Secure positioning of wireless devices with application to sensor networks

Srdjan Čapkun and Jean-Pierre Hubaux
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne (EPFL) , CH-1015 Lausanne, Switzerland
srdan.capkun@epfl.ch, jean-pierre.hubaux@epfl.ch

Abstract*— So far, the problem of positioning in wireless networks has been mainly studied in a non-adversarial setting. In this work, we analyze the resistance of positioning techniques to position and distance spoofing attacks. We propose a mechanism for secure positioning of wireless devices, that we call Verifiable Multilateration. We then show how this mechanism can be used to secure positioning in sensor networks. We analyze our system through simulations. *Keywords: System design, Simulations. [1]

I. Introduction

Recently, researchers have proposed a number of positioning and distance estimation techniques for wireless networks [1], [2], [3], [4], [5], [6]. However, they all studied these techniques in non-adversarial settings. Distance estimation and positioning techniques are, nevertheless, highly vulnerable to attacks from compromised nodes and external attackers. *Compromised nodes* can report false position and distance information in order to cheat on their locations. *External attackers* can modify (spoof) the measured positions and distances of wireless nodes.

Few proposals for secure distance and location verification have already been proposed. Brands and Chaum [7] propose a distance bounding protocol that can be used to verify the proximity of two devices connected by a wired link. Sastry, Shankar and Wagner [8] propose a new distance bounding protocol, based on ultrasound and radio wireless communication. Both proposals focused on the verification of the distance to a device, or on its presence in a region of interest. Poovendran and Lazos [9] proposed a set of techniques for secure positioning in sensor networks based on directional antennas. Kuhn [10] proposed an asymmetric security mechanism for navigation signals. Both proposals address secure position computation by a node, but not secure position verification (typically by an authority).

In this work, we propose a mechanism for secure position computation and verification of positions of wireless devices. We call our mechanism Verifiable Multilateration (VM). This mechanism is based on the measurements of the time of radio signal propagation (i.e., time-of-flight (ToF)). Verifiable Multilateration consists of conventional multilateration with distance bounding or distance estimation and enables verification of node positions by a set of (at least three) base stations, which do not need to be tightly synchronized.

In Verifiable Multilateration, we primarily make use of the distance bounding protocols; however, as we will show, Verifiable Multilateration can also be used with conventional radio frequency time-of-flight distance estimation techniques. We will show that by using conventional distance estimation instead of distance bounding, some security properties of the Verifiable Multilateration mechanism can still be preserved.

Because of its generality, Verifiable Multilateration can be used to secure positioning in a variety of systems. In this work, we focus on sensor network positioning and we propose SPINE, a system for **S**ecure **P**ositioning **I**n sensor **NE**tworks. This system is based on Verifiable Multilateration and ensures secure positioning of sensors in the presence of adversaries. We present a security and performance analysis of SPINE.

The organization of the rest of the paper is the following. In Section II, we provide a survey of positioning techniques and analyze attacks against them. In Section III, we describe a technique for radio frequency distance bounding. In Sections IV, we describe our technique for position verification called Verifiable Multilateration (VM). In Section V, we present a scheme for secure positioning of a network of sensors. In Section VI, we present an overview of current proposals and techniques for positioning in wireless networks, based on Verifiable Multilateration. We conclude the paper in Section VII.

II. Attacks against position and distance estimation techniques

We now review positioning and distance estimation techniques and analyze their vulnerabilities.

We first shortly present our attacker model. We call an attacker *external* if the attacker cannot authenticate itself as an honest network node to other network nodes or to a central authority. We call a node *compromised* if it is controlled by an attacker and it can authenticate itself to the authority and to other network nodes [11]. 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.

A. Global Positioning System (GPS)

The Global Positioning System is today the most widespread outdoor positioning system for mobile devices. The

[1]The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322 (`http://www.terminodes.org`).

	Dishonest node	Attacker
RSS (Received Signal Strength)	Distance enlargement and reduction	Distance enlargement and reduction
US time-of-flight (ToF)	Distance enlargement and reduction	Distance enlargement and reduction
RF time-of-flight (ToF)	**Distance enlargement and reduction**	**Distance enlargement only**
US distance bounding	Distance enlargement only	Distance enlargement and reduction
RF distance bounding	**Distance enlargement only**	**Distance enlargement only**
Civilian GPS	False position reports	Position spoofing

TABLE I

VULNERABILITIES OF THE POSITIONING AND DISTANCE ESTIMATION TECHNIQUES TO DISTANCE AND POSITION SPOOFING ATTACKS.

system is based on a set of satellites that provide a three dimensional positioning with accuracy of around 3 m. GPS also provides devices with an accurate time reference. GPS, however, has several limitations: it cannot be used for indoor positioning nor for positioning in dense urban regions: in those cases, because of the interferences and obstacles, satellite signals cannot reach the GPS devices. Furthermore, the civilian GPS was never designed for secure positioning. Civilian GPS devices can be "spoofed" by GPS satellite simulators, which produce fake satellite radio signals that are stronger than the real signals coming from satellites. Most current GPS receivers are totally fooled, accepting these stronger signals while ignoring the weaker, authentic signals. GPS satellite simulators are legitimately used to test new GPS products and can be bought for $10k-$50k or rented for just $1k per month. Some simple software changes to most GPS receivers would permit them to detect relatively unsophisticated spoofing attacks [12]. Nevertheless, more sophisticated spoofing attacks would still be hard to detect. Military GPS are protected from position spoofing by codes which cannot be reproduced by the attackers. Recently, Kuhn [10] proposed an asymmetric security mechanism for navigation signals that can be used to secure civilian GPS. This mechanism is, however, vulnerable to some sophisticated attacks involving jamming and fast wormholes.

Even if a mobile node is able to obtain its correct position from the GPS satellites, the authority or another mobile node have no way to verify the correctness of node's position, unless the mobile node is equipped with a trusted software or hardware module [13].

B. Ultrasound (US)

Ultrasound-based systems operate by measuring ToF of the sound signal measured between two nodes. An interesting feature of these systems is that, if used with RF signals, they do not require any time synchronization between the sender and the receiver. The limitations of the US-based systems are that, due to outdoor interferences, US systems can be mainly used indoors, and that the US signals can be animal-unfriendly.

US-based systems are vulnerable to distance reduction and distance enlargement attacks by external attackers and compromised nodes. To reduce the measured distance between two honest nodes, two external nodes can use a fast radio link to transmit the signals faster between the honest nodes. Furthermore, by jamming and replaying the signals at a later time, external attacker can enlarge the measured distances between honest nodes. If conventional US ToF technique is used, a compromised node can also reduce or enlarge the measured distance by laying about the signal sending/reception times or by simply delaying its response to honest nodes.

Recently, Sastry, Shankar and Wagner [8] have proposed a US-based distance bounding technique which resists to distance reduction attacks from compromised nodes; it does not, however, resist to attacks from external attackers. This does not mean that this technique is useless for secure applications; it can still be used for verifying location claims in systems in which attackers have no physical access to the localization region. In [14] Waters and Felten presented a similar technique.

C. Radio (RF)

In techniques based on the Received Signal Strength (RSS), the distance is computed based on the transmitted and received signal strengths. To cheat on the measured distance, a compromised node therefore only needs to report a false power level to an honest node. External attackers can also modify the measured distance between two honest nodes by jamming the nodes' mutual communication and by replaying the messages with higher or lower power strengths.

