. In this paper, our focus is on overcoming attacks where the malicious anchors attempt to successfully disorient the target node MT by providing false location and distance information. We will address Denial-of-Service (DoS) type attacks in future research efforts.

IV. Secure Localization using Jamming

A. Proposed Secure Localization Protocol

We propose a simple *request-response* strategy in order to secure localization in the presence of cheating anchors. In our proposal, as discussed earlier, the MT and anchors communicate on two separate channels. The MT asynchronously broadcasts localization requests on the CSMA/CA control channel and receives responses from anchors on the CDMA-based data channel. Each time the MT needs to determine its location, it periodically broadcasts *request frames* at random intervals (within the current OCS validity period ψ). The request frames contain a randomly generated *request number* (D) which is used by the MT to track the corresponding responses. Requests are broadcast and all requests contain the same fixed source address in order to maintain *sender anonymity*. Moreover, each such request is broadcast with a different radio power level (between some permitted E_{min} and E_{max} mW) and with a different random request number for further anonymity.

The (honest) anchors hear for localization requests on the CSMA/CA control channel and broadcast their *responses* on the CDMA data channel. These broadcast responses contain position information of the corresponding anchors, which is then used by the MT for its own position estimation. A *response number* of $D+1$ is used to identify the response frame corresponding to a request with request number D. This helps the MT identify and process only responses corresponding to its own requests. In order to confuse the malicious anchors, and to prevent them from selectively targeting requests from the MT, we follow a coordinated "*request confusion*" strategy. According to this strategy, in each time interval ψ, a small number of dummy request frames, similar to the ones sent by the MT, are periodically broadcast by a random group (at least one) of honest anchors. Such a request confusion strategy makes it extremely difficult for malicious anchors to distinguish valid requests sent by the MT from the dummy ones. Lets assume that in the time interval ψ, a group of ν_ψ honest anchors send a maximum of r_ψ requests each. The parameters of the request confusion strategy are carefully selected so that the total number ($\nu_\psi r_\psi$) of dummy requests in each time interval ψ provide reasonable confusion without overwhelming the system.

Location response frames sent by the anchors on the CDMA data channel are encoded with the valid OCS assigned to each anchor (during that time period). Optionally, all anchors could also sign their responses using the group signatures of their corresponding GC, which could be verified by other anchors in their GC. On receiving a response, the MT attempts to decode the received frame using an appropriate OCS (from the OCS table stored in its memory). Response frames that cannot be decoded correctly (because of being encoded with an OCS not in its table) or those that do not pertain to its own request are immediately discarded by the MT. As discussed earlier, the MT sends consequtive requests with different transmission power and each with a different request number. The MT waits for atleast *two* responses encoded with the same OCS within a fixed time duration (T_{res} seconds) and from the same fixed location (i.e., from the same anchor) before estimating its distance to each such anchor. We will soon see how this helps the MT to avoid using coordinates from malicious anchors. In our scheme, the MT uses a *Time-of-flight* based approach for location estimation. ToA-based approaches requires that the MT be time-synchronized with the anchors. To avoid this strict synchronization requirement, the MT employs multilateration by using TDoA-based distance estimates, which does not require knowledge of the absolute time of transmission.

Due to the "request confusion" strategy, where both the MT and honest anchors independently send requests with different power levels, malicious anchors are unable to accurately distinguish if a request was from the MT or not. Consequently, the malicious anchors are unable to estimate their distance from the MT and unable to collaboratively track the MT. Thus, it is non-trivial for the malicious anchors to selectively manipulate self location information in the response frames in order to successfully disorient the MT. The cheating behavior of the malicious anchors is thus restricted to following two kinds. First, where the malicious anchors send *random* false self-locations in the response frames, and second, where they send *fixed* false self-locations. Irrespective of this, the response frames by malicious anchors still need to be encoded by a valid OCS (assuming they are able to determine valid OCSs).

Both these kinds of cheating behavior results in inconsistent location information which can be easily detected by honest anchors that know their own locations. Many examples of such detection techniques can be found in the literature [17], [18]. As mentioned earlier, our protocol can be made further secure by requiring that all anchors sign the response frames with their group key. Obviously, malicious anchors will not possess a valid group key and would be unable to produce the correct group signature which would be detected by the honest anchors. After cheating is detected, honest anchors will selectively jam all future (request and response) frames encoded

with the OCS used by the malicious node by broadcasting a jamming signal until the expiration of the current time interval ψ. The jamming signal consists of an information signal (or data) consisting of all 1's encoded with the malicious anchor's OCS. This jamming signal is transmitted by all in-range honest anchors in the current GC. The transmitted jam signal by honest anchors adds together with the signal of the malicious anchor to produce a unique, but invalid, signal (or data). Thus, the data received by the MT on this particular OCS (used by the malicious anchor) will have out of range coordinates and/or corrupt header and will be discarded by the MT.

