

Secure RSS-based Localization in Sensor Networks

Srdjan Čapkun¹, Saurabh Ganeriwal², Farooq Anjum³ and Mani Srivastava²

¹Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland, capkuns@inf.ethz.ch.

²Networked and Embedded Systems Lab, University of California Los Angeles, CA 90095, {saurabh,mani}@ee.ucla.edu

³Applied Research, Telcordia Technologies Inc., 445 South St, Morristown, NJ 07960, fanjum@telcordia.com.

Abstract—We propose a secure localization scheme for sensor networks based on the received signal strength (RSS) ranging techniques. Our scheme enables the network authority to obtain locations of sensor nodes in the presence of an attacker. The proposed scheme uses a small number of anchor nodes with known locations that provide points of reference from which the sensors’ locations are computed. This scheme makes use of robust localization and time synchronization primitives which, appropriately combined, enable the detection of attacks on localization, within a realistic attacker model. We perform an in-depth security analysis of our scheme and we demonstrate its feasibility on Mica2 sensor platform.

I. INTRODUCTION

Sensor networks have increasingly become the subject of intense scientific interest over the past few years. An important problem in such networks is sensor localization. In the last decade, researchers have proposed a number of localization techniques for wireless networks [37], [38], [26], [3], [15], [6]. The use of these techniques is broad and ranges from enabling networking functions (such as routing) to enabling location-related applications (such as access control). It can be argued that these techniques can be adapted for sensor networks. However, recent analysis [35], [18] showed the vulnerability of the proposed techniques to attacks involving false position and range reports by internal attackers and position spoofing by external attackers.

It can be argued that the use of GPS would be a solution to this set of problems. But this is not really the case. This is not only because GPS is costly, since it adds a lot of complexity and cost to the sensors, but because its civilian version is also insecure [39]. In the light of this, a number of secure localization techniques were proposed specifically for sensor networks [19], [35], [21], [22], [36]. These mechanisms typically rely on high speed hardware (e.g., distance bounding with radio frequency (RF) signals [4]), directional antennas, and robust statistics. Although the reliance on RF distance bounding can result in secure localization schemes, these

incur additional costs associated with the high-speed and nanosecond-precise specialized hardware that distance-bounding requires. This might prohibit the use of such schemes with low-cost sensor nodes.

In this paper, we focus our attention on achieving RSS-based secure localization of sensor nodes. We show that secure localization can be achieved using (low-cost) ranging techniques based on the received signal strength (RSS) measurements and hence, requires no additional hardware support from what already exists on a typical sensor node. We note here that RSS-based localization schemes, although less precise (1m accuracy) than RF-based or ultrasound (US) based time-of-flight techniques (2cm accuracy), do not require any specialized hardware unlike those techniques. It is worth noting that in late 2005 Chipcon released its system on chip (SoC) solution for Zigbee based networks, (CC2431); this solution combines RF transceiver (CC2430) with a location engine that is based on RSS ranging. Recent advances in RSS-based ranging technologies [40], [41] also show that with appropriate calibration techniques, errors in RSS ranging can be significantly reduced.

RSS-based ranging techniques compute distances based on the transmitted and received signal strengths. To modify the measured distance between two honest nodes, an external attacker only needs to jam the nodes’ mutual communication and replay the messages with higher or lower power strengths. In this paper, we demonstrate that, by exploiting the availability of secure time synchronization and the secure and precise estimation of the packet propagation delay, we can detect jam-and-replay attacks on RSS-based ranging. We name this primitive Robust RSS Ranging. We show that robust RSS ranging is resilient to jam-and-replay attacks which could result in either distance enlargement or reduction.

However, robust RSS ranging is not resilient to signal amplification attacks wherein the attacker amplifies the signal of one node such that it reaches other nodes with a higher strength thereby resulting in a distance reduction. A slightly sophisticated attacker can amplify and transmit the signal at such high speeds so as not to be detected by

end-to-end delay measurements. In order to counter these attacks, we use a variant of verifiable multilateration[35], termed *inverse verifiable multilateration*.

We further observe attacks by internal attackers (i.e., compromised sensor nodes), which are simpler to perform and can be more harmful than those performed by external attackers. Compromised nodes can report non-existing links, false locations and can also modify ranges measured to them by their neighboring nodes. We detail an outlier detection scheme, termed as Neighborhood Consistency Check, that helps in removing the influence of internal attackers on the final location estimation.

The remainder of the paper is organized as follows. In Section II, we present our system and attacker models and we review attacks on localization in sensor networks. In Section III, we present our Robust RSS ranging protocol. In Section IV, we show how Robust RSS ranging can be used to detect signal amplification attacks. We describe our secure localization algorithm in Section V. In Section VI, we analyze the security of our algorithm. Related work is reviewed in Section VII. We conclude the paper in Section VIII.

II. MODEL

In this section, we describe our system and attacker models and briefly review attacks on sensor network localization.

A. System model

Our system consists of a set of sensor nodes¹, forming a network as also a set of Anchor Nodes (ANs) with known locations and a sink. The network is operated by an authority who controls the network membership and assigns a unique identity to each node. The Anchor Nodes and the sink are robust to compromise. We assume that every legitimate node shares a secret key with the sink and the ANs. We further assume that every sensor node holds the authentic roots of hash chains corresponding to every AN. This authentication material is established/obtained prior to secure localization execution through the authority controlling the localization infrastructure. Sensor nodes and ANs communicate using radio transmissions. We assume bidirectional radio links between neighboring devices.

In our system, ANs know their locations or can obtain their locations securely (e.g., through distance-bounding based techniques). Here, we assume that the attackers cannot tamper with these locations. We consider that each AN has a capability to vary its transmission power level

¹Referred as simply "nodes" throughout the paper unless an ambiguity exists.

(e.g., the transmission power of Mica2 nodes can be controlled and varied from -20 to 10 dBm [1]). Each power level p corresponds to a different transmission range, represented by a circle of radius ' r_p '.

We further assume that all nodes and ANs have internal clocks and can measure time with certain precision (μ seconds for Mica2 nodes). ANs are securely synchronized to the network sink, either through GPS [17], or through pairwise secure time synchronization techniques [14].

B. Attacker model

We adopt the following attacker model. We assume that the attacker controls the communication channel in a sense that it can eavesdrop messages, modify transmitted messages and schedule transmissions. We further assume that the attacker can jam the communication between two nodes by transmitting signals which will disrupt packet reception at the receiver. We consider stealthy, disruptive jamming that cannot be detected at the receiver. Currently available sensor network platforms use Chipcon1000, 2.4 GHz IEEE 802.15.4 compliant (Direct Sequence (DSSS)) or Bluetooth (Frequency Hopping (FHSS)) radios. DSSS and FHSS, because of their low transmitting RF power (1mW), are vulnerable to broadband jamming. Recently, Xu et al. [42] showed that jamming attacks are indeed feasible against Mica nodes, and that detecting these attacks requires significant resources.

