

SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks

Loukas Lazos and Radha Poovendran
Network Security Lab, Dept. of EE,
University of Washington, Seattle, WA 98195-2500
{l_lazos, radha}@ee.washington.edu

ABSTRACT

In many applications of wireless sensor networks (WSN), sensors are deployed un-tethered in hostile environments. For location-aware WSN applications, it is essential to ensure that sensors can determine their location, even in the presence of malicious adversaries. In this paper we address the problem of *enabling sensors of WSN to determine their location in an un-trusted environment*. Since localization schemes based on distance estimation are expensive for the resource constrained sensors, we propose a range-independent localization algorithm called SeRLoc. SeRLoc is distributed algorithm and does not require any communication among sensors. In addition, we show that SeRLoc is robust against severe WSN attacks, such as the *wormhole attack*, the *sybil attack* and *compromised sensors*. To the best of our knowledge, ours is the first work that provides a security-aware range-independent localization scheme for WSN. We present a threat analysis and comparison of the performance of SeRLoc with state-of-the-art range-independent localization schemes.

Categories and Subject Descriptors

C.2 [Computer System Organization]: Computer-Communication Networks; C.2.1 [Network Architecture and Design]: Distributed networks—*Network topology*

General Terms

Algorithm, Security, Performance, Design

Keywords

Secure Localization, Wireless sensor networks, range-independent

This work was supported in part by the following grants: NSF grand ANI-0093187, ARO grant DAAD19-02-1-0242 and by the Collaborative Technology Alliance (CTA) from ARL, DAAD19-01-2-0011. The views and conclusions contained here are those of the authors and should not be interpreted as representing the official policies or endorsements, either express or implied, of NSF, ARO, ARL, or the U.S. Government or any of its agencies.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WiSe'04, October 1, 2004, Philadelphia, Pennsylvania, USA.
Copyright 2004 ACM 1-58113-925-X/04/0010 ...\$5.00.

1. INTRODUCTION

Wireless ad hoc sensor networks (WSN) operate in the absence of a pre-deployed infrastructure, are self-configurable, low cost and can be rapidly deployed. Hence, such networks enable a variety of consumer applications, such as emergency rescue, disaster relief, smart homes and patient monitoring, industrial applications, such as structural health monitoring and environmental control, and military applications, such as target identification and tracking.

Many of the applications proposed for WSN require knowledge of the origin of the sensing information. For example, in order to identify a crack in the rotating part of a motor or in the arch of a bridge, we need the location of the sensors that detect high stress forces or structure anomalies. Furthermore, location is assumed known in the realization of many network operations such as, routing protocols where a family of geographically aided algorithms have been proposed [1, 2], or security protocols where location information is used to prevent threats against services [3–5].

WSN may be deployed in hostile environments with sensors operating unsupervised. Hence, an adversary can interrupt the functionality of location-aware applications by exploiting the vulnerabilities of the localization scheme. Though many localization techniques have been proposed for wireless ad hoc networks [6–16], almost no research has been presented in securing the localization scheme against security threats.

Since sensors are hardware and power limited, we consider computationally efficient methods such as symmetric cryptography and efficient hash functions [17], to prevent attacks against the localization scheme. In addition, since distance measurements are known to be susceptible to distance enlargement/reduction [18], we do not use any such measurements to infer the sensor location. We refer to methods that are not using distance measurements as range-independent localization schemes [6–9, 14, 16].

In this paper we make the following contributions: (i) We propose *SeRLoc*, a novel range-independent localization scheme for WSN, that achieves decentralized, resource-efficient sensor localization. (ii) We propose security mechanisms for SeRLoc that allow each sensor to determine its location *even* in the presence of well known threats on WSN such as the wormhole attack [3, 19, 20], the sybil attack [21, 22] and sensor compromise. (iii) We provide simulation studies that show that SeRLoc localizes sensors with higher accuracy than state-of-the-art decentralized range-independent localization schemes [6–9], and is robust against varying sources of error.

The remainder of the paper is organized as follows. In Section 2 we describe the secure localization problem and state our network model. Section 3 describes our localization scheme and Section 4 presents a threat analysis. In Section 5 we present the need for securing the state-of-the-art range-independent localization schemes.

In Section 6 we present related work. In Section 7 we evaluate the performance of SeRLoc compared to other range-independent localization schemes. Section 8 presents our conclusions.

2. PROBLEM STATEMENT & NETWORK MODEL

2.1 Problem Statement

We study the problem of *enabling nodes of a wireless sensor network to determine their location even in the presence of malicious adversaries*. This problem will be referred to as *Secure Localization*. Note that secure localization is a different problem from verifying the location claim of a sensor, known as location verification [23, 24]. Location verification is not addressed in this paper. We consider secure localization in the context of the following design goals: (a) decentralized implementation, (b) resource efficiency, and (c) robustness against security threats.

2.2 Network Model

Network generation: We assume that the network consists of a set of sensor nodes N of unknown location and a set of specially equipped nodes L we call *locators*, with known location and orientation. Locators' position can be acquired through GPS receivers¹ [25]. We assume that all network nodes are deployed randomly in a specific network region of area \mathcal{A} .

The random deployment of the network nodes can be modeled as a *spatial homogeneous Poisson point process* [26]. The random placement of the locators with a density $\rho_L = \frac{|L|}{|\mathcal{A}|}$ ($|\cdot|$ denotes the cardinality of a set) is equivalent to a sequence of events following a homogeneous Poisson point process of rate ρ_L . Given that L events occur in area \mathcal{A} , these events are uniformly distributed within \mathcal{A} . The random deployment of sensors with a density $\rho_s = \frac{|N|}{|\mathcal{A}|}$, is equivalent to a random sampling of the \mathcal{A} with rate ρ_s [26].

Let LH_s denote the set of locators heard by a sensor s . The probability that s hears exactly k locators $P(|LH_s| = k)$ is equal to the probability that k locators are deployed within an area of size πR^2 , where R is the locator-to-sensor communication range. Since locators deployment follows a spatial Poisson process (randomly deployed in a specific area):

$$P(|LH_s| = k) = \frac{(\rho_L \pi R^2)^k}{k!} e^{-\rho_L \pi R^2}. \quad (1)$$

Using (1), we compute the probability that *every* sensor hears *at least* k locators. The random sensor deployment implies statistical independence in the number of locators heard by each sensor and hence:

$$\begin{aligned} P(|LH_s| \geq k, \forall s \in N) &= P(|LH_s| \geq k)^{|N|} \\ &= (1 - P(|LH_s| < k))^{|N|} \\ &= \left(1 - \sum_{i=0}^{k-1} \frac{(\rho_L \pi R^2)^i}{i!} e^{-\rho_L \pi R^2}\right)^{|N|}. \end{aligned} \quad (2)$$

Antenna model: Sensors are assumed to be equipped with omnidirectional antennas having a sensor-to-sensor communication range r . Locators are assumed to be equipped with sectored antennas with M sectors. We assume a directivity gain $G(M)$ and an idealized angular reception [27]. However, the ideal sector assumption is relaxed during our security evaluation and simulation study,

¹Though GPS signals can be spoofed, knowledge of the coordinates of several nodes is essential to achieve any kind of node localization.

where we consider imperfect sectorization. We assume that locators transmit with higher power than sensors and hence have a locator-to-sensor communication range of $R > r$. Due to the directivity of the locators' antennas, and the higher locator transmitting power, the locator-sensor link is asymmetric. The sensor-to-locator communication range is $d = rG^{\frac{1}{\gamma}}$, where γ is the signal attenuation factor [27]. We also assume that locators can simultaneously transmit in all their antenna sectors.

Note that to achieve a communication range ratio $\frac{R}{r}$, locators need to transmit with power $P_L = (\frac{R}{r})^\gamma (P_s/G)$, where P_s is the sensor transmitting power. Given that sensors are very low power devices, the higher transmit power assumption on the locator side is a reasonable one. A typical sensor has a communication range from $3 \sim 30m$ with a transmission power of $P_s = 0.75mW$ [28]. Hence, guards needs to transmit with a power $P_g = 75mW$ to achieve a communication range ratio $\frac{R}{r} = 10$ when $\gamma = 2$ even without the use of directional antennas.

