Range Authentication Protocols for Localization

Catherine Meadows and Paul Syverson
Center for High Assurance Computer Systems
U.S. Naval Research Laboratory
{meadows,syverson}@itd.nrl.navy.mil

1 Introduction

A *distance bounding protocol* is a type of challenge-response protocol used to measure the distance of a responder (or prover) from an initiator (or verifier). The advantage of the protocol is that, assuming that the prover does not collude with anybody else, he cannot make himself appear any closer than he is. If he colludes with somebody else, he can only appear as close as his cohort.

Distance bounding protocols were originally proposed for the verification of proximity. More recently, they, and a similar type of protocol with somewhat different trust assumptions, the *authenticated ranging protocol*, have been proposed for secure localization in sensor networks. Many localization techniques relay on collecting ranging measurements, and distance bounding and authenticated ranging provide a means of doing this a degree of security. However, the security obtained is not perfect, as we shall see, and the trust assumptions must be considered carefully when deciding how to use them.

In this work we concentrate on the application of distance bounding and authenticated ranging to secure localization. We are interested in the case in which only part of the network (perhaps restricted to one geographical portion of the network) knows its location, and location information must be propagated securely to other parts of the network. We are interested in questions like such as the following: How is a node able to determine that another node may be lying about its location? Does it make sense to try to authenticate ranging information of even nodes that lie, or at somewhat less cost, only try to authenticate ranging information of honest nodes? How can we best employ different techniques for securing ranging information?

The rest of the paper is organized as follows. We begin with a survey of distance bounding and range authentication protocols, paying particular

attention to a discussion of their vulnerabilities and how they are relevant to our attended application. The next section is concerned with an in-depth analysis of the the type of scenario we are dealing with. With respect to that scenario, we discuss the effectiveness of three different hypothetical algorithms involving a verifier V trying to find out its distance from a prover P. In the first case, V is able to detect whether P is lying. In the second case, P is able to determine whether or not P is pretending to be closer than it is. This is the case of the distance bounding protocol. In the third case, V is not able to detect anything about the veracity of P, but a dishonest P is not able to impersonate an honest P' unless P''s authentication information has been compromised. This is the case of the range authentication protocol. The results from this analysis inform the last section of our paper, which describes a proposed protocol for securing ranging information, and gives a security analysis.

2 Overview of Distance Bounding and Range Authentication Protocols

In this section we give an overview of distance bounding and range authentication protocols, and the types of attacks they are subject to. But first, we give a word on notation.

The protocols presented in this section depend upon two types of challenges and responses: a rapid challenge and response, which is used to measure time of flight, and an authenticated challenge and response, which is untimed. We use the following notation to describe the rapid challenge and response, or rapid exchange:

$$V \leftrightarrow P : M \leftrightarrow N$$

A number of the protocols we describe in this and the next section will rely on the rapid bit-by-bit exchange of data. When a particular data element N is used, we denote the i'th bit of N by N_i. Also, i The bit-by-bit exchange of N and M by P and V is denoted by

$$\sum V \leftrightarrow P : N_i \leftrightarrow M_i$$

2.1 Distance Bounding Protocols Based on Time-of-Flight

2.1.1 The Echo Protocol

The first protocol to deal with location verification in a wireless environment is the Echo protocol of Sastry et al. [10]. In this protocol the verifier sends a

nonce to the prover, and the prover returns it, with the bits in reverse order, sending the last bit in the nonce as soon as it receives it. In the basic version of the Echo protocol, no authentication is used, although there is some discussion of how it could be added.

2.1.2 The Brands-Chaum Protocol and Its Descendants

The Brands-Chaum [1] distance bounding protocol actually predates the Echo protocol. It was designed in response to the grandmaster attack on zero-knowledge protocols, in which a man in the middle passes a challenge from A to B and passes off B's response as its own. Brands and Chaum developed a protocol that authenticates a principal P who is trying to prove that it is no more than a certain distance from a verifier V.

This protocol makes use of a technique that has become typical for protocols that authenticate distance, and we recall terminology that we originally used in [8] to describe it. Some responses to challenges are used to estimate distances, we refer to such a response as a *rapid response*, since it must be used to estimate distance via a round trip. Others are used to authenticate the rapid responses, and these we refer to as an *authenticated response*.

The Brands-Chaum protocols starts with the prover P committing to a nonce. Part of the commitment is sent to the verifier V. This is followed by a rapid bit-by-bit exchange between V and P in which P's rapid response is the bit-by-bit exclusive-or of its nonce with V's. At the end, P reveals its commitment and also signs V's nonce, as follows:

1. $P \rightarrow V$: commit(N_P)
2. $\sum V \leftrightarrow P : N_V \leftrightarrow N_{Vi} \oplus N_{Pi}$
3. $P \rightarrow V$: reveal commitment, $S_P(M)$

where M is the set of messages passed between V and P in the second step.

Note that there is nothing preventing a dishonest P' from observing the message exchange between V and P, preventing the last message (which may be sent over a slower channel) from reaching V, and substituting $S'_P(M)$. Brands and Chaum were not considering such interception attacks, but they can be prevented by having P's identity as well as N_P committed in the beginning.

2.1.3 Čapkun and Hubaux

Čapkun and Hubaux consider several variants of the Brands-Chaum protocol for use in distance bounding in wireless sensor networks for use in *verifiable*

multilateration. Their version of the distance bounding protocol is one in which the bit-by-bit exchange is replaced by the exchange of two messages, where the second is in reverse order, as in the Echo protocol. Their protocol is as follows:

1. P generates random nonce N_P and $c, d = \text{commitment}(N_P)$.
2. $P \rightarrow V : c$
3. $P \leftrightarrow V : N_V, N_V \oplus N_P$
4. $P \rightarrow V : E_{K_{PV}}(P||N_V||N_P), d$

V verifies the commitment and makes the measurement.