RF time-of-flight-based systems exhibit the best security properties. In these systems, nodes measure their mutual distance based on the time of propagation of the signal between them. Because RF signals travel at the speed of light, an external attackers can, by jamming and replaying the signals, only increase, but not decrease the measured time-of-flight between the nodes. A compromised node can further cheat on the distance by laying about the signal transmission and reception times.

An RF distance bounding technique proposed by Brands and Chaum [7] exhibits better security properties than conventional RF ToF distance estimation; it allows the nodes to upper bound their distances to other nodes, meaning that it prevents a compromised node from reducing the measured distance. As we will show in Section III in more detail, with RF ToF distance-bounding protocols, external attackers and compromised nodes can only increase, but not decrease the measured distances to honest nodes.

D. Conclusion

We conclude our review with the summary of vulnerabilities of positioning and distance measurement systems shown on

Table I, which illustrates that the RF ToF-based positioning solutions are best suited for secure positioning. The reason is that with RF it is generally possible to perform non-line-of-sight distance estimations; the precision of the system can be very high (15 cm error with Ultra Wide Band systems at a distance of 2 km [15]). Furthermore, the RF ToF distance estimation and distance bounding techniques are the most effective techniques to counter attacks from external attackers and compromised nodes. A potential drawback of these systems is that, because they rely on the speed of light, the devices need to have a fast-processing hardware. In the following section we present in more detail the protocols for RF ToF distance estimation and distance bounding, and we discuss how they can be implemented with current technologies.

III. Distance bounding

Distance bounding techniques are used to upper bound the distance of one device to another (compromised) device. As we indicated in Table I, RF-based distance bounding protocols are vulnerable to distance enlargement attacks but not to distance reduction attacks. Distance bounding protocols are used by a verifier v to verify that a claimant node u being at a distance d_{uv} from a verifier node v, cannot claim to be at a distance $d'_{uv} < d_{uv}$. These protocols were first introduced by Brands and Chaum [7] to prevent Mafia Fraud attacks.

The pseudocode of the distance bounding protocol is shown in figure 1. In the first step of the protocol, the claimant u commits to a random value N_u. The verifier replies with a challenge nonce N_v, sends it to u in a reverse bit order and starts its timer as soon as the last bit of the challenge has been sent. The claimant u responds immediately with $N_v \bigoplus N_u$, upon receiving the challenge from v. Once the verifier has received the response from u it stops the timer and converts the challenge-response time t_{vu} to a distance d_{vu}. In the last step of the protocol, u authenticates itself to v and reveals the decommit value $\hat{d}$. The authentication and the authenticity of d is ensured with a message authentication code (MAC), using a secret key K_{vu} that u and v share. Finally, v verifies if the value N_u received in the time-measuring phase corresponds to the received (commit, decommit) pair $(c, \hat{d})$.

The commitment scheme needs to satisfy two properties: (i) a user who commits to a certain value cannot change this value afterwards (we say that the scheme is *binding*), (ii) the commitment is hidden from its receiver until the sender "opens" it (we say that the scheme is *hiding*). A commitment scheme transforms a value m into a commitment/opening pair (c, d), where c reveals no information about m, but (c, d) together reveal m, and it is infeasible to find $\hat{d}$ such that $(c, \hat{d})$ reveals $\hat{m} \neq m$. Simple commitment schemes can be realized with hash functions, which do not impose high computational requirements on sensor nodes.

The described protocol is suitable for devices that can perform rapid message exchanges, execute XOR operations rapidly, and perform encryption. In the case of RF-based distance bounding, the most important assumptions are that the claimant needs to be able to bound its processing (XOR) to a few nanoseconds, and that the verifier v needs to be able to measure time with nanosecond precision ($1ns$ corresponds to the time that it takes an electromagnetic wave to propagate over 30 cm). This requirement allows the node to perform distance bounding with radio signals with an uncertainty of 30 cm. We are aware that a nanosecond processing and time measurements are achievable only with dedicated hardware. Recent developments in location system show that RF time of flight systems based on Ultra Wide Band (UWB) can achieve nanosecond precision of measured times of signal flight (and consequently of the distances). The tests with Multispectral solution's UWB Precision Asset Location system [16] consisting of active tags and tracking devices show that this system can provide two- and three-dimensional location of objects to within a few centimeters. The range of the system is 100 m indoor and 2km outdoor. The used UWB tags are active and roughly the size of a wristwatch, weighing approximately 40 grams each.

u : **Generate** random nonce N_u
: **Generate** commitment $(c, d) = \mathrm{commit}(N_u)$
$u \rightarrow v$: c
v : **Generate** random nonce N_v
$v \rightarrow u$: N_v *(bits sent from MSB to LSB)*
$u \rightarrow v$: $N_u \bigoplus N_v$ *(bits sent from LSB to MSB)*
v: **Measure** time t_{vu} between sending N_v and receiving $N_u \bigoplus N_v$
$u \rightarrow v$: $N_u, N_v, d, MAC_{K_{uv}}(u, N_u, N_v, d)$
v: **Verify** MAC and verify if $N_u = \mathrm{open}(c, d)$

Fig. 1. Pseudocode of the distance bounding protocol.

In the case of a US-based distance bounding, node processing speed and clock accuracy can be of the order of milliseconds. Thus, US distance bounding can be easily implemented with off-the-shelf components such as microphones and 802.11 wireless cards [8].

IV. Verifiable Multilateration

In Section II, we described security problems related to various positioning and distance estimation techniques and in Section III we showed how the devices can upper-bound their mutual distances. We now propose a technique for position verification that we call *Verifiable Multilateration* (VM). This technique enables a secure computation and verification of the positions of mobile devices in the presence of attackers. By *secure position computation* we mean that base stations compute a correct position of a node in the presence of attacker; by *secure position verification* we mean that the base stations verify a position reported by the node.

Multilateration is a technique for determining the position of a (mobile) device from a set of reference points whose positions are known, based on the distances measured between the reference points and the device. The position of the device in two (three) dimensions can be computed if the device measured its distance to three (four) reference points. As we

already detailed in Section II, distance estimation techniques are vulnerable to attacks from external attackers and from compromised nodes, which can maliciously modify the measured distances. Multilateration is equally vulnerable to the same set of attacks because it relies on distance estimations.

A. Algorithm

Verifiable Multilateration relies on distance bounding. It consists of distance bound measurements from at least three reference points (verifiers) to the considered device (the claimant) and of subsequent computations performed by an authority. For simplicity, we show the algorithm for two dimensional positioning; at the end of this subsection, we briefly explain how a similar algorithm can be applied to the three dimensional case.

The intuition behind verifiable multilateration algorithm is the following. Because of the distance bounding property, the claimant can only pretend that it is more distant from the verifier than it really is. If it increases the measured distance to one of the verifiers, to keep the position consistent, the claimant needs to prove that at least one of the measured distances to other verifiers is shorter than it actually is, which it cannot because of the distance bounding. This property holds only if the position of the claimant is determined within the triangle formed by the verifiers. This can be explained with a simple example: if an object is located within the triangle, and it moves to a different position within the triangle, it will certainly reduce its distance to at least one of the triangle vertices. The same properties hold if an external attacker enlarges distances between verifiers and an honest claimant. This basic intuition behind verifiable multilateration is illustrated in Figure 2a.

