Secure Verification of Location Claims with Simultaneous Distance Modification

Vitaly Shmatikov and Ming-Hsiu Wang

The University of Texas at Austin

Abstract. We investigate the problem of verifying location claims of mobile devices, and propose a new property called *simultaneous distance modification* (SDM). In localization protocols satisfying the SDM property, a malicious device can lie about its distance from the verifiers, but all distances can only be altered by the same amount. We demonstrate that the SDM property guarantees secure verification of location claims with a small number of verifiers even if some of them maliciously collude with the device. We also present several lightweight localization protocols that satisfy the SDM property.

1 Introduction

In wireless networks, the physical location of a mobile device such as a sensor, a mobile phone, or a small computer often has implications for location-based access control and security of the nearby devices. A malicious device may lie about its location in an attempt to appear either farther away than its true location (*e.g.*, in order to intercept other devices' communications), or closer than it really is (*e.g.*, to subvert a location-based access control mechanism). In this paper, we study the problem of verifying location claims of potentially malicious mobile devices in an environment where some parts of the localization infrastructure may have been compromised.

To verify location claims of mobile devices, most existing protocols employ *distance bounding* [BC93]. A verifying "beacon" challenges the device and measures the time elapsed until the receipt of its response. This gives a lower bound on the distance to the device, which therefore cannot claim to be closer than it really is. Measurements from multiple beacons can then be combined to estimate the device's location.

Our contributions. We define a new property called *simultaneous distance modification* (SDM). In distance estimation protocols with the SDM property, a malicious device being interrogated by multiple verifiers can increase its claimed distance from the verifiers, but all distances can only be altered by the same amount. The SDM property enables secure verification of location claims with a small number of verifiers. In contrast to previously proposed protocols, the device's location can be verified *anywhere* on the two-dimensional plane and not just in the area enclosed by the verifiers.

In addition to the generic security argument for protocols with the SDM property, we present two practical protocols satisfying this property: (1) a challenge-response protocol based on hash chains and time-of-flight estimation, and (2) a hyperbolic localization protocol based on time difference of arrival. In contrast to the previous work, we analyze security of both protocols in the presence of malicious verifiers.

Model. We use the standard model for location verification [WF03,SSW03,ČH05]. The goal of a malicious device is to be localized in a place other than its true location. Therefore, it participates in the protocol, but tries to mislead the verifiers. This model matches practical wireless security scenarios such as location-based access control, in which a device that refuses to respond to distance estimation requests is simply denied access. Our desired security property is as follows: *if the protocol produces a location for the device, then this location must be correct.*

The device is located on a two-dimensional coordinate grid. We will sometimes refer to the device's location as a *point*, even though in reality it is a small region rather than a point due to imprecision of distance measurements.

Several verifying *beacons* are located on the grid and exchange messages with the device. We assume that signals can be linked to the device that emitted them, *i.e.*, devices have "identities." This does not imply strong authentication; the device may have a unique code or dedicated frequency. The signal recognition assumption is essential, and is made by all localization protocols in the literature. In section 6, we discuss possible attacks if a signal cannot be linked to a particular device.

All beacons are connected to a trusted central processor, or the *base station*, which computes the location of the mobile device from the beacons' reports. We assume that the only way for the base station to communicate with the device is via the beacons, *i.e.*, localization must rely entirely on the information supplied by the (potentially malicious) beacons. By default, we assume that if the protocol detects an inconsistency in the device's responses to different beacons, it will not produce a location. Denial of service attacks are beyond the scope of this paper.

We abstract from the details of physical communication between the beacons and the device. It can be based on radio [BC93,WF03], ultrasound [SSW03], or any other suitable technology. An honest beacon's correct location is known to the base station via either static pre-configuration, or an on-board GPS, or from a previous instance of localization where the beacon itself acted as the device.

We will consider both honest and malicious beacons, but assume that there is a secure communication channel (*e.g.*, a secure wire) between each beacon and the base station. In particular, we assume that a malicious device cannot interfere with the information sent by an *honest* beacon to the base station. This is a realistic assumption for many sensor and mobile networks, where devices are low-powered and have no physical access to the communication network between the beacons and the base station.

Related work. Distance and angle estimation techniques include Time Difference of Arrival (TDoA) [PCB00,SHS01,LOR06], Time of Arrival [HWLC97,SHS01], Received Signal Strength [BP00], and Angle of Arrival [NN03a]. These methods are *not* designed to be secure in the presence of malicious devices and beacons. Range-free protocols [BHE00,NN03b,HHB+03] do not require distance or angle measurements, but are also insecure in adversarial environments.

In radio-based secure distance bounding by Brands and Chaum [BC93], the prover cannot pretend to be closer to the verifier than it really is. Similar protocols based on ultrasound and ultra-wideband appear in, respectively, [SSW03] and [HK05]. Variations include authenticated challenge-response [MSC06]. Distance bounding, however, does not prevent a device from enlarging the distance, *i.e.*, claiming to be farther away than

it really is, because a malicious device can delay its responses. Furthermore, standard distance bounding can be subverted by guessing attacks or by exploiting the relatively high latency of communication channels [CHKM06].