The protocol is used in verifiable multilateration as follows: Suppose that three verifier nodes V_1, V_2, and V_3 hear from a fourth node P that claims to be within the triangle formed by the three nodes. Each of the nodes performs a distance bounding protocol with P. If P is not actually at the location claimed, it must be pretending to be closer to one of the V_i than it is. The distance bounding protocol, will detect that P is actually farther from V_i than it claims to be.

The Brands-Chaum protocol is intended to be secure against a prover who tries to make itself look closer than it is by sending out its bits too soon. Čapkun and Hubaux note that, if we limit ourselves to verifying distance of trusted provers, then we can relax the requirement that the prover respond immediately. In their *authenticated ranging* protocol, they amend their distance bounding protocol to have the prover include an interval between the receipt of the verifier's challenge and its response.

Another use of distance bounding and verifiable multilateration is given in the ROPE [5] protocol. In this protocol an unseated node computes its distance from three trusted beacons using distance bounding. If it locates itself within a triangle formed by the three locators, then then it can perform verifiable multilateration, since none of the locators can appear, or can be made to appear, closer than they are. We note that since the beacons are trusted, this is a natural place to use authenticated ranging. We will discuss the merits of using authenticated ranging versus distance bounding in different contexts in more detail later in this paper.

2.1.4 Hancke and Kuhn

In the protocol of Hancke and Kuhn [4], the prover and verifier (or the verifier alone, in a variant) generate a psuedo-random number by taking a keyed hash over two nonces. The number is divided up into two bit-strings. In the rapid

bit exchange, the prover responds to the verifier's i'th challenge C_i by sending the i'th bit of sequence 0 if C_i is a zero, and the j'th bit of sequence 1 if it is a one, as follows:

1. $V \rightarrow P : N_V$
2. $P \rightarrow V : N_P$
   $h(K_{VP}, N_V||N_P) = R_1^0||R_2^0 \cdots R_n^0||R_1^1||R_2^1 \cdots R_n^1||$
3. $\sum V \leftrightarrow P : C_i \leftrightarrow R_i^{C_i}$

The main advantage of this protocol over Brands-Chaum is that, since all the bits are known, they do not have to be recovered to verify the hash. This gives greater tolerance for error. However, the security properties are essentially the same.

2.1.5 Meadows, Syverson, and Chang and Meadows, Poovendran, Pavlovic, Syverson, and Chang

These protocols [9, 8] are closely related to the Čapkun-Hubaux protocol. differing mainly in its use of the commitment. In the first, [9] the commitment is taken over the prover's name and its nonce, and used in the place of the nonce in the rapid exchange. In the second, [8] the commitment is done away with altogether, and the protocol instead relies on the assumption that an attacker could not modify a message from an honest prover without being at least as close to the verifier as the real prover and/or slowing the message down. This assumption will be discussed in more detail in the section on attacks.

The following is an abstraction of both the protocols:

1. $V \rightarrow P$: request
2. $V \leftrightarrow P : N_V \leftrightarrow F(P, N_P, N_V)$
3. $P \rightarrow P : Pos_P, N_P, N_V, MAC_{K_{PV}}(P, i, Pos_P, \{I_P\}, N_P, N_V)$

where MAC is a message authentication code. In [9] the function F is $h(P, N_P) \oplus N_V$, where h is a one-way function. In [8] F must have the properties that 1) V can verify that $F(P, N_P, N_V)$ was computed using P, N_P, and N_V, and 2) that the probability of any principal other than V guessing the first k bits of $f(P, N_P, N_V)$ is no greater than p^k, even in P and N_P are known, where $p < 1$ is a constant, and k is a parameter depending upon the length of N_V.

2.2 Distance Bounding Protocols Using Non-Interactive Range Measurement

All of the protocols we have discussed so far are based on interactive time of flight measurements. Non-interactive techniques are vulnerable to wormhole attacks, and would appear to be harder to secure. However, it is possible to secure them by introducing interaction on another level. We discuss some of these properties below.

2.2.1 Čapkun, Čagajl, and Srivastava's TDOA-Based Protools

In cite [14], Čapkun et al. describe a protocol using *time difference of arrival* (TDOA). In TDOA multiple receivers record the time of arrival of a signal. By comparing the differences between the time of arrival, two receivers can measure the difference between the distance of the source to the receivers and thus locate the source of the signal on a hyperboloid. Four or more receivers (in three-dimensional space) can compute three or more hyperboloids and thus locate the source of the signal.

In addition to wormhole attacks, this method of localization is vulnerable to an attacker using directional antennas to send different signals to different receivers at different times. If the attacker knows the location of the receivers, it can thus assume any position. The solution recommended by Čapkun et al. is to use either hidden or mobile base stations to do the localization. The solution recommended against wormhole attacks is to use a timed challenge-response scheme, so that the victim of a wormhole attack would not be able to respond in time. The challenge-response scheme is not as fine-grained.