Such a strategy prevents malicious anchors from sending multiple responses with false location information encoded with the same OCS. As long as $T_{res} \leq \psi$ seconds, the MT will never be able to utilize the location information sent by the malicious anchors for location estimation because it requires at least two responses encoded with the same OCS (or from the same anchor) within T_{res} seconds. It should be noted that every honest anchor is able to send the relevant location data (for MT localization) along with the jamming signal on the channel, i.e., the jamming signal (data consisting of all 1's multiplied or encoded with the OCS used by the malicious anchors) is added to the valid signal (valid location data of the honest anchor multiplied or encoded with the OCS assigned to it) by the honest anchors and transmitted on the channel. Our proposal is outlined in Protocols 1, 2 and 3 below.

```
1: Generate OCS table with Golay algorithm;
2: for each time period ψ_k do
3:    Randomly select OCSs from the set of OCSs valid for
      the flock;
4:    Prepare the valid OCS table for advertisement;
5:    Sign and Encrypt (with a pre-shared group key) the
      OCS table;
6:    Broadcast OCS table on the CSMA control channel;
7: end for
```

Protocol 1: Parent Anchor in a GC

```
1: while data on CSMA control channel do
2:    if data is from parent anchor then
3:       Verify group signature and decrypt data;
4:       Identify an OCS to use from the table of valid OCSs;
5:       Save the table of valid OCSs for the current time
         duration;
6:    else if data is a localization request then
7:       Create and ansynchronously send a dummy location
         request on the CSMA control channel with some
         probability p;
8:       Let D be the request number in the received request;
9:       Create a response packet with response number D+1
         and containing self location coordinates;
10:      Optionally, sign the packet with group key;
11:      Encode packet with the chosen OCS (bit-wise inner
         product);
12:      Synchronously send response packet on CDMA data
         channel;
13:   else
14:      Drop the packet;
15:   end if
16: end while
17: while data on CDMA data channel do
18:   if data contains location responses then
19:      if cheating detected in location responses then
20:         Create a jamming signal (packet consisting of all
            1's);
21:         Broadcast the jamming signal on CDMA data
            channel;
22:      else
23:         Drop the packet;
24:      end if
25:   end if
26: end while
```

Protocol 2: Honest Anchors

B. Analysis and Discussion

We now analyze the security provided by our scheme and discuss some of its shortcomings. Earlier, Jadliwala et al. [24] proved a lower bound for the minimum number of honest anchors required for secure (bounded-error) localization in the presence of cheating anchors. Their work focused on tolerating the effect of cheating anchors. Contrary to that, our proposal provides a mechanism to actively detect and disable *all* malicious localization data (or anchors). Thus, the localization error in our proposal depends only on the employed distance estimation and multilateration procedure and on communication related parameters such as OCS length (OCSL), but not on the number of malicious anchors in the network (as was the case in [24]).

Malicious anchors in our protocol could either be *outsiders*, i.e., not belonging to a particular GC or *insiders*, i.e., belonging to a particular GC at network initialization. As malicious outsiders do not possess the shared OCS table currently being used, they first need to determine the valid OCS for the flock they intend to cheat in. The probability of choosing the right flock by the uncoordinated outsider adversary depends on the CRF. This probability decreases as the CRF becomes smaller. In the worst case, the likelihood of picking a valid OCS in a GC depends on the number of OCSs used in that specific GC, which in turn depends on the OCSL. For example, in any GC g employing a Golay code of length L and with a current valid OCS table of $F_g(\psi_k)$, during any time period k, this probability is $\frac{|F_g(\psi_k)|}{L}$ (note: In Golay, the total number of valid OCSs for a code of length L is L). If an OCS currently in use by another anchor is used, it will corrupt the data received at the MT and thus cannot be used to disorient it. As the OCSs used by honest beacons are changed periodically, a *brute-force* type of attack would slowly become infeasible. Readers should note that the use of secret OCSs (at the PHY layer) by the anchors is not aimed towards providing strong authentication guarantees. Rather, we rely on higher

```
1: while data on CDMA data channel do
2:    Decode the packet or data frame, i.e., calculate inner
      product using all valid OCSs;
3:    if (location response packet) and (flock# and GC# match
      network plan) then
4:       if another response encoded with same OCS and
         from same coordinates received no earlier than T_res
         seconds then
5:          Save anchor coordinates;
6:       end if
7:    else
8:       Drop the packet;
9:    end if
10: end while
11: Select atleast three coordinates;
12: Perform Multilateration;
```

Protocol 3: Mobile Target (MT)

level cryptographic mechanisms, such as group signatures, to detect and disable communications from malicious outsiders.