Verification of location claims typically involves combining distance bounds from multiple verifiers. In previously proposed protocols [WF03,SSW03,ČH05], a malicious device can easily enlarge the distance in each instance of the distance-bounding protocol, and pretend to be *outside* the area enclosed by the verifiers. This is a serious security risk. For example, an untrusted device may claim to be far away from a wireless network, while locating itself in the middle in order to eavesdrop on messages.

All existing location verification protocols also assume that the verifiers are trusted. For example, the TDoA-based protocol of [ČČS06], which is superficially similar to one of our protocols, is insecure when some of the beacons (called "base stations" in [ČČS06]) are malicious. By contrast, we explicitly analyze the case when some of the beacons maliciously collude with the device whose location claims are being verified.

A complementary problem to location verification is *location discovery*: how to enable an *honest* device to determine its own location in the presence of malicious beacons [LP05,LND05a,LND05b,DFN06]. None of these protocols consider a malicious device colluding with malicious beacons to lie about its location. The only exception is the claim verification protocol of [LPČ05], which does not prevent a malicious device from pretending to be farther away than it really is.

Organization of the paper. We define the simultaneous distance modification (SDM) property and show how it guarantees secure localization in section 2. In section 3, we investigate which geometry of verifier placement prevents false location claims. In section 4, we present our protocols with the SDM property, and analyze their security in the presence of malicious beacons in section 5. In section 6, we survey attacks on the SDM property. Conclusions are in section 7.

2 Simultaneous Distance Modification (SDM)

Range measurement involves estimating the distance between a beacon and the mobile device from measurements of time, angle, or signal strength, then combining measurements from multiple beacons to localize the device. Intuitively, a range measurement protocol satisfies the *simultaneous distance modification* (SDM) property if a malicious device, by giving false responses to multiple beacons, can change each beacon's distance estimate, but all estimates can only be changed by the same amount.

Let s be the mobile device, and let $b_0, \ldots, b_n$ be the beacons within its broadcast range. Let d_i be the actual distance between s and b_i, and d'_i be the distance (possibly incorrect, due to malicious responses by s) as reported by the range measurement protocol. The $d_i - d'_i$ value is the *reported distance error* for beacon b_i. The SDM property states that, regardless of what s does, there is some constant k such that $d'_i - d_i = k$ for every honest beacon b_i. In other words, if the adversary changes the reported distance between s and some beacon by k, then he must also change the reported distance between s and every other beacon by k, or else the measurements will be inconsistent and the attack will be detected.

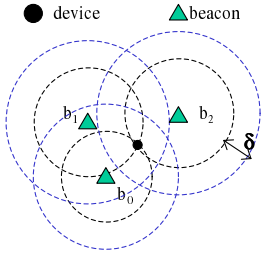

Fig. 1. Localization with three beacons.

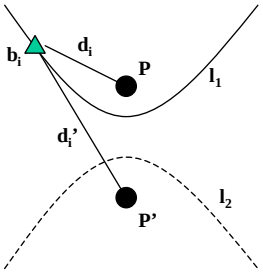

Fig. 2. Distance modification.

For the rest of this section, assume that all beacons are honest (we consider the case of malicious beacons in section 5). Recall that the goal of the malicious device is to convince the base station of a false location, *i.e.*, the reports of all beacons should be *consistent*, yet the resulting location should *not* be the device's true location.

The following lemma gives the sufficient and necessary conditions under which a false location claim by a malicious device may successfully pass verification.

Lemma 1 (Security of SDM). *Consider an honest-beacon localization protocol based on range measurement which satisfies the SDM property. A malicious device located at position p can cause the localization protocol to compute its location as p', where $p' \neq p$, if and only if* all *of the following conditions hold:*

1. All beacons within the device's range lie on the same lobe l of some hyperbola h.

2. Positions p and p' are the foci of the hyperbola h.

3. If distance bounding is used, l must be the lobe closest to p.

Proof sketch: Localization with three honest beacons is shown in fig. 1. Each circle represents a distance between the beacon and the device, as reported by the range measurement protocol. The three circles corresponding to the actual distances intersect in the device's true location. Due to imprecise measurements, the intersection is a small region rather than a single point (this does not affect our analysis). We say that two curves "intersect" if they pass within the measurement error of each other (see section 5.1). The simplest protocol is to take the intersection of the three circles corresponding to the reported distances as the device's location. If the circles don't intersect in a single location, report an inconsistency.

If the protocol for measuring the distances between the individual beacons and the device satisfies the SDM property, the device can alter each reported distance by $|d'_i - d_i| = \delta$. Intuitively, the radiuses of all three circles must expand or contract by the same δ. The protocol produces a false location if and only if the new circles "intersect" in a region other than the device's true location.

