Secure Verification of Location Claims*

Naveen Sastry Umesh Shankar David Wagner
UC Berkeley

Abstract

With the growing prevalence of sensor and wireless networks comes a new demand for location-based access control mechanisms. We introduce the concept of secure location verification, and we show how it can be used for location-based access control. Then, we present the Echo protocol, a simple method for secure location verification. The Echo protocol is extremely lightweight: it does not require time synchronization, cryptography, or very precise clocks. Hence, we believe that it is well suited for use in small, cheap, mobile devices.

1 Introduction

Computer scientists are used to studying access control mechanisms where one's identity determines what one is authorized to do. However, in the physical world, identity is not the only thing that matters: often, the requester's physical location also plays an important role in determining access rights. This suggests studying location-based access control.

Location-based access control in the physical world is easy, natural, and familiar. For example, being able to turn on or off the lights in a particular room using traditional technologies *requires* having a physical presence in the room. The very design of the light switch is what enforces the security policy. In contrast, achieving the same kind of guarantee with information systems, such as wireless networks, is less straightforward; it is not simply a matter of putting a switch in the right place. To enforce location-based access control policies on information resources, we need a way to perform *location verification*, where a principal's location is securely verified to meet certain criteria, e.g., being inside a particular room or a specific building.

Location verification enables location-based access control. After verifying a principal's location using a location verification protocol, the principal can be granted access to a particular resource according to the desired policy. This approach is naturally combined with physical security; guards or locks might be used to determine who is allowed to enter a building, then location verification employed to allow wireless access to all those inside. The location verification problem is the key technical challenge that must be surmounted to implement location-based access control.

Location-based access control has several benefits. Most importantly, it is natural for many applications. One simple policy might allow wireless control of only the lights for the room you are in, or might insist that a company server cease operating if it is taken outside the building. In addition, using location for access control obviates the need to establish shared secrets in advance. Visitors to a building need not obtain wireless encryption keys prior to their visit; instead,

*This work was supported in part by DARPA NEST contract F33615-01-C-1895, NSF CCR-0113941, and an equipment donation from Intel.

access could be granted automatically to all physical occupants of the building.

In this paper, we study the location verification problem. First, we introduce and define the location verification problem (Section 2). Then, we propose a new protocol for location verification, called *the Echo protocol* (Section 3), and we prove its security (Section 4). Additionally, we discuss the privacy implications of the Echo Protocol, observing that privacy is largely an orthogonal issue. This work provides a foundation for securely using location in wireless information systems.

2 Goals and Assumptions

2.1 Problem Statement

There are many natural variants of the secure location problem. We focus on solving the *in-region verification* problem: a set of *verifiers* V wish to verify whether a *prover* p is in a region R of interest. R may be a room, a building, a stadium, or other physical area. The region typically has some sort of physical control to restrict people's entry into it; the purpose, then, is to control access to resources that are not intrinsically constrained by physical security, such as wireless networks. The verifier infrastructure V may, in some cases, be a distributed system consisting of multiple nodes.

The protocol must run correctly in the face of adversaries. Thus, when p does not in fact have a physical presence inside R, the verifier must be careful not to accept p's claim to be in R. Furthermore, if p does have a presence in R, the verifier should accept p's claim; otherwise the protocol would not be useful in practice. We therefore require the following two properties to ensure that the protocol is useful and secure:

- *Completeness:* If p and V both behave according to the protocol, and p is in R, then V will accept that p is in R.
- *Security:* If V behaves according to the protocol and accepts p's claim, then p, or a party colluding with p, has a physical presence in R.

It is important to distinguish between the problem we are addressing, the in-region verification problem, and the *secure location determination problem*. In the latter problem, V attempts to securely discover the physical location of p. In contrast, in the in-region verification problem, p claims to be in a particular region, and V accepts or rejects the claim. The prover's location claim serves as a hint for the verifier to confirm or disprove. Framing the problem in terms of secure in-region verification, not secure location determination, simplifies the problem and allows different location determination algorithms to be used.

In fact, it is possible to compose an in-region verification protocol with any location determination algorithm, even a potentially insecure one, without compromising the security of the ultimate guarantee that a prover is in the region. The in-region verification algorithm verifies whether the claimed location is in R or not; thus, p can use an insecure localization algorithm to generate a claimed location that will be securely tested for accuracy by V. At worst, p's claim will be rejected; in no case will V believe something about p's location that has not been securely checked. The prover p thus has the flexibility to choose any appropriate location determination algorithm, even if it has not been proven secure. After running the determination algorithm, p will know which claims it can plausibly make.