2.2.2 Mayrhofer, Gellerson, and Hazas

Mayrhofer et al. [7] describe a non-interactive range authentication protocol using an out-of-band interactive authentication channel. We give a broad outline of the protocol as follows. It is assumed that the verifier is claiming a particular location, and that verifier and prover share a method of encoding digital messages as time intervals. The principals use both RF and ultrasound channels. The verifier and prover exchange keys using the RF channel. The prover encrypts a nonce with the key and sends it to the verifier, again over the RF channel. It then encodes the nonce as a time interval I. It then sends an RF signal and, I time units later, sends the ultrasound signal. The difference in velocity of RF and ultrasound is such that the time of flight of the RF signal is negligible with respect to the time of flight of the ultrasound signal. Thus, the time interval between the verifier's receiving the RF and ultrasound signals can be taken as equal to I plus the time it takes for the signal to travel

from prover to verifier. The prover, on receiving the signal from the verifier, subtracts the time corresponding to the claimed distance, and then retrieves the nonce. It then uses a combination of time of flight and angle of arrival to verify the position of the source of the signal. The angle of arrival is computed, not using difference in time of flight, but analysis of peak signal values, where an incoming pulse registers the highest peak with the receiver oriented most closely to the source. This, according to the authors, makes it less susceptible to directional antenna spoofing attack on TDOA.

There are two things that could go wrong. Either the owner of the key is claiming a false position, or an attacker at the true position is falsely claiming to be the owner of the key. In the former case, the attacker can lie about its distance but will not be able fake the angle of arrival unless it is colinear with the prover and verifier. In the latter case, the attacker will not know the key, so it will not be able to decrypt the nonce and encode it into the message.

2.2.3 Simultaneous Distance Modification

In [15] Wang and Shmatikov propose a primitive they call *simultaneous distance modification*. The idea is that an attacker claiming a false location can only misrepresent its distance from a beacon by a constant amount. If this property is achieved, Wang and Shmatikov show that an attacker can misrepresent its position only if the verifying beacons lie on an hyperbola with the real and pretended loci as the foci. They also demonstrate that in this case it is enough to have the beacons set up in a rectangular grid. They also study the case in which some of the beacons may be untrustworthy. They give a sample protocol that implements simultaneous distance modification. We will construct another protocol that has this property later in this paper.

2.3 Attacks on Distance Bounding Protocols

2.3.1 Attacks on Brands-Chaum and Related Protocols

There is a well-known attack on all distance bounding protocols that was originally pointed out by Desmedt [3]. This is the *terrorist attack*, in which a dishonest prover shares the information needed to perform the rapid exchange with a closer-in cohort, while sending the slower authenticated message itself. There are two types of solutions that have been proposed to this problem. One is to make it impossible for the dishonest prover to share the information needed to compute the rapid exchange without revealing the information needed to authenticate itself. However, a defense that relies upon the unwillingness to share secret information only makes sense in a the case in which the dishonest prover and its cohort are cooperating but mutually distrusting.

This is not likely in a sensor network environment, where they are both likely to be controlled by the same attacker. The other solution is to rely upon tamper-proof hardware. In this case we can rely on the weaker range authentication protocol since in such a case we are preventing dishonest provers from appearing at all.

There is also a more subtle attack on the Brands-Chaum protocol, Because it is designed to protect against somewhat different threats compared with its successors, the prover only commits to its nonce, N_P, not its nonce and its identity. This means that an attacker could witness the protocol and prevent the final authenticated message from reaching the verifier. It could then substitute a new message signed using a key belonging to the attacker.

If one reads the Brands and Chaum paper closely, one realizes that they were not considering the type of man-in-the-middle attacks in which traffic was actually destroyed. Instead, they were considering either dishonest provers who were responding to the protocol directly, or man-in-the-middle attacks in which the attacker was relaying responses that would not have reached the verifier otherwise.

We believe that man-in-the-middle attacks on the authenticated message that involves preventing legitimate traffic from reaching the verifier are realistic in our scenario, however. Although the rapid exchanges should take place at lower levels of the network stack, there is no reason not to implement the authenticated message at a higher level. This means that it will be vulnerable, not only to physical jamming, but to higher-level network attacks, which will make it easier to prevent the authenticated message from reaching its intended destination. This shows that it pays to be careful when adapting a protocol intended for one context for use in another.

2.3.2 Attacks Described by Clulow et al.

In [2] Clulow et al. describe a number of different attacks on distance bounding protocols. They present two main families of attacks: attacks on protocols that use packet-based instead of bit-by-bit rapid exchanges, and attacks involving the type of radio signal used and the way bits are encoded in the signal. We will discuss each of these in detail.

The first and most generally applicable attack applies to the Echo protocol and any protocol that uses the trick of having the verifier reverse the prover's nonce. A prover is supposed to start sending immediately after receiving the last bit of the verifier's nonce. However, a dishonest prover can gain a one-bit advantage with a 50 percent chance of success by guessing the verifier's last bit before it is received. Since the prover will already have seen all of the other bits, it will not need to guess those. Thus, the security of a rapid packet exchange

is closer to that of a single rapid exchange of a single bit than an exchange of multiple bits. This is further supported by Clulow et al.'s observation that it is the time interval between last bit of the challenge and first bit of the response for which the timing estimate is likely to be more accurate. We note that not only the Echo protocol but the packet-based protocols in [13, 9, 8] are subject to the last bit guessing attack.

Clulow at al. note that all of these attacks depend upon the freedom given to the prover to choose to respond earlier than the designated time. Thus they only apply to distance bounding protocols, not to range authentication protocols.