Fig. 2 shows a malicious device in position p. Let d_i be the true distance between p and beacon b_i. For the device to be localized in some $p' \neq p$, it is necessary (but not sufficient) to modify the distance reported by b_i so that d'_i is equal to the distance between b_i and p'. This must hold for *every* beacon b_i. Therefore, all beacons must lie

on the same lobe of a hyperbola whose foci are p and p'. (A *hyperbola* is the set of all locations x on a plane such that the absolute value of the difference between the distances from x to the two foci is a constant.) In fig. 2, the l_1 lobe is the set of all locations for which this difference is negative, the l_2 lobe is the set of all locations for which the difference is positive. With distance bounding (see section 1), a malicious device can pretend to be farther away, but not closer than it really is. Therefore, $d'_i > d_i$. In this case, the *only* situation in which a device located at p can successfully pretend to be located at p' is if all beacons lie on l_1, *i.e.*, the lobe of the hyperbola closest to p.

Lemma 1 says that a malicious device cannot choose an arbitrary false location. Its false location claim will pass verification *only* in the following case: if all beacons lie on a hyperbola and the device happens to be located in its focus, then it can successfully pretend to be located in the other focus. If any of the three conditions of lemma 1 is violated, the reported distances will be inconsistent, and the attack will be detected.

3 Preventing false location claims

We now investigate how many beacons and which placement geometry are sufficient to ensure that the conditions of lemma 1 can never be satisfied and, therefore, a false location claim by a malicious device can never pass verification.

Random beacon placement and pre-measurement selection. This is the most general scenario. Beacons are randomly scattered on the localization plane, and a subset of beacons must be chosen *before* the device's location claims are known. Beacon placement must be such that the chosen beacons cannot all lie on the same lobe of some hyperbola. Then, by the contrapositive of lemma 1, a false location claim cannot pass verification.

The straightforward approach is to start with the minimum number of beacons which uniquely identify a hyperbola lobe, then select one more beacon which does *not* lie on this lobe. In our setting, the lobe can lie at an arbitrary angle to the coordinate grid. To capture all possible rotations of the hyperbola, we resort to the general conic section equation, where A, B, C, D, E, F are constants:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

Since the base station knows the coordinates of all beacons, six randomly selected beacons uniquely determine some conic section. The base station solves the system of six equations and checks whether $B^2 - 4AC > 0$, *i.e.*, whether the resulting section is a hyperbola. If not, the selected set is sufficient for secure localization.

If the chosen beacons do lie on a hyperbola, the base station randomly selects the 7th beacon. With high probability, it will not lie on the same lobe, or else the base station chooses a different beacon. The minimal set for preventing false location claims thus consists of seven beacons. If the size of the beacon set must be minimized, the base station can re-sample the six beacons until they do not form a hyperbola.

Random beacon placement with post-measurement selection. In this scenario, each beacon reports its distance from the device, and the base station selects a subset of the beacons *after* receiving all distance reports. A different set of beacons can thus be used for each device. For each set, the base station computes the device position p' as

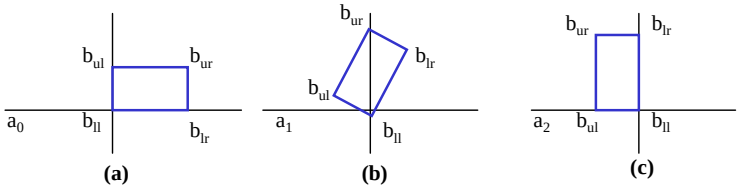

Fig. 3. Rectangular arrangement of beacons.

the "intersection" of the circles whose centers are the beacons and the radiuses are the reported distances. Three beacons are sufficient.

If the beacons' reported distances are inconsistent, *i.e.*, the circles do not intersect in a single location, then the base station aborts the protocol. Otherwise, the base station assumes that p' is a *false* location and attempts to derive a contradiction. If the latter succeeds, it concludes that p' is the device's true location.

The polar-coordinate equation for a hyperbola with a focus at the origin is:

$$r = \frac{a(e^2 - 1)}{1 - e * cos(\theta + \phi)}$$

By setting p' as the origin and using the beacons' polar coordinates with respect to that origin, this system can be solved for a, e, ϕ, uniquely identifying some hyperbola h. As before, the base station checks whether the three beacons all lie on the same lobe of h, and, if distance bounding is used, that this is *not* the lobe closest to p'. If either condition fails, p' cannot be a false location, and the device is securely localized.

If the three beacons all lie on the same hyperbola lobe, then the base station randomly selects a 4th beacon which does not lie on the lobe, and checks whether its distance report is consistent with those of the three original beacons. If it is, then p' cannot be a false location, and the device's location claim is securely verified.

Controlled beacon placement. If placement of beacons on the localization grid is not random, but controlled by some trusted entity, then the *same set of four beacons* can be used to securely verify the claims of any device. It is sufficient to find a placement topology such that the beacons cannot all lie on the same hyperbola lobe. Consider a rectangle. Observe that for every hyperbola lobe, there exists some Cartesian coordinate system such that (1) the hyperbola lobe is a function in this coordinate system, and (2) the derivative of this function changes sign only once. In any Cartesian coordinate system, a curve that passes through the four points forming the corners of a rectangle is either not a function, or requires more than one sign change in the derivative. Therefore, four beacons placed in a rectangle cannot lie on the same hyperbola lobe.

