Secure Localization and Authentication in Ultra-Wideband Sensor Networks

Yanchao Zhang, *Student Member, IEEE*, Wei Liu, Yuguang Fang, *Senior Member, IEEE*, and Dapeng Wu, *Member, IEEE*

***Abstract*—The recent Federal Communications Commission regulations for ultra-wideband (UWB) transmission systems have sparked a surge of research interests in the UWB technology. One of the important application areas of UWB is wireless sensor networks. The proper operations of many UWB sensor networks rely on the knowledge of physical sensor locations. However, most existing localization algorithms developed for sensor networks are vulnerable to attacks in hostile environments. As a result, attackers can easily subvert the normal functionalities of location-dependent sensor networks by exploiting the weakness of localization algorithms. In this paper, we first analyze the security of existing localization techniques. We then develop a mobility-assisted secure localization scheme for UWB sensor networks. In addition, we propose a location-based scheme to enable secure authentication in UWB sensor networks.**

***Index Terms*—Authentication, localization, security, sensor networks, ultra-wideband (UWB).**

I. Introduction

ULTRA-WIDEBAND (UWB) has a number of unique merits such as low probability of interception and detection, resilience to multipath fading, high penetration probability, and fine time resolution for accurate location determination. Therefore, it is finding ever-increasing uses in wireless communications, networking, radar imaging, and localization systems [1]. This paper is concerned with UWB wireless sensor networks (WSNs), which are important UWB applications.

Many WSNs require sensor nodes to know their physical locations. Examples include those for target detection and tracking, precision navigation, search and rescue, geographic routing, security surveillance, and so on. Driven by this demand, many localization schemes have been proposed in recent years, with most assuming the existence of a few *anchors* that are special nodes knowing their own locations, e.g., via global positioning system (GPS) or manual configuration. These proposals can be divided into two categories: *range-based* such as [2]–[3] and *range-free* [4]–[5]. The former are characterized by using absolute point-to-point distance (range) or angle estimates in location derivations, while the latter depend on messages from neighboring sensors and/or anchors. Range-based solutions can provide more accurate locations, but have higher hardware requirements for performing precise range or angle measurements. By contrast, although having lower hardware requirements, range-free approaches only guarantee coarse-grained location accuracy. Due to space limitations, we limit the paper scope to range-based approaches and leave the investigation on range-free ones as the future work.

We observe that almost all existing range-based proposals were designed for benign scenarios where nodes cooperate to determine their locations. As a result, they are ill-suited for unattended and often hostile settings such as tactical military operations and homeland security monitoring. Under such circumstances, attackers can easily subvert the normal functionalities of WSNs by exploiting the weakness of localization algorithms [6], [7]. In this paper, we do not intend to provide brand-new localization techniques for UWB sensor networks. Instead, we focus on analyzing and enhancing the security of existing approaches when applied in adversarial settings.

The rest of this paper is structured as follows. We start with analyzing the vulnerability of existing approaches in Section II. Next, we present a novel mobility-assisted secure localization scheme (SLS) in Section III. Section IV illustrates a location-based authentication scheme designed for security-sensitive UWB sensor networks. We then review related work in Section V and end with conclusions and future work.

II. Vulnerability Analysis of Two-Way Time-of-Arrival (ToA) Localization

Popular range-based localization techniques include received-signal-strength-indicator (RSSI), angle-of-arrival (AoA), time-of-arrival (ToA), and time-difference-of-arrival (TDoA). Readers are referred to [3] for a nice review. Among these techniques, ToA is the most commonly used one whose requirement for fine time resolution can be satisfied by the UWB technique. Therefore, our study focuses on a two-way ToA approach, which is illustrated with Fig. 1.

In the shown example, anchors *A*, *B*, and *C* intend to determine the two-dimensional (2-D) location of sensor *S*. To do so, *A* transmits at time t_1 a challenge to sensor *S*, which immediately echoes a response received by *A* at time t_2. Anchor *A* can then estimate its distance to *S* as $d_{AS} \approx (t_2 - t_1)c/2$, where c is the speed-of-light. In the same way, *B* and *C* can obtain distance estimates to *S*, denoted by d_{BS} and d_{CS}, respectively. Let (X_A, Y_A), (X_B, Y_A), (X_C, Y_C) be the known locations of *A*, *B*, and *C*, and (X_S, Y_S) be *S*'s location to be decided. Assume that *A* is the leader which collects d_{BS} and d_{CS}, and then sets up the following equations:

$$\begin{cases} f_A = d_{AS} - \sqrt{(X_S - X_A)^2 + (Y_S - Y_A)^2} \\ f_B = d_{BS} - \sqrt{(X_S - X_B)^2 + (Y_S - Y_B)^2} \\ f_C = d_{CS} - \sqrt{(X_S - X_C)^2 + (Y_S - Y_C)^2} \end{cases}. \quad (1)$$

Manuscript received March 1, 2005; revised October 15, 2005. This work was supported in part by the U.S. Office of Naval Research under Young Investigator Award N000140210464, and in part by the U.S. National Science Foundation under Grant ANI-0093241 (CAREER Award) and Grant DBI-0529012.

Y. Zhang, Y. Fang, and D. Wu are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: yczhang@ufl.edu; fang@ece.ufl.edu; wu@ece.ufl.edu).