Network initialization: We assume that sensors and locators can be pre-loaded with cryptographic quantities before deployment.

Attacks not addressed: In this paper we do not consider attacks against the physical layer such as frequency jamming. Spread spectrum [31] and coding [32] is known to be an efficient mechanism to shield the physical layer against jamming attacks. Also, we do not consider any attack against the Medium Access Control (MAC) protocol that may lead to a denial-of-service (DoS). *We also assume that locators are trusted and cannot be compromised by an adversary*. Dealing with compromised locators is an on-going work.

3. SERLOC: A SECURE LOCALIZATION SCHEME

In this Section we describe how our SEcure Range-independent LOcalization scheme (*SeRLoc*) enables sensors to determine their location based on beacon information transmitted by the locators, and present the security mechanisms that protect the location computation, in the presence of malicious adversaries.

3.1 Location Determination

In SeRLoc, sensors determine their location based on the beacon information transmitted by the locators. Figure 1 illustrates the idea of the scheme. Each locator transmits different beacons at each antenna sector with each beacon containing, (a) the locator's coordinates, (b) the angles of the antenna boundary lines, with respect to a common global axis.

If a sensor hears a beacon transmitted at a specific antenna sector of a locator L_i , it has to be included within that sector. Given that sensors know the locators' communication range R a priori, and acquire the locator's coordinates and sector boundary lines through the beacons, they can identify the region within which they reside by computing the overlap between all the sectors that they hear. Each sensor determines its location as the center of gravity (CoG) of the overlapping region computed via the beacon information. The CoG is the least square error solution given that a sensor can lie with equal probability at any point of the overlapping region.

In Figure 1(a), sensor s is able to hear locators $L_1 \sim L_4$. Sensor s determines its position by executing a four step algorithm. In *Step 1*, the sensor collects the beacons from all locators it can hear. In *Step 2*, it determines an approximate search area within which it is located based on the coordinates of the locators heard. In *Step 3*, it computes the overlapping antenna sector region using a majority vote scheme. In *Step 4*, it determines its location as the center of gravity of the overlapping region. We now describe all four steps in detail.

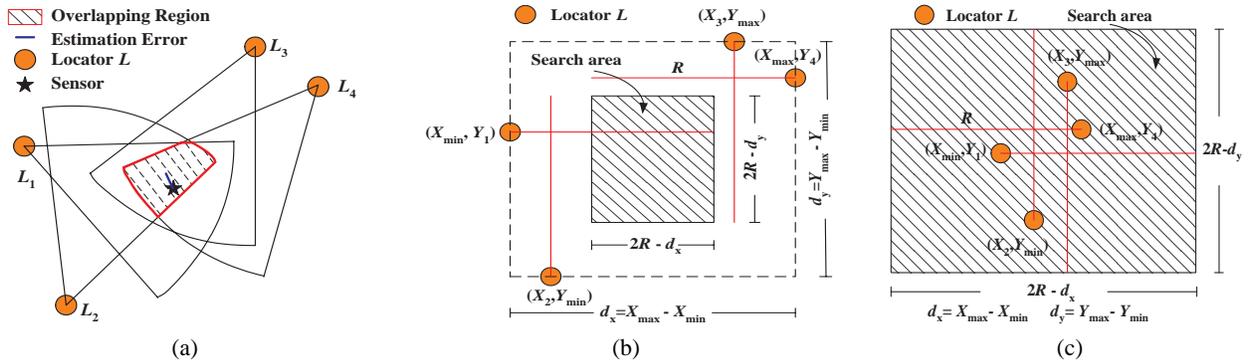


Figure 1: (a) Locators $L_1 - L_4$ transmit beacons at each sector. Sensor s estimates its location as the Center of Gravity C_oG of the overlapping region of the sectors that include it. (b) Step 2: determination of the search area, a rectangular area of size less than R^2 , (c) a rectangular area of size greater than R^2 .

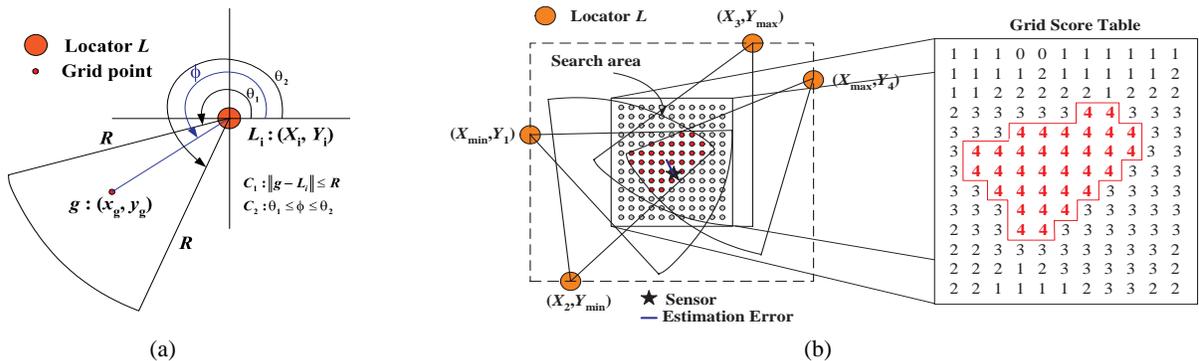


Figure 2: (a) Step 3: Grid-sector test for a point g of the search area. (b) Steps 3,4: Placement of a grid of equally spaced points in the search area, and the corresponding grid score table. The sensor estimates its position as the centroid of all grid points with the highest score.

- *Step 1: Locators heard:* The sensor collects information from all the locators that it can hear. A sensor s with coordinates (x_s, y_s) can hear all locators LH_s with coordinates (X_i, Y_i) that lie within a circle of radius R , centered at (x_s, y_s) .

$$LH_s = \{\|s - L_i\| \leq R, \quad i = 1 \dots |L|\}. \quad (3)$$

- *Step 2: Search area:* The sensor computes a search area where it will attempt to locate itself. Initially, the sensor finds the minimum X_{min}, Y_{min} and maximum X_{max}, Y_{max} locator coordinates form the set LH_s .

$$\begin{aligned} X_{min} &= \min_{L_i \in LH_s} X_i, & X_{max} &= \max_{L_i \in LH_s} X_i, \\ Y_{min} &= \min_{L_i \in LH_s} Y_i, & Y_{max} &= \max_{L_i \in LH_s} Y_i. \end{aligned} \quad (4)$$

Observe in Figure 1(b) that if sensor s can hear locator i with coordinates (X_{min}, Y_i) , s has to be located *left* from the vertical boundary of $(X_{min} + R)$. Similarly, s has to be located *right* from the vertical boundary of $(X_{max} - R)$, *below* the horizontal boundary of $(Y_{min} + R)$, and *above* the horizontal boundary of $(Y_{max} - R)$.

The dimensions of the rectangular search area are $(2R - d_x) \times (2R - d_y)$ where d_x, d_y are the horizontal distance $d_x = X_{max} - X_{min} \leq 2R$ and the vertical distance $d_y = Y_{max} - Y_{min} \leq 2R$, respectively. In Figure 1(b), the search area is $A_s < R^2$, since $d_x > R$ and $d_y > R$. In Figure 1(c), $d_x < R$ and $d_y < R$, and hence, $A_s > R^2$.

- *Step 3: Overlapping region-Majority vote:* In Step 3, sensors determine the overlapping region of all sectors. Since it would

be computationally expensive for each sensor to attempt to analytically determine the overlapping region, based on the line intersections, we employ a grid scoring system that defines the overlapping region based on majority vote. A grid scoring scheme was also used in [6], though the grid was placed at a fixed region around the sensor, regardless of the positions of the locators.