Lemma 2 (Rectangular topology prevents false localization). *If the localization protocol satisfies the SDM property, and four verifying beacons are placed in a rectangular grid, then a false location claim can never pass verification.*

Proof sketch: Denote the four beacons as b_{ll} (lower left), b_{lr} (lower right), b_{ul} (upper left), and b_{up} (upper right) and let $b_i.x$ and $b_i.y$ be, respectively, the x and y coordinates

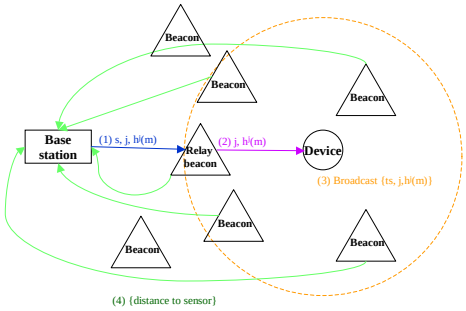

Fig. 4. SDM protocol based on hash chains.

of beacon b_i. Consider the family of Cartesian coordinate systems with the origin at b_{ll}. Fig. 3 shows some of the rotations (technically, we are rotating the coordinate system, but it is easier to visualize the rotations of the beacon rectangle around the b_{ll} origin). Denote these rotations as a_0, a_1, a_2, respectively. In all of them, two of the beacons lie on the y-axis. Therefore, no curve that goes through all four beacons can be a function.

Now consider all coordinate systems where the (rotated) principal axis lies between that of a_0 and that of a_1. In all of these systems, $b_{ul}.x < b_{ll}.x < b_{ur}.x < b_{lr}.x$. Similarly, $b_{ul}.y > b_{ll}.y$ and $b_{ll} < b_{ur}$, implying one sign change in the derivative. And $b_{ll}.y < b_{ur}.y$ and $b_{ur} > b_{lr}$, implying another sign change in the derivative. Therefore, no hyperbola lobe can pass through all four beacons in this coordinate system.

A similar argument applies to all coordinate systems where the rotated principal axis lies between a_1 and a_2. Therefore, no hyperbola can pass through all four beacons in any coordinate system rotated between 0 and 90 degrees. The proof for rotations between 90 and 360 degrees is similar. Since any Cartesian coordinate system can be x- and y-translated into a system in the above family, this completes the proof.

4 Protocols with the SDM property

4.1 Challenge-response with hash chains

We present a localization protocol based on hash chains, in which the SDM property is achieved by a simple challenge-response mechanism. The protocol can be ultrasound-based as in [SSW03] or radio-based as in [BC93,ČH05] (the latter requires extremely precise clocks in order to measure propagation time of speed-of-light signals, and cannot be used in many practical scenarios).

The base station sets up a hash chain $h^k(m)$, where m is a secret, k is a parameter (how many localizations can be performed before a new chain must be created), and h is a cryptographic hash function. The $h^k(m)$ value is distributed to all beacons. Each beacon must maintain the current chain counter c (initialized to k) and $h^c(m)$ value.

The protocol is shown in fig. 4. To localize a device, the base station sends the message $\langle j, h^j(m)\rangle$ to a randomly chosen *relay* beacon, along with a future time t_0. The only requirement on j is that it has to decrease from one instance of the protocol to the next (*i.e.*, the hash chain should be monotonically unrolled).

At time t_0, the relay beacon challenges the device with $\langle j, h^j(m)\rangle$. The device responds by broadcasting the message $\langle t_d, j, h^j(m)\rangle$ where t_d is its current timestamp. Each beacon within the broadcast range, upon receiving this message, records the current time t_i and verifies the $\langle j, h^j(m)\rangle$ value by applying hash function h to $h^j(m)$ $c-j$ times and comparing the result to $h^c(m)$. If verification succeeds, the beacon sets $c = j$, $h^c(m) = h^j(m)$, and sends the timestamp pair $\langle t_d, t_i\rangle$ to the base station.

The base station computes the reported distance from each beacon to the device as $d'_i = \frac{t_i - t_d}{v}$, where v is the speed of signal propagation. The base station constructs a circle for each beacon, with the beacon in the center and d'_i radius. If the circles do not "intersect" (*i.e.*, pass within the measurement error of each other) in a single location, the protocol is aborted, and the location claim does not pass verification. Otherwise, the device is considered localized in the small region where all circles intersect. The case of malicious beacons is discussed in section 5.