Malicious insiders will be able to effectively communicate with the MT using a valid OCS (and of course, a valid signature), but would be easily detected (in a collaborative fashion) by the honest anchors based on the discrepancy of the location information transmitted in the response packet. The honest anchors will use the proposed CDMA-based jamming approach to prevent consecutive malicious responses within T_{res} seconds from these insiders. In order to avoid jamming, malicious insiders could use a different valid OCS for every transmission. But with such a behavior, they will still not be able to disorient the target node because at least two responses (within T_{res}) encoded with the same OCS are required for successfully disorienting the MT. Even if the outsiders are able to obtain valid OCSs (and sign the messages using a valid signature), discrepancy in location information can be verified at the honest anchors, who will jam future responses originating from these compromised anchors (or OCSs). Finally, compromising the MT is also not useful because the MT, similar to an outsider, does not possess the details of the valid OCSs used by the honest anchors in the current, as well as, future time periods.

One of the advantages of our scheme over existing approaches is that we are not constrained by the number of malicious anchors within a GC. A single honest anchor has the capability of jamming more than one malicious anchors (within the same GC) and, at the same time, is able to provide honest localization information to the MT. In order for the multilateration to work, at least three honest anchors within the GC are needed. Our scheme is also resistant to *replay attacks* where a malicious anchor may attempt to manipulate the distance between the honest anchors and the target by replaying responses from honest anchors. Due to the use of a random number D in each request, outsider malicious anchors would be unable to replay later a response sent earlier by an honest anchor. The MT only uses responses that are consistent with the random ID (D) in its request. Insider malicious anchors should be able to replay the message from a honest anchor by carefully re-crafting and re-signing the message. But once again, location inconsistencies in the replayed message will be detected by the honest anchors, who should be able to jam further instances of these replayed messages.

Our proposal is also resistant to other stronger variants of the replay attacks, such as the *wormhole attacks* [42], where an adversary replays packets from one part of the network in other parts of the network (probably, where the MT currently resides). As the group signatures used in different GCs in the network are different and are agreed during network initialization, messages created in a particular GC would not correctly authenticate in other parts of the network. Moreover, random request IDs used in each message, as well as, different OCSs used in different GCs in a given time period provide further protection against wormhole attacks.

V. Evaluation

We further evaluate our proposal using extensive simulation experiments, as discussed below.

A. Simulation Setup

In our simulations, we consider a $1000m \times 1000m$ network area where anchors (both honest and malicious) are distributed uniformly over the network area. One such distribution of 200 honest and 200 malicious anchors is shown in Figure 3. The position of the MT is chosen randomly in the network area. Table I outlines the parameter values used in our

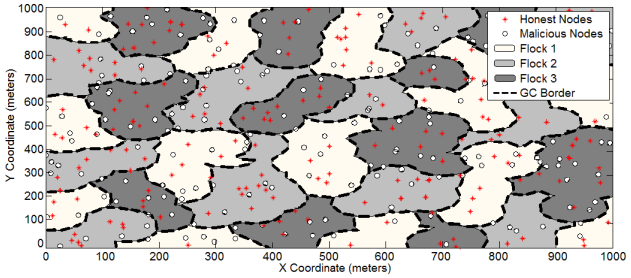


Fig. 3. Distribution of Anchors in the network

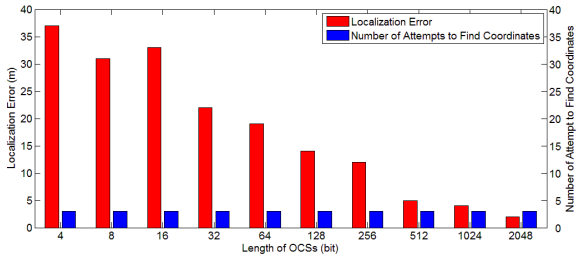
simulations. After deployment, we first tessellate the network and cluster the honest anchors as discussed in Section III-B. After the network tessellation and initialization, the network infrastructure is ready for the localization service. The MT begins its location discovery process by sending two initial requests on the control channel, and then additional ones, as needed. For location computation, the MT uses multilateration by estimating the time difference of arrival of valid coded bits from the various anchors.