Grid score table: The sensor places a grid of equally spaced points within the rectangular search area as shown in Figure 2(b). To determine the overlapping area, the sensor keeps a score for every grid point in a grid score table. Initially, all grid points have a zero score. If a point is included in a sector according to a *grid-sector test* described below, the sensor increments its score by one. If not, its score does not change. This process is repeated for all locators heard LH_s , for the same grid. The overlapping region is defined by the points that have the highest score in the grid. In Figure 2(b), we show the grid score table and the overlapping region according to the highest scoring points.

Note that due to the finite grid resolution, the use of grid points for the definition of the overlapping region induces error in the calculation. The resolution of the grid can be increased to reduce the error at the expense of energy consumption due to the increased processing time.

Grid-sector test: Let the coordinates of a grid point g be denoted as (x_g, y_g) . Point g is included in a sector of angles $[\theta_1, \theta_2]$ originating from locator $L_i : (X_i, Y_i)$ if it satisfies two conditions: (C_1) : g has to lie within the communication range of L_i , (C_2) : The angle ϕ of the line connecting L_i and g , has to lie within

$[\theta_1, \theta_2]$.

$$C_1 : \|g - L_i\| \leq R, \quad C_2 : \theta_1 \leq \phi \leq \theta_2. \quad (5)$$

Note that the sensor *do not have to* perform any angle-of-arrival (AOA) measurements. Both the coordinates of the locators and the grid points are known, and hence the sensors can analytically calculate ϕ . In Figure 2(a), we illustrate the grid-sector test, with all angles measured with reference to axis $y = 0$.

- *Step 4: Location estimation:* The sensor determines its location as the centroid of all the grid points that define the overlapping region (highest score in the grid):

$$\tilde{s} : (x_{est}, y_{est}) = \left(\frac{1}{n} \sum_{i=1}^n x_{g_i}, \frac{1}{n} \sum_{i=1}^n y_{g_i} \right), \quad (6)$$

where n is the number of grid points of the overlapping region.

3.2 Security Mechanisms of SeRLoc

We now describe the security mechanisms of SeRLoc that enable the secure location computation.

Encryption: To protect the localization information, we encrypt all beacons transmitted from locators. Since sensors are constrained in both computational power and energy resources, we do not consider asymmetric key cryptography solutions. Instead, sensors and locators share a global symmetric key K_0 , pre-loaded before deployment.

In addition, every sensor s shares a symmetric pairwise key $K_s^{L_i}$ with every locator L_i , also pre-loaded. Since the number of locators deployed is relatively small, the storage requirement at the sensor side is within the storage constraints (a total of $|L|$ keys). For example, mica motes [28] have 128Kbytes of programmable flash memory. Using 64-bit RC5 [29] symmetric keys and for a network with 200 guards, a total of 1.6Kbytes of memory is required to store all the symmetric pairwise keys of the node with all the guards. In order to save storage space at the locator side (locators would have to store $|N|$ keys), the pairwise key $K_s^{L_i}$ is derived by a master key K_{L_i} , using a pseudo-random function [30] h and the unique sensor Id_i : $K_s^{L_i} = h_{K_{L_i}}(Id_i)$. Hence, given an Id_i , a locator can compute its pairwise key with a sensor whenever needed, without having to store any pairwise keys.

Locator ID authentication: The use of a shared symmetric key does not identify the source of the messages that each sensor hears. Hence, in the absence of additional security features, a malicious sensor may impersonate multiple locators. To prevent sensors with access to the shared key K_0 , from injecting false localization information into the network, we require sensors to authenticate the source of the beacons *using collision-resistant hash functions*.

We use the following scheme based on *efficient one-way hash chains* [33], to provide locator ID authentication². Each locator L_i has a unique password PW_i . The password is blinded with the use of a *collision-resistant* hash function such as MD5 [17]. Due to the collision resistance property, it is computationally infeasible for an attacker to find a value PW_j , such that $H(PW_i) = H(PW_j)$, $PW_i \neq PW_j$. The hash sequence is generated using the following equation:

$$H^0 = PW_i, \quad H^i = H(H^{i-1}), \quad i = 1, \dots, n,$$

with n being a large number and H^0 never revealed to any sensor. Each sensor is pre-loaded with a table containing the Id of each locator and the corresponding hash value $H^n(PW_i)$. For a

²Hash chains have been widely used to authenticate the source of a message in many applications including wireless ad hoc networks [35, 36].

network with 200 guards, we need 8 bits to represent node Ids. In addition, hash functions such as MD5 [17] have a 128-bit output. Hence, the storage requirement of the hash table at any node is only 3.4Kbytes. To reduce the storage needed at the locator side, we can employ an efficient storage/computation method for hash chains of time/storage complexity $\mathcal{O}(\log^2(n))$ and compute any hash chain values when needed [34]. We now describe how the hash values authenticate the locator's ID.

Assume that a locator L_i wants to transmit its first beacon. Initially, sensors only know the hash value $H^n(PW_i)$. The locator includes $(H^{n-1}(PW_i), j)$ in the beacon transmission, with the index $j = 1$ (first hash value published). Every sensor that hears the beacon can authenticate the locator ID only if $H(H^{n-1}(PW_i)) = H^n(PW_i)$. After verification, the sensor replaces $H^n(PW_i)$ with $H^{n-1}(PW_i)$ in its memory and increases the hash counter by one, so as to perform only one hash operation in the reception of a second message from the same locator L_i . The index j is included in the beacons, so that sensors can re-synchronize with the current published hash value, in case of loss of some intermediate hash values. The beacon of Locator L_i has the following format:

$$L_i : \{ (X_i, Y_i) \parallel (\theta_1, \theta_2) \parallel (H^{n-j}(PW_i), j) \}_{K_0},$$

where \parallel denotes the concatenation operation and $\{m\}_K$ denotes the encryption of message m with key K . Note that our location Id authentication does not prevent the replay of a message that originated from a locator. However, it allows a sensor to ensure that the message was generated by a locator. We will show that this condition is sufficient to secure our localization scheme against possible attacks.

4. THREAT ANALYSIS

In this Section we show that SeRLoc is resilient to severe types of attacks such as the wormhole attack [3, 19, 20], the sybil attack [21, 22] and compromised sensors. *Note that our goal is not to prevent the attacks that may be harmful in many network protocols, but to allow sensors to determine their location, even in the presence of such attacks.*

4.1 The Wormhole Attack

Threat model: To mount a wormhole attack, an attacker initially establishes a direct link between two points in the network. We will refer to the attacker's link as *wormhole link*. Once the wormhole link is established, the attacker eavesdrops messages at one end of the link, referred as the *origin point*, tunnels them through the wormhole link and replays them at the other end, referred as the *destination point*. The wormhole attack is very difficult to detect, since it can be launched without compromising any host, or the integrity and authenticity of the communication [3, 19, 20].

Existing solutions: The authors in [3] describe a solution for the wormhole attack, based on geographical and temporal packet leases. The use of geographical leases assumes known node location, and hence is not adequate for securing a localization algorithm. The use of temporal leases requires all nodes to have tightly synchronized clocks and demands computational power which, according to the authors, is beyond the capability of sensors [3]. The authors in [19, 37] propose a defense against wormholes based on measuring the time of flight of a message in a challenge-reply scheme. Such a solution assumes that sensors are able to perform time measurements of nanosecond precision and hence, requires very accurate clocks at each sensor. In addition, distance estimates based on time of flight are sensitive to distance enlargement errors.