We note three other things about the attacks against packet exchanges. First of all, the degree of success obtained by the attacker is bounded by the time to process one bit. Thus the vulnerability is implementation-dependent. Secondly, it can be prevented by using multiple rapid packet exchanges, where the probability of that attack's success is reduced to $1/2^k$, where k is the number of rapid exchanges. Finally, there may be other good reasons for using rapid packet exchanges instead of rapid bit exchanges. In particular, rapid packet exchanges can be useful in preventing pulse delay attacks, which, while they are not relevant to the original application of distance bounding protocols, are relevant to the type of setting we are working in. We will discuss pulse delay attacks, and how rapid packet exchanges can help prevent them, later in this section.

The other attack on packet-based protocols occurs when the prover's packet begins with some header information before the prover's rapid response. This could include synchronization bits, protocol identification, etc. Because the prover can construct these before receiving the verifier's nonce, it can send its response before it has finished receiving that nonce, and thus appear closer than it is. Thus, in any implementation the prover's rapid response packet must begin with the rapid response information specified in the protocol.

The signal-based attacks in [2] rely on the time to process reception of a bit. If the signal representing a bit is of a non-trivial length, then an attacker with a more sensitive receiver can detect early whether a signal represents a one or a zero, and respond early, thus appearing closer. This can be exploited by a dishonest prover who uses it to respond early, or two attackers, one positioned near the verifier, one position near the prover, who can use it make make it appear that an honest prover appear close to an honest verifier by guessing the honest principals' signals early and relaying them to each other. The remedy recommended by Clulow et al. is to use a narrow-width signal, such as in ultra-wideband.

2.3.3 Pulse Delay Attacks

For our applications we are interested in more than dishonest nodes attempting to appear closer than they are or to make honest nodes appear closer than they are. We are also be interested in the case in which dishonest nodes attempt to make honest nodes appear further away than they are. Thus we may need ways of countering the *pulse delay* attack of [6], in which an attacker uses jamming to prevent a signal from reaching a verifier and later replays it. This makes it appear as if the signal took a longer time to reach the verifier. A similar attack can be launched against the prover, delaying the prover's response by delaying its reception of the verifier's challenge. In [6] the pulse delay attack is used against synchronization protocols, but it could just as easily be used against range authentication.

We follow the approach of Sun at al. [11], in which an upper bound is put on the delay after which a response will be accepted. This delay is less than or equal to the time it should take to jam and replay the response. We note that, for narrower signals such as recommended by Clulow et al., this will decrease the usefulness of the protocol. However, we can increase the size of the bound by requiring that both the challenge and response be more than one bit long. An attacker must continue jamming the frequency for as long as it would take the challenge or response to arrive, and cannot send the delayed challenge or response until after this is done. This use of multi-bit challenge and responses gives a defense against this attack, even if the actual distance measurement is done only on the basis of the first bit sent or received.

2.3.4 Message Integrity Attacks

Generally, when we are dealing with a cryptographic protocol, whether wired or wireless, we assume that an attacker can modify any message in transit any way it pleases. This is because it could conceivably have control over a node in the route from source to destination, and could modify a packet while it has control over it. This assumption is clearly not realistic for distance bounding and range authentication protocols, since such an attack would slow the signal down so much that it would be obviously bogus. However, this does not rule out the possibility that an attacker might be able to modify the signal directly while it is in flight. We note that in [12] Čagalj et al. address this problem by constructing coding techniques that make it harder for attackers to mount such attacks. These consist of three steps. First, on-off binary encoding is used so that, while an attacker can change a zero to a one by inserting a signal, it can only change a one to a zero by canceling a signal. Next, error-correction schemes are used such that it is impossible to alter one codeword to another except by changing a zero to a one. Finally, it is made more difficult for the

attacker to cancel a one by having the sender send multiple copies signals each with a different randomly chosen phase, whenever a one is to be sent.

3 Threat Model and Scenario Analysis

3.1 Basic Scenario

In this section we give the threat model and scenario analysis that we use as a basis for our protocol design. We start by describing the basic scenario. We then give the types of assumptions that we make about nodes' ability to authenticate distance, and then we evaluate the effects of these assumptions with respect to these scenarios.

In our model we assume that there is a set of nodes (seated nodes) that know and advertise their location, as well as a number of unseated nodes that try to learn their location from the unseated nodes. Some of the seated nodes are honest and give the correct location and/or distance when asked. Some are dishonest and give incorrect location and/or distance information in order to mislead the unseated nodes. No a priori assumptions about trust are made, and, to simplify our argument, we assume that the only error in reported location and distance is the error purposefully introduced by the dishonest node. In this case (as in most conventional localization schemes), the unseated node's strategy is to go with the location recommended by the majority of nodes. In order to determine its location, an unseated node V determines its distance $d(V, P_i)$ from a number of seated nodes P_i, each advertising its location L_i.[1] For each P_i, V draws a circle of radius $d(V, P_i)$, and casts one vote for each point on the circle. Thus, if k circles intersect at a point, that point gets k votes. The point with the most votes becomes P's location.

Of course, there are many other ways of performing multilateration besides voting. However, all these algorithms have the desired effect that, if a majority of nodes endorse a location, then that location should be chosen, or some location close to it. Thus, we can use voting as a rough approximation to inform our thought experiments.

There are two types of nodes from which an unseated sensor can learn its location. One is the set of nodes it can communicate with the sensor directly. We refer to these as *helper* nodes. The other is the set of nodes that cannot communicate with the sensor directly, but can observe the behavior of the helper nodes and forward their opinions to that sensor. We refer to these nodes as *recommender* nodes. We assume that a sensor can directly

[1] The nodes are named 'V' for verifier and 'P' for prover, for reasons that should become apparent later in the paper.