B. Simulation Results and Discussion

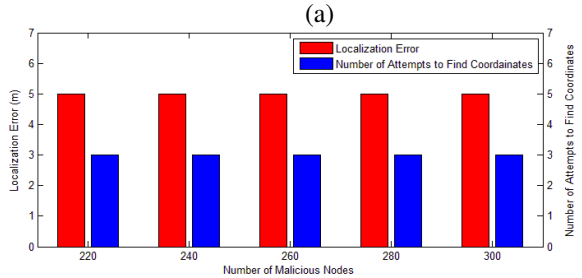
We evaluate our proposed secure localization protocol under several different network conditions. In our first set of experiments, we deploy 200 honest anchors and 200 malicious anchors (as shown in Figure 3). Our first goal is to verify the

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Simulation area	$1000\ m^2$
Tx power on CSMA/CA-based control channel	$E_{min} = 1\ mW$ to $E_{max} = 15\ mW$
Tx power on CDMA-based data channel	$15\ mW$
Carrier frequency	$2.4 - 2.48\ GHz$ (Zigbee)
Bit rate	$250\ Kbit/sec$
Packet rate	$5208\ Packet/sec$
Orthogonal Chip Code generator	Golay
Chip Code Size	Varies between 4 to 2048 $bits$ - Asynchronous OCS
CRF	$\frac{1}{3}$
Radio propagation model	Free Space
Maximum delay spread	$3\ \mu sec$
Bandwidth Efficiency	$84\ \%$
Cluster layout	3 GCs
OCS transmission duration or T_c	$0.1 \times 10^{-3}\ secs$
T_{res}	$\psi\ secs$



(a)



(b)

Fig. 4. Simulation Results (a) Localization Errors versus OCS Length and (b) Localization Errors versus Number of Malicious Nodes

effectiveness of our protocols in eliminating the cheating effect of malicious anchors. One of the first observation that we make is that, in all our simulation runs, all malicious or cheating anchors are successfully jammed, thus preventing them from disorienting the MT during the multilateration procedure. With only the data received from the honest anchors, the MT is able to accurately localize itself. Thus, it appears that the localization error (Euclidean distance between the calculated location and MT's real location) observed in our simulations is not due to the malicious anchors, but rather due to the properties of the CDMA communication channel and/or the multilateration technique used for localization.

We observe that the localization error decreases considerably when the OCS length used by the anchors increases (Figure 4 (a)). The localization error is relatively high (roughly 37 meters) when a 4-bit OCS is employed. On increasing the OCS length to 2048 bits, the localization error decreases to less than 2 meters, which is a significant improvement. Golay OCS has certain fault-tolerance properties due to which it can tolerate bit errors up to a certain threshold. This fault-tolerance depends on the length of the OCS (OCSL) used. Due to interference among the anchors (both malicious and honest), encoded transmissions from the honest anchors can be partially corrupted. Depending on the number of corrupted bits, it can be either tolerated at the MT or it can result in the manipulation of the coordinate information (from the honest anchors) received by the MT. In other words, encoded transmissions with shorter OCSs result in a larger error in the received coordinate information, which translates to a larger error in the multilateration procedure. This in turn results in a larger localization error. We can conclude that by choosing an OCS of appropriate length, fairly accurate multilateration-based localization of the MT is possible in an adversarial network environment containing malicious anchors. However, the number of attempts required by the MT, defined as the number of distinct localization requests sent by the MT, to overcome the cheating effect of malicious anchors and to compute its position does not depend on the OCSL, as seen in Figure 4 (a). We also observe that the total number of requests needed by the MT is minimum. The responses from the third request are not even used during the location computation as all malicious anchors are neutralized before that.

When the number of malicious anchors increases and the distribution of the honest anchors is the same, we can see from Figure 4 (b) that, for a particular OCSL (in this case, 512-bits), the increase in the number of malicious anchors has no effect on the localization accuracy and the number of requests needed for secure localization. This also shows that in our scheme, a smaller number of honest anchors can successfully disable a relatively larger number of malicious or cheating anchors, further proving its robustness in highly insecure environments.