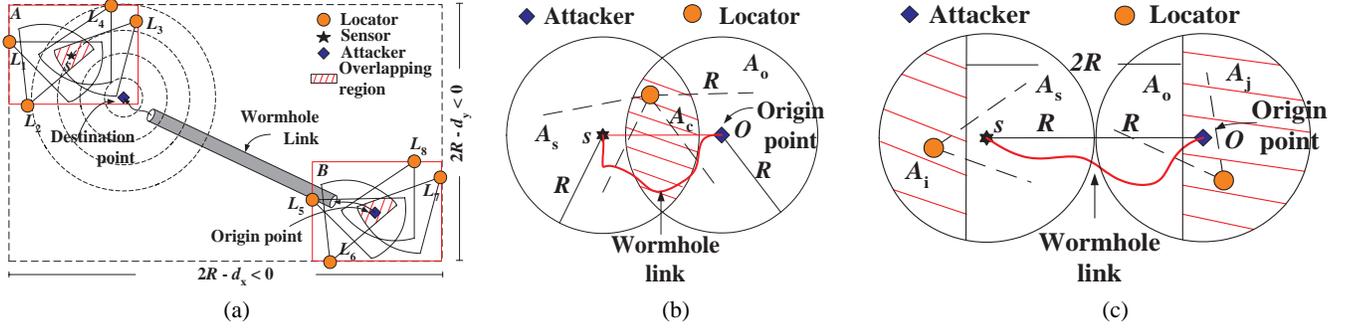


Figure 3: (a) Wormhole attack: An attacker records beacons in area B , tunnels them via the wormhole link in area A and re-broadcasts them. (b) Sector uniqueness: a sensor s cannot hear two sectors from the same locator. (c) Communication range violation: a sensor cannot hear two locators that are more than $2R$ apart.

Wormhole attack against SeRLoc: In the case of SeRLoc, an attacker records the beacons transmitted from locators at the origin point and replays them at the destination point, thus providing false information about the locators heard in a specific neighborhood. In Figure 3(a), an attacker records beacons at region B , tunnels them via the wormhole link in region A and replays them, thus leading sensor s to believe that it can hear locators $\{L_1 \sim L_8\}$.

Detecting wormholes: We now show how a sensor can detect a wormhole attack using two properties of SeRLoc.

1. *Sector uniqueness property:* If an attacker replays a transmission of a locator L_i that is directly heard to sensor s , the sensor can detect the attack using the sector uniqueness property. The attacked sensor will detect that it is infeasible to hear two sectors of a single locator³ (replay of the same sector is of no use to the attacker). The beacon of L_i directly heard to sensor s will reach s earlier than any replay, assuming that the locator transmits in all sectors simultaneously. In addition, the sensor will acquire the latest published value of the hash chain of L_i through the direct link. Hence, any replay containing an already published hash value will not be authenticated.

In Figure 3(b), A_s denotes the area where locators heard to sensor s can reside (circle of radius R centered at s), A_o denotes the area where locators heard at the origin point of the attack can reside (circle of radius R centered at O) and A_c denotes the common area $A_c = A_s \cap A_o$. The detection probability $P(SU)$ due to the sector uniqueness property is equal to the probability that at least one locator lies within an area of size A_c . Using equation (2) from Section 2.2,

$$\begin{aligned} P(SU) &= P(|LH_{A_c}| \geq 1) = 1 - P(|LH_{A_c}| = 0) \\ &= 1 - e^{-\rho_L A_c}, \end{aligned} \quad (7)$$

where LH_{A_c} denotes the set of locators heard by sensor s that lie inside area A_c . In Figure 4(a), we show the detection probability $P(SU)$ for locator densities ρ_L , for distances $0 \leq \|s - O\| \leq 3R$, normalized over R , and for $\frac{R}{r} = 10$. We observe that if $\|s - O\| \geq 2R$, the sector uniqueness property cannot be used to detect a wormhole attack ($P(SU) = 0$).

2. *Communication range violation property:* Every locator directly heard to a sensor s is less than R units away from s as stated

³We treat multipath effects and imperfect sectorization as replay attacks and execute the location resolution algorithm to be presented, to determine the sensor location.

by (3), i.e. $\|s - L_i\| < R, \forall L_i \in LH_s$. Hence, two locators $L_i, L_j \in LH_s$, heard to s , cannot be more than $2R$ apart, i.e. $\|L_i - L_j\| \leq 2R$. If the sensor hears two locators for which the communication constraint is violated, i.e. $\|L_i - L_j\| > 2R$, it detects that is under attack.

We now compute the detection probability $P(CR)$ due to the communication range violation property. Consider Figure 3(c), where $\|s - O\| = 2R$. If any two locators within A_s, A_o have a distance larger than $2R$ the attack is detected. Though $P(CR)$ is not easily computed analytically, we can derive a lower bound on $P(CR)$ as follows: consider the event where there is at least one locator in the shaded area A_i (the vertical lines defining A_i, A_j , are perpendicular to the line connecting s, O) and at least one locator in the shaded area A_j . If such an event occurs, then $\|L_i - L_j\| > 2R$ and the attack is detected. Hence,

$$\begin{aligned} P(CR) &= P(\|L_i - L_j\| > 2R) \\ &\geq P(CR \cap (|LH_{A_i}| > 0 \cap |LH_{A_j}| > 0)) \quad (8) \\ &= P(CR | (|LH_{A_i}| > 0 \cap |LH_{A_j}| > 0)) \\ &\quad P(|LH_{A_i}| > 0 \cap |LH_{A_j}| > 0) \quad (9) \\ &= P(|LH_{A_i}| > 0 | |LH_{A_j}| > 0) \quad (10) \\ &= (1 - P(|LH_{A_i}| = 0))(1 - P(|LH_{A_j}| = 0)) \\ &= (1 - e^{-\rho_L A_i})(1 - e^{-\rho_L A_j}), \quad (11) \end{aligned}$$

where (8) follows from the fact that the probability of the intersection of two events is always less or equal to the probability of one of the events, (9) follows from the definition of the conditional probability, (10) follows from the fact that when $LH_{A_i} > 0$ and $LH_{A_j} > 0$, we always have a communication range violation ($P(CR | (|LH_{A_i}| > 0 \cap |LH_{A_j}| > 0)) = 1$), and (11) follows from (2). It can be proven that the lower bound on $P(CR)$ is maximized when $A_i^* = \max_i \{A_i\}$ subject to the constraint $A_i = A_j$. In Figure 4(b), we show the lower bound on $P(CR)$, by setting $A_i^* = \max_i \{A_i\} \ni A_i = A_j$.

3. *Detection probability P_{det} :* By combining the two detection techniques, we compute a lower bound on the detection probability P_{det} of a wormhole attack as:

$$\begin{aligned} P_{det} &= P(SU \cup CR) \\ &= P(SU) + P(CR) - P(SU)P(CR) \\ &\geq (1 - e^{-\rho_L A_c}) + (1 - e^{-\rho_L A_i^*})^2 - \quad (12) \\ &\quad (1 - e^{-\rho_L A_c})(1 - e^{-\rho_L A_i^*})^2, \quad (13) \end{aligned}$$

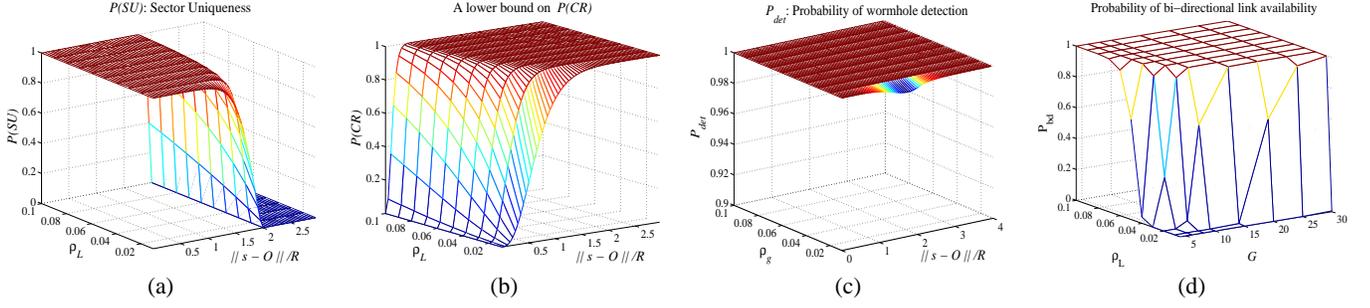


Figure 4: Wormhole detection probability for $\frac{R}{r} = 10$ based on, (a) sector uniqueness: $P(SU)$, (b) communication range violation, a lower bound on $P(CR)$. (c) Wormhole detection probability P_{det} (Z axis ranges from $[0.9, 1]$). (d) Probability that all sensors have a bi-directional link with at least one locator for $r = 4$. G denotes the antenna directivity.

where (12), follows from the fact that A_c, A_i^*, A_j do not overlap, and (13) follows from (7), (11). In Figure 4(c), we show the lower bound on P_{det} for $R \in [0, 4R]$, based on (13). For values of $R > 4R$, $P_{det} = P_{det}(4R)$, since $A_i = A_j = \pi R^2$. Note that the lowest detection probability is $P_{det} \geq 99.48\%$, attained at $\rho_L = 0.01$. From Figure 4(c), we observe that a wormhole attack is detected with a probability very close to unity.