We distinguish two attacker models: internal and external. In the external attacker model, we assume that none of the nodes involved in the protocol are compromised. An external attacker thus cannot authenticate itself as an honest network node to other network nodes or to the central authority. An internal attacker controls one or more network nodes. We assume that when a node is compromised, its secret keys and other secrets that it shares with other nodes are known to the attacker; subsequently, compromised nodes can authenticate themselves to the authority and to other network nodes. We further put no restrictions on the colluding abilities of compromised nodes. Thus, internal attackers can exchange the security material present on the nodes they control.

C. Attacks

One of the most obvious threats to sensor networks is the physical displacement of nodes. Detection of these attacks require periodic execution of localization protocols in the sensor network. Even if localization is performed periodically, if the network is not properly protected, an attacker can create the impression to the displaced node and to its neighbors that the node did not move; a simple approach for the attacker is to create a communication

link (a wormhole[16]) to the new location of the honest node.

Even without displacing the nodes, external attackers can still perform a number of attacks on node positions and network topology. An attacker can permanently or temporarily jam the communication between pairs of nodes and thus remove links that would normally exist. Similarly, by creating wormholes, an attacker can establish links between nodes that are not in each others' power range. Furthermore, an attacker can change measured distances between the nodes by appropriately modifying their ranging communication. We detail attacks on RSS-based ranging in Section III-B.

Finally, if an attacker controls several sensor nodes, it can simply manipulate their locations by allowing the nodes to exchange their authentication material. A larger number of colluding compromised nodes can influence location computation of entire sections of the network if not of the whole network.

III. ROBUST RSS RANGING

In this section, we extend the basic mechanism of RSS-based ranging to make it resilient to jam-and-replay attacks from external attackers. Robust RSS-based ranging uses entity authentication and end-to-end delay measurements to detect the distance enlargement and reduction attacks by external attackers. We analyze the security properties of these mechanisms in Section VI. We first overview the basic idea behind RSS-based ranging and discuss the attacks that can be mounted on this ranging technique.

A. RSS-based ranging

The basic idea here is to compute the distance between an anchor node and a sensor node using RSS measurements. We next present one possible instantiation of the (authenticated) ranging protocol².

Authenticated RSS-based ranging

1 $S \rightarrow AN : S, AN, N_S$
 2 $AN \rightarrow S : AN, S, N_S, P_t$
 $MAC\{K_{ANS}\}[AN, S, N_S, P_t]$
 3 $S : \text{from } P_t \text{ and } P_r, \text{ compute } d_{AN,S}$

Here we assume that the anchor node and the sensor node share a secret key denoted by K_{ANS} . N_S denotes a nonce while P_t denotes the strength of the transmitted signal at the anchor node and P_r denotes the RSS power measurement of the received signal at the sensor

²We present the authenticated version of the protocol to rule-out trivial attacks (e.g., impersonation and attacks due to lack of authentication).

node. Knowing the power P_t at which the anchor node transmitted the signal and the measured power P_r of the received signal, the sensor can estimate its distance to the anchor node denoted as $d_{AN,S}$. This requires that each sensor node make use of either a propagation model or a signal strength map. A log-normal shadowing model has been verified to represent the propagation model for sensor nodes deployed in an obstacle free environment [23].

The message (numbered 2 in the authenticated RSS-based ranging protocol above) used to measure the distance between AN and S is authenticated and its integrity is protected to prevent impersonation and message forging³. Here, we implicitly assume that the AN and the sensor node are mutually trusted; by misbehaving, any of the entities can arbitrarily modify the measured distance.

RSS-based ranging is vulnerable to attacks from external attackers. An external attacker can influence ranging by jamming signals, and by modifying the signal strength. These signal characteristic modifications result in distance enlargement or reduction. We consider these attacks next.

B. Attacks on RSS-based ranging

There are two types of attacks possible, namely, distance enlargement attacks and distance reduction attacks. **Distance enlargement** attacks can be performed by two techniques: (i) jam-and-replay and (ii) signal annihilation. In the first technique the attacker jams the original signal (message 2 in the protocol) and replays it with a lower signal strength. The second technique is performed by introducing signals on the channel of the phase opposite to the phase of the original signal; the addition of the original and the introduced signal results in a signal which is the same as the original signal but of a lower strength. The authors in [33] have shown that the attacker cannot successfully annihilate the signal except with a negligible probability; this is due to the unpredictability of the message (introduced through the message authentication code), and due to the unpredictability of the channel conditions (because of phase delays and multipath effects). However, jamming attacks can be performed even by a non-sophisticated attacker [42].

Distance reduction attacks can also be performed by two techniques (i) jam-and-replay and (ii) relaying/amplifying (i.e., creating wormholes [16]). The first technique consists in jamming the original signal and replaying it with a higher signal strength than that of the original signal. The second technique requires deploying relays and/or amplifiers, whose role is to receive the signal and send it amplified towards the receiver. Thus, the signal is heard

³ $MAC\{K\}[A]$ denotes the message authentication code of message A using the key K

further than it is supposed to be, and/or with a higher strength.

Finally, attackers can, by introducing obstacles, change the propagation of the signals, resulting in inaccurate location estimates, given that the signal power decay will no longer correspond to the models used by the nodes. This attack is however unlikely, given that it requires that the attacker be able to accurately estimate the impact of obstacles on signal power decay. Additionally, the attacker also needs to have full access to the location of sensors and has to be able to hide the obstacles from the network authority. Thus, we will not consider this attack in this paper.

C. Robust RSS ranging

In this section, we demonstrate that, by measuring packet propagation delay, we can detect jam-and-replay attacks on RSS-based ranging. We present a robust RSS ranging protocol that detects distance reduction and enlargement attacks by jam-and-replay. We take our motivation from the fact that jamming the original signal and replaying it later, with enhanced/reduced signal power, consumes time. We use this fact to our advantage. In addition to RSS measurements, robust RSS ranging also calculates the end-to-end delay for packet transmissions between the anchor node and the sensor node. If this delay exceeds an expected value, we conclude that the protocol has been subjected to a jam-and-replay attack. The protocol is executed as follows:

Robust RSS ranging

- 1 $S(T1) \rightarrow (T2)AN : S, AN, N_S$
- 2 $AN(T3) \rightarrow (T4)S : AN, S, N_S, T2, T3, P_t$
 $MAC\{K_{ANS}\}[AN, S, N_S, T2, T3, P_t]$
- 3 S : from P_t and P_r , compute $d'_{AN,S}$
: calculate end-to-end delay $d = [(T2 - T1) + (T4 - T3)]/2$
: If $T^- \leq d \leq T^*$ then $d_{AN,S} = d'_{AN,S}$ else abort

d represents the calculated end-to-end delay between the anchor and the node. $T1, T2, T3$ and $T4$ represent the times at either the node or the anchor as indicated.