W. Liu is with Scalable Network Technologies, Los Angeles, CA 90045 USA (e-mail: liuw@ufl.edu).

Digital Object Identifier 10.1109/JSAC.2005.863855

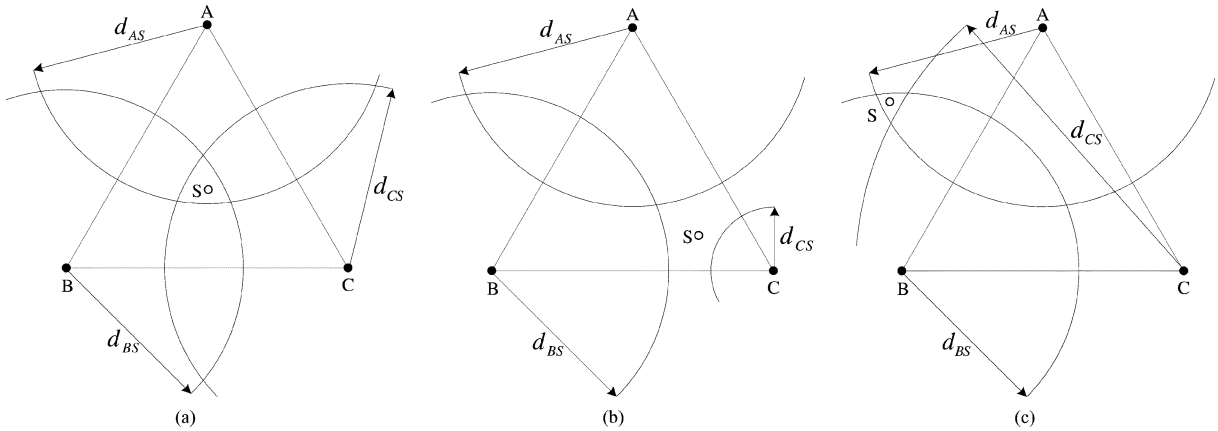

Fig. 1. An exemplary two-way ToA localization process, where anchors *A*, *B*, and *C* are determining the location of sensor *S*. (a) No attacks. (b) d_{CS} is reduced. (c) d_{CS} is enlarged.

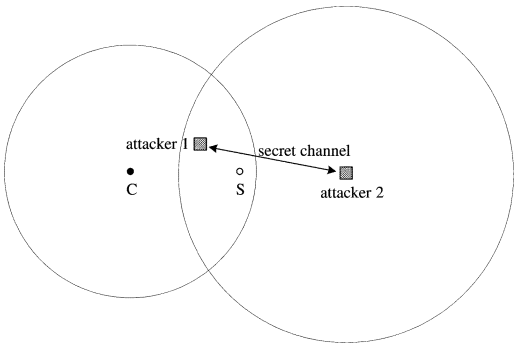

Fig. 2. Topology of an exemplary distance enlargement attack.

If there is no measurement error, f_A, f_B, and f_C are all equal to zero, and (X_S, Y_S) is the common intersection point of the three circles defined by the above equations. Since measurement errors inevitably exist in reality, however, (X_S, Y_S) will be somewhere in the intersection area formed by the three circles, as shown in Fig. 1(a). It can be obtained via the minimum mean-square error (MMSE) method [2], i.e., minimizing $F(X_S, Y_S) = f_A^2 + f_B^2 + f_C^2$.

The above process is vulnerable to distance reduction and enlargement attacks, in which attackers attempt to reduce and enlarge distance estimates, respectively, so as to maliciously increase the location inaccuracy. For example, attackers can impersonate sensor *S* to answer anchor *C*'s challenge before *S* does, and then jams the later genuine response from *S*. As a result, d_{CS} would be intentionally reduced. In addition, Fig. 2 shows the topology of an exemplary distance enlargement attack, where the two circles indicate the transmission ranges of anchor *C* and attacker 2, respectively. In this attack, the challenge from *C* is correctly received by attacker 1, but not by sensor *S* whose reception activities are interfered by attacker 2. Subsequently, attacker 1 sends the unmodified challenge via a secret channel to attacker 2 which, in turn, forwards the challenge to sensor *S* after some time. Sensor *S* will consider it a challenge from anchor *C* and respond to it. In doing so, attackers can increase the challenge-response time difference measured at *C*, and thus the distance estimate d_{CS}. Both distance reduction and enlargement attacks may make the location estimate of sensor *S* far from its true location, as can be seen from Fig. 1(b) and (c), respectively. To satisfy the requirement for high location accuracy by many WSN applications, we must therefore seek ways to mitigate the impact of such attacks.

III. Mobility-Assisted Secure Localization for UWB Sensor Networks

In this section, we present a mobility-assisted secure localization scheme (SLS) for UWB sensor networks. To ease our illustration, we focus on how to ensure secure 2-D location estimates, but SLS can be easily extended to the three-dimansional case.

A. Network Model

We consider a WSN that consists of randomly deployed sensor nodes, e.g., via random aerial scattering. Sensor localization is normally done during the network initialization phase, in which we assume that a set of anchors, denoted by $\mathcal{A}$, perform coordinated group movement across the whole sensor field. Typical examples of anchors are mobile robots or unmanned aerial vehicles (UAVs) flying at low levels. The number of anchors, denoted by $n_a = |\mathcal{A}|$, should be at least three for determining a 2-D location. Intuitively, the more anchors (i.e., distance estimates) are available, the more precise location estimates are at the cost of increased communication and computational overhead. We also indicate anchor i by A_i for $i \in \{1, \ldots, n_a\}$.

Each A_i is assumed to know its own location (X_{A_i}, Y_{A_i}) at any time and place through GPS receivers or other means. In addition, there is always a leader in $\mathcal{A}$ that takes charge of the localization process. In practice, each anchor should take turns to act as the leader to balance their resource usage. For convenience, however, we assume A_1 to be always the anchor leader hereafter. We further assume that anchors and sensor nodes have the same transmission range r_0.

