Secure positioning of wireless devices with application to sensor networks

Srdjan Čapkun and Jean-Pierre Hubaux
Laboratory for Computer Communications and Applications (LCA)
Swiss Federal Institute of Technology 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.

I. Introduction

Recently, researchers have proposed a number of positioning and distance estimation techniques for wireless networks [25], [26], [21], [4], [16], [7]. However, they all studied these techniques in non-adversarial settings. Distance estimation and positioning techniques are, nevertheless, highly vulnerable to attacks from dishonest nodes and malicious attackers. *Dishonest nodes* can report false position and distance information in order to cheat on their locations. *Malicious attackers* can modify the measured positions and distances of wireless nodes.

Some proposals for secure distance and location verification have already been proposed. Brands and Chaum [5] 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 [22] 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.

In this work, we focus on secure verification of positions (instead of presence in a region) of wireless devices. We propose a mechanism for secure verification of device positions called 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 in 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 a 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 through simulations.

The organization of the rest of the paper is the following. In Section II, we overview 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 overview 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 a node *malicious* if it is controlled by an attacker but cannot authenticate itself as an honest network node to other network nodes or to a central authority. We call

	Dishonest node	Malicious 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**
GPS	False position reports	Position spoofing

TABLE I

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

a node *dishonest* or *compromised* if it is controlled by an attacker and it can authenticate itself to the authority and to other network nodes [17]. 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. We further assume that when a node is malicious or compromised, this is not detected by other honest nodes, nor by the central authority (at least for some time).

A. Global Positioning System (GPS)

The Global Positioning System is today the most widespread outdoor positioning system for mobile devices. The 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, 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 [28]. 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.

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 [2].

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 malicious and dishonest nodes. To reduce the measured distance between two honest nodes, two malicious attackers 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, malicious attackers can enlarge the measured distances between honest nodes. If conventional US ToF technique is used, a dishonest 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 [22] have proposed a US-based distance bounding technique which resists to distance reduction attacks from dishonest nodes; it does not, however, resist to attacks from malicious 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 malicious attackers have no physical access to the localization region. In [29] 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 dishonest node therefore only needs to report a false power level to an honest node. Malicious 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, a malicious attacker can, by jamming and replaying the signals, only increase, but not decrease the measured time-of-flight between the nodes. A dishonest 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 [5] 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 dishonest node from reducing the measured distance. As we will show in Section III in more detail, with RF ToF distance-bounding protocols, malicious attackers and dishonest 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 which is shown on Table I, which illustrate 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 [14]). Furthermore, the RF ToF distance estimation and distance bounding techniques are the most effective techniques to counter attacks from malicious attackers and dishonest nodes. A potential drawback of these systems is that, because they operate with the speed of light, the devices require 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 (dishonest) device. As we indicated in Table I, RF-based distance bounding protocols are vulnerable to distance enlargement attacks but not to distance reduction attacks. In this section, we present a distance bounding protocol, similar to the ones proposed by Brands and Chaum [5]. The protocol is shown in Figure 1. It is performed between a verifier v and a claimant u; the verifier performs distance bounding of the claimant, i.e., upper-bounds its distance to the claimant. The protocol begins by a commitment of the claimant u to the random values by hashing them with a collision-resistant one-way hash function h. The verifier then generates a challenge nonce N_v, sends it to u in a reverse order (by sending the last bit first and the first bit last) and starts the timer when it sends the last bit of the challenge. Upon receiving the challenge, u is expected to respond immediately with $N_v \oplus N_u$. When it receives all the bits of the response, the verifier stops the timer, computes the challenge-response time-of-flight t_{vu} and estimates the distance to u. Since u cannot send the correct response before receiving the challenge, it can either delay the response, or send it immediately after receiving the challenge. In the last stage of the protocol, u authenticates itself to v by encrypting the second part of its commitment N'_u with the key K_{uv} that it shares with v. N'_u is then used by the verifier to authenticate u and to verify if the commitment corresponds to u's response.

u : generate random nonces N_u, N'_u
: generate commitment $commit = h(N_u, N'_u)$
$u \rightarrow v$: $commit$

v : generate random nonce N_v
$v \rightarrow u$: N_v
$u \rightarrow v$: $N_v \oplus N_u$
v : measure the time t_{vu} between sending N_v and receiving $N_v \oplus N_u$

$u \rightarrow v$: $E_{K_{uv}}(u, N_v, N'_u)$

v : decrypt the message and verify if $commit = h(N_u, N'_u)$

Fig. 1. A distance bounding protocol.