This protocol detects jam-and-replay attacks by measuring the end-to-end delay and comparing it with an expected maximal delay T^* and minimal delay T^- . If the attacker jams and replays the original packet, the new packet will be delayed by (at least) the time that it takes to transmit the original (jammed) packet. The delay introduced is even larger when considering the processing at the attacker. Clearly the performance of this scheme relies on the fact that the value of T^* and T^- , referred to as the maximal and minimal delay respectively, can be estimated. We next (in Section III-C.1) show that T^*

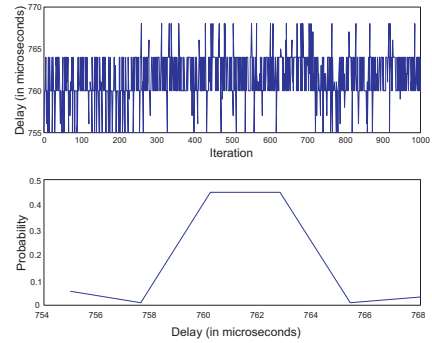


Fig. 1. End-to-end Delay over a link

and T^- can be accurately estimated, which enables the detection of the attack. We provide a testimony to this claim by carrying out a detailed empirical evaluation on Mica2 motes.

| Maximum delay (μs) | Minimum delay (μs) | Average delay (μs) | Standard deviation (μs) |
|---------------------------|---------------------------|---------------------------|--------------------------------|
| 768 | 755 | 762 | 2.82 |

TABLE I

PROPAGATION DELAY STATISTICS BETWEEN PAIRS OF MICA2 MOTES.

1) *End-to-end delay estimation*: When a packet traverses over a wireless link, the variation in channel access delay, becomes a critical bottleneck in estimating a stable value of the end-to-end delay. A detailed breakdown of the end-to-end packet delay is given in [13]. A way around this problem is to timestamp the packets below the MAC (Medium Access Control) layer. This approach has been used by the existing time synchronization approaches in sensor networks to achieve an accuracy of a few microseconds. In addition, we need a cryptographic library that can also calculate the Message Authentication Code (MAC) on-the-fly as the packets are being transmitted. Note that the message authentication code is also calculated over the timestamps. TinySec, a symmetric cryptographic library on motes, enables this MAC calculation. We have developed a prototype implementation that integrates the functionality of lower level timestamping with TinySec. We use this implementation to calculate the end-to-end delay between a pair of Mica2 motes. In all, we computed the delay for 200 independent runs. We then repeated the complete procedure for 5 different pair of motes to remove any hardware specific bias. The gathered statistics are averaged over these 1000 independent runs. The first plot in Figure 1 shows the actual delay measured in every run and the second plot shows the distribution of the end-to-end delay. Table I summarizes the statistics of the measurements. The histogram of the computed delay

resembles a Gaussian distribution $N(d_{avg}, \sigma)$. The delay will therefore fall in the interval $[d_{avg} - 3\sigma, d_{avg} + 3\sigma]$ with 99.97% confidence. Thereby, maximal delay T^* is set to $d_{avg} + 3\sigma$ ($\approx 770\mu s$) and minimal delay T^- to $d_{avg} - 3\sigma$ ($\approx 750\mu s$).

2) *Properties of T^** : It is important to realize that the absolute value of the calculated end-to-end packet delay is not of much significance to us. We are time-stamping the first byte of the SFD (Start Frame Delimiter) at the transmitter and the end of the second byte of the SFD at the receiver. Thereby, we expect the calculated end-to-end delay to be roughly equal to twice the byte transmission time. The thing that we most importantly care is the standard deviation of the estimated end-to-end delay. It is of the order of a few microseconds and roughly 0.5% of the absolute value of d_{avg} . This implies that the end-to-end packet delay, in a non-malicious setting, will always be in a range that is 3% wide (with a 99% confidence).

We also measured the complete packet transmission time, as observed by the application layers at the transmitter and receiver respectively in the 1000 experimental runs, mentioned in the previous section. This time varies from few milliseconds to hundreds of milliseconds and the main variability comes due to the channel access delay. Thereby, even sophisticated attackers that try to perform jam-and-replay distance modification attacks will be detected the calculated end-to-end delay will be increased by at least the duration of the original packet transmission, which is roughly in the order of milliseconds.

An interesting point to note out here is the fact that the value of T^* or T^- does not depend on the actual distance between the sensor nodes. The reason for this are twofold: (1) The actual value of T^* and T^- is in the order of hundreds of microseconds, whereas RF propagation takes only a nanosecond to travel a distance of one foot. Most of the time is taken in transmitting the packet bit by bit at the physical layer, due to the relatively slower radios (maximum speed of 250 Kbps) in these type of systems. Thereby, even if a node is communicating with a nearby node ($< 10cm$) or a distant node ($> 10m$), the relative difference in the end-to-end delays for the two scenarios will be a nominal factor, of the order of a few nanoseconds, and (2) Sensor nodes typically have clocks that can only measure to an accuracy of a few microseconds, making it infeasible to even calculate this nominal difference.

We carried out an empirical evaluation of this assertion by measuring the value of end-to-end delay between node pairs, which were kept at different distances from one another. As anticipated, the distribution of measured delay was same for all the mote pairs. This has a strong implication. There is no need to estimate the value of

T^* and T^- within the network at runtime. It can be calculated before the deployment of the actual network and the nodes can be pre-configured with the value of T^* and T^- , greatly reducing the overhead. We do note that the value of T^* and T^- will be different for different sensor networking platforms. For example with Micaz motes, we expect the value to be much less as they use a faster radio.

D. Robust RSS ranging with one-way communication

In the previous section, we proposed a protocol for achieving robust RSS ranging using a two-way message exchange between an anchor and a node. Note that this requires no prior synchronization between the two nodes. However, if the nodes are synchronized, then robust RSS ranging can be achieved by a single message transfer from the anchor as follows:

Robust RSS ranging with one-way communication

```

1 AN(T1) → (T2)S : AN, S, T1, P_t
                    MAC{K_{ANS}}[S, T1, P_s]
3 S : from P_t and P_r, compute d'_{AN,S}
    : calculate end-to-end delay d = (T2 - T1) - abs(δ_{ANS})
    : If T^- ≤ d ≤ T^* then d_{AN,S} = d'_{AN,S} else abort

```

Here, δ_{ANS} represents the relative clock offset between the anchor node and the sensor node. Recently, Ganerival et al. [14] proposed a secure time synchronization algorithm that achieves an accuracy of a few tens of microseconds. The one way robust RSS ranging protocol will therefore achieve the same performance in countering the jam-and-replay attacks as the two-way RSS ranging. For completeness, we review secure time synchronization protocol in the following subsection.