An important property of this protocol is that the device cannot broadcast a valid response *before* receiving the challenge, since this requires finding a pre-image of $h^c(m)$. Instead of $\langle t_d, j, h^j(m)\rangle$, the device can broadcast, for example, $\langle t_d, j+i, h^{j+i}(m)\rangle$ for some i such that $j+i < c$, but this can only happen *after* the device has received $h^j(m)$. Therefore, elapsed time can be artificially increased by delaying the response, but not shortened. Furthermore, because localization is based on a *single* message emitted by the device, the reported distances will change by the same amount for each beacon (under the assumption that distance is linear in time). Therefore, the protocol satisfies the SDM property. Its security then follows directly from lemmas 1 and 2.

This protocol assumes that the clocks of the beacons and the device are synchronized. (Changing the timestamp cannot help a false location claim to pass verification, but clock skew can prevent an honest claim from being verified.) To remove this requirement, the protocol can be slightly modified so that both challenge and response include t_0. The base station can then compute t_d as $\frac{t_r - t_0}{2v}$, where t_r is the time the device's response was received at the relay beacon.

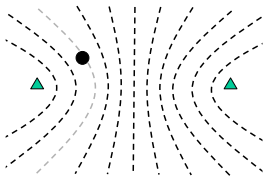

Fig. 5. Family of hyperbolas between two beacons.

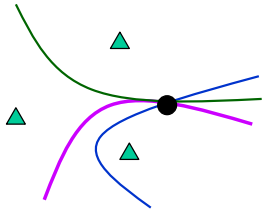

Fig. 6. Intersection of three hyperbolas.

4.2 Time difference of arrival

Time difference of arrival (TDoA) inherently possesses the SDM property. The device broadcasts an identifying signal. All beacons within range record the time of signal arrival and relay it to the base station. For each pair of beacons, the base station computes

the hyperbola corresponding to the difference between their timestamps (for any two beacons A and B, all locations whose distances from A and B differ by a constant form a hyperbola). The "intersection" of all hyperbolas is the location of the device.

This assumes that a constant difference in *time* (of signal arrival) implies a constant difference in *distance* (to the location from which the signal was emitted). The signal must travel at a constant speed v, as is the case for radio or ultrasound signals. Distance d_i between the device and a beacon is v multiplied by the time difference between the (unknown) time t_0 when the signal was sent and the time it was received.

Let t_{bi} be the time when beacon b_i received the signal. Then $v \cdot (t_{bi} - t_0) = d_i$. Even though t_0 is not known, given two timestamps t_{bi} and t_{bj} from different beacons, the base station can subtract it out to obtain the equation $v \cdot (t_{bi} - t_{bj}) = d_i - d_j$. This equation defines a hyperbola on which the signal-emitting device must be located, as shown in figure 5. Given multiple hyperbolas (one per each pair of beacons), they must "intersect" in the device's true location (see fig. 6).

TDoA-based localization satisfies the SDM property because all beacons' measurements are based on a *single* signal broadcast by the device. Observe that the time of signal emission does not enter into the TDoA calculation. Therefore, unlike distance bounding protocols, TDoA localization is not vulnerable to the distance enlargement attack, in which the device delays its response to a challenge in order to pretend that it is located farther than it really is.

4.3 Signal strength

Signal strength drops off as the inverse square of the distance [SHS01,Rap96]. A constant difference between the relative strengths of received signals does not imply that the device lies on a certain hyperbola, and the protocol of section 4.2 does not work.

The protocol based on hash chains from section 4.1 can still be used. All that is needed is some way of converting the received signal into distance. Suppose that the malicious device artificially modifies its response, *e.g.*, emits at a lower than normal signal strength in order to pretend that it is located farther away than it really is. As long as the modification is the same for all receiving beacons, as will be the case when localization is based on a single broadcast response, the protocol works.

Technically, this is not the same property as SDM, as the error in reported distances is not constant across all beacons (a constant difference in signal strength does not imply a constant difference in distance). Nevertheless, the same general principle applies. For all beacons which receive the same signal, the reported distance will differ from the true distance by a fixed amount, which depends on the true distance. Therefore, the adversary cannot pass verification for an arbitrary false location claim.

All of the above protocols assume that the signal sent by the device which is being localized cannot be modified or delayed before reaching the beacons. For example, if signal strength is artificially boosted in transit by some colluding device, localization will be incorrect. Similarly, if a non-radio signal is used, it can be artificially "speeded up" by one or more colluding devices who talk to each other by radio. Finding effective defenses against these attacks is an interesting topic for future research.

5 Preventing false location claims when beacons can be malicious

We now consider verification of location claims of a potentially malicious device in the scenario where some of the beacons may collude with it. The SDM property improves security of localization in this case, too. We emphasize that none of the existing protocols for verifying location claims provide any security guarantees in this scenario.

Naturally, even with the SDM property, secure verification of location claims is not guaranteed unless there is a bound on the number of malicious beacons. Let n be the number of beacons within the range of the device being localized, b the maximum number of malicious beacons, $g = n - b$ the minimum number of honest beacons.