Before network deployment, we assume that the network planner picks a sufficiently long secret $\mathcal{K}$, and loads each sensor *S* with a secret key $K_S = h_{\mathcal{K}}(ID_S)$. Here, ID_s is the unique identifier of node *S*, h indicates a fast hash function such as SHA-1, and $h_{\mathcal{K}}(M)$ refers to the message integrity code (MIC) of message M under key $\mathcal{K}$. We further postulate that each anchor knows the network secret $\mathcal{K}$ and is trusted and unassailable to attackers during the node localization phase which

TABLE I
K-DISTANCE ALGORITHM

1: $\mathcal{T} = \phi$
2: **for** ($j = 1$; $j \leq K$; $j++$) **do**
3: A_i sends a random challenge nonce N_j to S
4: S responds with N_j and another random nonce M_j
5: A_i sets t_j = time elapses between challenge and response
6: S sends to A_i a number $v = h_{K_S}(N_j \parallel M_j)$
7: **if** $h_{K_S}(N_j \parallel M_j) == v$ **then** /*by A_i*/
8: $t_{p,j} = (t_j - t^{A_i}_{proc} - t^{S}_{proc} - t_{tran})/2$
9: $\mathcal{T} = \mathcal{T} \cup \{t_{p,j}\}$
10: **end if**
11: **end for**
12: t_{A_iS} = median($\mathcal{T}$)
13: **return** $d_{A_iS} = ct_{A_iS}$ /*c is the light speed*/

usually does not last too long. This assumption is reasonable in that anchors are usually much fewer than sensor nodes, so we can spend more on them by enclosing them in high-quality tamper-resistant enclosures and putting them under perfect monitoring. How to deal with compromised anchors is part of our ongoing work.

B. Overview of SLS

After sensor nodes are deployed, anchors are instructed to perform strategic group movement along preplanned routes to localize all the sensor nodes. Anchors are required to always maintain an n_a-vertex polygon with the longest distance between any two vertices no larger than r_0. This means that anchors and sensors inside the polygon can directly communicate with each other. To localize a node, say *S*, anchors first measure their respective distance to *S* with a modified two-way ToA approach, called K-distance. The anchor leader A_1 then collects all the distance estimates whereby to derive a MMSE location estimate. Subsequently, A_1 runs a *validity test* on the location estimate to detect possible attacks.

Unlike traditional localization methods such as AHLos [2], our mobility-assisted approach does not require each sensor node to accurately measure distances to anchors and do the MMSE estimation. Instead, each node just needs to answer the challenges from anchors, and the tasks of time (distance) measurement and location derivation are shifted to resource-rich anchors. This is highly desirable for lowering the requirements on sensor hardware and thus the manufacturing costs. In the rest of this section, we will detail the operations of SLS with a to-be-localized sensor node *S* as an example.

C. K-Distance: A K-Round Distance Estimation Algorithm

To obtain a distance estimate to node *S*, anchor A_i first calculates $K_S = h_{\mathcal{K}}(ID_S)$ based on the preloaded network secret $\mathcal{K}$. It then executes the K-distance algorithm outlined in Table I. A_i begins with sending to *S* an l-bit random nonce N_j and starts a timer when the last bit of N_j is sent. Upon receiving N_j, node *S* needs to immediately echo N_j concatenated by another l-bit random nonce M_j picked by itself. Next, *S* sends to A_i a MIC $v = h_{K_S}(N_j \parallel M_j)$, where $\parallel$ means message concatenation.

When receiving the last bit of the response, A_i stops the timer and sets t_j equal to the elapsing time. It then uses K_S to compute a MIC on N_j and M_j. If the result is not equal to v which arrives later, A_i considers the response a bogus one and simply ignores it. Otherwise, it believes that the response indeed came

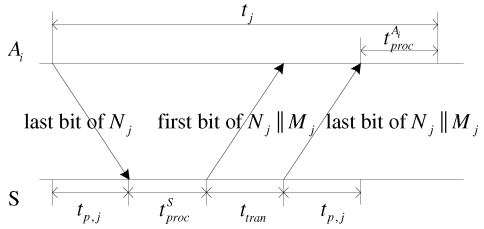

Fig. 3. Time plot of the challenge-response process.

from *S*, and proceeds to calculate the one-way signal propagation time as $t_{p,j} = (t_j - t^{A_i}_{proc} - t^{S}_{proc} - t_{tran})/2$. Here, $t^{A_i}_{proc}$ represents the time duration from when the last bit of the response hits the antenna of A_i until the response is completely decoded (cf. Fig. 3); t^{S}_{proc} is the time duration from when the last bit of the challenge reaches the antenna of *S* until *S* transmits the first bit of the response. $t^{A_i}_{proc}$ and t^{S}_{proc} are device-dependent and usually are constant or vary in a tiny scale. Both can be predetermined and preloaded to A_i to calibrate the time measurements to certain precision. Assume that transmission links from *S* to anchors have a bandwidth of b b/s. Then, the response transmission time t_{tran} is approximately equal to $2l/b$ seconds.

The above process offers strong defense against distance reduction attacks in the sense that attackers cannot reduce $t_{p,j}$, and thus the distance estimate $ct_{p,j}$. One reason is that the MIC check ensures that an authentic response can only be sent by node *S*. Another important reason is that nothing can travel faster than light so that attackers are unable to make the challenge arrive at *S* earlier than it should.

Attackers, however, can still launch the distance enlargement attack, i.e., enlarging $t_{p,j}$, and thus the distance estimate. To mitigate this attack, we require A_i to perform K times of distance measurements. The motivation is that attackers might not be able to actively affect all K time measurements, and thus distance estimates. It is also worth noting that our method can help mitigate sporadic measurement errors. K is a design parameter that determines the tradeoff between algorithm overhead and resilience to distance enlargement attacks and measurement errors. Assume that all the K time measurements are stored in an initially empty set $\mathcal{T}$. The next question is how to securely use them. The naive use of the average is insecure because attackers can easily make the calculated average quite different from the true one by merely enlarging one-time measurement to be sufficiently large.