1) *Secure time synchronization*: Similar to robust RSS ranging, this protocol relies on the precise estimation of end-to-end packet delays by time stamping the packets below the MAC layer. A node S, runs this protocol to synchronize with an anchor AN in its neighborhood set as follows:

Secure pairwise synchronization

```

1 S(T1) → (T2)AN : AN, S, N_S, sync
2 AN(T3) → (T4)S :
  S, AN, N_S, T2, T3, HMAC{K_{ANS}}[S, AN, N_S, T2, T3, ack]
3 S : calculates end-to-end delay
    d = [(T2 - T1) + (T4 - T3)]/2
    : If d ≤ T^* then δ = (T2 - T1) - (T4 - T3)/2, else abort

```

Here δ is the computed clock offset between AN and S. The protocol counters the packet modification attacks by attaching Message Authentication Codes (denoted as HMAC) generated using appropriate secret keys at the end

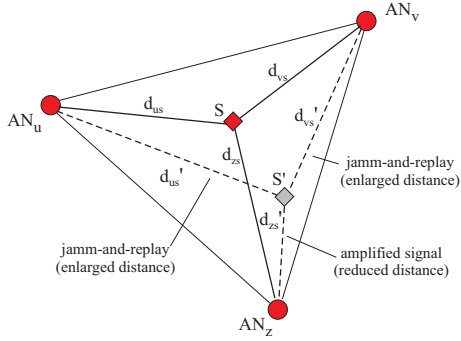


Fig. 2. Inverse verifiable multilateration: signal amplification attacks are prevented using robust RSS ranging and distance consistency check.

of a packet. A more subtle attack on the time synchronization process is the pulse-delay attack (similar to jam-and-replay attacks), wherein the attacker delays the receipt of the packet at the nodes by jamming the original signal and then replaying it at some later time in the future. Since the attacker is simply replaying the signal without modifying it, these attacks cannot be countered using traditional cryptographic techniques. Instead, pulse delay attacks are prevented by a comparison of the measured end-to-end delay d with the maximal delay, T^* . If the measured delay is unrealistically high, the nodes detect an attack and the synchronization fails. As shown in the previous section, with a 99% confidence interval, the value of the maximal delay can be estimated within $10\mu s$. The worst-case synchronization error of this protocol is about $10\mu s$. This is similar to the numbers achieved by non-secure time synchronization protocols [10], [32], [13], [28].

IV. DETECTING SIGNAL AMPLIFICATION ATTACKS

In III-B, we discussed four ways by which an attacker can modify measured distances between node pairs; distance reduction can be achieved by jam-and-replay attacks combined with signal amplification, whereas distance enlargement can be achieved by jam-and-replay with signal annihilation.

In the previous section, we have shown that by robust RSS ranging, which relies on end-to-end propagation delay measurements we can efficiently detect jam-and-replay attacks. Furthermore, in [33], the authors have shown that signal annihilation attacks can be efficiently prevented by signal randomization, which in our protocol is achieved by the use of HMAC. The unpredictability of the message will, in most modulation schemes, make the modulated signal on the channel sufficiently unpredictable for the attacker, thus preventing its annihilation. This is termed as the *anti-blocking* property of the wireless channel.

One attack that is probably the hardest to detect is the distance reduction attack by signal amplification. This attack consists of amplifying the signal such that it reaches the neighboring nodes with a higher strength. This also ensures that its reach is extended to the nodes that would normally not receive it. These signals might not be detected through end-to-end delay measurements due to the high speeds with which signals can be amplified and transmitted. We do not expect such attacks to be possible with attackers equipped with mote-type devices (e.g., Mica2 motes [2]), although these attacks can be performed by sophisticated attackers.

In order to detect such signal amplification attacks, we use an *inverse verifiable multilateration*. Verifiable multilateration [35] is a technique developed to secure RF and US time-of-flight multilateration localization schemes. In that technique, anchor nodes verify the location of a node within a triangle formed by their locations. Verifiable multilateration uses the observation that if one of the node's measured distances to the anchor nodes is enlarged (say by an attacker), one of the distances to the remaining anchor nodes needs to be reduced for the location to be consistently computed. In verifiable multilateration, distance reduction is prevented by the use of RF time-of-flight techniques and distance-bounding [4], which then leads to the detection of distance enlargement attacks (in verification triangles).

In inverse verifiable multilateration, distance reduction attacks (by signal amplification) are detected within verification triangles using the fact that an attacker cannot perform distance enlargement (i.e. by using robust ranging and anti-blocking) without being detected. We illustrate this by an example shown in Figure 2. In this example, anchor nodes AN_u , AN_v and AN_z form a verification triangle within which the location of a sensor S is verified. In order to convince the anchor nodes that the sensor is at a false location S' , an attacker reduced the distance between AN_z and S (from d_{zs} to d'_{zs}). However, to make the attack work, the attacker now needs to increase sensor's measured distances to AN_u and AN_v , from d_{us} and d_{vs} to d'_{us} and d'_{vs} , respectively. As these distance enlargement attacks are detected through robust RSS ranging, anchor nodes will also detect the reduction of the measured distance. As a result the signal amplification attack is detected.

This approach can easily accommodate errors in range measurement. To detect an attack, it is sufficient to observe inconsistencies between the measured ranges and the estimated location; if these inconsistencies exceed the expected errors, an attack is detected. Namely, for the location of the sensor to be accepted by the anchor nodes, the measured distances need to intersect in a single point

or enclose a small area within which the location can be estimated.

V. SECURE LOCALIZATION IN SENSOR NETWORKS

In this section, we develop a Secure RSS-based localization algorithm (SLA). Our algorithm is based on robust RSS ranging and inverse verifiable multilateration, and imposes no additional hardware requirement on the sensor nodes, besides the hardware available on Mica2 sensor platforms.

A. Protocol

We next outline the basic protocol⁴. We show the operation of the algorithm through an example depicted in Figure 3. Algorithm details are in the following pseudocode. The protocol is initiated either by the sink or is executed according to a predefined schedule.

Secure RSS-based Localization Algorithm (SLA)

- 1 AN-to-node secure pairwise synchronization
- 1a Nodes send to the sink reports containing identities of nodes in their neighborhoods.
- 2 Anchors broadcast messages.
 - $AN_u(t_s^u) \rightarrow (t_r^u)S : b_u = V_i^u, E_{K_u}(u, V_i^u, t_s^u, P^u, L^u)$
 - $AN_v(t_s^v) \rightarrow (t_r^v)S : b_v = V_i^v, E_{K_v}(v, V_i^v, t_s^v, P^v, L^v)$
 - $AN_z(t_s^z) \rightarrow (t_r^z)S : b_z = V_i^z, E_{K_z}(z, V_i^z, t_s^z, P^z, L^z)$
- 3 S : verify the authenticity and freshness of V_i^u, V_i^v, V_i^z : note power levels and times at which AN broadcasts are received
- $S \rightarrow (t_r)Sink : E_{K_S}(b_u, b_v, b_z, t_r^u, t_r^v, t_r^z, O_S, P_r^u, P_r^v, P_r^z)$
- 4 $Sink$: extract $t_s^u, t_s^v, t_s^z, t_r^u, t_r^v, t_r^z, P^u, P^v, P^z, L^u, L^v, L^z, O_S, P_r^u, P_r^v, P_r^z$: compute location L_S of node S using robust RSS ranging and inverse verifiable multilateration
- 5 $Sink$: After computing the location of all the sensors run the neighborhood consistency check to detect the internal attackers and discard their locations.