We deliberately consider an extremely strong attack model. All malicious beacons collude and choose a false location for the device which is the *worst possible location* from the viewpoint of the localization protocol. In other words, the attack succeeds if malicious beacons can convince the base station that the device is located in *any* position other than its true location. In reality, a malicious device may wish to be localized in a *specific* false location, so "insecurity" in our model does not always imply insecurity in practice. Vice versa, if the protocol is secure in our model, then it is also secure in any realistic deployment scenario.

Depending on the beacon placement procedure, malicious beacons may not freely choose their own locations on the grid (*e.g.*, if the beacons' layout is configured by the base station). With static beacons, the topology may enable malicious beacons to produce false locations for some devices, but not others. It is much more difficult for a coalition of malicious beacons to convince the base station of false locations for multiple devices. We will further strengthen the attack model by assuming that the base station does not notice inconsistencies between multiple runs of the localization protocol. Finally, we will assume that malicious beacons can eavesdrop on all distance and time measurements reported by the honest beacons. This is too strong in many scenarios, *e.g.*, when each beacon is connected to the base station by a dedicated wire.

5.1 Challenge-response

As before, we require that if the protocol produces a location, then the location must be correct. If the device is malicious, the protocol may fail to provide an answer. This is not a significant limitation, because in the standard location claim verification scenario [WF03,SSW03,ČH05], the objective of a malicious device is to convince the base station of a false location.

We add the following voting scheme to the protocol of section 4.1.

1. Let t be a threshold value, which is a parameter of the protocol. It is equal to the fraction of reported distances that must be consistent before the base station decides that the device has been localized.
2. For each beacon that reported distance d'_i to the device, the base station computes a circle of radius d'_i centered at that beacon.
3. Let P be the set of locations in which at least $t \cdot n$ distance circles "intersect" (*i.e.*, pass within the measurement error of each other).
4. If set P is empty, return a special symbol, indicating that the answer is inconclusive.

5. For each location $x_i \in P$, define $c(x_i)$ to be the number of distance circles "intersecting" in that location. Note that $c(x_i) \geq t \cdot n$.
   Let $X = \{x_i \in P \text{ s.t. } \forall j \neq i \; c(x_j) \leq c(x_i)\}$ be the set of locations where most circles "intersect."
6. If $|X| > 1$, the answer is inconclusive; else let p be the single location contained in X.
7. Return p as the device's location.

Security analysis (honest device, malicious beacons). If the device is honest, the base station will receive at least g correct distances from the good beacons. All corresponding circles intersect in the true location.

Can colluding malicious beacons produce a false location in which the number of intersecting circles exceeds the threshold as well? First, the malicious beacons have to find the region p' in which the second highest number of honest beacons' circles pass within the measurement error of each other (the region with the highest number of intersections is the true location). Note that (a) such a region may not exist, and (b) malicious beacons cannot freely choose an arbitrary point as the false location. Let m be the number of honest beacons' circles intersecting in p'. Each malicious beacon modifies its distance report so that the resulting circle passes through p'.

The number of votes for the false location p' is $b + m$, where b is the number of bad beacons. The number of votes for the correct location is at least g (some of the malicious beacons' circles may pass through the correct location in addition to the false location). Correct localization is only guaranteed if $g > b + m$.

Deriving a theoretical upper bound on m is difficult, as it depends on the layout of the beacons, device location, and precision of distance measurement. We use simulation instead. Our setup consists of a square grid, with the device being localized positioned in its center, and n beacons randomly scattered within the device's broadcast range. The hyperbolas or distance circles (depending on the localization protocol) are computed for each beacon and overlaid on the grid. Two curves are considered to intersect at position p if both pass within the distance measurement error of p. Our simulation parameters are consistent with the specification of PAL650 UWB Precision Asset Location system [FRB03]: the communication range between a device and a beacon is 200 feet (indoor) or 600 feet (outdoor), measured with 1-foot precision. By default, the device is falsely localized if the location produced by our protocol differs from the correct location by more than 20 feet.

Fig. 7 shows the number of circles intersecting in the false location p' with the second highest number of intersections, averaged across 5000 simulations, assuming a 200-feet communication range. It is much smaller than the number of intersections in the correct location, which is equal to the number of beacons.

Security analysis (malicious device, malicious beacons). This case is difficult because *all* reported distances, including those reported by the honest beacons, may be incorrect. The SDM property ensures, however, that the distances reported by the honest beacons are changed by the same amount viz. correct distances. Therefore, if the device attempts to alter its reported distance to one of the honest beacons, it has no control over the distances reported by the other honest beacons.

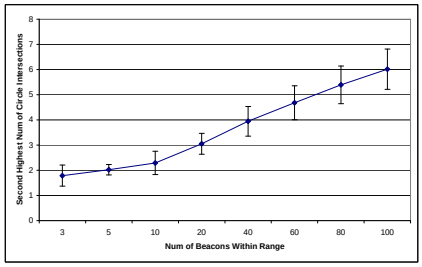

Fig. 7. Number of circles intersecting in the false location.