As pointed out in [8], the median is a safer replacement for the average, so K-distance uses the median of K time measurements to calculate d_{A_iS}.[1] For brevity only, we assume $K \geq 3$ to be odd in what follows and the extension to the case that K is even is straightforward. Let $t_{(1)}, \ldots, t_{(K)}$ denote trustful time estimates (without attacks) in $\mathcal{T}$ placed in an increasing order. We then have $t_{A_iS} = median(\mathcal{T}) = t_{(r)}$ for r equal to $(K+1)/2$. Consider first the simple case that attackers enlarged just one-time estimate from $t_{(j)}$ to $t'_{(j)}$. If $t_{(j)}$, $t'_{(j)} < t_{(r)}$, the median t_{A_iS} remains unchanged; otherwise, it changes to some value between $[t_{(r-1)}, t_{(r+1)}]$. It is easy to see that K-distance is vulnerable to single distance enlargement attack when K is equal to one (as all previous TOA-based

[1]We notice that there might exist other methods such as least median squares (LMS) to deal with outliers (distance estimates enlarged in our case). However, they are less computationally efficient than the median method.

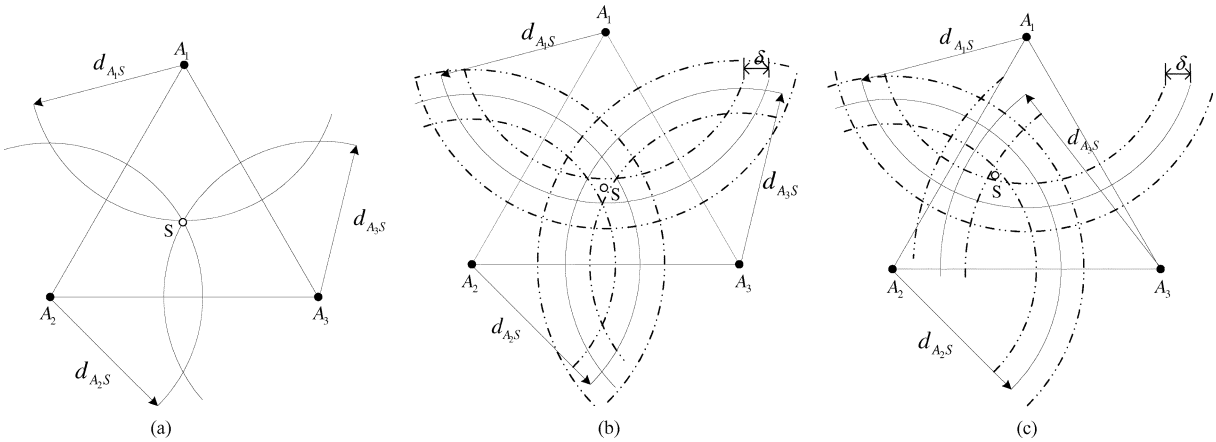

Fig. 4. Location validity test with three anchors. (a) No measurement errors. (b) Measurement errors exist. (c) d_{A_3S} is enlarged.

proposals) or two. In general, if m time measurements were enlarged, t_{A_iS} either remains unchanged or changes to some value between $[t_{(r-m)}, t_{(r+m)}]$, depending on how attackers contaminated the time measurements. It is obvious that the median method can tolerate the enlargement of up to about half of the time measurements.

A_i then calculates $d_{A_iS} = ct_{A_iS}$ and sends to anchor leader A_1 a message of format $\{d_{A_iS}, h_{\mathcal{K}}(d_{A_iS})\}_{\mathcal{K}}$, where $\{M\}_K$ means encrypting data M with key $\mathcal{K}$. Upon receipt of it, A_1 decrypts d_{A_iS} and checks its authenticity via the preloaded $\mathcal{K}$. Once obtaining all n_a distance estimates, A_1 can then derive a MMSE location estimate (X_S, Y_S).

D. Location Validity Test

The median approach may be enough for withstanding less powerful attackers. However, if K assumes a small value, attackers launch persistent attacks, and m is greater than $(K+1)/2$, some distance estimates used for deriving (X_S, Y_S) might have still been enlarged, leading to the invalidity of (X_S, Y_S). Therefore, we require A_1 to run a validity test on (X_S, Y_S).

Consider first the simple case that there are no measurement errors. If all the n_a distance estimates were not enlarged by attackers, (X_s, Y_s) should be exactly the intersection point of n_a circles $\{(x - X_{A_i})^2 + (y - Y_{A_i})^2 = d^2_{A_iS} | 1 \le i \le n_a\}$. To test the validity of (X_S, Y_S), A_1 merely needs to check whether (X_S, Y_S) is inside the n_a-vertex polygon formed by all the anchors. The underlying logic is very simple. If attackers want to make *S* appear to be at any location other than its true location, they have to enlarge certain distance measurements, while at the same time reduce some others so as to keep the resulting location estimate inside the polygon. As mentioned before, however, our K-distance algorithm can prevent attackers from launching distance reduction attacks. Therefore, anchors can be assured that the location estimate is trustable as long as it resides in the n_a-vertex polygon. We refer to Fig. 4(a) for an example with three anchors ($n_a = 3$).

To determine the inclusion of a point inside a polygon, we select the *ray-tracing* method for its simpleness and computational efficiency. This method works by starting at the point in question and drawing a straight line in any direction. If the number of times the ray intersects the polygon edges is odd, the starting point is inside the polygon and is outside otherwise.