Location resolution algorithm: Although a wormhole can be detected using one of the two detection mechanisms, it creates location ambiguity to the sensor. To resolve the location ambiguity a sensor under attack executes the *Attach to Closer Locator Algorithm* (ACLA).

Assume that a sensor authenticates a set of locators LH_s , but detects that it is under attack. Initially, the sensor computes the rectangle A defined by the intersection of the lines in (4). The sensor places a point grid within A and performs the grid point test to acquire the grid score table (Step 2 of SeRLoc). Then, using the grid score table, it identifies disjoint regions D_i that score higher than a threshold th . The threshold th is a design parameter determined in relation to the locator density ρ_L . For all regions D_i the sensor computes the CoG_i and identifies the locators $L_i \in LH_s$ closest to each CoG_i . For each L_i the sensor encrypts a nonce η_i , with the pairwise key $K_s^{L_i}$, concatenates its Id_s and buffers the message. Once, all messages have been created, the sensor broadcasts them sequentially and awaits for the first authentic reply. The sensor identifies the locator L'_i closest to it by the first reply, and determines its location as CoG'_i of the region D'_i , closest to L'_i .

Attach to Closer Locator Algorithm (ACLA)

1. $A : \{X \geq X_{min}, X \leq X_{max}, Y \geq Y_{min}, Y \leq Y_{max}\}$.
2. Place a point grid in A and execute the grid score test.
3. Compute $CoG_i \forall D_i$ with score $\geq th$.
4. $\forall D_i$, find $L_i^* \in LH_s$

$$L_i^* = \min \|L_i - CoG_i\|.$$
5. $\forall L_i^*$, broadcast $\{\eta_i || q\}_{K_s^{L_i^*}} || Id_s$.
6. Identify $L'_i \in L^*$ that replies first with the correct nonce.
7. Sensor location: CoG'_i of the region corresponding to L'_i .

Note that in Step 5 of ACLA, the messages sent to different locators have a time difference up to $(k-1)T_s$, where T_s is the message transmission time and k is the total number of messages transmitted. To account for the time difference we propose the following enhancement. The sensor attaches to each message a sequence

number q . The sequence number indicates the reverse order by which the encrypted messages are transmitted. The first message has a sequence number $q = k - 1$, while the last message has a sequence number $q = 0$. Locators that hear the sensor's message, wait for a $q * T_s$ time before they reply with the nonce η_i .

By adding transmission delay at the locator side, we compare messages transmitted at different times only based on their round-trip time, without using any high accurate clocks to measure time of flight⁴.

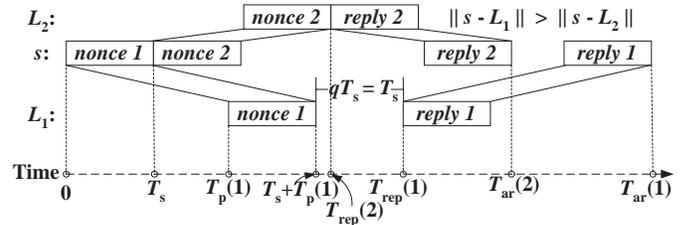


Figure 5: T_s denotes transmission time, $T_p(i)$ denotes the propagation time of one bit for L_i , $T_{rep}(i)$ denotes the reply time for L_i and $T_{ar}(i)$ denotes the arrival time of reply $T_{rep}(i)$. Since $\|s - L_1\| > \|s - L_2\|$, $T_{ar}(1) > T_{ar}(2)$.

In Figure 5, we assume that for the two locators L_1, L_2 , $\|s - L_1\| > \|s - L_2\|$ and hence, $T_p(1) > T_p(2)$ where $T_p(i)$ denotes the propagation time for one bit to arrive from s at locator L_i . If the sensor starts transmitting the nonces at time $T = 0$, nonce η_1 with sequence number $q = 1$ arrives at L_1 at time $T_s + T_p(1)$, while nonce η_2 with $q = 0$ arrives at L_2 at time $2T_s + T_p(2)$. Locator L_1 waits for $qT_s = T_s$ time before replying, thus starting its reply at time $T_{rep}(1) = 2T_s + T_p(1)$, while L_2 replies immediately at time $T_{rep}(2) = 2T_s + T_p(2)$, where $T_{rep}(i)$ is the time when L_i replies. Since $T_p(1) > T_p(2)$, it follows that $T_{rep}(1) > T_{rep}(2)$. Hence, using our delay scheme the closest locator will always reply first independent on the transmission sequence of the nonces. In addition, since the closest locator replies first it is guaranteed that its message will arrive at s first with a time difference of $2(T_p(1) - T_p(2))$ (we ignore any processing time which is equal to both locators). Note that if L_2 is heard to s through a direct link and L_1 is heard through a wormhole link, $T_p(1) \gg T_p(2)$.

⁴Note that we have assumed that locators are trusted and have not been compromised by the attacker. In case of locator compromise a more elaborate algorithm involving multiple locators and a majority vote scheme can be employed at the expense of algorithmic complexity and higher number of locators needed.

To execute ACLA, a sensor must be able to communicate bi-directionally with at least one locator. The probability $P_{s \rightarrow L}$ of a sensor having a bi-directional link with at least one locator can be computed as :

$$P_{s \rightarrow L} = 1 - e^{-\rho_L \pi r^2 G^{\frac{2}{\gamma}}}. \quad (14)$$

The probability P_{bd} that *all* sensors can bi-directionally communicate with at least one locator is:

$$P_{bd} = (1 - e^{-\rho_L \pi r^2 G^{\frac{2}{\gamma}}})^{|N|}. \quad (15)$$

In Figure 4(d), we plot P_{bd} vs. the locator density ρ_L and the antenna directivity G , for $\frac{r}{\lambda} = 10$. From (15), we can properly choose ρ_L, G, r , so as every sensor has a bi-directional link with at least one locator with probability very close to unity, and hence, resolve any location ambiguity.

4.2 Sybil Attack and Compromised Sensors

Threat model: In the sybil attack [21,22] an attacker impersonates multiple network entities by assuming their identities. Unlike the wormhole attack, in the sybil attack model the attacker is able to compromise the communication (gain access to the cryptographic quantities usually by compromising network entities), obtain multiple node identities and insert bogus information into the network. A solution for the sybil attack for WSN was recently proposed in [22].

Sybil attack against SeRLoc In SeRLoc, sensors do not rely on other sensors to compute their location. Hence, an attacker has no incentive to assume sensor identities. Similarly compromised sensors cannot directly impact the localization. An attacker can only affect the localization mechanism, if it successfully impersonates several locators.

To impersonate locators, an attacker has to compromise the global key K_0 used by locators to transmit beacons. Once K_0 has been compromised, the attacker can obtain published values of the hash chains of the locators it hears. Since the sensor always has the latest published values from the locators that it can directly hear, an attacker can only impersonate locators that are not directly heard by the sensor under attack. Using the acquired hash values not heard at the sensor under attack, the attacker can impersonate multiple locators, and create arbitrary beacon messages.