In the first phase of the protocol, the nodes securely synchronize with their neighboring nodes and the ANs. Note that we do not require that all the nodes in the network reach a unique time of reference, but only that the nodes are aware of the clock offsets with their respective neighboring nodes and ANs. This is done using secure pairwise time synchronization algorithm described in Section III-D.1. The node stores the clock offsets with all the ANs within whose range the node is within a set, represented by O_S . As shown in Section III-D, secure time synchronization enables the ANs and sensors to perform

⁴Our description assumes two dimensional space and extends trivially to the three dimensional space

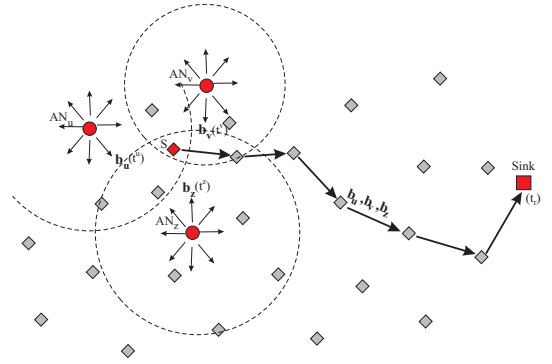


Fig. 3. RSS-based Secure Localization. The anchor nodes issue beacons. The beacons are collected by sensor nodes and reported to the network sink. The network sink computes the locations of the nodes.

one-way robust RSS ranging. Additionally, every node sends an encrypted report about the list of nodes in its neighborhood to the sink. The encryption is done using the key shared between the node and the sink.

Following this, the anchor nodes (for simplicity we assume three ANs denoted as AN_u, AN_v, AN_z) broadcast localization beacons to the sensor nodes. Each AN broadcasts at a power level of its choosing. The sink would have knowledge of this power level. Each beacon consists of the following values: (i) the hash chain value V_i^u used by the Anchor Node AN_u to authenticate itself to the sensor nodes, (ii) the time t_s^u at which the beacon was sent from AN_u , (iii) the power level P^u at which the beacon from AN_u was transmitted and (iv) the location L^u of the anchor node AN_u at time t_s^u . Note that all data except for the hash chain value is encrypted with a key K_u shared between AN_u and the sink and is thus not accessible to a node or to any external/internal attacker.

After receiving a beacon (step 3 of the algorithm), the node registers the time (e.g., t_r^u) and the power level (e.g., P_r^u) at which it receives the beacon. It then verifies the authenticity and the freshness of the beacon by checking the positions of the received hash values on their respective hash chains; if the received hash value is more recent than the values that the sensor received so far, the sensor will accept the message as being fresh. If the hash value with the same index or with an older index is received or if the sensor node does not have a valid hash chain corresponding to the received hash value, the sensor will reject the beacon and will not process it. This procedure is not crucial for the security of the positioning system, but does limit the impact of battery draining attacks. The authentication of the ANs by the sensor is detailed in Section V-B.

Upon receiving three or more beacons, the sensor forwards the information obtained from the beacons to the sink in a separate packet that is encrypted with

the pairwise secret key between the sensor and the sink, K_S . This packet contains the following information: (i) complete beacon information that is received by the sensor in the previous step from the respective anchor nodes (AN_u, AN_v, AN_z), (ii) beacon reception times (t_r^u, t_r^v, t_r^z), (iii) the received signal strengths of the beacons (P_r^u, P_r^v, P_r^z), and (iv) the time offset set O_S (the clock differences) of the sensor to its neighboring nodes and ANs.

In the final phase of the protocol the sink decrypts the message received from the sensor. Based on the known locations of the anchor nodes and the beacon information, transmit and receipt times as well as the RF power levels, the sink estimates the location of the sensor node using robust RSS ranging and inverse verifiable multilateration.

Finally, the sink performs a neighborhood consistency check to establish if computed locations of sensors are consistent with the neighborhood information reported in the first step. We provide details of this in Section V-C. This check aims to detect attacks from internal attackers.

B. Authentication of ANs

To prevent trivial battery draining of sensor nodes by the attacker, beacons sent by the ANs are authenticated by the sensor nodes. This authentication is performed using hash chains generated at ANs. Each Anchor Node AN_u creates a hash chain $V_0^u, V_1^u, \dots, V_K^u$ by choosing the initial value V_0^u uniformly at random and computing $V_i^u = H(V_{i-1}^u)$ for $i = 1, 2, \dots, K$, where H is a one-way hash function. V_K^u is called the root of the hash chain and it is distributed to sensor nodes and to the sink in an authenticated way. AN_u discloses the elements of its hash chain in reverse order (with respect to generation) starting with V_{K-1}^u and proceeding towards V_0 . The i^{th} beacon from the AN contains V_{K-i}^u . A neighboring sensor node, which receives the beacon, can verify the authenticity of V_{K-i}^u by hashing it iteratively i times and comparing the result $H^{(i)}(V_{K-i}^u)$ to the pre-distributed authentic root V_K^u . This mechanism can be very efficiently implemented at ANs and the sensor nodes as shown in [8]. Once all elements of the chain are disclosed, the roots of new chains are distributed in an authenticated manner.

C. Neighborhood Consistency check

All the messages in our protocol are authenticated using appropriate secret cryptographic keys, removing the possibility of malicious external entities actively joining the protocol (by impersonating an AN). Furthermore, robust RSS ranging and inverse verifiable multilateration provides resiliency against passive jam-and-replay and signal amplification attacks respectively from malicious

external entities. However, a location computed by the sink can be still incorrect if the information in the messages being reported back from the sensor node to the sink is incorrect. This could be because either the node that reported these messages is compromised or because the node is impacted due to the properties of RF signals.

At the conclusion of step 4 of the SLA the sink has estimated the locations of the different sensor nodes in the network. The sink at this point is also assumed to have authentic information about the neighborhood of each sensor node (from step 1a of SLA). Given this, the sink has to determine which of the estimated locations are correct and which are incorrect. The sink does this by executing step 5 of the SLA which is the neighborhood consistency check (NCC). Thus the objective of NCC is to determine the set of correctly estimated locations and the set of incorrectly estimated locations. NCC achieves this by using information about the set of neighbors and the system as we explain later. Note that NCC is generic in the sense that it can be applied irrespective of the location estimation techniques used⁵. The pseudo code for NCC is as follows:

Consider a sensor node j with k_j neighbors and let us denote the neighborhood set for this node as $NS_j = \{NS_j^1, NS_j^2, \dots, NS_j^{k_j}\}$.

Neighborhood consistency check (NCC)

- 1) Let the computed locations of the various neighbors of j be $\{L_{j1}, L_{j2}, \dots, L_{jk}\}$. Let the estimated location of j be L_j . Let a counter N_c be initialized to 0. N_c is a measure of the number of neighbors whose estimated location is inconsistent with node j .
- 2) For each neighbor N_j^n of j if $|L_j - L_{jn}| > r$, where r is the transmission range of a node, then increment N_c .
- 3) The estimated location of node j is accepted as true depending on the relative values of N_c and T where T is the acceptance threshold⁶.

We see from the above that the value of T is very important as also the relationship between N_c and T . For example consider the required relationship for the estimated location of node j to be accepted as true to be $N_c \leq T$ and let $T = 0$. Then the estimated location of a node j is assumed to be false if it is inconsistent with any of its neighbors. Thus, in this case we will not have any missed detection (a wrongly estimated location

⁵The resolution associated with the location estimation technique though would impact the decision of a correctly estimated location. Thus, if the technique has a average location resolution of 1 cm, then any estimated location, at a distance larger than 1 cm from the true location, can be considered incorrect

⁶The precise relationship will be specified later.

accepted as correct) unless an adversary corrupts an entire neighborhood of nodes. On the other hand, when T is a large number (greater than the maximum number of neighbors of any node in the network) and $N_c \leq T$ is the requirement, then the estimated location of a node j is always assumed to be true. Thus, in this case we will not have any false positives whereby a correctly estimated location is rejected as false. Another simple strategy is to accept the estimated location of a node j as correct if it is consistent with a majority of its neighbors. Thus, in this case we will have the risk of both missed detection as well as false positive.