TABLE II
TESTING IF A POINT IS INSIDE A $|\mathcal{B}|$-VERTEX POLYGON

Inputs: $\mathcal{B}$: an anchor set, (X_S, Y_S): a location estimate
Output: 0 if outside, else 1

```
1: u = 0
2: for (i = 1, j = |B|; i ≤ |B|; j = i++) do
3:     if ((((Y_i ≤ Y_S)&&(Y_j > Y_S)) || ((Y_i > Y_S)&&(Y_j ≤ Y_S)))
4:     &&(X_S > (X_i − X_j) * (Y_S − Y_j)/(Y_i − Y_j) + Y_j)) then
5:         u =!u
6:     end if
7: end for
8: return u
```

This is easy to understand intuitively. Each time the ray crosses a polygon edge, its in-out parity changes because each edge always separates the inside of a polygon from its outside. Eventually, any ray must end up beyond and outside the bounded polygon. Therefore, if the point is inside, the sequence of crossings "$\rightarrow$" must be: $\text{in} \rightarrow \text{out} \rightarrow \cdots \rightarrow \text{in} \rightarrow \text{out}$, and there are an odd number of them. Similarly, if the point is outside, there are an even number of crossings in the sequence: $\text{out} \rightarrow \cdots \rightarrow \text{in} \rightarrow \text{out}$. Table II gives the pseudocode implementation for the ray-tracing method, which uses a horizontal ray extending to the left of (X_S, Y_S) and parallel to the negative x axis.

In practical scenarios, however, time measurement errors and thus distance estimate errors occur inevitably. The n_a circles centered at anchors will, therefore, not have a common intersection point, but form an intersection area in which the location estimate is located, as shown in Fig. 4(b). This would introduce room for distance enlargement attacks. Consider again the three-anchor example in Fig. 4(c). Suppose the distance estimate d_{A_3S} was maliciously enlarged, while d_{A_1S} and d_{A_2S} are just a little larger than the actual distances due to measurement errors. It is obvious that, by adjusting the level of enlarging d_{A_3S}, attackers might be able to freely enlarge the intersection area of the three circles, and thus make the MMSE distance estimate (though still inside the triangle) deviate much from the true location. Fortunately, we can alleviate this issue by imposing certain reasonable constraints. Let δ be the two-sided maximum allowable measurement error with respect to distance estimates. Now, (X_s, Y_s) should reside in the intersection area of n_a rings, $\{(d_{A_iS} - \delta)^2 \le (x - X_{A_i})^2 + (y - Y_{A_i})^2 \le (d_{A_iS} + \delta)^2 | 1 \le i \le n_a\}$ [see Fig. 4(b)]. This means that, in addition to performing the point-inclusion test, A_1 needs to check whether the inequality

$|d_{A_iS} - \sqrt{(X_S - X_{A_i})^2 + (Y_S - Y_{A_i})^2}| \leq \delta$ holds for each d_{A_iS}. If so, (X_s, Y_s) is considered valid and invalid otherwise.

With our method in place, attackers might only be able to enlarge any d_{A_iS} a little bit to make the resulting (X_S, Y_S) appear to be valid, leading to tolerable location imprecision. However, if they enlarge d_{A_iS} by a relatively large amount, the resulting (X_S, Y_S) will be identified as invalid. One such example is shown in Fig. 4(c). Therefore, although our method cannot completely eliminate distance enlargement attacks, which is believed to be impossible for any security mechanism, it does constrain the impact of attackers to a tolerable level.

If (X_S, Y_S) does not pass either the point-inclusion test or the δ-error check, A_1 recomputes a MMSE location estimate based on any $(n_a - 1)$ distance estimates and checks its validity via these two tests. If all the sets of $(n_a - 1)$ distance estimates are traversed and still no valid location estimate is generated, A_1 tries the sets of $(n_a - 2)$ distance estimates. A_1 continues this process until either a valid (X_S, Y_S) is found or all the 3° subsets of n_a distance estimates are examined (three is the minimum number of distance estimates required to derive a 2-D location estimate). If the latter case occurs without yielding a valid location estimate, A_1 may consider that the localization process was attacked and should take certain actions, e.g., reporting this abnormality to the control center, as stipulated by concrete WSN applications.

If a valid (X_S, Y_S) is derived, anchor A_1 transmits it securely to node *S* in a message, $\{X_S, Y_S, h_{K_S}(X_S \parallel Y_S)\}_{K_S}$. Upon receiving it, node *S* uses the preloaded secret key K_S to decrypt (X_S, Y_S) and compute a MIC. If the result matches with what A_1 sent, *S* considers (X_S, Y_S) trustable and saves it for subsequent use.

E. Discussion

1) Overhead Analysis: So far, we have elaborated the operations of SLS, by which a valid location estimate can be obtained despite the presence of attacks as long as there are at least three unattacked distance estimates. The desirable security improvement does not come for free. Specifically, the K-distance algorithm requires each anchor to obtain K-distance estimates instead of one as in previous schemes. Besides the tunability of K, however, K-distance cannot only mitigate distance enlargement attacks, but also smooth sporadic measurement errors in the first place. Also note that, if some distance estimates were maliciously enlarged, A_1 may need to perform the MMSE estimation for up to $\sum_{j=3}^{n_a} \binom{n_a}{j}$ times. In practical scenarios, n_a should be carefully chosen to be a small number that can guarantee a certain level of resilience to attacks, while not incurring too much overhead. For instance, when $n_a = 5$ anchors are used, SLS can tolerate two (40%) maliciously enlarged distance estimates that are not filtered by K-distance. Then, A_1 needs to calculate at most 16 distance estimates. Since anchors have more powerful computational capacities than sensor nodes and node localization is a one-time process, we believe such overhead to be acceptable for security-sensitive UWB sensor networks.