The verifiable multilateration algorithm is executed by the verifiers and by the authority as follows.

Verifiable multilateration

$\mathcal{T} = \emptyset$; set of verification triangles around u
$\mathcal{V} = \{v_1, ..., v_n\}$; set of verifiers in the power range of u
1 *For all* $v_i \in \mathcal{V}$, perform distance bounding
from v_i to u and obtain db_i
2 With all $v_i \in \mathcal{V}$, compute the estimate (x'_u, y'_u) of the position by MMSE
3 *If for all* $v_i \in \mathcal{V}$, $|db_i - \sqrt{(x_i - x'_u)^2 + (y_i - y'_u)^2}| \leq \delta$ *then*
for all $(v_i, v_j, v_k) \in \mathcal{V}^3$, *if* $(x'_u, y'_u) \in \triangle(v_i, v_j, v_k)$
then $\mathcal{T} = \mathcal{T} \cup (v_i, v_j, v_k)$
if $|\mathcal{T}| > 0$ *then* position is accepted and $x_u = x'_u$, $y_u = y'_u$
else the position is rejected
else the position is rejected

In step 1 of the algorithm, the verifiers $v_1, ..., v_n$ which are in the power range of the claimant u perform distance bounding to the claimant u and obtain distance bounds $db_1, ..., db_n$. These distance bounds as well as the positions of the verifiers (which are precisely known) are then reported to the authority.

In step 2, the authority computes an estimate (x', y') of the claimant's position; this position is computed by using distance bounds from all verifiers in u's neighborhood, typically by the MMSE of the following system of equations:

$$\text{Let } f_i(x'_u, y'_u) = db_i - \sqrt{(x_i - x'_u)^2 + (y_i - y'_u)^2}$$

The position of u is obtained by minimizing
$F(x'_u, y'_u) = \sum_{v_i \in \mathcal{T}} f_i^2(x'_u, y'_u)$
over all estimates of u

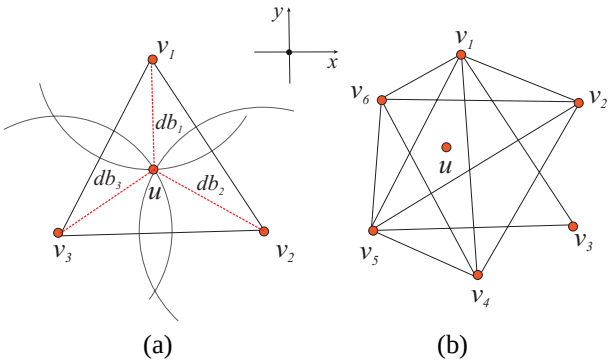

Fig. 2. Examples of Verifiable Multilateration. a) with three verifiers. b) with six verifiers.

In step 3 of the algorithm, the authority runs the following two tests: (i) *δ-test*: for all v_i, does the distance between (x'_u, y'_u) and v_i differ from the measured distance bound db_i by less than the expected distance measurement error δ and (ii) *point in the triangle test:* does (x'_u, y'_u) fall within at least one physical triangle formed by a triplet of verifiers. Note also that we call the triangle formed by the verifiers the *verification triangle*. If both the δ and the point in the triangle tests are positive, the authority accepts the estimated position (x'_u, y'_u) of the claimant as correct; else, the position is rejected.

The expected error δ is a system parameter and depends on the number of verifiers and on the distance estimation techniques used. This error becomes smaller as more verifiers are used to compute (x'_u, y'_u). In most cases, δ can be approximated as 3σ, where σ is the expected standard deviation of the computed position. The following well known test is run to detect if the claimants estimated position (x'_u, y'_u) falls within the verification triangle $\triangle(v_i, v_j, v_k)$:

Point in the triangle test

$f_{ij}(u) = (y'_u - y_i)(x_j - x_i) - (x'_u - x_i)(y_j - y_i)$
$f_{ki}(u) = (y'_u - y_k)(x_i - x_k) - (x'_u - x_k)(y_i - y_k)$
$f_{jk}(u) = (y'_u - y_j)(x_k - x_j) - (x'_u - x_j)(y_k - y_j)$
If $f_{ij}(u) \cdot f_{jk}(u) > 0$ and $f_{jk}(u) \cdot f_{ki}(u) > 0$
then u is in $\triangle(v_i, v_j, v_k)$

The logic behind this test is the following. Three functions $f_{ij}(u), f_{ik}(u), f_{jk}(u)$ are defined, one for each edge of the triangle. $f_{ij}(u)$ is zero for all points u on the line v_i, v_j, and non-zero for all other points. In fact, looking from v_j at v_i, $f_{ij}(u)$ is negative for all points (x, y) on the left side of the edge v_i, v_j, and positive for all points (x,y) on the right side of the edge. The same applies for the other two edges and functions. By combining the output from the three

functions we can compute if a point falls in or out of the triangle $\triangle(v_i, v_j, v_k)$.

If both the δ and the point in the triangle tests are positive, this means that the claimant falls in at least one verification triangle v_i, v_j, v_k, and that distance bounds (db_i, db_j, db_k) are consistent with the estimated position and with each other (Figure 2a). This means that none of the distance bounds (db_i, db_j, db_k) were enlarged.

If any of the distance-bounds db_i differs from the estimated position (x'_u, y'_u) by more than δ, this indicates that there is a possible distance enlargement attack on one or more of the distance bounds that caused such an unexpectedly high error to occur. If a larger number of verification triangles can be formed around u, the authority can try to detect which of the distances are enlarged. Those distances can then be filtered-out and the position can be computed with the remaining set of distances. This detection is performed such that the position of u is computed independently in each triangle. If in a given triangle the computation is successful, then all the distance bounds from the verifiers forming that triangle are considered correct; otherwise, all three distance bounds are considered suspicious.

Detection of enlarged distances:
$\mathcal{C} = \emptyset$; set of verifiers with correctly measured bounds
$\mathcal{NC} = \emptyset$; set of verifiers whose bounds are suspicious
1 *For all* $v_i \in \mathcal{T}$
  if in at least one of the verification triangles
    with v_i the position of u is computed correctly
  then db_i is correct, $\mathcal{C} = \mathcal{C} \cup \{v_i\}$
  else $\mathcal{NC} = \mathcal{NC} \cup \{v_i\}$
2 *For all* $v_i \in \mathcal{NC}$
  if v_i can create a verification triangle
    with any pair $(v_j, v_k) \in \mathcal{C}^2$
  then db_i *is subject to an enlargement attack*
3 With all $v_i \in \mathcal{C}$, compute the estimate (x''_u, y''_u) of the position by MMSE
4 *For all* $v_i \in \mathcal{NC}$, *if* $|db_i - \sqrt{(x_i - x'_u)^2 + (y_i - y'_u)^2}| \leq \delta$
  then db_i *is subject to an enlargement attack*

In this algorithm, the number of verification triangles and the number of enlarged distances will determine if the algorithm can detect which distance(s) is(are) enlarged. Nevertheless, in all cases, even if the number of verifiers is strictly equal to three, the Verifiable Multilateration algorithm will detect any distance enlargement attack (even if only one distance is enlarged), but it will not always be able to detect which distance it is.

Verifiable Multilateration can be also applied to three dimensional positioning. For this, the system requires a minimum of four verifiers, that form a triangular pyramid, within which the secure determination of the claimant's position is possible. The algorithm is then executed in way similar to the two-dimensional case.