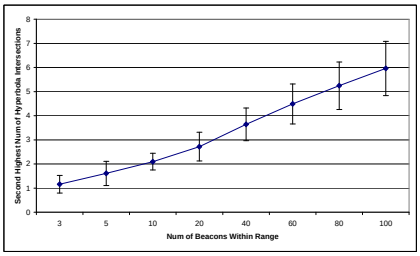

Fig. 8. Number of hyperbolas intersecting in the false location.

As explained in section 2, the number q of honest beacons' circles that will intersect in the false location is equal to the number of honest beacons that happen to lie on the same lobe of a hyperbola whose focus is the true location of the device, and whose other focus is the false location. This number is very small relative to the total number of beacons (see fig. 9 for beacons with 200-feet communication range).

The total number of circles that intersect in the false location is $b + q$. The protocol will output the false location if $\frac{b+q}{n} \geq t$. Therefore, our protocol guarantees secure localization of a malicious device even in the presence of malicious beacons as long as $g \geq b + max(m, q) + 1$.

5.2 Time difference of arrival

An important advantage of TDoA localization (see section 4.2) is that it doesn't matter whether the device is malicious or honest. We adopt the following voting protocol.

1. For each beacon b_i, the base station constructs $n - 1$ hyperbolas as described in section 4.2), one per each beacon b_j where $j \neq i$.
2. Let P be the set of locations in which at least two of the constructed hyperbolas "intersect," *i.e.*, pass within the measurement error of each other.
3. For each location $x_l \in P$, define $h(x_l)$ to be the number of hyperbolas "intersecting" in that location. Let $X = \{x_m \in P \text{ s.t. } \forall l \neq m \; h(x_l) \leq h(x_m)\}$ be the set of locations where most hyperbolas "intersect."
4. If $|X| > 1$, the beacon abstains. Otherwise, its vote is the single location contained in X.
5. The location with the most beacon votes is determined to be the device's location.

Security analysis. For each beacon pair when both beacons are honest, the hyperbola passes through the true location p. Therefore, for each honest beacon, at least $g - 1$ hyperbolas will intersect in the true location.

As in the challenge-response protocol, the worst possible false location p' is the region where the second highest number of hyperbolas intersect. Let m be this number. The only situation in which an honest beacon will abstain or vote for a false location is when $g - 1 \leq b + m$. The probability of this happening is very small when beacons are

scattered randomly on the localization grid (see fig. 8). Therefore, as long as there are slightly more honest beacons than malicious beacons, each honest beacon will vote for the correct location.

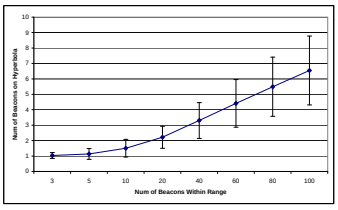

Fig. 9. Number of beacons lying on a hyperbola.

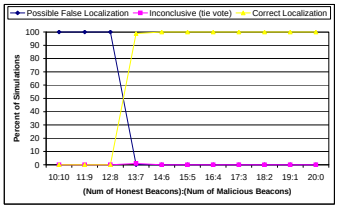

Fig. 10. TDoA localization with malicious beacons (numbers for false localization are a conservative upper bound).

The $b+m$ upper bound on the number of hyperbolas intersecting in the false location is very conservative. To achieve it, *every* malicious beacon must report the signal-receipt timestamp such that *all* of the resulting TDoA hyperbolas pass through p'. This can only happen if *all* honest beacons lie on the same lobe of a hyperbola whose foci are p and p'. The probability of this is very small (see fig. 9).

As long as $g > b$, and the vote of each honest beacon is correct, the protocol will produce the correct location. Even if some of the honest beacons' votes are incorrect, the protocol produces the correct location as long as fewer than $\frac{m}{2}$ of the honest beacons lie on a hyperbola whose foci are the true and false locations. Finally, the attack will fail completely if the false location is anything other than a focus of this hyperbola.

Simulation results with 200-feet communication range are shown in fig. 10. As mentioned above, these numbers are a very conservative upper bound on the attackers' ability to have a false location claim successfully pass verification.

Existence of more than one location with a non-trivial number of votes should be treated as an anomalous event. In particular, if location p received the highest number of votes v, location p' has the second-highest number of votes v', and v' is close to v, the base station should suspect that an attack is in progress and verify whether a large number of reporting beacons happen to lie on a hyperbola whose foci are p and p'. Once the attack is confirmed, all subsequent reports from these beacons should be ignored.

6 Attacks on the SDM property

SDM property fundamentally relies on the assumption that all beacons' reports are based on a single signal sent by the device. To break the SDM property, a malicious device must be able to send different signals to different beacons. This requires the device to carry directional antennas, or else this can be achieved by device *cloning*, where multiple physical devices pretend to be the same device for the purposes of localization. Note that direct attacks on distance bounding, such as those described in [CHKM06], do not violate the SDM property.