2) Other Applications: In addition to securely localizing sensor nodes, SLS can find uses in many other applications. One example is critical asset tracking. Many organizations, particularly defense contractors, have parts and equipment of a sensitive, secure, or hazardous nature. These parts need to be monitored and audited to record their movements and who had access to them, as proof that they have not been tampered with or viewed by unauthorized personnel. We can accomplish this task by deploying a tracking infrastructure composed of a set of anchors and attaching to critical assets some sensors that are difficult to remove without being detected. Anchors and sensors communicate with each other through UWB radios. SLS can then be used by anchors to keep tracking the locations of critical assets (in fact, attached sensors).

IV. Location-Based Secure Authentication in UWB Sensor Networks

We have detailed SLS that works due to the high time resolution (nanosecond scale) of UWB radio. Once sensors obtain their respective secure locations, it is reasonable to consider leveraging such secure location information to further improve the security and survivability of WSNs. In this section, we present a novel location-based secure authentication scheme for UWB sensor networks.

Secure mutual authentication between neighboring sensor nodes is of vital importance for sensor network security. For example, a node should only accept and/or forward messages from authenticated neighbors. Otherwise, attackers can easily inject bogus messages into the network to deplete scare network resources and interrupt normal network functionalities. Common authentication techniques can be classified into symmetric-key and public-key solutions. The former require each pair of neighboring nodes to share a pairwise symmetric key. Recent years have witnessed a growing body of work on how to establish pairwise shared symmetric keys between neighboring sensor nodes (see, for example, [9]–[10]). However, almost all of them fail to provide strong node-to-node authentication as opposed to public-key techniques, [11].

It was a common belief that public-key techniques are too complex, slow, and power hungry for WSNs. However, many recent proposals such as [12] and [13] have challenged this seemingly proper belief by showing that public-key techniques are rather tractable on low-end sensor nodes. In addition, each sensor node usually has a limited number of neighbors and just needs to perform public-key authentication once with each neighbor during the network bootstrapping phase. Subsequent message encryption and authentication can be fulfilled through symmetric-key techniques. Therefore, we believe that it is appropriate to employ public-key techniques to enable one-time neighborhood authentication. Our location-based authentication scheme is built upon the ID-based cryptography (IBC) [14], [15], rather than the all-too-familiar certificate-based cryptography (CBC) such as RSA.

In what follows, we first briefly introduce IBC, and then illustrate the generation process of location-based keys (LBKs), as well as the location-based authentication scheme. At last, we analyze the security of the proposed scheme.

A. Introduction to IBC

IBC is receiving extensive attention as a powerful alternative to CBC, as it allows public keys of entities to be directly derived from their publicly known identity information. IBC thus eliminates the need for authenticated public-key distribution conventionally realized via public-key certificates. This inbred feature makes IBC particularly suitable for the resource-constrained wireless arena, e.g., WSNs, [11], mobile ad hoc networks [16],

and wireless mesh networks. Although the idea of IBC dates back to 1984 [17], only recently has its rapid development taken place due to the application of the following pairing technique.

Let $\mathbb{G}_1$ be an additive cyclic group of prime order q and $\mathbb{G}_2$ be a multiplicative cyclic group of the same order. Assume that the discrete logarithm problem (DLP) is hard[2] in both $\mathbb{G}_1$ and $\mathbb{G}_2$. A pairing is a *bilinear map* $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ if, for all $P, Q, R, S \in \mathbb{G}_1$, we have[3]

$$\hat{e}(P+Q, R+S) = \hat{e}(P,R)\hat{e}(P,S)\hat{e}(Q,R)\hat{e}(Q,S). \quad (2)$$

Modified Weil [14] and Tate [15] pairings are examples of such bilinear maps, for which the *Bilinear Diffie–Hellman Problem* (BDHP) is believed to be hard.[4] We refer readers to [14] and [15] for further details on pairing.

B. Generating Location-Based Keys (LBKs)

A core component of our location-based authenticate scheme is to generate LBKs for individual nodes. To do this, a trusted authority (TA), e.g., the network planner, decides the pairing parameters $(q, \mathbb{G}_1, \mathbb{G}_2, \hat{e})$ prior to network deployment. It also selects a random $g \in \mathbb{Z}_q^*$ as a master key and a cryptographic hash function H_1 that maps arbitrary strings to nonzero elements in $\mathbb{G}_1$. Both sensor nodes and anchors are preloaded with parameters $(q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, H_1)$, but only anchors have the knowledge of g.

The generation of LBKs can be well integrated with the secure localization process addressed previously. Consider again the example in Section III-D. After deriving a location estimate (X_S, Y_S) for node *S*, the anchor leader A_1 proceeds to compute a LBK for *S* as $LK_S = gH_1(X_S \parallel Y_S)$. A_1 then sends message $\{X_S, Y_S, LK_S, h_{K_S}(X_S \parallel Y_S \parallel LK_S)\}_{K_S}$ to node *S* which, in turn, can decrypt and authenticate the message. $\langle (X_S, Y_S), LK_S \rangle$ is called the public/private key pair of node *S*. Since the DLP is difficult in $\mathbb{G}_1$, it is impossible to deduce the master key g from any given $\langle (X_S, Y_S), LK_S \rangle$ pair. This also means that, even after compromising an arbitrary number of sensor nodes, attackers are still unable to exploit the locations and LBKs of compromised nodes to derive g, and thus cannot calculate the LBKs of noncompromised nodes.