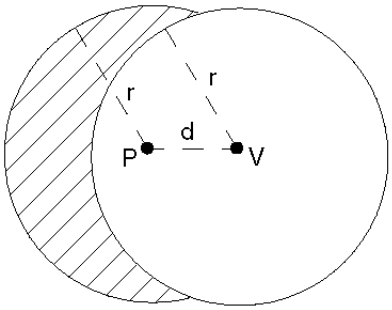

Figure 1: Recommenders must be within shaded region

authenticate any recommendation from a recommender node. Otherwise, the recommendation would only be as good as the entity that forwards it.

Thus we attach to each unseated node a set of helper nodes which consists of the nodes within its range. To each unseated node A and helper node B we attach a set of recommender nodes that is within range of B but out of rang of A, as is illustrated in Figure 1. As we can see, the closer a helper node is to A, the smaller the set of recommender nodes. This will have an effect on the measurement of a helper node's ability to mislead A.

How many recommendations against a dishonest D can V receive? That depends on the number of nodes within range of D and out of range of V, as illustrated in Figure 1.

The area of the shaded region in Figure 1 is

$$\pi r^2 - 2\,(\,(r^2)\arccos(d/(2r)) - (d/2)\sqrt{r^2 - (d/2)^2}\,).$$

If we express d as qr, where q is a number between 0 and 2, this becomes

$$r^2 \cdot (\,\pi - 2(\arccos(q/2) - q/2\sqrt{1 - (q/2)^2})\,).$$

If we look at the graph in Figure 2, we see that when r is fixed, and d varies between 0 and $2r$, the function is almost linear except when d is close to $2r$, when it flattens out. If the density of honest nodes is k per square with side-length r, then the number of honest recommenders is $k\pi - 2k(\,\arccos(q/2) - q/2\sqrt{1 - (q/2)^2}\,)$. Thus, dishonest nodes that are close to A have the advantage, because fewer recommenders are available to tell A that the dishonest node is lying.

Taking these area calculations into consideration, a key observation is that using recommenders diminishes the relative impact of the total number of

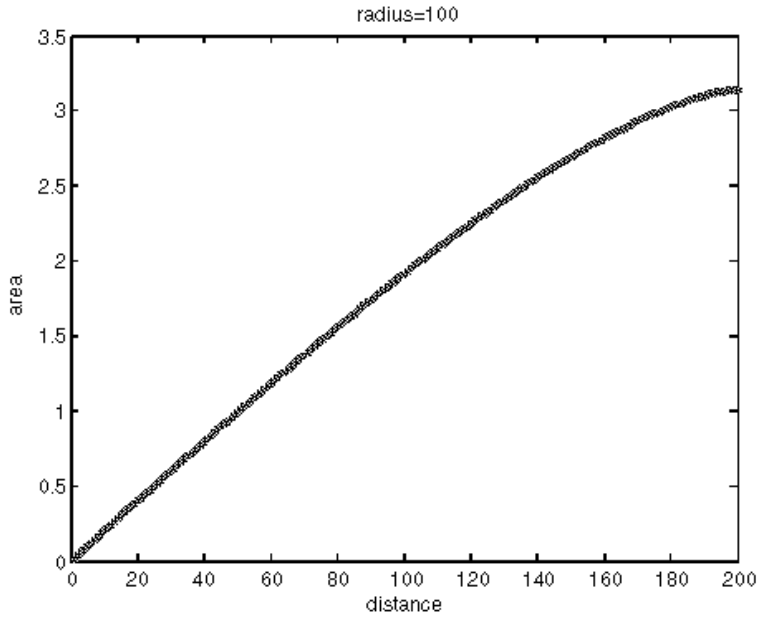

Figure 2: Recommender area vs. distance between honest and dishonest nodes

dishonest nodes in the network. By using recommenders it is no longer the case that their input must simply outweigh that of the honest nodes in range r of V; now their input must outweight that of the honest nodes within the recommender range of V, an area four times as large.

What strategy should be used in weighing input from the recommenders? Each recommender can be allowed one vote as in the local location calculation, but can they recommend about only one helper or can one recommender vote on each helper it can hear? The latter will allow dishonest recommenders to have an amplifying effect by casting votes against every honest helper node in range of V. In this case it is even effective for all but three of the dishonest helpers to remain silent and allow their identities to be used as recommenders. If they are thus powerful enough to successfully discredit a single honest helper node, they are also powerful enough to knock all of them down. It is possible for A to discount a node that claims to be adjacent to all of her helpers, but the effect is still strong.

Thus, it would seem prudent to limit the effect of dishonest recommenders to at most one helper node. There may be other more optimal strategies, but this will prevent dishonest recommenders from having any amplifying effect. However, this need not imply one recommender can only provide input on one helper. The recommender will not necessarily even be aware of V or which

are her helper nodes. He will only be saying to the world that a given node is lying about its location. V can simply weigh each vote concerning a node by the reciprocal of the number of distinct recommedations V has from the same recommender. This may allow for some diffusion attacks where honest recommenders do not have enough weight to discredit any single dishonest node, but dishonest nodes can coordinate and focus to knock down specific helpers of V. This in turn can be countered to some extent by weighting the votes in support of helpers independently from the weighting of votes against helpers.

As previously observed, helper nodes farther away from V will be in range of more recommender nodes, while helper nodes closer to V will be in range of fewer. A dishonest helper node can avoid some or all of the recommender nodes if it reduces its signal strength, but it can avoid them all by doing so only if its distance from V is less than $\frac{1}{2}r$.