A better strategy is to determine T based on the concept of minimizing the expected risk. The expected risk is a weighted sum of the probability of not detecting false locations (missed detection) and the probability of categorizing a correctly estimated location as false (false positive). The weights can be chosen based on the relative importance placed on missed detection as well as on false positives. Hence, we next develop an approach to determine T based on this idea of minimizing the expected risk.

Let p indicate the probability that an incorrect location is accepted to be correct in step 2 of the NCC above thereby resulting in no increment in N_c . This could happen if both L_{jn} and L_j are incorrect but the values themselves are consistent. It could also happen if L_j is incorrect but L_{jn} is correct with the values being consistent. These events could happen because the node in question is compromised while the neighbor of interest might or might not have been compromised. Variations due to the vagaries of wireless signal propagation might also result in such events.

We also let q be the probability that a correct location is rejected in step 2 of the neighborhood consistency check thereby resulting in an increment in N_c . One way due to which this could happen is because the neighbor in question has been compromised and hence the estimated location of the neighbor is incorrect while the node has a correctly estimated location with the values being inconsistent.

Let π_G indicate the a-priori probability that the estimated location of a sensor node is correct. We also let C_m denote the cost of missed detection and C_f denote the cost of false positive. Then the expected risk is given by $C_m P_m (1 - \pi_G) + C_f \pi_G P_f$ where P_m is the probability of missed detection and P_f is the probability of false positive. Then the value of threshold T which minimizes the expected risk for node j is given by the following

theorem⁷.

Theorem 1: Let threshold $T_{opt} = \left\lceil \frac{\ln \frac{C_f \pi_G}{C_m (1 - \pi_G)} + k_j \ln \frac{1 - q}{p}}{\ln((1 - p)(1 - q)/pq)} \right\rceil$. The following is the optimal strategy for each sensor node j .

When $p + q < 1$,⁸ we conclude that the estimated location is false if and only if $N_c \geq T_{opt}$

When $p + q > 1$, we conclude that the estimated location is false if and only if $N_c < T_{opt}$

Let $p + q = 1$. If $\pi_G \geq \frac{C_m}{C_f + C_m}$ the estimated location is assumed to be correct and otherwise incorrect.

Proof of Theorem 1: Follows from standard decision theory [27] (pp.5-9).

We next provide an example to illustrate the above concepts. Consider a sensor network with 200 nodes. Each node has a variable number of neighbors. We focus on one of the nodes say node with id 10. Assume that this node has 10 neighbors. Consider the counter N_c of this node. Assume that the counter value is $N_c = 4$ after the verification is done. So the question is whether the estimated location of the sensor node can be accepted. In the absence of any other information, we could follow the majority rule according to which the estimated location is accepted if it is consistent with the majority of the neighbors. Following this rule will lead to acceptance of the estimated location of this node.

But now consider extra information known by the system. Assume that the estimated location is wrong for 1 percent of the nodes in the system (this value could be based on observing historical data). This could also imply that the attacker compromises 1 percent of the nodes. Hence $\pi_G = 0.99$. Also let $p = 0.1$ and $q = 0.1$. Further let equal weightage be placed on both missed detection as well as false positives. Hence, $C_m = C_f = 1$. In such a case the threshold $T_{opt} = 3.96$ which implies that the estimated location of this sensor node should not be accepted following Theorem 1.

As remarked earlier, there are several unrealistic assumptions made in the derivation of T_{opt} in Theorem 1. Specifically, p and q are not independent. This is due to the fact that the locations are symmetric. In addition, the errors in the estimated locations (which govern each decision under step 2 of NCC) might also not be independent due to the correlations associated with wireless transmission characteristics. Further, when a compromised insider targets a neighborhood, the errors in the estimated locations of a node and its neighbors might

⁷We assume that every decision in step 2 of NCC is independent here which is not true in reality. We will investigate the impact of relaxing this assumption via simulations later

⁸Note that p and q are probabilities associated with different events. Thus, $p + q$ can exceed 1.

also not be independent. We use simulations to study the impact of all this.

For the simulations, we consider a sensor network with 500 nodes and an average node density of 63 nodes in every node neighborhood. The neighborhood of a sensor is governed by its transmission radius which is 0.2units. We assume that information about π_G is available. We show results averaged over 10 trials in figure 4. We plot π_G on the x-axis while we show the expected risk on the y-axis. In this case we assume that both missed detection and false positives are equally important.

In this scenario, we assume that every node can be compromised with a probability $(1 - \pi_G)$ after an initial interval. A compromised node will have an estimated location anywhere outside its true neighborhood. Given the primitives such as robust ranging and inverse verifiable multilateration, this could possibly require that the node collude with other compromised nodes in order to provide the consistent set of values to the sink. A non-compromised node on the other hand will have an estimated location perturbed from the actual location. The perturbation is assumed to be given by a gaussian variable with zero mean and a standard deviation proportional to the transmission radius. In such a case, we have observed that the p and q values vary in a small range for all trials. We consider a point in this range as the estimated values of p and q denoted as p_e and q_e . Note that we also calculate the real p and q values for every trial.

We study the expected risks associated with five different strategies and show the performance of these strategies in figure 4. We next explain the five strategies. The first strategy is the analytical strategy. This corresponds to the theoretical expected risks calculated as $C_m P_m (1 - \pi_G) + C_f \pi_G P_f$ while using p and q and not p_e and q_e . The next four strategies relate to simulations. Note that the expected risk for simulations is given by the sum of missed detections and false positives. In case of "sim-ideal", we consider simulations where the threshold T_{opt} is calculated using p and q while also adhering to the assumptions of independence for every decision (in step 2 of NCC) used in the derivation of T_{opt} . This implies that every decision is independent and is impacted by the p and q values; this corresponds to a hypothetical situation. In case of "sims-perfect information", we calculate T_{opt} using p and q values while considering the dependency among the various decisions. The fourth strategy "sims-imperfect information" is also similar except that T_{opt} is calculated using p_e and q_e values. The fifth strategy is the simple strategy where we use the majority rule. This is denoted as "sims-majority" in the figure. Thus, here T for every node j corresponds to $k_j/2$ where k_j is the number of neighbors of node j .

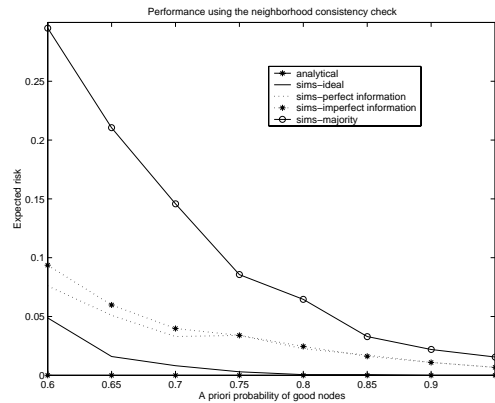


Fig. 4. The expected risks associated with various strategies

We observe from this figure that the analytical risk is in the range of 10^{-3} and hence effectively zero. We also observe that the majority rule performs very badly when the number of compromised nodes is very large. We have observed that this gap increases when the node density increases. The performance of Sim-perfect and sim-imperfect is also seen to be very close thereby showing the insensitivity of the approach to the exact values of p and q . The impact of lack of independence though is somewhat significant as can be seen from the difference between "sims-ideal" and "sims-imperfect information"; yet we do much better than the majority rule.