C. Location-Based Neighborhood Authentication

After securely localized and armed with LBKs, sensor nodes can fulfill mutual authentication with their neighbors. In many WSNs under consideration, sensor nodes are fixed at where they were deployed, so they can be uniquely identified by their locations [3], provided that no nodes have the same locations. This fact is the main motivation of our location-based neighborhood authentication scheme, which is illustrated with neighboring nodes *R* and *S*, as an example

1. $R \rightarrow * : X_R, Y_R, n_R$
2. $S \rightarrow R : X_S, Y_S, n_S, h_{K_{S,R}}(n_R \parallel n_S \parallel 1)$
3. $R \rightarrow S : h_{K_{R,S}}(n_R \parallel n_S \parallel 2)$.

[2]It is computationally infeasible to extract the integer $x \in \mathbb{Z}_q^* = \{i | 1 \leq i \leq q-1\}$, given $P, Q \in \mathbb{G}_1$ (respectively, $P, Q \in \mathbb{G}_2$) such that $Q = xP$ (respectively, $Q = P^x$).

[3]In particular, $\forall\ P, Q \in \mathbb{G}_1, \forall\ a, b \in \mathbb{Z}_q^*$, $\hat{e}(aP, bQ) = \hat{e}(aP, Q)^b = \hat{e}(P, bQ)^a = \hat{e}(P, Q)^{ab}$, etc.

[4]It is believed that, given $\langle P, xP, yP, zP \rangle$ for random $x, y, z \in \mathbb{Z}_q^*$ and $P \in \mathbb{G}_1$, there is no algorithm running in expected polynomial time, which can compute $\hat{e}(P, P)^{xyz} \in \mathbb{G}_2$ with nonnegligible probability.

Assume that node *R* starts the authentication process by broadcasting its location (X_R, Y_R) and a random nonce n_R. Upon receiving the message, node *S* first ascertains that the claimed location is inside its transmission range r_0 by checking whether $(X_R - X_S)^2 + (Y_R - Y_S)^2 \leq r_0^2$ holds. This check is necessary to filter bogus authentication requests that contain sensors' locations outside the transmission range of *S* (cf. Section IV-D). If the inequality does not hold, *S* simply ignores the authentication request because the initiator is by no means its neighbor. Otherwise, *S* calculates

$$K_{S,R} = \hat{e}(LK_S, H_1(X_R \| Y_R)) = \hat{e}(gH_1(X_S \| Y_S), H_1(X_R \| Y_R))$$

and returns a unicast packet that consists of its location (X_S, Y_S), a random nonce n_S, and an authenticator $V_S = h_{K_{S,R}}(n_R \parallel n_S \parallel 1)$.

Upon receipt of the reply, *R* also checks if $(X_R - X_S)^2 + (Y_R - Y_S)^2 \leq r_0^2$ holds. If so, it calculates

$$K_{R,S} = \hat{e}(H_1(X_S \| Y_S), LK_R) = \hat{e}(H_1(X_S \| Y_S), gH_1(X_R \| Y_R)).$$

According to the bilinearity of $\hat{e}$, $K_{R,S}$ is equal to $K_{S,R}$ if and only if *S* has the authentic K_S and *R* holds the authentic K_R. Therefore, if the recomputed MIC $h_{K_{R,S}}(n_R \parallel n_S \parallel 1)$ is equal to what *S* sent, node *R* considers *S* an authentic neighbor. Next, *R* unicasts to *S* a new MIC $h_{K_{R,S}}(n_R \parallel n_S \parallel 2)$ to prove its knowledge of LK_R. Node *S* then recalculates the MIC and, if the result matches what was received, ascertains that *R* is an authentic neighbor. Following the similar procedures, all the neighboring sensor nodes can fulfill mutual authentication.

D. Discussion

Our location-based authentication scheme can withstand a variety of attacks. For example, it is immune to the *location impersonation attack* in which an attacker impersonates nodes whose locations are within the transmission range of another node under attack, say *S*. The reason is that the attacker will not be in possession of the corresponding LBKs. Our scheme is also impervious to the *wormhole attack* [18] in which two powerful collusive attackers tunnel authentication messages received at one location of the network over an invisible, out-of-band, low-latency channel to another network location which is typically multihop away. By doing that, they attempt to make two victim nodes far apart from each other to believe that they are authentic neighbors. With our scheme in place, the wormhole attack is no longer feasible in that each legitimate node will deny authentication requests from sensors that are not physically within its transmission range. Due to the same reason, our scheme can as well defend against the *node replication attack*, where attackers put clones of a compromised node into multiple locations distant from its original location. Interested readers are referred to for more attacks that the LBKs and location-based authentication scheme can deal with.

We would like to point out that our scheme itself cannot prevent a compromised node or its replicas from achieving mutual authentication with its legitimate neighbors, which is a difficult (if not impossible) task for any security solution. However, our scheme can guarantee that the compromised node or its replicas receive nothing more than some random numbers and locations from legitimate nodes. Therefore, the compromised node cannot impersonate its legitimate neighbors to other

nodes, and attackers cannot utilize the keying material of compromised nodes to launch network-wide attacks at arbitrary locations. This is in contrast to the case that public and private keys of sensor nodes are bound to their IDs. In other words, our LBKs and location-based authentication scheme reduce the impact of compromised nodes from the otherwise network-wide scale to their vicinity, more specifically, within a circle with radius $2r_0$ centered at their current locations. This greatly facilitates the design of efficient localized intrusion detection mechanisms.