While the above results are for a single distribution of honest and malicious nodes, we did run our simulations for varying distributions of honest and malicious anchors. In our next set of experiments, we simulated 10 different uniform distributions of honest and malicious anchors, where each distribution was simulated 100 times. In these simulations, we considered an OCSL of 512 bits and the total number of honest and malicious anchors were fixed at 200. In these simulations, we observed that the average localization error was 4.36 meters and the average number of MT requests before localization was around 4, which is very similar to the earlier results (Figures 4 (a) and (b)). These average results show that our scheme consistently performs well under various distributions of honest and malicious anchors. For one of the above distribution, Fig. 5 shows the position estimation time in comparison to the parameter T_{res}. We can see from the figure that our secure protocol executes efficiently (on average, it takes $0.1485\ msec$) and is always able to securely localize the MT within the first time period ($\psi\ secs$) of its request.

In summary, our simulation results confirm that DSSS or

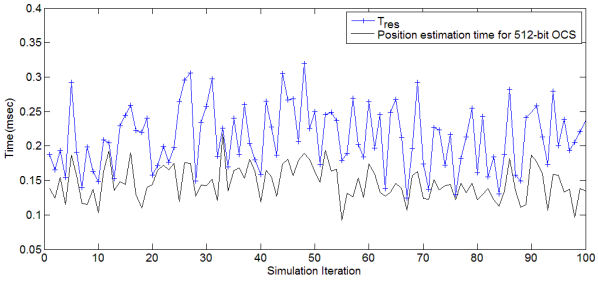


Fig. 5. Position Estimation Time

CDMA-based localization schemes are efficient and reactive jamming is an effective strategy to disable cheating anchors in location discovery systems that utilize anchor devices.

VI. Conclusion

In this paper, we presented a new approach for securing localization in anchor-based positioning systems. The proposed approach implemented a request confusion strategy in order to anonymize localization requests and a reactive jamming strategy on the CDMA response channel to actively disable malicious or cheating anchors. Our simulations showed that, if appropriate parameters are chosen, the proposed technique is effective in eliminating cheating anchors and is fairly accurate. An extensive literature review shows that our technique is one of the first such techniques that deploys jamming on a DSSS or CDMA communication channel for actively securing location discovery in wireless networks.

References

[1] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *IEEE Computer*, 2001.

[2] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The Active Badge Location System," *ACM Transaction on Information Systems*, 1992.

[3] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based User Location and Tracking System," in *IEEE INFOCOM*, 2000.

[4] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *ACM MOBICOM*, 2000.

[5] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications Magazine*, 2000.

[6] D. Niculescu and B. Nath, "DV based Positioning in Ad hoc Networks," *Journal of Telecommunication Systems*, 2003.

[7] R. Stoleru and J. A. Stankovic, "Probability Grid: A Location Estimation Scheme for Wireless Sensor Networks," in *IEEE SECON*, 2004.

[8] M. W. Carter, H. H. Jin, M. A. Saunders, and Y. Ye, "Spaseloc: An Adaptive Subproblem Algorithm for Scalable Wireless Sensor Network Localization," *SIAM J. on Optimization*, 2006.

[9] J. Liu, Y. Zhang, and F. Zhao, "Robust Distributed Node Localization with Error Management," in *ACM MobiHoc*, 2006.

[10] G. Mao, B. D. O. Anderson, and B. Fidan, "Path Loss Exponent Estimation for Wireless Sensor Network Localization," *Computer Networks*, 2007.

[11] R. Moses, D. Krishnamurthy, and R. Patterson, "A self-localization method for wireless sensor networks," *Eurasip Journal on Applied Signal Processing, Special Issue on Sensor Networks*, 2003.

[12] J. Xiao, L. Ren, and J. Tan, "Research of TDOA Based Self-localization Approach in Wireless Sensor Network," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[13] N. O. Tippenhauer, K. B. Rasmussen, C. Pöpper, and S. Čapkun, "Attacks on Public WLAN-based Positioning Systems," in *ACM MobiSys '09*, 2009.

[14] M. Williams, "Spoofed! fake gps signals lead yacht astray," *Computer World*, 2013.

[15] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust Statistical Methods for Securing Wireless Localization in Sensor Networks," in *ACM/IEEE IPSN*, 2005.

[16] D. Liu, P. Ning, and W. Du, "Attack-Resistant Location Estimation in Sensor Networks," in *ACM/IEEE IPSN*, 2005.