Defense against the sybil attack: Though we do not provide any mechanism to prevent an attacker from impersonating locators (except for the ones directly heard to a sensor), we can still determine the position of the sensor in the presence of a sybil attack, as long as the pairwise keys between the locators and sensors are not compromised. The sensor will detect ambiguity between its actual location and the location(s) indicated by the impersonated locators. Using the *location resolution algorithm* presented in the wormhole attack, the sensor can determine its actual location, since the closest locator to its actual position, will always reply first. The attacker has no way to decrypt the nonce, encrypted with the pairwise key, or encrypt any kind of reply. Hence, the sensor will successfully compute its location, assuming that locators are not compromised.

5. THREATS ON EXISTING LOCALIZATION SCHEMES

We now provide examples of exploitation of the vulnerabilities of state-of-the-art range-independent localization schemes, that lead to incorrect location computation. Vulnerabilities for range-dependent schemes have been presented in [18]. Due to space limitation we

examine three popular schemes, Dv-hop [9], Amorphous localization [7] and APIT [6]

5.1 Dv-hop and Amorphous localization

In Dv-hop [9], following a strategy similar to distance vector routing, each node discovers the shortest path in number of hops to every other node. Reference points compute the average length of one hop, based on the hop count to other reference points, and flood the network with the hop estimate. Nodes use the hop size estimate and the number of hops to compute their distance to at least three reference points and perform triangulation to determine their location. Amorphous localization [7] employs a similar strategy with the exception of computing the average hop size offline through an approximate formula [38]. Dv-hop and Amorphous localization

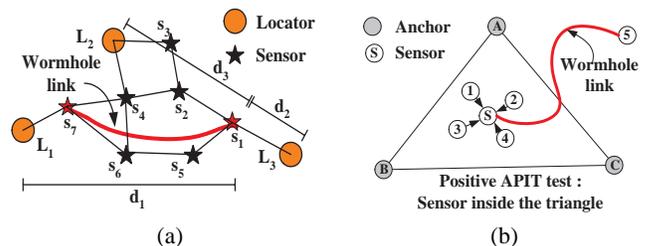


Figure 6: (a) Dv-hop, Amorphous localization (b) APIT localization test.

face the same security threats as any distance vector routing protocol. Attacks against the distance vector routing protocols in a wireless ad hoc network have been documented in [19, 35, 39, 40].

As an example, consider a wormhole link between nodes s_1 and s_7 of Figure 6(a). Both s_1 and s_7 will assume false number of hops (two) to reference points L_1, L_3 , respectively. In addition, L_1, L_3 will make an incorrect estimate on the average hop size, when they acquire false hop count measurements. Nodes s_1, s_2 will compute a much smaller distance to reference points L_1, L_3 and triangulation will provide a highly inaccurate position estimate.

5.2 APIT Localization

In APIT localization [6], a sensor relies on neighbor sensor information to determine if it is located inside or outside a virtual triangle defined by three reference points called anchors. For example, in Figure 6(b), sensor s measures the power to three anchors A, B, C , and also gathers the measurements of all neighboring sensors 1 ~ 4. If no neighbor of s is further from/closer (according to the power measurements) to all three anchors A, B, C simultaneously, s assumes that it is inside the triangle $\triangle ABC$. Otherwise, s assumes it is outside $\triangle ABC$. The sensor s repeats the APIT test for all 3-tuples of anchors heard, and estimates its position as the center of gravity of the overlapping region of the triangles for which the APIT test was positive (The sensor was inside the triangle).

Assume a wormhole link between s and node 5. Node five is further from all three anchors A, B, C and hence, the APIT test will indicate the s is outside $\triangle ABC$. In another attack a malicious sensor can fail an arbitrary number of APIT tests for its neighbors, by advertising false power levels and hence, critically increase the localization error.

From our examples it becomes evident that localization schemes need to be enhanced with security mechanisms to ensure the correct location determination in the presence of adversaries.

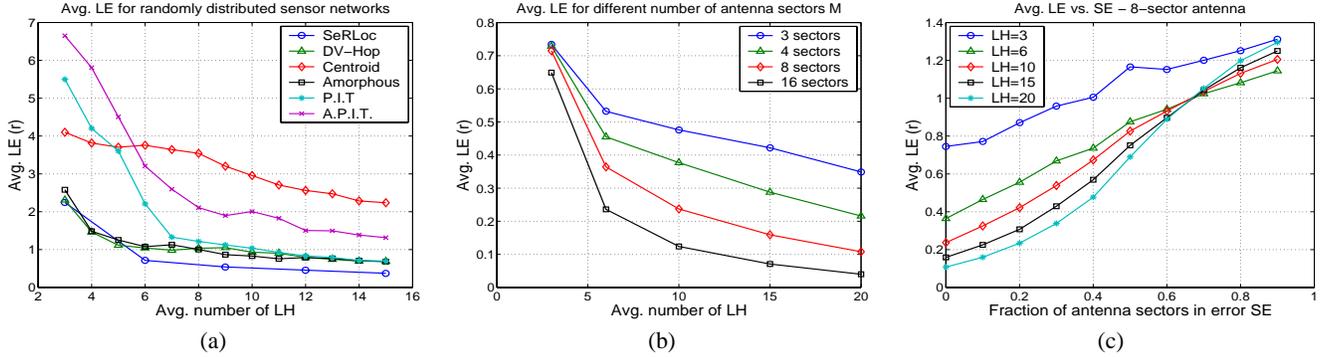


Figure 7: (a) Average localization error \overline{LE} vs. average number of locators heard \overline{LH} for a network of $|N| = 5,000$ and locator-to-sensor ratio $\frac{R}{r} = 10$. (b) \overline{LE} vs. \overline{LH} for different number of antenna sectors. (c). \overline{LE} vs. sector error SE for different number of locators heard \overline{LH} .

6. RELATED WORK

Localization schemes can be classified to range-dependent and range-independent based schemes. In range-dependent schemes, nodes determine their location based on distance or angle estimates to some reference points with known coordinates. Such estimates may be acquired through different methods such as time of arrival (TOA) [15, 25], time difference of arrival (TDOA) [11, 13], angle of arrival (AOA) [10], or received signal strength indicator (RSSI) [12]. Attacks and countermeasures for range-dependent schemes have been presented in [18]. To the best of our knowledge no prior research has been presented in securing range-independent schemes. Due to space limitation, we focus on range-independent schemes, since SeRLoc belongs to that category.

In the range-independent localization schemes, nodes determine their location without use of time, angle, or power measurements. Nodes depend on beacons, or connectivity information to compute their location. We already presented DV-hop [9], amorphous localization [7], and APIT [6] in Section 5. In [8], the authors propose *Centroid*, an outdoor localization scheme, where reference points broadcast beacons with their coordinates. Nodes estimate their position as the centroid of the locations of all the reference points that they hear. Centroid has a very simple implementation and low communication cost. However, it results in a crude approximation of node location.

Two methods have been proposed that utilize connectivity information to determine the node location. In [14], the authors formulate a semi-definite program, based on the connectivity constraints and obtain the optimal position estimates that satisfy all constraints. In [16] the authors use multidimensional scaling to acquire an arbitrary rotation of the network topology. If three nodes know their location, the network topology can be mapped to the absolute node location. Both schemes in [14, 16], require centralized computation and hence are not used for comparison in the performance evaluation.

7. PERFORMANCE EVALUATION

In this section we compare the performance of SeRLoc with state-of-the-art localization techniques, namely Dv-Hop [9], Amorphous localization [7], Centroid localization [8], APIT [6] and its theoretical ideal version PIT [6]. We show that SeRLoc has superior performance in localization error and requires significantly fewer resources than other methods. We also show that SeRLoc is robust against both error in the locators' coordinates and estimation of the antenna sector that includes the sensors.

7.1 Simulation Setup

We randomly distributed 5,000 sensors within a 100x100 rectangular area. We also randomly placed locators within the same area and computed the average localization error as:

$$\overline{LE} = \frac{1}{|N|} \sum_i \frac{\|\tilde{s}_i - s_i\|}{r}, \quad (16)$$

where N is the set of sensors, \tilde{s}_i is the sensor estimated position, s_i is the real position and r is the sensor communication range.