We have so far implicitly assumed that each insider knows π_G, p , and q . Note that π_G is the most difficult to ascertain as it is directly controlled by the intruders and they can dynamically vary π_G . The optimum aggregation rule at a given π_G can be substantially suboptimal at a different value of π_G . Thus, the adversary can significantly increase the system risk by selecting a π_G which is different from that assumed by the sink. The aggregation strategies need to be robust to such dynamic variations. We can then use strategies such as the min-max strategy [27] which do not require any knowledge of π_G .

VI. SECURITY ANALYSIS

In this section, we analyze the security of SLA.

A. Robust RSS ranging security and tradeoffs

As we show in Sections III-C.1 and III-D.1 the precision of the end-to-end delay estimation depends on the speed and accuracy of the used hardware. With Mica2 motes, the variation in the estimation of the end-to-end delay can be approximately $20\mu s$ and the synchronization error can be up to $30\mu s$, giving a total leverage of around $50\mu s$ with the attacker. Thereby, robust RSS ranging will be resilient to jam-and-replay attacks that cannot amplify or reduce the signal power within $50\mu s$.

If an attacker jams and replays the packet, the end-to-end delay will at least be increased by a complete packet transmission time at the physical layer. Note that we are already accounting for a very strong attacker model, wherein we are completely neglecting the channel access delay in replaying the packet. The maximum radio speed of Mica2 motes is 38.4Kbps. Thereby even a small packet payload of 16 bytes (with a fixed TinyOS header) will take a few milliseconds to get transferred at the physical layer and hence, even in the best case scenario (for the attacker), the end-to-end delay will get increased by a few milliseconds. This is roughly one-two orders of magnitude more than the maximum leverage available with the attacker. Hence, we conclude that robust RSS ranging is resilient to an external mote-class attacker. However, we do not neglect the possibility of these attacks if the attacker employs sufficiently fast and sophisticated hardware that can perform jam-and-replay attacks within $50\mu s$. To counter these attacks, the sensors would need to implement more precise clocks.

Comparison with RF distance bounding. Radio(RF) distance bounding and radio(RF) ranging are two primitives, similar to robust RSS ranging, that have been used by researchers to accurately counter distance reduction attacks against localization schemes [4], [35]. These primitives rely on the speed of the radio channel (i.e. the speed of light) to prevent distance reduction (i.e., the attacker cannot reduce the measured distance as it cannot *speed-up* radio signals). Although both these primitives are more accurate than robust RSS ranging, they both require nanosecond precision in time measurements and in some instances even nanosecond processing. If the hardware does not support these, the protocols cannot be implemented. Robust RSS ranging enables a good tradeoff between security and complexity. Based on an estimated attacker speed and sophistication (and the value put on the localization service), one can design a system with an appropriate precision of the end-to-end delay estimation. Given its low complexity and cost, robust RSS ranging therefore provides, in a number of scenarios, a more viable and lightweight solution than RF distance-bounding.

B. Resiliency to external attacks

Since robust RSS ranging can detect jam-and-replay range modification attacks, and inverse verifiable multilateration can prevent signal amplification attacks, this implicitly means that attacks by external attackers are entirely prevented within our localization scheme. Namely, if each distance modification is detected by the sink, and if the sensor is trusted, the location of the sensor will be correctly computed (based on the correctly computed

distances). This holds assuming that the attacker cannot jam-and-replay within the expected variation time of the end-to-end delay. The security of our localization scheme, in terms of external attacks, can therefore be directly linked with the precision of the time synchronization. We empirically calculated this precision to be around $50\mu s$ on mote-class devices. The more precise the time synchronization between ANs and sensors is, the harder it is for the attacker to spoof the position of the nodes. In conclusion, the location of each non-compromised sensor node will be securely computed within our scheme and our attacker model.

C. Resiliency to internal attacks

Although robust RSS ranging does provide protection against external attacks on localization, it does not shield against attacks by compromised nodes (internal attackers). Note that these attacks are much harder to protect against. A compromised sensor node can manipulate the computation of its own location by reporting false readings in step 3 of the SLA. Such manipulation though has to be consistent with the robust RSS ranging and inverse verifiable multilateration mechanisms made use of by the sink. This typically ⁹ would require the compromised nodes to collude with each other. In such a case, two colluding compromised nodes have to be able to combine their information to spoof their presence in a different location. If not, the attack cannot be considered successful. Even assuming that this is possible, we have seen that NCC algorithm with proper choice of parameters can minimize the system risks. It does this by detecting false locations while also minimizing the rejection of correct locations. Note though that it is possible that some false locations might not be detected which is a risk associated with this low cost strategy.

VII. RELATED WORK

In the last decade, a number of indoor localization systems were proposed, based notably on infrared [37], ultrasound [38], [26], received radio signal strength [3], [15], [6] and radio time-of-flight [20], [12] techniques. These localization techniques were also extended to wireless ad hoc networks [9], [5], [34], [25], [30], [24], [11], [7].

Recently, a number of secure distance and location verification schemes have been proposed. Brands and Chaum [4] proposed a distance bounding protocol that can be

⁹A single compromised node can only fake its position within the vicinity of the ANs it hears. This might not be a security risk given the large resolution associated with the RSS technique as well as the ranges associated with ANs and sensor nodes

used to verify the proximity of two devices connected by a wired link. Sastry, Shankar and Wagner [29] proposed a distance bounding protocol, based on ultrasonic and radio wireless communication. In [16], the authors propose a mechanism called “packet leashes” that aims at preventing wormhole attacks. Kuhn [17] proposed an asymmetric security mechanism for navigation signals. Capkun and Hubaux [35] propose a technique called verifiable multilateration, based on distance-bounding, which enables a local infrastructure to verify positions of the nodes. Lazos et al. [19] proposed a set of techniques for secure positioning of a network of sensors based on directional antennas and distance bounding. Li et al. [21] and Liu et al. [22] propose statistical methods for securing localization in wireless sensor networks. In [36], Capkun et al. propose a secure localization scheme based on hidden and mobile base stations. In [31], Sedihpour et al. demonstrated the feasibility of distance reduction and enlargement attacks on ultrasonic ranging systems.

VIII. CONCLUSION

In this paper, we presented a RSS-based secure localization scheme for sensor networks. We have shown that this scheme can be implemented on current sensor networking platforms (e.g., Mica2) at no additional hardware cost to the existing nodes. The proposed scheme relies on end-to-end propagation delay measurements to detect distance enlargement attacks and on inverse verifiable multilateration to detect distance reduction attacks by external attackers. Internal attacks are addressed through a mechanism called neighborhood consistency check, which can tolerate a high fraction of compromised nodes.