B. Properties

In this subsection, we summarize the most important properties of the Verifiable Multilateration mechanism. These are the following:

1) A node located at position p within the triangle/pyramid formed by the verifiers cannot prove to be at another position $p' \neq p$ within the same triangle/pyramid.
2) A node located outside the triangle/pyramid formed by the verifiers cannot prove to be at any position p within the triangle/pyramid.
3) An external attacker performing a distance enlargement attack cannot trick the verifiers into believing that a claimant located at a location p in the triangle/pyramid is located at some other position $p' \neq p$ in the triangle/pyramid.
4) An external attacker performing a distance enlargement attack cannot trick the verifiers into believing that a claimant is located at any position p within the triangle/pyramid, if the claimant is located outside of the triangle/pyramid.

To prove properties 1 and 3 we propose the following theorem, that applies to (two dimensional) trilateration. Note that a similar theorem can be constructed for properties 2 and 4.

Theorem 1: For any two points p and p' ($p \neq p'$), located within a triangle (v_i,v_j,v_k), at least one, but not more than two of the following inequalities hold:

$$db_{ip} > db_{ip'}; \quad db_{jp} > db_{jp'}; \quad db_{kp} > db_{kp'};$$

where db_{ip} represents the distance between the verifier v_i and node p.

Proof:
We observe the three circles C_i, C_j and C_k with centers at v_i, v_j and v_k and radiuses db_{ip}, db_{jp} and db_{kp} respectively. We assume that the three circles intersect in a point p and that this point is in the triangle $\triangle(v_i, v_j, v_k)$.

We now consider another point $p' \neq p$ in $\triangle(v_i, v_j, v_k)$. We observe that p' can be in one of the two disjoint regions: (i) in the circle C_i, (i.e., $db_{ip} > db_{ip'}$) (ii) outside of C_i, or on its border (i.e., $db_{ip} \leq db_{ip'}$).

If $d_{ip} > d_{ip'}$, the theorem holds directly. If $d_{ip} \leq d_{ip'}$, it necessarily follows that one or both of the following equations hold $d_{jp} > d_{jp'}$; $d_{kp} > d_{kp'}$. To show this, it is sufficient to notice that if C_i, C_j and C_k intersect at a single point within the triangle, the triangle is located in the region $C_i \cup C_j \cup C_k$. From this, it follows that if $d_{ip} < d_{ip'}$, then p' is not in C_i, or on the border of C_i but in $C_j \cup C_k$. p' cannot be situated at the borders of C_j and C_k because the only intersection of C_j and C_k in the triangle is in p. From this it follows that p' needs to be in C_j or in C_k or in both circles. From this it directly follows that at least one of the following equations holds: $d_{jp} > d_{jp'}$; $d_{kp} > d_{kp'}$. □

An equivalent theorem can be proposed for the three dimensional multilateration. The proof would then consist in showing that if a claimant located within the triangular

pyramid moves at a different position within the pyramid, it will certainly reduce its distance to one of the verifiers.

C. Verifiable Multilateration with distance estimation

Verifiable Multilateration can also be performed with authenticated distance estimation, instead of distance bounding. Authenticated distance estimation enables nodes to securely associate estimated distances to true node identities. A possible implementation of authenticated distance estimation is to base it on classical three-pass authentication protocols. If the nodes are tightly synchronized, they can measure the signal time of flight to estimate their mutual distance. In the packets they send, nodes include timestamps of the times at which they sent the packets. Upon receiving a packet, each node registers the packet reception time, and estimates the distance based on the difference between the sending and the reception time. If the nodes' clocks are not tightly synchronized, but the nodes can measure time precisely, they can measure message roundtrip times and processing times, and estimate their distance accordingly. The implementation of the authenticated distance estimation can be based on symmetric-key or public-key cryptography, depending whether the nodes share secret keys, or hold each others' authentic public keys.

If implemented with authenticated distance estimation, VM offers protection only from external attackers, but not from compromised nodes. This is why it could be used only in cooperative scenarios in which the claimant and the verifiers cooperate to securely determine the position of the claimant. In the following sections, we will mainly make use of VM with distance bounding; at appropriate places, we will comment on the possible use of VM with distance estimation.

D. The threat of device cloning

With verifiable multilateration, an authority can prevent a single compromised node from cheating about its position. However, if an attacker owns several devices and each device looks to the authority as the same node, the attacker can still successfully cheat on its position. One attack assumes that the attacker places three/four devices within the triangle/triangular pyramid, such that each device is close to one of the verifiers. Each of the devices can then show to its corresponding base station (by delaying the messages) that it is positioned at *any* distance larger than their actual distance (which is small). As to the base stations these devices appear to be a single claimant, the attacker can prove to be at any distance to the base stations, and thus at any position in the verification triangle/triangular pyramid.

A solution that prevents this attack is to make claimanr devices tamper-proof such that their authentication material is not revealed to the attacker and that they cannot be cloned; however, as shown in [13] tamper-proofness has its limitations. Another possibility is that the base stations perform device fingerprinting [17] by which they identify each device as unique. In that case, the base stations can identify a claimant device by the unique "fingerprint" that characterizes its signal transmission. This process is used by cellular network operators to prevent cloning fraud; namely, a cloned phone does not have the same fingerprint as the legal phone with the same electronic identification numbers.

E. Secure node tracking

One of the most direct applications of Verifiable Trilateration mechanism is the secure tracking of mobile devices. This can be enabled by creating a tracking infrastructure that consists of a set of verifiers, which can be fixed, with predetermined positions, or randomly distributed over the area of interest, or even mobile. For the simplicity of presentation, we will analyze this infrastructure in a two-dimensional case; the generalization to the three-dimensional case is straightforward.

The number of verifiers needed to cover an area, such that position verification can be performed in the whole area, depends on the number of verifiers and their (and mobile nodes') power ranges. So far, we have assumed that the power range of each verifier can cover the verification triangle and that the position verification is thus enabled in the whole triangle. This is, however, not true in general; the verification triangle is the largest possible region in which three verifiers can verify node positions. If the power ranges of the verifiers are such that they do not cover the whole triangle, the verification region can be significantly smaller than the verification triangle. Only if the verifiers are in each others' power ranges will the verification region be equal to the verification triangle.

For this reason, the optimal way to cover an area of interest is to place the verifiers within the area such that they form regular triangles with sides equal to their power ranges. In this case, the number n of verifiers needed to cover a square area of $L \times L$ is

$$n \quad = \quad [2L/R + 3][2L/R + 1]/2$$

where L is the area width and length, R is the power range of the verifiers and mobile nodes. In this way, each verifier (except for the boundary verifiers) will be a verifier in six triangles (i.e., in a hexagon).

We now consider the case in which, instead of being predeployed on fixed locations, the verifiers are uniformly distributed over the area of interest. We performed simulations to determine the number of verifiers necessary to cover the area. This coverage will depend on the sizes and the positions of the verification triangles formed by the verifiers. Our simulations were performed on areas of variable sizes (from 500×500 to 2000×2000 m^2 with verifiers power ranges of 250 m). To avoid boundary effects, the verifiers were uniformly distributed in the area and in a boundary region outside the area, whose width was 10% of the area width.

The results of an average of 100 simulations are shown in Figure 3 and are displayed with confidence intervals of 95%. As expected, an optimal placement of verifiers is much more efficient than their random placement, in terms of number of nodes.