3.2 Threat Analysis

3.2.1 Threat Models

The goal of the dishonest nodes is to convince the unseated node that it is somewhere other than where it actually is. In order to do this with a minimum of dishonest nodes, they will need to all vote for the same location. They can accomplish this easily if they perform some rigid motion, e.g. if they translate, rotate, or flip the plane that they lie in (assuming a 2D distribution of nodes).

We also consider three types of threat models:

1. Nodes can only give their turedistance from another node, although they can lie about their location;

2. Nodes can pretend to be farther away than they are, and can also lie about their location, and;

3. Nodes can say anything they like about their distance and their location.

3.2.2 Reflection Attack

We note that, even if nodes cannot lie about their distance from a node, they can still dislocate the node by performing a rigid motion of the plane. Moreover, they can even pick up the votes of two honest nodes, as illustrated in Figure 3.

The idea is that the dishonest helper nodes, D_i, advertise their locations as the reflection of the locations across the line between H_1 and H_2. This allows them to cast their votes for the other point at which the circle around H_1 and

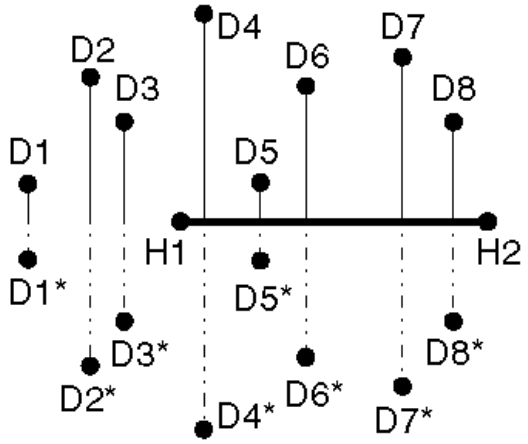

Figure 3: Dishonest nodes reflecting location about line between honest nodes

H_2 interesect. By picking up two honest votes this way, the dishonest nodes can assure that they can win as long as $p + 2 > h$, where h is the number of honest nodes, and p is the number of dishonest nodes.

Now, how do we make use of the nodes' ability to find out true distances? It does not give the honest helper nodes any advantage. The honest nodes (except for H_1 and H_2) can tell that the dishonest nodes are lying, but even if they report the lies to V, they will not provide her with any new information: she already knows that they disagree with each other. Similarly, if the dishonest nodes vote against the honest nodes, that will not provide any new information either, unless they vote against H_1 and H_2.

Unfortunately, the dishonest nodes have an advantage. Since they are willing to share keys with each other, a dishonest node can masquerade as any dishonest node anywhere in the network. In a worst case, it must mean that for the (local) honest nodes to win, h must be greater than $p' + 1$ where p' is the number of dishonest nodes anywhere.

However, we can make use of the recommender nodes. Such a node C may be able to tell that D's distance does not match its advertised location. It could then cast a vote against D that does provide new information to V. At this point, to keep things simple, we ignore the possibllity of recommenders casting votes for or against other recommenders and consider the case in which recommenders cast their votes against nodes contributing to V's initial location calculation. Recommender votes cannot be given directly to V, but they can be passed along to any helper node.

3.2.3 Impact of Distance Verification Assumptions

We have seen that dishonest nodes are able to combine to give misleading location information to unseated nodes no matter what the distance verification assumptions are. There are two other things that dishonest nodes are able to do in all three cases. First of all, if a node is a helper node, it can limit the number of recommender nodes that can hear it by reducing its signal strength. It can only avoid all recommender nodes if it is less than $\frac{1}{2}r$from V, however. Secondly, if a dishonest node is a recommender node, it can impersonate other dishonest nodes whose keying information it shares. Since recommender nodes do vote for or against other recommender nodes, this will not be detected.

The main impact of the different assumptions about the ability to detect cheating on distance is on the ability of helper nodes to impersonate other nodes. In the first case where any lie about distance can be detected, recommender nodes can detect dishonest helper nodes. Moreover, dishonest helper nodes are somewhat limited in the ways they can impersonate other dishonest nodes to the unseated node V. If a helper node D_1 attempts to impersonate D_2, it will have to pretend that D_2 is the same distance from V as D_1 is.

What happens if nodes can pretend to be further away but not closer than they are? In that case, all the dishonest nodes can share their keys with the one closest to V, call it D_0. D_0 can impersonate the other dishonest helper nodes pretending to be further away while using their keying material. In that case the vulnerability of V is thus closely related to its distance from the closest dishonest node.

Moreover, the recommender nodes now have only two ways of telling if D_0 is lying about its location. One is if D_0 is pretending to be closer to a recommender node than it is. The other is if D_0 pretends to be more than r away from a recommender node. D can avoid both of these if it positions itself and its cohorts away from the recommender nodes, but no more than r away. However, again in both cases, it can mitigate the effect of the constraints by reducing its signal strength so that at least some of the recommender nodes do not detect its signal, but V is still able to detect it. This is of course easier the closer D is to V. Again, D can avoid all the recommender nodes if its distance from V is less than $\frac{1}{2}r$.

Finally, we consider the case in which a node can pretend to be any distance from another node. In that case, the only constraint left is the constraint that the new locations be no more than r away from the recommender nodes, which can again be mitigated if D reduces its signal strength.

To sum up, not allowing nodes to appear closer than they are does not, for this particular application, seem to provide much more protection against node displacement attacks then allowing dishonest nodes to say anything that

they like about their distance. In the first case, the dishonest helper node must avoid placing itself and its cohorts too near the recommender nodes, while in the second case it doesn't have to worry about this. But even in the first case it can avoid some of the recommender nodes by reducing its signal strength, and all of them if its distance from V is less than $\frac{1}{2}r$.