REFERENCES

- [1] <http://www.chipcon.com>.
- [2] Mica sensor platform. <http://www.xbow.com>.
- [3] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *Proceedings of the IEEE Conference on Computer Communications (InfoCom)*, volume 2, pages 775–784, 2000.
- [4] S. Brands and D. Chaum. Distance-bounding protocols. In *Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, pages 344–359. Springer-Verlag New York, Inc., 1994.
- [5] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, October 2000.
- [6] P. Castro, P. Chiu, T. Kremenek, and R. Muntz. A Probabilistic Room Location Service for Wireless Networked Environments. In *Proceedings of the Third International Conference Atlanta Ubiquitous Computing (Ubicomp)*, volume 2201. Springer-Verlag Heidelberg, September 2001.
- [7] Haowen Chan, Mark Luk, and Adrian Perrig. Using Clustering Information for Sensor Network Localization. In *Proceedings of IEEE Conference on Distributed Computing in Sensor Systems (DCOSS 2005)*, June 2005.
- [8] D. Coppersmith and M. Jakobsson. Almost Optimal Hash Sequence Traversal. In *Proceedings of the 6th International Conference on Financial Cryptography (FC)*. Springer, March 2002.
- [9] L. Doherty, K. Pister, and L. El Ghaoui. Convex position estimation in wireless sensor networks. In *Proceedings of the IEEE Conference on Computer Communications (InfoCom)*, April 2001.
- [10] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. *SIGOPS Operating System Review*, 36(SI):147–163, 2002.
- [11] T. Eren, D. Goldenberg, W. Whiteley, Y.R. Yang, A.S. Morse, B.D.O. Anderson, and P.N. Belhumeur. Rigidity, computation, and randomization in network localization. In *Proceedings of the IEEE Conference on Computer Communications (InfoCom)*, 2004.
- [12] R.J. Fontana. Experimental Results from an Ultra Wideband Precision Geolocation System. *Ultra-Wideband, Short-Pulse Electromagnetics*, May 2000.
- [13] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the ACM Conference on Networked Sensor Systems (SenSys)*, pages 138–149. ACM Press, 2003.
- [14] S. Ganeriwal, S. Čapkun, S. Han, and M. Srivastava. Secure Time Synchronization Service for Sensor Networks. In *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, 2005.
- [15] J. Hightower, G. Boriello, and R. Want. SpotON: An indoor 3D Location Sensing Technology Based on RF Signal Strength. Technical Report 2000-02-02, University of Washington, 2000.
- [16] Y.-C. Hu, A. Perrig, and D. B. Johnson. Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks. In *Proceedings of the IEEE Conference on Computer Communications (InfoCom)*, San Francisco, USA, April 2003.
- [17] M. G. Kuhn. An Asymmetric Security Mechanism for Navigation Signals. In *Proceedings of the Information Hiding Workshop*, 2004.
- [18] L. Lazos and R. Poovendran. Serloc: Secure range-independent localization for wireless sensor networks. in *Proceedings of WISE*, page 2130, October 2004.
- [19] L. Lazos, S. Čapkun, and R. Poovendran. ROPE: Robust Position Estimation in Wireless Sensor Networks. In *Proceedings of IPSN*, 2005.
- [20] J.-Y. Lee and R.A. Scholtz. Ranging in a Dense Multipath Environment Using an UWB Radio Link. *IEEE Journal on Selected Areas in Communications*, 20(9), December 2002.
- [21] Z. Li, W. Trappe, Y. Zhang, and B. Nath. Robust Statistical Methods for Securing Wireless Localization in Sensor Networks. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*, 2005.
- [22] D. Liu, P. Ning, and W. Du. Attack-Resistant Location Estimation in Sensor Networks. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*, 2005.
- [23] D. Lymberopoulos, Q. Lindsey, and A. Savvides. An empirical analysis of radio signal strength variability in ieee 802.15.4 networks using monopole antennas. In *Proceedings of the European Workshop on Wireless Sensor Networks (EWSN)*, 2006.
- [24] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of the ACM Conference on Networked Sensor Systems (SenSys)*, pages 50–61. ACM Press, 2004.
- [25] D. Niculescu and B. Nath. DV Based Positioning in Ad hoc Networks. *Journal of Telecommunication Systems*, 22(4):267–280, 2003.
- [26] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *Proceedings of the*

- ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 32–43. ACM Press, 2000.
- [27] H. V. Poor. *An Introduction to Signal Detection and Estimation*. Springer-Verlag, 1994.
- [28] K. Romer. Time synchronization in ad hoc networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 173–182. ACM Press, 2001.
- [29] N. Sastry, U. Shankar, and D. Wagner. Secure Verification of Location claims. In *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, pages 1–10. ACM Press, September 2003.
- [30] A. Savvides, C.-C. Han, and M. B. Srivastava. Dynamic fine-grained localization in Ad-Hoc networks of sensors. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 166–179. ACM Press, 2001.
- [31] S. Sedihpour, S. Čapkun, S. Ganeriwal, and M. Srivastava. Implementation of attacks on ultrasonic ranging systems (demo). In *Proceedings of the ACM Conference on Networked Sensor Systems (SenSys)*, 2005.
- [32] J. van Greunen and J. Rabaey. Lightweight time synchronization for sensor networks. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 11–19. ACM Press, 2003.
- [33] M. Čagalj, S. Čapkun, RamKumar Rengaswamy, Ilias Tsigkogiannis, M. Srivastava, and Jean-Pierre Hubaux. Integrity (I) codes: Message Integrity Protection and Authentication Over Insecure Channels. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2006.
- [34] S. Čapkun, M. Hamdi, and J.-P. Hubaux. GPS-free Positioning in Mobile Ad-Hoc Networks. *Cluster Computing*, 5(2), April 2002.
- [35] S. Čapkun and J.-P. Hubaux. Secure positioning of wireless devices with application to sensor networks. In *Proceedings of the IEEE Conference on Computer Communications (InfoCom)*, 2005.
- [36] S. Čapkun, M. Srivastava, and M. Čagalj. Secure Localization with Hidden and Mobile Base Stations. In *Proceedings of the IEEE Conference on Computer Communications (InfoCom)*, 2006.
- [37] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The Active Badge Location system. *ACM Transactions on Information Systems*, 10(1):91–102, 1992.
- [38] A. Ward, A. Jones, and A. Hopper. A New Location Technique for the Active Office. *IEEE Personal Communications*, 4(5), October 1997.
- [39] J. S. Warner and R. G. Johnston. Think GPS Cargo Tracking = High Security? Think Again. *Technical report, Los Alamos National Laboratory*, 2003.
- [40] K. Whitehouse and D. Culler. Macro-calibration in Sensor/Actuator Networks. *Mobile Networks and Applications Journal (MONET): Special Issue on Wireless Sensor Networks*, June 2003.
- [41] K. Whitehouse, C. Karlof, A. Woo, F. Jiang, and D. Culler. The effects of ranging noise on multihop localization: an empirical study. In *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks (IPSN)*, 2005.
- [42] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 46–57, New York, NY, USA, 2005. ACM Press.