However, for security purposes, in some scenarios, it might be advantageous for the verifiers to be randomly placed, to

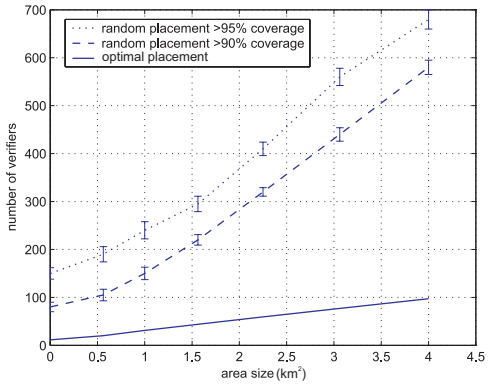

Fig. 3. Number of verifiers required to cover an area ($L \times L$) with verification triangles. The power range is 250 m.

randomly move within the area of interest and thus not to have their positions known at all times. Verifier mobility could also prevent the cloning attack and would facilitate the reconstruction of the device trajectory. Furthermore, to reconstruct the trajectory of a node, the verifiers do not need to know the positions of the node at all times; the positions that are not verified can often be reconstructed from the known ones.

As we already noted, if the verifiers are placed only within the area of interest, because of the boundary effects, verification triangles cannot cover the whole intended area. Therefore, verifiers need to be distributed also around the boundaries outside of the area of interest. In the case of a carefully planned tracking infrastructure, the verifiers can be placed either outside of the area or exactly on its borders.

V. SPINE: Secure Positioning In Sensor Networks

One of the main challenges in sensor networks [18], [19], [20], [21] is sensor positioning. Knowing the positions of sensors is important for relating the measured data with the physical location. Researchers have recently proposed a number of positioning algorithms for sensor and ad hoc networks (see Section VI). The majority of the proposed algorithms rely on insecure local distance measurements and on cooperation between the nodes that are not necessarily trustworthy.

In this section, we present SPINE, a system for secure positioning of a network of sensors, that is based on Verifiable Multilateration. We first shortly describe attacks on sensor network positioning systems.

A. Threat analysis

We characterize attackers according to the number of external and compromised nodes that they control. By Attacker-x-y we denote attacker that controls x compromised and y external nodes.

1) Node physical displacement and removal: One of the most obvious threats to sensor networks is the physical displacement of nodes. An attacker can physically displace nodes from their original positions to other positions in the network, or can temporarily or permanently remove the nodes from the network while this remains undetected to the nodes or to the network authority. These attacks are especially harmful in sensor networks, in which the nodes are, given their size and purpose, in most cases easily accessible to the attacker. It would be naive to believe that this problem can be solved only by a simple exchange of authenticated beacons between the nodes, or by conventional positioning techniques. If the network is not properly protected, an external 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 replace the network node with a fake one, and to create a communication link to the new position of the honest node. Typically, this attack can be performed by Attacker-0-2. By enabling communication between two honest nodes, the attacker easily creates the impression to the nodes that their positions remained unchanged. This attack, that we call the *node displacement* attack is illustrated in Figure 4, case *a)*.

2) Attacks on node positioning: Even without displacing the nodes, an external attacker can still perform a number of attacks on node positions and network topology. An example of this behavior is the *wormhole attack* shown in Figure 4, case *b)*, by which the attacker establishes links between nodes that are not in each others' power range. This attack can be typically performed by Attacker-0-2. Besides the establishment of new links, attackers can permanently or temporarily jam the communication between pairs of nodes and thus by remove links that would normally exist. This attack can be even performed by Attacker-0-1. These two attacks could easily jeopardize the security of sensor positioning systems that rely exclusively on beacons.

Attacks by compromised nodes are simpler to perform and can be more harmful than those performed by external nodes. Compromised nodes can modify the computed network topology by reporting non-existing links, or by not establishing or not reporting the links that would normally be established. A set of compromised nodes controlled by the same attacker can, by disseminating false information from the nodes that it controls, influence the view of other network nodes or of the central authority about the network topology and node positions. As we already detailed in Section II, an Attacker-1-0 can report false signal strength or time-of-flight values and can thus easily spoof the distance that other nodes measure to it. The *false position and distance dissemination attack* is illustrated in Figure 4, case *d)*.

B. System model

Our system consists of a set of sensor nodes and a set of reference nodes (landmarks) with known locations. Nodes and verifiers communicate using radio transmissions. If two nodes reside within the power range of each other, they are considered neighbors. We assume that the radio link between neighbors is bidirectional. Nodes measure local information, which is then collected by the central authority. Communication between nodes may involve multiple wireless hops; we do

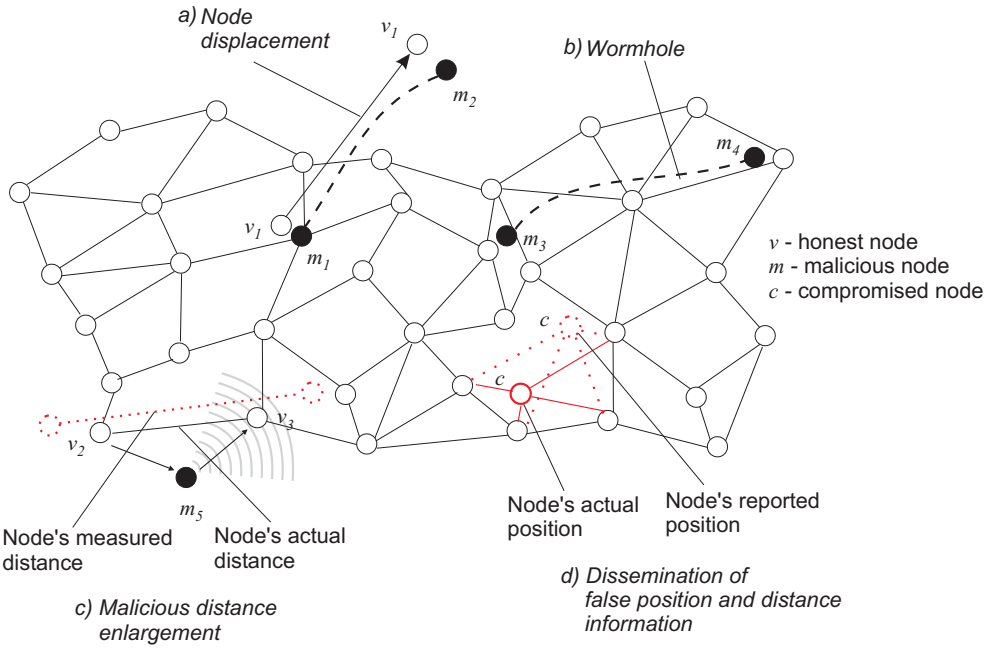

Fig. 4. Attacks on sensor network positioning.

not make any specific assumptions about the routing protocol used to transfer packets from their source to their destination.

We assume that the sensor nodes have distance-measuring capabilities, but are not equipped with GPS receivers. We assume, notably, that the nodes are able to measure the distances to their neighbors or to the landmarks by using time-of-arrival or round-trip time measurements with radio signals. We also assume that the nodes are able to bound their processing delays to a few nanoseconds.