It is also worth noting that our authentication scheme implicitly achieves key agreement between neighboring sensor nodes. Consider nodes *R* and *S* as an example. Since $K_{R,S}$ is equal to $K_{S,R}$, it can be used as their shared key to encrypt and authenticate subsequent messages between them via efficient symmetric-key techniques. Therefore, each node just needs to perform the public-key-based three-way handshake with each of its neighbors once during the whole network lifetime. Such overhead should be acceptable for security-sensitive UWB sensor network applications.

V. Related Work

In this section, we briefly review some important work that is closely related to this paper. Brands and Chaum [19] propose a TOA-based distance bounding protocol that can be used to verify the proximity of two devices connected by a wired link. Sastry *et al.* [20] present a similar distance bounding approach based on ultrasound and RF signals to verify the presence of a wireless device in a region of interest. In [21], Waters and Felten propose a scheme that uses round-trip ToF RF signals to prove the locations of tamper-resistant devices. Their scheme cannot be directly applied in UWB sensor networks because individual sensors are usually not tamper-resistant due to cost limitations. More recently, Lazos and Poovendran [6] present an approach to secure range-free sensor localization techniques [4]–[5]. By contrast, this paper concentrates on securing range-based localization techniques [2]–[3]. The closest work to our SLS can be found in [7], in which a scheme called verifiable multilateration (VM) is proposed for secure positioning of wireless devices. However, SLS differs significantly from VM in several major aspects. First, SLS is able to mitigate the impact of attacks and sporadic measurement errors in the first place, which is a nice property not provided by VM. Second, VM calculates location estimates on the basis of three anchors or triangles. By contrast, we consider a more general case by using an n_a-vertex polygon formed by n_a anchors for $n_a \geq 3$, which allows for higher location accuracy. Last, we propose to utilize mobile anchors instead of static anchors, which can greatly reduce the number of required anchors.

VI. Conclusion and Future Work

An important application of UWB in sensor networks is to localize sensor nodes due to its high temporal resolution. In this paper, we present SLS, a novel mobility-assisted secure localization algorithm that can furnish sensor nodes with secure, accurate locations despite the presence of attacks. In addition, we propose the novel notion of LBKs and a location-based neighborhood authentication scheme which can withstand many attacks and fulfill pairwise key agreement. As the future research, we plan to extend our approach to range-free localization techniques. We also intend to further investigate the potentials of LBKs and the UWB technology in securing sensor networks.

References

[1] R. C. Qiu, H. Liu, and X. Shen, "Ultra-wideband for multiple access communications," *IEEE Commun. Mag.*, vol. 43, no. 2, pp. 80–87, Feb. 2005.

[2] A. Savvides, C. Han, and M. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proc. ACM MobiCom*, Rome, Italy, Jul. 2001, pp. 166–179.

[3] X. Cheng, A. Thaeler, G. Xue, and D. Chen, "TPS: A time-based positioning scheme for outdoor wireless sensor networks," in *Proc. IEEE INFOCOM*, Hong Kong, China, Mar. 2004, pp. 2685–2696.

[4] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher, "Range-free localization scheme in large scale sensor networks," in *Proc. ACM MobiCom*, San Diego, CA, Sep. 2003, pp. 81–95.

[5] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proc. ACM MobiCom*, Philadephia, PA, Sep./Oct. 2004, pp. 45–57.

[6] L. Lazos and R. Poovendran, "Serloc: Secure range-independent localization for wireless sensor networks," in *Proc. ACM WiSe*, Philadelphia, PA, Oct. 2004, pp. 21–30.

[7] S. Čapkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, pp. 1917–1928.

[8] D. Wagner, "Resilient aggregation in sensor networks," in *Proc. ACM SASN*, Washington, DC, Oct. 2004, pp. 78–87.

[9] L. Eschenauer and V. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. ACM CCS*, Washington, DC, Nov. 2002, pp. 41–47.

[10] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proc. IEEE Symp. Security Privacy*, Oakland, CA, May 2003, pp. 197–213.

[11] ——, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 247–260, Feb. 2006.

[12] G. Gaubatz, J. Kaps, and B. Sunar, "Public keys cryptography in sensor networks – Revisited," in *Proc. ESAS*, EURESCOM, Heidelberg, Germany, Aug. 2004, pp. 2–18.

[13] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUS," in *Proc. CHES*, Boston, MA, Aug. 2004.

[14] D. Boneh and M. Franklin, "Identify-based encryption from the Weil pairing," in *Proc. CRYPTO*, vol. 2139, LNCS, 2001, pp. 213–229.

[15] P. Barreto, H. Kim, B. Bynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," in *Proc. CRYPTO*, vol. 2442, LNCS, 2002, pp. 354–368.

[16] Y. Zhang, W. Liu, and W. Lou, "Anonymous communications in mobile ad hoc networks," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, pp. 1940–1951.

[17] A. Shamir, "Identity based cryptosystems and signature schemes," in *Proc. CRYPTO*, vol. 196, LNCS, 1984, pp. 47–53.

[18] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2, pp. 293–315, 2003.

[19] S. Brands and D. Chaum, "Distance-bounding protocols (extended abstract)," *Theory and Application of Cryptographic Techniques*, pp. 344–359, 1993.

[20] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *Proc. ACM WiSe*, San Diego, CA, Sep. 2003.

[21] B. Waters and E. Felten, "Proving the Location of Tamper-Resistant Devices," Dept. Comput. Sci., Princeton Univ., Princeton, NJ, Tech. Rep. TR-667-03, 2003.

Yanchao Zhang (S'03), photograph and biography not available at the time of publication.

Wei Liu, photograph and biography not available at the time of publication.

Yuguang Fang (S'92–M'99–SM'99), photograph and biography not available at the time of publication.

Dapeng Wu (S'98–M'04), photograph and biography not available at the time of publication.