When it estimates the distance to u, the verifier also takes into account the processing delay of u. In this case, this time is relatively small, given that u needs to perform only an XOR operation, and does not need to perform any cryptographic operations until the end of the protocol. The security of this distance bounding protocol relies on the fact that it is difficult for the claimant to guess the random nonce created by the verifier. The probability that the claimant succeeds in proving that it is closer to the verifier than it really is, can therefore be computed as $\frac{1}{2^\ell}$, where ℓ is the length of the nonce in bits.

The described protocol is suitable for devices that

can perform rapid message exchanges, execute XOR operations rapidly, and perform encryption. 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 30 cm). This requirement allows the node to perform distance bounding with radio signals with an uncertainty of 30 cm. We are aware that with today's mobile devices, a nanosecond processing and time measurements are achievable only with dedicated hardware. Recent developments in location system show that RF ToF 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 [15] consisting of active tags and tracking devices show that this system can provide two- and three-dimensional location of objects to within several 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 over 40 grams each.

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 techniques 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 an attacker.

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 "outside" malicious attackers and from dishonest 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. The algorithm

Verifiable Multilateration relies on distance bounding. It consists of distance bound measurements from at least three reference points (verifiers) to the mobile 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 the section, we briefly comment on how a similar algorithm can be applied to the three dimensional case. The algorithm is executed by the verifiers and by the authority as follows.

Verifiable multilateration

$\mathcal{T} = \{\emptyset\}$; set of verifiers that form 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 *for all* triplets $(v_i, v_j, v_k) \in \mathcal{V}^3$, compute the
position (x'_u, y'_u) with db_i, db_j, db_k by MMSE
if (x'_u, y'_u) in $\triangle(v_i, v_j, v_k)$ *then* $\mathcal{T} = \mathcal{T} \cup \{v_i, v_j, v_k\}$

3 with all $v_i \in \mathcal{T}$, compute the position (x''_u, y''_u) by MMSE

4 *if for all* $v_i \in \mathcal{T}$, $|db_i - \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2}| \leq \delta$
then $x_u = x'_u$, $y_u = y'_u$
else the position is rejected

In the initial phase of the algorithm, the verifiers $v_1, ..., v_n$ which are in the power range of the claimant u independently 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 known) are reported to the central authority, which then computes securely the position of the claimant. For each triplet of verifiers (v_i, v_j, v_k), the authority computes wether the claimant u is positioned within the triangle $\triangle(v_i, v_j, v_k)$ by making use only of distance bounds db_i, db_j, db_k measured by that triplet of verifiers. This approximative position of the claimant (x'_u, y'_u) is computed with the Minimum Mean Square Estimate (MMSE) method, which we describe later. Subsequently, two tests are run: (i) does the computed position differ from the measured distance bounds db_i, db_j, db_k by less than the expected distance measurement error δ and (ii) does the computed position fall within the physical triangle $\triangle(v_i, v_j, v_k)$ formed by that triplet of verifiers. If both tests are positive, the verifiers v_i, v_j, v_k will participate in the computation of the final position of the claimant. Note also that we call the triangle formed by the verifiers the *verification triangle*.

The following computations are performed to detect if the positioning error of the position (x'_u, y'_u) is lower than the expected error δ.

Distance enlargement (δ) test

if $|db_i - \sqrt{(x_i - x'_u)^2 + (y_i - y'_u)^2}| \leq \delta$ and
$|db_j - \sqrt{(x_j - x'_u)^2 + (y_j - y'_u)^2}| \leq \delta$ and
$|db_k - \sqrt{(x_k - x'_u)^2 + (y_k - y'_u)^2}| \leq \delta$
then the position is valid
else the position is rejected

If the test is negative, 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.

The following well known test is run to detect if (x'_u, y'_u) coordinates fall within the 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 triangle test are positive, this means that the verifiers v_i, v_j, v_k are able to securely compute the position of the claimant u and that this position is (x'_u, y'_u). The intuition behind this is the following. If a node is dishonest, it might try to cheat about its position. 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. Specifically, 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. This is illustrated in Figure 2a.