One simple attack is to send multiple signals at different strength so that far-away beacons do not receive the weaker signals. This naive attack is easily detected by the honest beacons located close to the device because they will receive multiple signals. A more sophisticated attack involves *beam forming*. While broadcast is usually omni-directional, beam forming allows the signal to be sent directionally. To succeed, the malicious device must form a separate beam for each honest beacon. The device must not only have the physical capacity for beam forming (not feasible for many mobile devices), but also to know the locations of all honest beacons within range. Moreover, if localization is based on time-of-flight measurements, all targeted signals must be sent within a relatively short interval.

Another attack involves colluding devices who jam and/or replay each other's signals. This requires a large number of malicious devices, and is not realistic in many practical scenarios. If multiple devices at different locations share the same identity, they can each send a different message to a subset of the honest beacons.

Defending against cloning and directional signals is a difficult challenge, and an interesting topic for future research. Proposed defenses include hiding locations of the beacons [ČČS06]. Our protocols are compatible with this defense, and the generic security argument given in sections 2 and 3 holds when the beacons' locations are hidden. The analysis in [ČČS06], however, does not consider the case of malicious beacons colluding with the device.

In this paper, we focused on verifying location claims of a single device. When multiple devices are being localized, interference and missed signals are possible. Because our protocols require that a sufficient number of honest beacons receive the device's signal, the protocol may need to be repeated several times. Each protocol session must include a unique session id so that different sessions can be distinguished.

We assumed that communication between the beacons and the base station is secure. If the adversary has the ability to block the reports of honest beacons, verification of location claims does not appear feasible since the base station will be computing the location solely from the reports of malicious beacons.

7 Conclusions

We proposed a new *simultaneous distance modification* property for distance estimation protocols, and demonstrated that this property enables secure verification of location claims of mobile devices with a small number of verifiers, and regardless of the device's position relative to the verifiers. We also presented two lightweight localization protocols based on, respectively, challenge-response and time difference of arrival. These protocols prevent false location claims even if some of the verifiers are malicious.

References

[BC93] S. Brands and D. Chaum. Distance-bounding protocols (extended abstract). In *EUROCRYPT*, 1993.

[BHE00] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. Technical Report 00-729, Computer Science Department, University of Southern California, April 2000.

[BP00] P. Bahl and V. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, 2000.

[ČČS06] S. Čapkun, M. Čagalj, and M. Srivastava. Secure localization with hidden and mobile base stations. In *INFOCOM*, 2006.

[ČH05] S. Čapkun and J-P. Hubaux. Secure positioning of wireless devices with application to sensor networks. In *INFOCOM*, 2005.

[CHKM06] J. Clulow, G. Hancke, M. Kuhn, and T. Moore. So near and yet so far: distance-bounding attacks in wireless networks. In *ESAS*, 2006.

[DFN06] W. Du, L. Fang, and P. Ning. LAD: localization anomaly detection for wireless sensor networks. *J. Parallel Distrib. Comput.*, 66(7):874–886, 2006.

[FRB03] R. Fontana, E. Richley, and J. Barney. Commercialization of an ultra wideband precision asset location system. *IEEE Conf. on Ultra Wideband Systems and Technologies*, 2003.

[HHB+03] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *MOBICOM*, 2003.

[HK05] G. Hancke and M. Kuhn. An RFID distance bounding protocol. In *SecureComm*, 2005.

[HWLC97] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice*. Springer-Verlag, 1997.

[LND05a] D. Liu, P. Ning, and W. Du. Attack-resistant location estimation in sensor networks. In *IPSN*, 2005.

[LND05b] D. Liu, P. Ning, and W. Du. Detecting malicious beacon nodes for secure location discovery in wireless sensor networks. In *ICDCS*, 2005.

[LOR06] LORAN. LORAN-C general information. `http://www.navcen.uscg.gov/loran/`, 2006.

[LP05] L. Lazos and R. Poovendran. SeRLoc: Robust localization for wireless sensor networks. *ACM Trans. Sensor Networks*, 1(1):73–100, 2005.

[LPČ05] L. Lazos, R. Poovendran, and S. Čapkun. ROPE: Robust position estimation in wireless sensor networks. In *IPSN*, 2005.

[MSC06] C. Meadows, P. Syverson, and L. Chang. Towards more efficient distance bounding protocols for use in sensor networks. In *SecureComm*, 2006.

[NN03a] D. Niculescu and B. Nath. Ad hoc positioning system (APS) using AoA. In *INFOCOM*, 2003.

[NN03b] D. Niculescu and B. Nath. DV based positioning in ad hoc networks. *J. Telecommunication Systems*, 2003.

[PCB00] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *MOBICOM*, 2000.

[Rap96] T. Rappaport. *Wireless Communications: Principle and Practice*. Prentice Hall, 1996.

[SHS01] A. Savvides, C-C. Han, and M. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Mobile Computing and Networking*, pages 166–179, 2001.

[SSW03] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *WiSe*, 2003.

[WF03] B. Waters and E. Felten. Secure, private proofs of location. Technical Report 667-03, Department of Computer Science, Princeton University, January 2003.