We assume that the network is operated by an authority. This authority can be on-line, meaning that the authority operates on-line servers (by single hop or multi-hop communication), or off-line, meaning that the services of the authority cannot be reached via the network. In any case, the authority controls the network membership and assigns a unique identity to each node. We further assume that each node is able generally to accomplish any task required to secure its communications. We do not assume, however, that the nodes are able to generate or verify public-key signatures. We assume that all network nodes can establish pairwise secret keys. This can be achieved by manually pre-loading all keys into the nodes in a network setup phase, by probabilistic key pre-distribution schemes [22], [23], or through an on-line key distribution center [11].

C. SPINE algorithm

Our secure positioning algorithm (SPINE) is based on Verifiable Multilateration. The algorithm is executed in three phases: (i) the sensors measure distance bounds to their neighbors, (ii) the distance bounds are verified through verifiable multilateration and (iii) the positions of the nodes are computed with a distributed or centralized range-based positioning algorithm. We note here that sensor distance bounding can be performed simultaneously between two sensors; a protocol that enables this was proposed by Čapkun, Buttyan and Hubaux in [24].

The algorithm is executed as follows.

SPINE algorithm:
$\mathcal{VD} = \{\emptyset\}$; set of verifiable distance bounds
$\mathcal{NV} = \{\emptyset\}$; set of non-verifiable distance bounds
$\mathcal{DB} = \{$all distance bounds$\}$
For all distances $db_i \in \mathcal{DB}$
  if db_i can be verified with BDV *then* $\mathcal{VD} = \mathcal{VD} \cup \{db_i\}$
  else $\mathcal{NV} = \mathcal{NV} \cup \{db_i\}$
Compute the positions of the nodes with $db_i \in \mathcal{VD}$
Compare the computed positions with $db_i \in \mathcal{NV}$

BDV stands for *Basic Distance Verification (BDV)*. BDV is illustrated in Figure 5; it relies on Verifiable Multilateration. BDV of the distance between v and u is performed by (i) forming verification triangles around u with v and its neighbors, (ii) by forming verification triangles around v with u and its neighbors and (iii) by forming verifiable triangles around u and v. In our example, the following triangles are formed around v: $\triangle(u, v_1, v_2)$, $\triangle(u, v_3, v_4)$, $\triangle(u, v_1, v_4)$, and $\triangle(u, v_2, v_3)$; only a single triangle $\triangle(v, v_5, v_6)$ is formed around u. Finally, a triangle $\triangle(v_4, v_5, v_6)$ is formed around both u and v. After forming the triangles, the measured distance bounds db_{uv} (from u to v) and db_{vu} (from v to u) are verified in all triangles, by performing verifiable multilateration over u and v, respectively. This is done in such a way that the nodes forming a triangle define a local coordinate system, in which they then compute the position of u or v, or the positions of both u and v. The computation of the position of u and v is performed with verifiable multilateration through which the

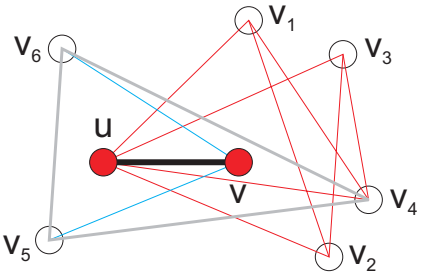

Fig. 5. Basic distance verification (BDV). To verify a distance, a set of triangles is formed around the measured distance bound.

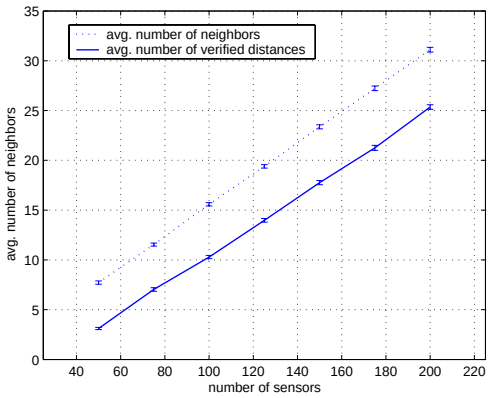

Fig. 6. An average number of neighbors per node and an average number of verifiable distances adjacent to a node.

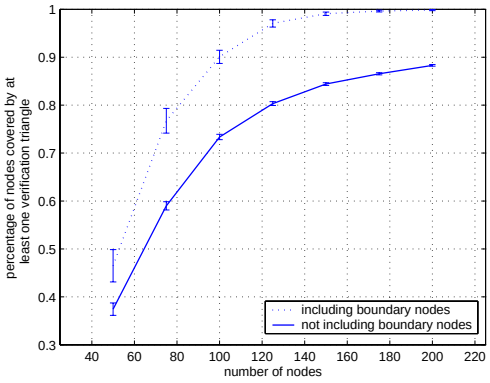

Fig. 7. The average percentage of nodes covered by at least one verification triangle, with and without boundary nodes.

distance bounds db_{uv} and db_{vu} are then verified. Verification of the distance bound is successful within BDV only if in all verification triangles the measured distance bounds db_{uv} and db_{vu} match the computed positions (with a tolerance of δ).

The algorithm is executed as follows.

Basic Distance Verification:
Measure db_{uv} by u and measure db_{vu} by v
1 Triangles are formed with v and its neighbors;
$\mathcal{US}$ is the set of triangles in which u lays
2 Triangles are formed with u and its neighbors;
$\mathcal{VS}$ is the set of triangles in which v lays
3 Triangles are formed with neighbors of u and v;
$\mathcal{UVS}$ is the set of triangles in which u and v lay
4 *In all* $\triangle_\ell \in \mathcal{US} \cup \mathcal{VS} \cup \mathcal{UVS}$ compute d^ℓ_{uv} with VM
5 *If for all* $\triangle_\ell \in \mathcal{US} \cup \mathcal{VS} \cup \mathcal{UVS}$, $|d^\ell_{uv} - db_{uv}| \leq \delta$ and $|d^\ell_{vu} - db_{uv}| \leq \delta$, *then* $\{db_{uv}, db_{vu}\}$ are verified *else* db_{uv}, db_{vu} cannot be verified

The set $\mathcal{VD}$ contains those distance bounds that can be verified by at least one triangle. The distance bounds that cannot be verified are included into a set $\mathcal{VD}$ of non-verified distances. Once the selection process is finished, the positions of the nodes can be computed by using only verified distances from the set $\mathcal{VD}$. Finally, the computed positions of the nodes are compared with the non-verified distances from $\mathcal{NV}$.

The computation of the positions of the nodes can be performed by a number of centralized or distributed range-based positioning algorithms (see Section VI). Note here that the BDV algorithm can be executed locally as the nodes forming a triangle are in each other's power ranges.

The effectiveness of any of the used positioning techniques (and consequently of SPINE) depends on the number of node neighbors (node density) and on the number and the spatial distribution of landmarks. The number of node neighbors is crucial to ensure that the positions of most of the nodes can be computed. The requirements for secure positioning are higher: it is necessary that the network is sufficiently dense to ensure that the positions of most nodes can be *securely* computed.

To show the difference between the density requirements for secure and non-secure positioning, we observe an average number of distance bounds to the neighbors that can be verified with BDV (the distances that are used for secure positioning), and an average number of node neighbors (the distances used for non-secure positioning). We performed simulations on an area of 100×100m, with 50 to 500 uniform randomly distributed nodes with power ranges of 25 m. The results are presented in Figure 6 with 95% confidence intervals.