The claimant can nevertheless, by enlarging the response times to all the verifiers, pretend to be outside the triangle. However, in this case the verifiers will not be able to verify its position and will not consider it to be securely computed.

If there are more than three verifiers that can form triangles around the claimant (Figure 2b), the final position of the claimant (x''_u, y''_u) is computed by using distance bounds from all those verifiers (six on the figure). Note that in this computation the distance bounds of the verifiers that cannot form at least one triangle around u are not taken into account. The position is computed by the MMSE of the following system of equations:

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

Minimum Mean Square Estimate
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

After the position (x''_u, y''_u) of u is computed, the distance enlargement (δ) test is run for distance bounds of verifiers in $\mathcal{T}$. If the difference between the computed position and each distance bound is smaller than δ, the position is considered as valid; otherwise, the position is rejected. We note here that the expected error δ can also be reduced with the increase of the number of verifiers to reflect the higher expected accuracy of the system.

If the position is rejected that means that one or more of the measured distance bounds have been enlarged either by the claimant or by a malicious attacker. If a larger number of verification triangles can be formed around u, the authority can try to detect which of the distances are enlarged:

Detection of enlarged distances:
$\mathcal{C} = \{\emptyset\}$; set of verifiers with correctly measured bounds
$\mathcal{NC} = \{\emptyset\}$; set of verifiers whose bounds are suspicious
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 $\mathcal{NC} = \mathcal{NC} \cup \{v_i\}$
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*

In this algorithm, the number of verification triangles will determine if the distance(s) which is(are) enlarged can be identified. 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.

In the same way as in two dimensions, 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 the same way as in the two dimensional case, but with positioning within triangular pyramids being the basic element.

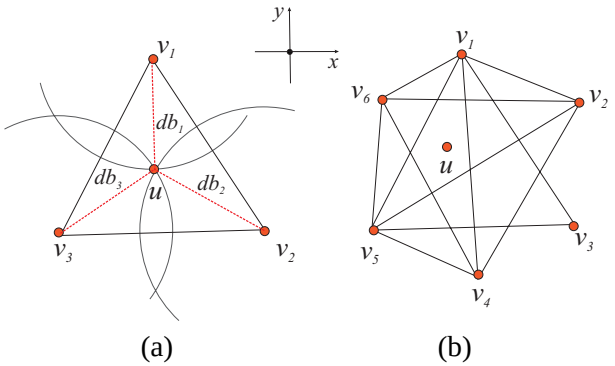

Fig. 2. Examples of Verifiable Multilateration. a) with three verifiers. b) with six verifiers.

B. The properties

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

1) A node located at the 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 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 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. One 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 wether the nodes share secret keys, or hold each others' authentic public keys.

If implemented with authenticated distance estimation, VM offers protection only from malicious 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 device from cheating about its position. However,

if an attacker owns several devices and each device seems 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). Since, 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.

One solution that prevents this attack is to make mobile devices tamper-proof such that their authentication material is not revealed to the attacker and that they cannot be cloned; however, as shown in [2] tamper-proofness has its limitations. Another possibility is that the base stations perform device fingerprinting [24] by which they identify each device as unique. In that case, the base stations can identify a mobile 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 to 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 and 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 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

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

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

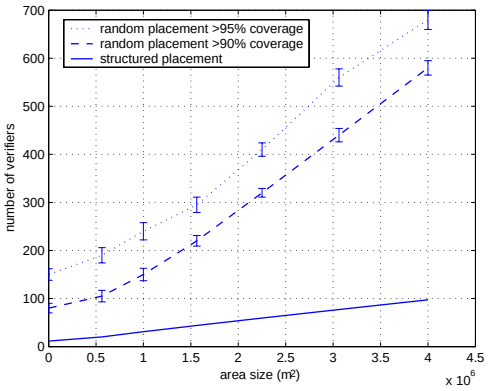

Fig. 3. Number of verifiers required to cover an area ($L \times L$) with verification triangles. The power range is 250 m.

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, a planned placement of verifiers is much more efficient than their random placement, in terms of the number of the nodes.

However, for security purposes, in some scenarios, it might be advantageous for the verifiers to be randomly placed, to 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 carefully placed either outside of the area or exactly on its borders.