7.2 Localization Error vs. Locators heard

In our first experiment, we investigated the impact of the average number of locators heard \overline{LH} in the localization error. In order to provide a fair comparison of SeRLoc with other methods, we normalize \overline{LH} for SeRLoc by multiplying \overline{LH} with the number of sectors used. Hence, for example, when $\overline{LH} = 9$, with SeRLoc using three sectors, every sensor hears on average three locators.

In Figure 7(a), we show the \overline{LE} vs. \overline{LH} with SeRLoc using three sectors and $\frac{R}{r} = 10$. We observe that SeRLoc is superior to all other range-independent algorithms compared. Note that SeRLoc achieves a localization error of $0.5r$, with very few locators ($\overline{LH} = 12$ which is equivalent to four locators with 3-sectored antennas). From equation (1) in Section 2.2, we can calculate the locator density required to achieve a specific \overline{LH} as $\rho_L = \frac{\overline{LH}}{A_s}$. For example to achieve $\overline{LE} = 0.5r$, we need a locator density of $\rho_L = \frac{4}{\pi R^2} = 0.0032$ ($R = 20$).

7.3 Localization Error vs. Antenna Sectors

In our second experiment, we examined the impact of the number of antenna sectors M on the average localization error \overline{LE} . In Figure 7(b), we show the \overline{LE} vs. \overline{LH} , for different number of antenna sectors. We can observe that for $\overline{LH} = 3$ the \overline{LE} is comparable for all values of M . However, as the value of \overline{LH} increases, the \overline{LE} decreases more rapidly for higher number of antenna sectors, due to the fact that the overlapping region becomes smaller when the antenna sectors become narrower.

The gain in the localization accuracy, comes at the expense of hardware complexity at the locator, since more complex antenna designs have to be employed to generate the sectoring. Additionally, errors in the estimation of the antenna sector where a sensor is included, become more frequent, since more sensors are located at the boundary between two sectors.

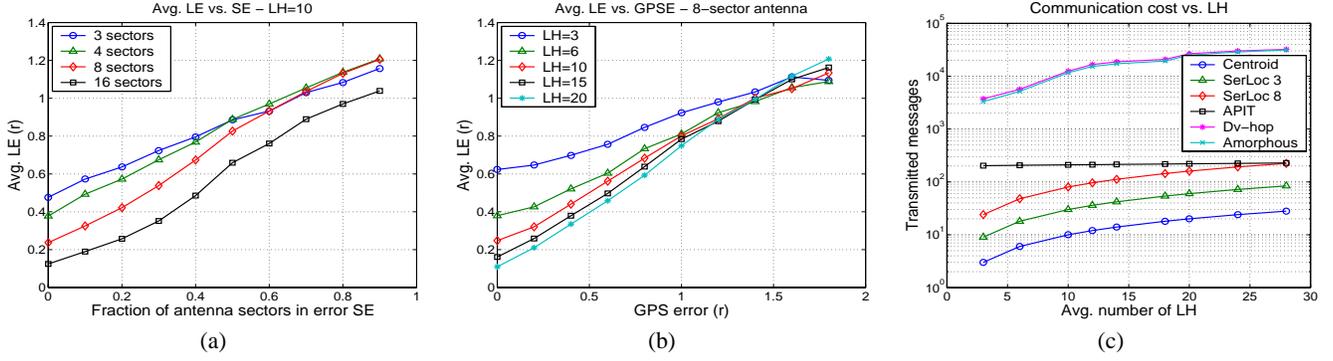


Figure 8: (a) Average localization error \overline{LE} vs. sector error SE for different number of antenna sectors for a network of $|N| = 5,000$ and $\frac{R}{r} = 10$. (b) \overline{LE} vs. locator GPS error in units of r for different average number of locators heard \overline{LH} . (c) Communication cost vs. \overline{LH} , for a network of 200 sensors.

7.4 Localization Error vs. Sector Error

Sensors may be located close to the boundary of two sectors of a locator, or be deployed in a region with high multipath effects. In such a case, a sensor may falsely assume that it is located in another sector, than the actual sector that includes it. We refer to this phenomenon as sector error (SE) and define it as:

$$SE = \frac{\# \text{ of sectors falsely estimated}}{\overline{LH}}. \quad (17)$$

A sector error of 0.5 indicates that *every* sensor falsely estimated the sectors of half the locators heard. In Figure 7(c), we show the \overline{LE} vs. the SE for different values of \overline{LH} , and 8-sector antennas. We observe that the \overline{LE} does not grow significantly large (larger than the sensor communication range r), until a fraction of 0.7 of the sectors are falsely estimated.

SeRLoc algorithm is resilient to sector error due to the majority vote scheme employed in the determination of the overlapping region. Even if a significant fraction of sectors are falsely estimated, these sectors do not overlap in the same network area and hence, score low in the grid-sector test (Step 3 of SeRLoc).

Note that the for a $SE > 0.7$, \overline{LE} increases with \overline{LH} . When the SE grows beyond a threshold, the falsely estimated sectors dominate in the location determination. As \overline{LH} grows, the falsely estimated overlapping region, shrinks due to more overlapping sectors. Hence the center of gravity that defines the sensor estimated location gets further apart than the actual sensor location.

In Figure 8(a), we show the \overline{LE} vs. SE for $\overline{LH} = 10$ and varying number of antenna sectors. We can observe that the narrower the antenna sector the smaller the \overline{LE} , even in the presence of sector error. For a small SE the overlapping region is dominated by the correctly estimated sectors and hence, shrinks with increasing antenna sectors. For large SE the overlapping region is dominated by the falsely estimated sectors and hence, an increase in \overline{LH} does not improve the localization accuracy.

Summarizing our findings for the sector error, we note that SeRLoc is resilient to sector error due to the majority vote mechanism employed in the overlapping region determination. When the sector error becomes very large, most of the sector estimations are in error. Hence, an increase in the number of locators heard, or number of antenna sectors does not decrease the localization error.

7.5 Localization Error vs. GPS Error

GPS, or any alternative localization scheme used to provide locators with their location, may have limited accuracy. To study the

impact of the error in the locators' position, on \overline{LH} , we induced a GPS error ($GPSE$) to every locator of the network. A value of $GPSE = r$ means that every locator was randomly placed at a circle of radius r centered at the locator's actual position.

In Figure 8(b), we show the average localization error \overline{LE} vs. the $GPSE$ in units of r , for varying number of \overline{LH} when locators use 8-sector antennas. We can observe that even for large $GPSE$ the \overline{LE} does not grow larger than $1.2r$. For example, when $GPSE = 1.8r$ and $\overline{LH} = 3$, $\overline{LE} = 1.1r$. According to Figure 7(a), Dv-hop and amorphous localization require $\overline{LH} = 5$ to achieve the same performance in complete absence of $GPSE$, while APIT requires $\overline{LH} = 12$ to reduce the $\overline{LE} = 1.1r$, with no $GPSE$ induced in the locators' positions. Note that once the $GPSE$ error becomes significantly large (over $1.6r$) an increase in \overline{LH} does not improve the accuracy of the position estimation.

7.6 Communication Cost vs. Locators Heard

In this section we analyse the communication cost of SeRLoc and compare it with the communication cost of the existing range-independent localization algorithms. In Figure 8(c), we show the communication cost in number of transmitted messages vs. \overline{LH} , when 200 sensors are randomly deployed.

We observe that DV-hop and Amorphous localization, have significantly higher communication cost compared to all other algorithms, due to the flood-based approach for the beacon propagation. The centroid scheme, has the lowest communication cost ($|L|$) since it only transmits one beacon from each locator to localize the sensor. APIT requires $|L| + |N|$ beacons to localize the sensors, while SeRLoc requires $|ML|$ number of beacons, where L is the set of locators and M is the number of antenna sectors.

Under the assumption that the number of sensors is much higher than the number of locators, ($|N| \gg |L|$), SeRLoc has a smaller communication than APIT, since SeRLoc is independent of the number of sensors deployed. In addition, SeRLoc achieves low localization error for smaller values of \overline{LH} , and hence requires a smaller number of reference points.