[17] ——, "Detecting Malicious Beacon Nodes for Secure Location Discovery in Wireless Sensor Networks," in *ICDCS*, 2005.

[18] W. Pires, T. H. de Paula Figueiredo, H. C. Wong, and A. A. Loureiro, "Malicious Node Detection in Wireless Sensor Networks," in *IPDPS*, 2004.

[19] S. Čapkun and J.-P. Hubaux, "Secure Positioning in Wireless Networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, 2006.

[20] S. Misra, G. Xue, and S. Bhardwaj, "Secure and Robust Localization in a Wireless Ad Hoc Environment," *IEEE Transactions on Vehicular Technology*, 2008.

[21] N. Safa, S. Sarkar, R. Safavi-Naini, and M. Ghaderi, "Secure Localization using Dynamic Verifiers," in *ESORICS*.

[22] S.-H. Fang, C.-C. Chuang, and C. Wang, "Attack-resistant wireless localization using an inclusive disjunction model," *Communications, IEEE Transactions on*, 2012.

[23] R. Feng, X. Guo, N. Yu, and J. Wan, "Robust multihop localization for wireless sensor networks with unreliable beacons," *International Journal of Distributed Sensor Networks*, 2012.

[24] M. Jadliwala, S. Zhong, S. J. Upadhyaya, C. Qiao, and J.-P. Hubaux, "Secure Distance-based Localization in the Presence of Cheating Beacon Nodes," *IEEE Transactions on Mobile Computing*, 2010.

[25] N. Sastry, U. Shankar, and D. Wagner, "Secure Verification of Llocation Claims," in *ACM WiSe '03*, 2003.

[26] H. H. Chen, *The Next Generation CDMA Technologies*. John Wiley and Sons, 2007.

[27] T. Wigren, "Adaptive Enhanced Cell-ID Fingerprinting Localization by Clustering of Precise Position Measurements," *IEEE Transactions on Vehicular Technology*, 2008.

[28] W. Cao, Q. Zhang, and A. Nallanathan, "A UWB Localization Scheme for LOS and NLOS Environments using Orthogonal Codes," in *IEEE International Conference on Ultra-Wideband (ICUWB)*, 2011.

[29] L. Doherty, L. E. Ghaoui, and K. S. J. Pister, "Convex Position Estimation in Wireless Sensor Networks," in *IEEE INFOCOM*, 2001.

[30] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from Connectivity in Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, 2004.

[31] X. Ji and H. Zha, "Sensor Positioning in Wireless Ad-hoc Sensor. Networks using Multidimensional Scaling," in *IEEE INFOCOM*, 2004.

[32] L. Fang, W. Du, and P. Ning, "A Beacon-Less Location Discovery Scheme for Wireless Sensor Networks," in *IEEE INFOCOM*, 2005.

[33] L. Lazos, R. Poovendran, and S. Čapkun, "Rope: RObust Position Estimation in Wireless Sensor Networks," in *ACM/IEEE IPSN*, 2005.

[34] S. Ray, R. Ungrangsi, F. de Pellegrini, A. Trachtenberg, and D. Starobinski, "Robust Location Detection in Emergency Sensor Networks," in *IEEE INFOCOM*, 2003.

[35] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan, "Ecolocation: A Sequence Based Technique for RF-only Localization in Wireless Sensor Networks," in *ACM/IEEE IPSN*, 2005.

[36] A. Boustani, J. Sabet, M. Azizi, N. Mirmotahhary, and S. Khorsandi, "Persian Code: A New Orthogonal Spreading Code Generation Algorithm for Spread Spectrum CDMA Systems," in *Wireless Advanced (WiAD), 2010 6th Conference on*, 2010.

[37] D. Chaum and E.V.Heyst, "Group Signatures," in *EUROCRYPT*, 1991.

[38] R. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," in *ASIACrypt*, 2001.

[39] U. Sawant, "Grid-based coordinated routing in wireless sensor networks," Master's thesis, University of North Texas, 2006.

[40] A. Boukerche, H. Oliveira, E. Nakamura, and A. Loureiro, "A Voronoi Approach for Scalable and Robust DV-hop Localization System for Sensor Networks," in *ICCCN*, 2007.

[41] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. John Wiley and Sons, 2005.

[42] Y.-C. Hu, A. Perrig, and D. Johnson, "Wormhole Attacks in Wireless Networks," *IEEE Journal on Selected Areas in Communications*, 2006.