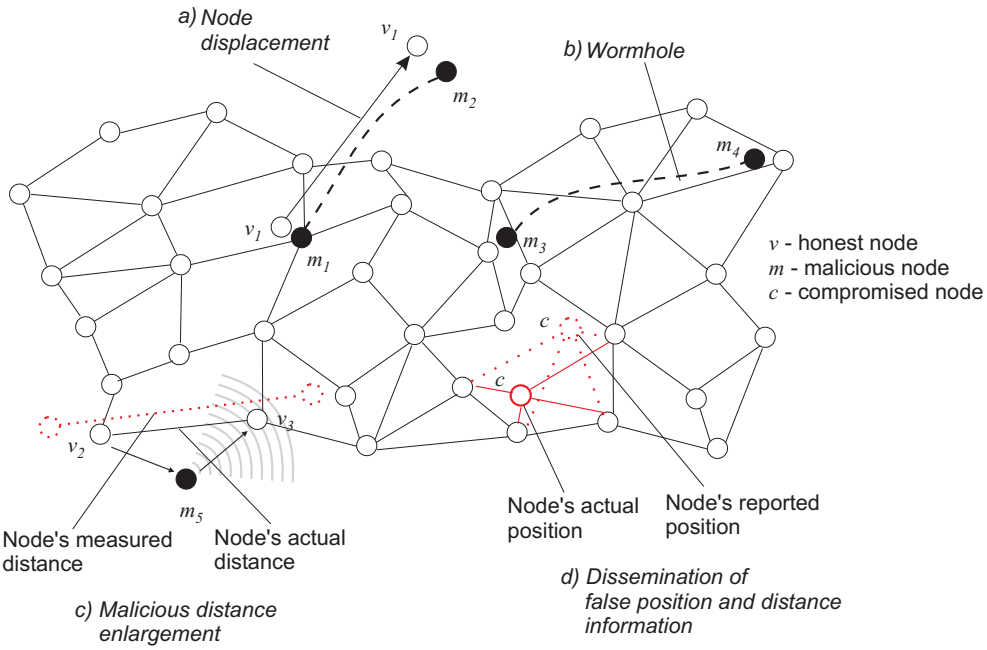

Fig. 4. Attacks on sensor network positioning.

V. SPINE: Secure Positioning In Sensor Networks

One of the main challenges in sensor networks [13], [3], [27], [1] is sensor positioning. Knowing the positions of sensors is important for relating the measured data with the physical position at which it was measured. 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 overview attacks on sensor network positioning systems.

A. Threat analysis

We characterize attackers according to the number of malicious and compromised nodes that they control. By Attacker-x-y we denote the attacker that controls x compromised and y malicious nodes [17].

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 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, the attackers 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, malicious 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 malicious 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 (Attacker-x-y) 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

In this section, we describe our 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 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. This requirement is important so that the nodes can perform distance bounding with radio signals. We are aware that with today's sensors a nanosecond processing bound is achieved only with dedicated hardware modules; recent advances in positioning with Ultra-Wide Band (UWB) technologies show that this is achievable and that these systems can provide high precision both for indoor and outdoor scenarios [15].

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 to generate symmetric cryptographic keys and more 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 [12], [8], or through an on-line key distribution center [17].

C. SPINE algorithm

Our secure positioning algorithm (SPINE) is a centralized algorithm based on distance bounding and on Verifiable Multilateration. The algorithm is executed in two phases: (i) the sensors measure distance bounds to their neighbors and (ii) the distance bounds are then collected at a central authority and the positions of the nodes are computed. 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 [9].

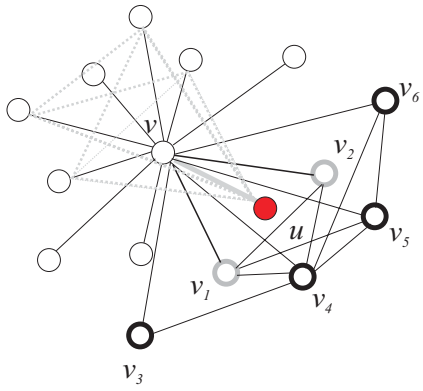

Fig. 5. Basic distance verification (BDV). To verify a distance, a set of triangles is formed around the distance.