4 Our Protocol

Our protocol uses the basic set-up described in the previous section. It has two stages. In the first stage, the unseated node, or prover, interacts with the helper nodes alone. In the second stage, the unseated node interacts with the recommender nodes, but only if it is unable to get a clear answer from the first stage, e.g. there is a tie between two locations. In the first stage, we use a protocol that guarantees only the weakest assumption for distance verification. In the second phase, we use something that comes very close to the second assumption, but only for recommenders.

4.1 First Stage

In the first stage, a verifier node V trying to locate itself uses an authenticated ranging protocol to find its distance from a prover node P, similar in construction to the distance bounding protocols described in the previous section. Our use of authenticated ranging means that we do not have to concern ourselves with the attacks of Clulow at al. on packet-based exchangees which allows us to have a packet-based as well as a bit-exchange version—subject to the assumption that an attacker is not able to modify packets in flight. It also allows us to simplify the protocol in other ways as well. In particular, the prover does not have make its rapid response depend on the verifier's challenge. It is enough that the authenticated response does; the authenticated response is taken as a statement by the prover that its rapid response was sent after the verifier's challenge.

The Stage One protocol comes in three versions: a packet-based version similar in form to Čapkun-Hubaux, a bit exchange version more similar in form to Brands and Chaum, and a bit exchange protocol similar to Hanke and Kuhn. We first describe the packet-based protocol. V begins by sending a packet to P in containing her name and a nonce N_V. It can contain other information as well, but this is all that is relevant to our protocol. P's rapid response is a packet containing its name and nonce N_P. This does not need to be an immediate response, but the delay should be small enough so that it does not interfere with measuring the distance from round-trip time when

accounted for. Note that P's response does not depend upon A's challenge in any way, since we are Finally, in the authenticated response, P sends its position, its delay in responding, a keyed hash taken over these and N_V and N_P, thus verifying that it is the one that responded to V's challenge.

1. $V \leftrightarrow P : V||N_V \leftrightarrow P||NP$
2. $P \rightarrow V : \text{Pos}||\text{Delay}||h(K_{VP}||\text{Pos}||\text{Delay}||N_V||N_P)$

A key assumption behind this version of the protocol is that an attacker cannot modify a message in real time and thus could not interfere with the first two messages without delaying the response so that the computation of any distance becomes impossible. This could be guaranteed, for example, by the techniques developed in [12]. If we do not wish to make this assumption, we can replace the identity sent in P's rapid response with a hash over P's identify and nonce.

Next, we consider the bit-by-bit version of the protocol.

1. $P \rightarrow V : h(P, N_P||M_P)$
2. $\sum V \leftrightarrow P : (N_V)_i \leftrightarrow (N_P)_i$
3. $P \rightarrow V : \text{Info}||N_P||M_P||h(K_{VP}, \text{Info}||N_V||N_P)$

In this protocol the prover first commits to a nonce and sends it to V. V and P then engage in a bit-by-bit exchange. The main difference between this and Brands-Chaum is that P's response depends only upon its own nonce, not V's and so can be readied in advance. This is because the protocol is designed for communication with an honest P, and so there is no need to prove (beyond P's word) that P is responding to V's nonce. P's final message opens the commitment, and also proves to V that P was responding to N_V with N_P. The Info field contains P's position, as well as P's delays in responding to V's bit-by-bit challenges, if delays are used.

We next give the Hancke-Kuhn version:

1. $V \rightarrow P : N_V$
2. $P \rightarrow V : N_P$
   $h(K_{VP}, N_V||N_P) = R_1^0||R_2^0 \cdots R_n^0||R_1^1||R_2^1 \cdots R_n^1||M$
3. $\sum V \leftrightarrow P : R_i^0 \leftrightarrow R_i^1$
4. $P \rightarrow V : \text{Info}||h(K_{VP}, \text{Info}||M)$

In this version of the protocol, the rapid bit exchange consists of the exchange of the first $2n$ bits of a shared secret. The remaining part of the shared secret is used to authenticate P's authenticated response, to tie its information (position and delays), with the rapid exchange.

4.2 Second Stage

The second stage is one in which the helper and recommender nodes verify the distance of the prover P. This stage may or may not be optional. We would expect it to be only engaged in if requested by V, e.g., when V is unable to determine a position by means of local information (i.e., from helpers only). On the other hand, as noted in Section 3, the adversary has a disproportionate advantage in being able to masquerade as all dishonest nodes throughout the sensor network. Using recommenders changes this advantage by up to a factor of four on average. Assume d' is the number of dishonest nodes anywhere in the sensor network and h is the number of local honest nodes. If the probability of node compromise is such that the risk that $d' > h + 2$ is too great, then it will be necessary to run the second stage even when a definitive location is reached based on local information.

The second stage requires the assumption that synchronization exists between honest provers and honest helpers and recommenders. This proceeds as follows:

The verifier V sends out a request for a location claim verification of the prover node P. This is forwarded to the appropriate recommender nodes, along with the approximate time this will occur, e.g. by one-hop flooding. V then engages in the Stage 1 protocol with P again, with one difference; P appends to each response the local time at which it was sent. Each helper and recommender that receives the signal computes its distance from P using the time sent by P. It then computes its distance from P's claimed location, and returns to V an authenticated message containing P's name and nonce, and the difference between the two distances. A difference under a certain threshold can be interpreted as a vote for P's location, and a distance above can be seen as a vote against.