8. CONCLUSION

We addressed the problem of secure localization in WSN. We proposed a range-independent, decentralized, localization scheme called SeRLoc, that allows sensors to determine their location in an un-trusted environment with the assistance of a small number of trusted entities. We showed how the security mechanisms of SeRLoc combined with its inherent geometric properties, can pro-

vide accurate location estimation even in the presence of severe security threats in WSN, such as the wormhole and sybil attack. For the wormhole attack, we provided an analytical evaluation of the probability of success against SeRLoc. Our simulation studies showed that SeRLoc localizes sensors with higher accuracy than state-of-the-art range-independent localization schemes, while requiring fewer reference points and lower communication cost. Moreover, our simulation showed that SeRLoc is resilient to sources of error such as error in the location of the reference points as well as error in the sector determination.

9. REFERENCES

- [1] Y. Ko and N. Vaidya, Location-Aided Routing (LAR) in Mobile Adhoc Networks, In *Proc. of MOBICOM 1998*, Dallas, TX, USA, October 1998.
- [2] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward, A Distance Routing Effect Algorithm for Mobility (DREAM), In *Proc. of MOBICOM 1998*, Dallas, TX, USA, October 1998.
- [3] Y. Hu, A. Perrig, and D. Johnson, Packet Leashes: A Defense Against Wormhole Attacks in Wireless Ad Hoc Networks, In *Proc. of INFOCOM 2003*, San Francisco, CA, USA, April 2003.
- [4] D. Liu, P. Ning, Location-based pairwise key establishments for static sensor networks, In *Proc. of SASN 2003*, Fairfax, VA, October 2003.
- [5] L. Lazos and R. Poovendran, Energy-Aware Secure Multicast Communication in Ad-hoc Networks Using Geographic Location Information, In *Proc. of IEEE ICASSP 2003*, Hong Kong, China, April 2003.
- [6] T. He, C. Huang, B. Blum, J. Stankovic and T. Abdelzaher, Range-Free Localization Schemes in Large Scale Sensor Network, In *Proc. of MOBICOM 2003*, San Diego, CA, USA, September 2003.
- [7] R. Nagpal, H. Shrobe, J. Bachrach, Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network, In *Proc. of IPSN 2003*, Palo Alto, USA, April, 2003.
- [8] N. Bulusu, J. Heidemann and D. Estrin, GPS-less Low Cost Outdoor Localization for Very Small Devices, In *IEEE Personal Communications Magazine*, 7(5):28-34, October 2000.
- [9] D. Nicolescu and B. Nath, Ad-Hoc Positioning Systems (APS), In *Proc. of IEEE GLOBECOM 2001*, San Antonio, TX, USA, November 2001.
- [10] D. Niculescu and B. Nath, Ad Hoc Positioning System (APS) using AoA, In *Proc. of INFOCOM 2003*, San Francisco, CA, USA, March 2003.
- [11] A. Savvides, C. Han and M. Srivastava, Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors, In *Proc. of MOBICOM 2001*, Rome, Italy, July 2001.
- [12] P. Bahl and V. Padmanabhan, RADAR: An In-Building RF-Based User Location and Tracking System, In *Proc. of the IEEE INFOCOM 2000*, Tel-Aviv, Israel, March 2000.
- [13] N. Priyantha, A. Chakraborty and H. Balakrishnan, The Cricket Location-Support System, In *Proc. of MOBICOM 2000*, Boston, MA, USA, August 2000.
- [14] L. Doherty, L. Ghaoui and K. Pister, Convex Position Estimation in Wireless Sensor Networks, In *Proc. of the IEEE INFOCOM 2001*, Anchorage, AK, USA, April 2001.
- [15] S. Čapkun, M. Hamdi and J. Hubaux, GPS-Free Positioning in Mobile Ad-Hoc Networks, In *Proc. of HICSS 2001*, Maui, Hawaii, USA, January 2001.
- [16] Y. Shang, W. Ruml, Y. Zhang and M. Fromherz, Localization from Mere Connectivity, In *Proc. of MOBIHOC 2003*, Annapolis, MD, USA, June 2003.
- [17] R. Rivest, The MD5 Message-Digest Algorithm, RFC 1321, April 1992.
- [18] S. Čapkun, J. Hubaux, Secure Positioning in Sensor Networks, Technical report EPFL/IC/200444, available at, <http://www.terminodes.org/micsPublications.php>.
- [19] S. Čapkun, L. Buttyan, J. Hubaux, SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks, in *Proc. of SASN 2003*, Fairfax, Virginia, October 2003.
- [20] P. Papadimitratos and Z. J. Haas, Secure Routing for Mobile Ad Hoc Networks, in *Proc. of CNDV 2002*, January 2002.
- [21] J. Douceur, The Sybil Attack, In *Proc. of IPTPS 2002*, Cambridge, MA, USA, March 2002.
- [22] J. Newsome, E. Shi, D. Song and A. Perrig, The Sybil Attack in Sensor Networks: Analysis and Defenses, In *Proc. of IPSN 2004*, Berkeley, CA, April 2004.
- [23] B. Waters and E. Felten, Proving the Location of Tamper Resistant Devices, http://www.cs.princeton.edu/bwaters/research/location_proving.ps.
- [24] N. Sastry, U. Shankar and D. Wagner, Secure Verification of Location Claims, In *Proc. of WISE 2003*, San Diego, CA, USA, September 2003.
- [25] B. Hofmann-Wellenhof, H. Lichtenegger and J. Collins, Global Positioning System: Theory and Practice, Fourth Edition, Springer-Verlag, 1997.
- [26] N. Cressie, *Statistics for Spatial Data*, John Wiley & Sons, 1993.
- [27] C. Balanis, *Antenna Theory*, John Wiley & Sons, 1982.
- [28] MICA Wireless Measurement System, available at: http://www.xbow.com/Products/Product_pdf_files/Wireless.pdf/MICA.pdf.
- [29] R. L. Rivest, The RC5 encryption algorithm, In *Proc. of the first Workshop on Fast Software Encryption*, pp. 86-96, 1995.
- [30] D. Stinson, *Cryptography: Theory and Practice*, 2nd edition, CRC Press, 2002.
- [31] R. Pickholtz, D. Schilling, and L. Milstein. Theory of Spread Spectrum Communications - A Tutorial, In the *IEEE Transactions on Communications*, 30(5):855-884, May 1982.
- [32] S. B. Wicker and M.D. Bartz, Type-II Hybrid-ARQ Protocols Using Punctured MDS Codes, In *Proc. of IEEE Transactions on Communications*, April 1994.
- [33] L. Lamport, Password Authentication with Insecure Communication, In *Communications of the ACM*, 24(11):770-772, November 1981.
- [34] D. Coppersmith and M. Jakobsson, Almost optimal hash sequence traversal, In *Proc. of the FC 2002*, Lecture Notes in Computer Science, IFCA, Springer-Verlag, Berlin Germany, 2002.
- [35] Y. Hu, D. Johnson and Adrian Perrig, SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks, In *Proc. of WMCSA 2002*, Calicoon, NY, USA, June 2002.
- [36] A. Perrig, R. Szewczyk, V. Wen, D. Culler and J. Tygar, SPINS: Security Protocols for Sensor Networks, In *Proc. of MOBICOM 2001*, Rome, Italy, July 2001.
- [37] Y. Hu, A. Perrig and D. Johnson, Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols, In *Proc. of WISE 2003*, October 2003.
- [38] L. Kleinrock and J. Slivester, Optimum transmission radii for packet radio networks or why six is a magic number, In *Proc. of the National Telecom Conference*, Pages 4.3.1-4.3.5, 1978.
- [39] S. Čapkun, J. P. Hubaux, BISS: Building secure routing out of an incomplete set of security associations, In *Proc. of WISE 2003*, San Diego, CA, September 2003.
- [40] C. Hu, A. Perrig, D. Johnson, Ariadne: A Secure On Demand Routing Protocol for Ad Hoc Networks, In *Proc. of MOBICOM 2002*, Atlanta, GA, USA, September 2002.