Once the central authority obtains the distance bounds from the sensors, it starts the computation of their positions. 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 all $db_i \in \mathcal{VD}$

Here, BDV is a mechanism that we call *Basic Distance Verification (BDV)*. The principle of BDV is Verifiable Multilateration. The BDV mechanism is illustrated in Figure 5. The central authority performs *basic verification* of the distance between v and u by forming verification triangles of v and its neighbors in which u falls, and vice-versa, by forming around v the triangles of u and its neighbors (or better visibility, this second set of triangles is shown shaded). In our example, the triangles are formed between v and the following pairs of its neighbors: (v_1, v_2), (v_4, v_6), (v_4, v_2), (v_4, v_5) and (v_1, v_5); equivalently, a set of triangles is formed with u and its neighbors around v. The output of BDV is binary: the BDV either confirms the measured distance bound, or rejects it as wrong. If within all the triangles, a distance bound is confirmed, the authority accepts that distance as a correct one; otherwise, it rejects the distance.

The central authority goes through all the received distance bounds in the network (set $\mathcal{DB}$), and includes those distance bounds that can be verified by at least one triangle into the set $\mathcal{VD}$ of verified distances. Other distance bounds, that cannot be verified are included into a set $\mathcal{VD}$ of non-verified distances. Once that the selection process is finished, the central authority computes the positions of the nodes by using only verified distances from the set $\mathcal{VD}$. Finally, the authority compares the computed positions of the nodes with non-verified distances

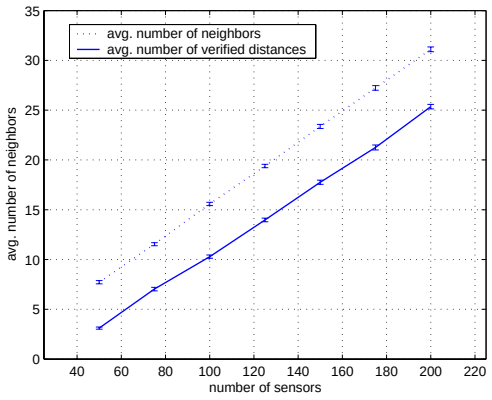

Fig. 6. An average number of neighbors per node and an average number of verifiable distances adjacent to a node.

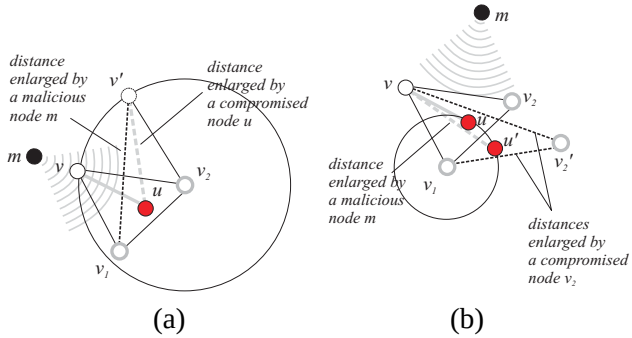

Fig. 7. Examples of attacks on the verification of the distance between v and u a) by a compromised claimant u and a malicious node m, and b) by a compromised verifier v_2 and a malicious node m.

from $\mathcal{NV}$.

The computation of the positions of the nodes can be performed by a number of positioning algorithms (see Section VI); here we consider an iterative multilateration mechanism proposed by Savvides, Han and Srivastava in [23]. This algorithm is executed as follows.

Iterative Multilateration:
iterativeM(){
Select u with the maximum $NoBeacon(u)$
if $NoBeacon(u) \geq 3$
compute position of u with verified distances by MMSE
convert u into a beacon
iterativeM()
else stop }

We adopt the following terminology: a beacon node is a node with a known position (be it computed or pre-measured); all landmarks are beacon nodes. With iterative multilateration algorithm, the position of a node with the highest number of beacon neighbors in their neighborhood is computed first. Once its position is computed, a node becomes a beacon and the algorithm is repeated until for all the nodes with at least three beacon neighbors the position is computed. If the computed position of a node differs from the verified distances by more than a predictable error δ, the authority can still reject the computed position and try to compute the position of another node. Moreover, if the number of verifiable distances (the number of beacons) with which the position is computed is larger than three, the authority can even use this redundancy to detect which of the distances significantly differs.

The effectiveness of iterative multilateration (and consequently of SPINE) depends on the number of node neighbors (node density) and on the number and the distribution of landmarks. The number of node neighbors is crucial to ensure that 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.