As expected, the results show that to perform secure positioning equivalently to non-secure positioning (meaning with approximately the same number of distances), a higher density of nodes is required. For non-secure positioning, the average of 10 distances per node (10 neighbors) is reached already with 80 nodes/100×100 m^2, whereas for secure positioning, the average of 10 verifiable distances requires at least 110 nodes/100×100 m^2.

We further computed the average percentage of nodes covered by at least one verification triangle. These results are shown in Figure 7. This figure is important as it shows that at node density of 120 nodes/100×100 m^2, most of the nodes are covered by at least one verification triangle, meaning that their adjacent distances and their position can be verified. As expected, the figure shows that the boundary nodes are not covered by verification triangles. This is an important indication that the landmark stations need to be specifically placed at the boundaries of the area to protect boundary nodes from attacks by enabling the formation of verification triangles

around them.

D. Security analysis

The resistance of SPINE relies on the resistance of BDV to attacks; it depends on the ability of the attacker to modify the verified distances, but also depends on the positioning algorithm used to compute node positions with verified distances.

Here, we primarily analyze the resistance of BDV to attacks. We then discuss security implications of using BDV with several positioning algorithms.

The resistance of BDV to attacks depends on the number and on the mutual dependance of triangles that are formed around the distance. To spoof a distance verified by a single triangle, it is sufficient that an external attacker enlarges two distances (the distance d_{uv}, and one additional distance between the nodes forming a triangle). This is illustrated on Figure 8, where distances d_{uv} and d_1 are enlarged. By enlarging these two distances, all the distances in the verification triangle remain mutually consistent. This attack can be performed by an external attacker.

If only a single node in a triangle is compromised, this node can enlarge distances to the claimant and to other nodes forming the verification triangle. This is illustrated on Figure 9. In this example, node v is compromised, and enlarges distances to u, v_2 and to v_3 such that all the distances in the verification triangle remain mutually consistent. Similarly to the attack on Figure 8, if an attacker controls one compromised and one external node, it can enlarge the measured distance even if the compromised node is not adjacent to the distance. This essentially means that a single-triangle BDV resists only to attacks that enlarge only a single distance.

If k verification triangles can be formed around a distance, the resistance of BDV to attacks can be expressed in terms of k. If the triangles are node-disjoint, then BDV resists to up to $2k$ distance enlargements. This is intuitive, as the distance is verified by k disjoint triangles, and an attacker needs to spoof the verification process in each of the triangles to successful cheat on the measured distance.

If the triangles are node-joint and edge-disjoint, then BDV also resists to up to $2k$ distance enlargements by external attackers, but it does not resist attacks by a single compromised node adjacent to the spoofed distance. Essentially, if all triangles have a common (compromised) node, the distance adjacent to that node can be successfully spoofed. We note here, however, that the triangles formed around a distance are almost never node-joint, given that some are formed with u and its neighbors around v, others are formed with v and its neighbors around u, whereas the third set of triangles is formed by the neighbors of u and v around the two nodes.

If the triangles are edge-joint, then BDV resists to up to $k+1$ distance enlargements by external attackers. If the nodes are positioned favorably for the attacker, the attacker can enlarge the joint edge and enlarge one additional edge from every triangle. We note here that this attack will not always be possible.

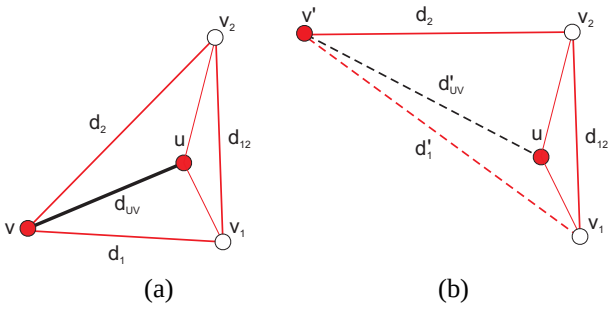

Fig. 8. An example of a distance enlargement attack by external nodes on a single-triangle BDV. Distance d_{uv} a) before enlargement and b) after enlargement.

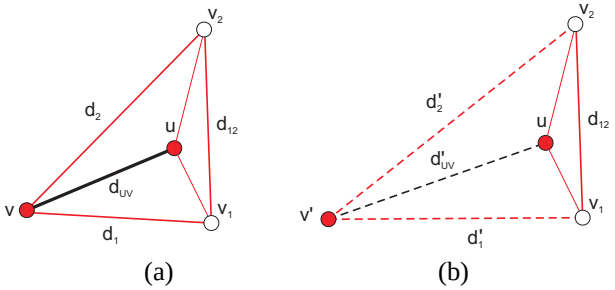

Fig. 9. An example of a distance enlargement attack by a compromised node (v) on a single-triangle BDV. Distance d_{uv} a) before enlargement and b) after enlargement.

We performed simulations on a network of sensors with densities from 50 to 500 nodes$/100 \times 100$ m^2 and a power range of 25 m. We computed the average number of verification triangles and an average number of edge-disjoint verification triangles that can be formed around a distance. The results show that BDV, depending on the node density and node positions, can resist to attacks up to 100 distance enlargements.

To compromise the computation of the position of a single node, an attacker needs to modify the computation and the verification of the (verified) distances surrounding the node. Furthermore, the attacker needs to make all the modified distances and positions consistent with the positions of other nodes in the network. The difficulty for the attacker here is in distance enlargement. Essentially, when the attacker enlarges distances, it makes some nodes to appear further from each other, but also makes some unavoidably to appear closer. This is why in a very dense network, the attacker could only scale-up all the distances in the network, but it would not be able to, by changing a smaller number of distances, successfully modify the computed positions of the nodes.

E. Discussion

SPINE is designed for both centralized and for distributed secure positioning. As a core part of SPINE, BDV relies solely on local communication between the nodes. All the triangles are formed within the local neighborhoods of the nodes. It is important to observe that the nodes that form a verification triangle are in each others' power ranges.

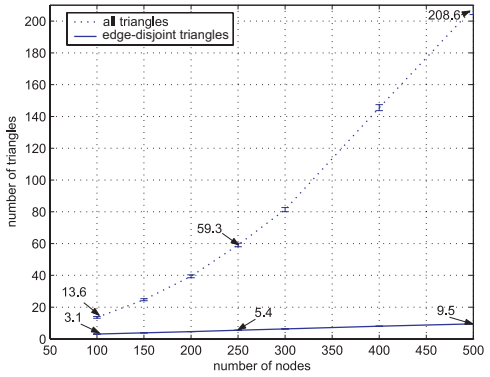

Fig. 10. An average number of verification triangles and an average number of edge-disjoint verification triangles that can be formed around a distance.

Besides SPINE, other approaches based on Verifiable Multilateration can be devised for secure position. If positioning is performed in a centralized manner, one appealing approach consists in a posteriori verification of node positions. This means that the central authority first collects distance bounds from the nodes, and computes the positions of the nodes (insecurely). The position of each node is then verified by Verifiable Multilateration assuming that the positions of its neighboring nodes are computed correctly. Unlike SPINE, this approach enables only verification, but not secure computation of node positions, and might be hard to implement securely in a decentralized manner.