We note that the stage two protocol satisfies the simultaneous distance modification property of Shmatikov and Wang. If P lies about its time, each node that receives its signal will calculate the same error in its distance. Thus, a set of honest helper nodes can only be fooled if they lie on a hyperbola with P's real and claimed location as its foci.

There are some other ways that P can cheat that we mentioned in Section 3. If P is close to V, it can reduce its signal strength so that the recommender nodes do not receive P's signal. V can counter this by giving a lower weight

to weaker signals, especially those that claim to be close to V.

Since the technique used in the stage 2 protocol has much in common with TDOA (indeed the recommender nodes could use TDOA to compute P's location if they so desired) it is potentially vulnerable to the same attacks as TDOA, that is, wormhole attacks and directional antenna attacks. We consider the wormhole attack first. In this attack P can also make use of a repeater C positioned at P's claimed position to repeat P's transmissions. The nodes close to C will report measurements consistent with P's claimed location. However, any other node that uses that repeater will have to claim the same location. Thus, if V queries multiple dishonest nodes, they will each have to make use of repeaters in different locations, and the dishonest nodes will need to communicate with the repeaters to tell them where to move (if a mobile repeater is used) or to tell the repeaters which one needs to be transmitting (if multiple stationary repeaters are used). The number of repeaters within range of the verifier now becomes a limiting factor, rather than the number of dishonest nodes anywhere in the network. And, if V is expected to execute stage 2 of the protocol only if it encounters multiple dishonest nodes, this should make the attack more difficult to carry off successfully.

The directional antenna attack is more serious. Since the positions of the recommender nodes are used by other nodes to compute their location, it is likely that they will be known by the attacker. Thus, a single node with multiple directional antennas could conceivably fake a large number of locations. However, we note that such an attack requires a greater investment on the part of the attacker, and such a directional antenna itself may be easier detect. We leave this as a question for further research.

5 Conclusion

We have given a framework for using distance bounding and range authentication for use in localization. We have used this, not only to compare the use of distance bounding and range authentication in localization, but to design a two-stage mechanism that takes advantage of the special features of localization.

For further work, it would be useful to integrate both the framework and the algorithms more closely conventional techniques for localization. Many range-based localization methods, for example, depend not only upon distance a node measures itself, but distances between other nodes that are reported to it. We can think of these reports, also, as recommendations. It would be useful to look at these from the point of view of security and see how our framework can be applied.

6 Acknowledgments

We would like to thank LiWu Chang for constructing the formulas and graphs used in Section 3.

References

[1] S. Brands and D. Chaum. Distance-bounding protocols. In *Advances in Cryptology - Eurocrypt '93*. LNCS 765, Springer-Verlag, 1995.

[2] Jolyon Clulow, Gerhard Hancke, Markus Kuhn, and Tyler Moore. So near and yet so far: Distance-bounding attacks in wireless networks. In *Proc. ESAS 2006*, volume 4357 of *LNCS*. Springer Verlag, 2006.

[3] Yvo Desmedt. Major security problems with the 'unforgeable' (Fiat-Shamir) proofs of identify and how to overcome them. In *Securicom 88*, pages 15–17, 1988.

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

[5] L. Lazas, S. Čapkun, and R. Poovendran. ROPE: Robust position estimation in wireless sensor networs. In *The Fourth International Conference on Information Processing in Sensor Networks (ISPN '05)*, April 2005.

[6] Michael Manzo and Tqnyq Roosta. Time synchronization attacks in sensor networks. In *Proc. SASN 2005*. ACM, 2005.

[7] R. Mayrhofer, H. Gellersen, and M. Hazas. Security by spatial reference: Using relative positioning to authenticate devices for spontaneous interaction. In *Proc. Ubicomp 2007: 9th International Conference on Ubiquitous Computing*, volume 4717 of *LNCS*, pages 199–216. Springer-Verlag, September 2007. *to appear*.

[8] Catherine Meadows, Radha Poovendran, Dusko Pavlovic, Paul Syverson, and LiWu Chang. Distance bounding protocols: Authentication logic and collusion attacks. In R. Poovendran, C. Wang, and S. Roy, editors, *Secure Localization and Time Synchronization in Wireless Ad Hoc and Sensor Networks*, pages 279–298. Springer Verlag, 2007.

[9] Catherine Meadows, Paul Syverson, and LiWu Chang. Towards more efficient distance bounding protocols. In *SecureComm 2006*, August 2006.

[10] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *ACM Workshop on Wireless Security (WiSe 2003)*, pages 48–61. ACM, September 19 2003.

[11] Kun Sun, Peng Ning, Cliff Wang, An Liu, and Yuzheng Zho. TinySeRSync: Secure and resilient time synchronization in wireless sensor networks. In *Proc. CCS 2006*. ACM, 2006.

[12] Mario Čagalj, Srdjan Čapkun, Ramkamur Rengaswamy, Ilias Tsigkogiannis, Mani Srivastava, and Jean-Pierre Hubaux. Integrity (i) codes: Message integrity protection and authentication over insecure channels. In *IEEE Symposium on Security and Privacy*, 2006.

[13] S. Čapkun and J. P. Hubaux. Secure positioning in wireless networks. *IEEE Journal on Selected Areas in Communication*, 24(2), February 2006.

[14] S. Čapkun, M. Čagalj, and M. Srivastava. Securing localization with hidden and mobile base stations. In *Proc. IEEE Infocom 2006*, 2006.

[15] Ming-Hsiu Wang and Vitaly Shmatikov. Secure verification of location claims using simultaneous distance modfication. In *Asian 07*, 2007.