We observe two values: 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 uniformly 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 the non-secure positioning, the average of 10 distances per node (10 neighbors) can be achieved already with 80 nodes/100×100 m^2, whereas for secure positioning, the average of 10 verifiable distances can be achieved at 110 nodes/100×100 m^2. This shows that to have the same percentage of the nodes positioned with with secure and non-secure positioning, in the case of non-secure positioning, the network needs to be from 30% (higher node densities) up to 50% (lower node densities).

We further computed the average percentage of nodes covered by at least one verification triangle. These results are shown in Figure 8. This value is important to show 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. In each step of iterative multilateration, the positions of nodes are computed from the verified distances to their neighbors. To compromise the computation of a position of a single node, the attacker needs to modify the computation and the verification of the verified distances surrounding the

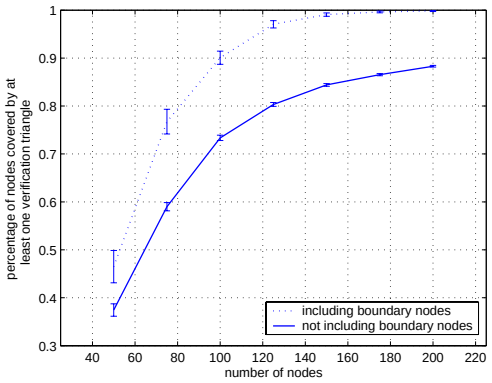

Fig. 8. The average percentage of nodes covered by at least one verification triangle, with and without boundary nodes.

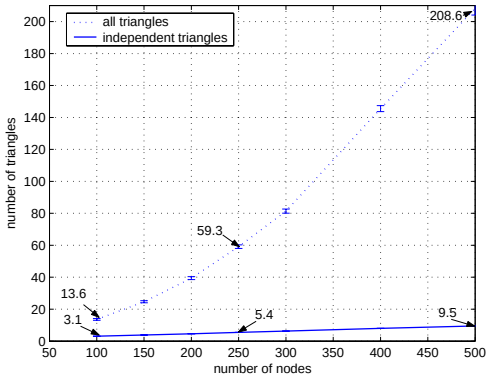

Fig. 9. An average number of verification triangles and an average number of independent verification triangles that can be formed around a distance.

node. If the number of those distances is d, the attacker needs to modify $n-3$ distances to successfully modify the computed distance.

Now we analyze in more detail how a single verifiable distance can be modified. The resistance of BDV to attacks depends on the number of triangles that can be formed around the distance. If only a single triangle can be formed around a distance, BDV will be resistant to attacks from a single malicious attacker (Attacker-0-1), and to attacks from an attacker controlling a single compromised node (Attacker-1-0). However, BDV with single triangle does not resist attacks from Attacker-2-0, Attacker-0-2, or Attacker-1-1. If one of the verifiers or the claimant u is compromised and the attacker controls an additional malicious node the attacker will be able to successfully cheat on the distance. Two examples of this are shown in Figure 7. In the first example, the claimant is compromised, and it enlarges its distance to v. Because the claimant is aided by a malicious node, which enlarges one of the distances between the verifiers, the distances between verifiers are consistent with the enlarged distance and the attack is successful. The second example shows a similar distance enlargement attack, but this time is performed by a compromised verifier v_2 and by a single malicious node (Attacker-1-1). An attacker controlling two malicious nodes (Attacker-0-2) can successfully perform the same attack; it is sufficient that it enlarges two distances (one between the claimant and the verifier, and the other between two verifiers).

However, if several verification triangles can be formed around the distance, the resistance of BDV to attacks is higher. If these triangles are dependent, meaning that, besides v, they share another common verifier, (e.g., triangles (v, v_1, v_2) and (v, v_4, v_2) in Figure 5), the resistance of BDV will not necessarily increase. Specifically, if the nodes are positioned favorably for the attacker, Attacker-1-1 can still successfully perform the distance enlargement attack. It is sufficient that the compromised claimant u enlarges the distance measured to v, and that the malicious node enlarges the distance that is common to all dependent verification triangles (in our example, the distance between v and v_2). However, in most cases, the attacker will need to control as many malicious nodes as there are verification triangles; BDV will thus be *in most cases* resistant to attacks from at least Attacker-0-k, where k is the number of verification triangles. If the verification triangles are independent (e.g., (v, v_1, v_2) and (v, v_3, v_4)), the resistance of BDV will increase, and it will be resistant to attacks from Attacker-0-k in all cases, no matter how the nodes are positioned.