VI. Related work on positioning techniques

One of the first indoor localization systems called Active Badge [1] was infrared(IR)-based. In this system, the location of each badge (e.g., attached to a person) is determined by its proximity to the nearest of the fixed receivers installed throughout the building. Indoor positioning systems based on the measurements of the propagation of sound were also proposed. Two examples of such systems are Active Bat [2] and Cricket [3]. The use of received radio signal strength for positioning was proposed in [4]. Other techniques based on the received signal strength include SpotON [5] and Nibble [6]. Time-of-flight radio signal propagation techniques were also used in systems based on ultra-wide band radio [25], [15].

Researchers also proposed positioning algorithms for wireless ad hoc networks. In [26], Doherty, Pister and El Ghaoui present a scheme in which the position of each node is computed in a centralized manner. In [27], Bulusu, Heidemann and Estrin propose a positioning system based on a set of landmark base stations with known positions. In [28], Čapkun, Hamdi and Hubaux present a GPS-free positioning system in which the nodes compute their positions by a collaborative action. In [29], Niculescu and Nath present a distributed ad hoc positioning system that provides approximate positions for all nodes in a network where only a limited fraction of nodes have self-positioning capabilities. In [30], the same authors present a positioning system based on the angle of arrival. In [31], Savvides, Han and Srivastava propose a dynamic fine-grained localization scheme for sensor networks in which groups of nodes collaborate to resolve their positions. In [32], Moore et al. describe a distributed, linear-time algorithm for localizing sensor network nodes in the presence of range measurement noise. The authors introduce the probabilistic notion of robust quadrilaterals as a way to avoid flip ambiguities that otherwise corrupt localization computations. In [33], Eren et al. provide a theoretical foundation for the problem of network localization. They apply graph rigidity theory to test the conditions for unique localizability and to construct uniquely localizable networks, and they study the computational complexity of network localization.

VII. Conclusion

In this work, we have analyzed positioning and distance estimation techniques in adversarial settings. We have shown that most proposed positioning techniques are vulnerable to position spoofing attacks from external attackers and compromised nodes. We have further shown that positioning and distance estimation techniques based on radio signal propagation exhibit the best properties for position verification. We have proposed a novel mechanism for position verification, called Verifiable Multilateration (VM). Verifiable Multilateration enables secure computation and verification of node positions in the presence of attackers. We have further proposed SPINE, a system for secure positioning in a network of sensors, based on Verifiable Multilateration. We have shown that this system resists against distance modification attacks from a large number of attacker nodes.

Our future work includes a detailed analysis and possible implementation of distance bounding and position verification techniques. Furthermore, we intend to investigate the applicability of our basic distance verification scheme to a number of existing positioning algorithms.

Acknowledgments

The authors would like to thank Virgil Gligor, Markus Jakobsson and Adrian Perrig for useful comments and suggestions.

References

[1] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The Active Badge Location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, 1992.

[2] A. Ward, A. Jones, and A. Hopper, "A New Location Technique for the Active Office," *IEEE Personal Communications*, vol. 4, no. 5, October 1997.

[3] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket location-support system," in *Proceedings of MobiCom*. ACM Press, 2000, pp. 32–43.

[4] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," in *Proceedings of Infocom*, vol. 2, 2000, pp. 775–784.

[5] J. Hightower, G. Boriello, and R. Want, "SpotON: An indoor 3D Location Sensing Technology Based on RF Signal Strength," University of Washington, Tech. Rep. 2000-02-02, 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)*, vol. 2201. Springer-Verlag Heidelberg, September 2001.

[7] S. Brands and D. Chaum, "Distance-bounding protocols," in *Workshop on the theory and application of cryptographic techniques on Advances in cryptology*. Springer-Verlag New York, Inc., 1994, pp. 344–359.

[8] N. Sastry, U. Shankar, and D. Wagner, "Secure Verification of Location claims," in *Proceedings of WiSe*. ACM Press, September 2003, pp. 1–10.

[9] L. Lazos and R. Poovendran, "SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks," in *Proceedings of WiSe*, 2004.

[10] M. G. Kuhn, "An Asymmetric Security Mechanism for Navigation Signals," in *Proceedings of the Information Hiding Workshop*, 2004.

[11] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: a secure on-demand routing protocol for ad hoc networks," in *Proceedings of MobiCom*. ACM Press, September 2002, pp. 12–23.

[12] J. S. Warner and R. G. Johnston, "Think GPS Cargo Tracking = High Security? Think Again," *Technical report, Los Alamos National Laboratory*, 2003.

[13] R. Anderson and M. Kuhn, "Tamper resistance - a cautionary note," in *Proceedings of the Second Usenix Workshop on Electronic Commerce*, 1996.

[14] B. Waters and E. Felten, "Proving the Location of Tamper-Resistant Devices," Princeton University, Tech. Rep. [Online]. Available: http://www.cs.princeton.edu/∼bwaters/research/location_proving.ps

[15] R. Fontana, "Experimental Results from an Ultra Wideband Precision Geolocation System," *Ultra-Wideband, Short-Pulse Electromagnetics*, May 2000.

[16] R. Fontana, E. Richley, and J. Barney, "Commercialization of an Ultra Wideband Precision Asset Location System," in *IEEE Conference on Ultra Wideband Systems and Technologies*, November 2003.

[17] D. Shaw and W. Kinsner, "Multifractal Modeling of Radio Transmitter Transients for Clasification," in *Proceedings of the IEEE Conference on Communications, Power and Computing*, May 1997, pp. 306–312.

[18] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks," in *Proceedings of MobiCom*. ACM Press, 1999, pp. 263–270.

[19] G. Asada, M. Dong, T. Lin, F. Newberg, G. Pottie, W. Kaiser, and H. Marcy, "Wireless Integrated Network Sensors: Low Power Systems on a Chip," in *Proceedings of the European Solid State Circuits Conference*, 1998.

[20] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister, "Smart dust: Communicating with a cubic-millimeter computer," *Computer*, vol. 34, no. 1, pp. 44–51, 2001.

[21] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[22] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the ACM Conference on Computer and Communications Security*. ACM Press, 2002, pp. 41–47.

[23] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks," in *Proceedings of the IEEE Symposium on Research in Security and Privacy*. IEEE Computer Society, May 2003, p. 197.

[24] S. Čapkun, L. Buttyán, and J.-P. Hubaux, "SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks," in *Proceedings of SASN*, Washington, USA, October 2003.

[25] J.-Y. Lee and R. Scholtz, "Ranging in a Dense Multipath Environment Using an UWB Radio Link," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 9, December 2002.

[26] L. Doherty, K. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proceedings of Infocom*, April 2001.

[27] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, October 2000.

[28] S. Čapkun, M. Hamdi, and J.-P. Hubaux, "GPS-free Positioning in Mobile Ad-Hoc Networks," *Cluster Computing*, vol. 5, no. 2, April 2002.

[29] D. Niculescu and B. Nath, "DV Based Positioning in Ad hoc Networks," *Journal of Telecommunication Systems*, vol. 22, no. 4, pp. 267–280, 2003.

[30] D. Niculescu and B. Nath, "Ad Hoc Positioning System using AoA," in *Proceedings of Infocom*, 2003.

[31] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in Ad-Hoc networks of sensors," in *Proceedings of MobiCom*. ACM Press, 2001, pp. 166–179.

[32] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proceedings of SenSys*. ACM Press, 2004, pp. 50–61.

[33] T. Eren, D. Goldenberg, W. Whiteley, Y. Yang, A. Morse, B. Anderson, and P. Belhumeur, "Rigidity, computation, and randomization in network localization," in *Proceedings of Infocom*, 2004.