To test the resistance of BDV to attacks, 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 independent verification triangles that can be formed around a distance. The results show that, BDV, depending on the node density is resistant to Attacker-0-2 to at most Attacker-0-100, depending on node density and node positions. We note here that to successfully attack SPINE, an attacker needs to successfully modify a larger number of measured distances.

VI. Related work on positioning techniques

One of the first indoor localization systems called the Active Badge [25] 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 [26] and Cricket [21]. The use of received radio signal strength for positioning was proposed in [4]. Other techniques based on the received signal strength include SpotON [16] and Nibble [7]. Time-of-flight radio signal propagation techniques were also used in systems based on ultra-wide band radio [18], [14].

Researchers also proposed positioning algorithms for wireless ad hoc networks. In [11], Doherty, Pister and El Ghaoui present a scheme in which the position of each node is computed in a centralized manner. In [6], Bulusu, Heidemann and Estrin propose a positioning system based on a set of landmark base stations with known positions. In [10], Čapkun, Hamdi and Hubaux present a GPS-free positioning system in which the nodes compute their positions by a collaborative

action. In [20], 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 [19], the same authors present a positioning system based on the angle of arrival. In [23], Savvides, Han and Srivastava propose a dynamic fine-grained localization scheme for sensor networks in which groups of nodes collaborate to resolve their positions.

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 dishonest and malicious attackers. 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 mechanisms 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 a SPINE, a system for secure positioning in a network of sensors, based on Verifiable Multilateration. We have showed that this systems resists against distance modification attacks from a large number of attackers.

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 the position and distance verification techniques to positioning in mobile ad hoc networks.

References

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

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

[3] 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.

[4] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *Proceedings of InfoCom*, volume 2, pages 775–784, 2000.

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

[6] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, October 2000.

[7] P. Castro, P. Chiu, T. Kremenek, and R. Muntz. A Probabilistic Room Location Service for Wireless Networked Environments. In *Proceedings of the Third International Conference Atlanta Ubiquitous Computing (Ubicomp)*, volume 2201. Springer-Verlag Heidelberg, September 2001.

[8] 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*, page 197. IEEE Computer Society, May 2003.

[9] 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.

[10] S. Čapkun, M. Hamdi, and J.-P. Hubaux. GPS-free Positioning in Mobile Ad-Hoc Networks. *Cluster Computing*, 5(2), April 2002.

[11] L. Doherty, K. Pister, and L. El Ghaoui. Convex position estimation in wireless sensor networks. In *Proceedings of InfoCom*, April 2001.

[12] 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*, pages 41–47. ACM Press, 2002.

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

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

[15] R.J. 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.

[16] J. Hightower, G. Boriello, and R. Want. SpotON: An indoor 3D Location Sensing Technology Based on RF Signal Strength. Technical Report 2000-02-02, University of Washington, 2000.

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

[18] J.-Y. Lee and R.A. Scholtz. Ranging in a Dense Multipath Environment Using an UWB Radio Link. *IEEE Journal on Selected Areas in Communications*, 20(9), December 2002.

[19] D. Niculescu and B. Nath. Ad hoc positioning system (aps) using aoa. In *Proceedings of InfoCom*, San Francisco, USA, April 2003.

[20] D. Niculescu and B. Nath. DV Based Positioning in Ad hoc Networks. *Journal of Telecommunication Systems*, 22(4):267–280, 2003.

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

[22] N. Sastry, U. Shankar, and D. Wagner. Secure Verification of Location claims. In *Proceedings of WiSe*, pages 1–10. ACM Press, September 2003.

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

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

[25] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The Active Badge Location system. *ACM Transactions on Information Systems*, 10(1):91–102, 1992.

[26] A. Ward, A. Jones, and A. Hopper. A New Location Technique for the Active Office. *IEEE Personal Communications*, 4(5), October 1997.

[27] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister. Smart dust: Communicating with a cubic-millimeter computer. *Computer*, 34(1):44–51, 2001.

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

[29] B. Waters and E. Felten. Proving the Location of Tamper-Resistant Devices. Technical report, Princeton University.