2.2 Assumptions

It is worth considering in more detail what our particular protocol is and is not attempting to do:

- **Regions, not points.** We are not attempting to verify the exact location of the prover. In other words, the location claims we verify are not claims of particular *point* locations (plus or minus some error distance), but rather just presence in a particular region R of interest. This model accords well with our anticipated applications. We assume that before the verification protocol begins, both the prover and verifier know the definition of the region R.

- **Only "local" regions.** It is not a requirement to verify *all* location claims. More specifically, we only attempt to verify location claims for regions R that are "near" V. We will explore more precisely what this means in Sections 3 and 4. The restriction makes sense in light of the proposed application domain: controlling access to wireless resources once physical access to an area has been granted.

- **RF and sound capability.** The verifier and prover must each be able to communicate using both radio frequency (RF) and sound (typically ultrasound frequencies). We will use both transmission media in our protocol.

- **Bounded processing delay.** The prover must be able to bound its processing delay. We will describe the effects that a loose bound will have on the protocol in Section 4.

2.3 Threat Model

In order to verify the security property, we must consider the protocol with respect to a particular threat model. We assume the verifier nodes are all trusted, and they can communicate securely amongst themselves. In contrast, the prover p might behave maliciously, and we will consider an adversarial prover consisting of multiple colluding nodes, arbitrary computing power, and secure RF (speed of light) communication amongst its own nodes as well as sound generation and detection capability on each of its nodes. Each adversarial node can generate directional signals. Furthermore, the verifiers will not be able to detect the presence of an adversary by monitoring the RF communications since an adversary can easily use encryption or send its data on different RF frequencies.

Lastly, by definition, the adversary must not actually have any presence in the region R. Otherwise, it would be able to make a legitimate claim and would not need to attack the protocol.

2.4 Design Principles

We designed our protocol according to the following design principles:

- **Make few resource demands on the prover and verifier.** We would like to minimize the computational power and hardware resources necessary to participate in the protocol. The real goal is to enable location proofs for a large class of devices.

- **No prearranged setup required.** It should not be necessary for the prover to have previously engaged in a setup or registration step with the verifier. This excludes many cryptographic solutions; even public-key cryptography requires prearranged trust relationships, and thus is not suitable for our purposes. By eliminating the setup step, we are enabling access to resources to be granted based on physical presence alone.

  In settings where keys have been previously set up, we can use them to complement our protocol. In the full version of this paper[14], we discuss a variant of the Echo protocol where a challenge-response protocol can be used to verify that a particular principal is inside a given region.

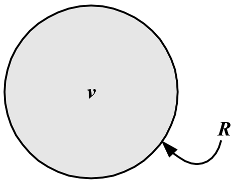

Figure 1: An illustration of our first simplification of the problem. The prover (not shown here) will try to convince the single verifier node v that it is inside the region R (depicted as a shadowed circle, which in this first scenario is assumed to be centered at v).

- **Quantitative guarantees.** We would like to provide precise bounds on the uncertainty in the protocol.

3 Our Design: The Echo Protocol

Next, we describe the design of our proposal for location verification, which we dub the Echo protocol. For expository purposes, we start by considering a simplified toy scenario and developing a simple protocol for this scenario (Section 3.1); then, we extend it repeatedly (Section 3.2) until we obtain the full protocol (Section 3.3).

Notation We define s to be the speed of sound, or 331 m/s. Likewise, we will take c to be the speed of light (which is the same as the speed of propagation of electromagnetic waves), or 3×10^8 m/s. Define $d(x, y)$ to be the distance between x and y. We define R to be the area in which we would like to verify the location of a prover p. The set of all verifier nodes is denoted by V. N denotes a nonce, i.e., an unpredictable random value.

1. $p \xrightarrow{\text{radio}} v : \ell$
2. $v \xrightarrow{\text{radio}} p : N$
3. $p \xrightarrow{\text{sound}} v : N$

v accepts iff $\ell \in R$ and
elapsed time $\leq d(v, \ell) \cdot (c^{-1} + s^{-1})$.

Figure 2: A protocol that solves our first simplification of the problem.

3.1 Protocol Intuition

Consider first a simplified case, where we have only a single verifier node v, where the region R is a circle[1], and where this circle is centered at v. This scenario is shown pictorially in Figure 1. Now, suppose that the prover claims to be at some location $\ell \in R$ inside the region.

We present a simple protocol for validating the location claim in this restricted case. First note that if the claimed location ℓ is not inside R, then the verifier can reject the claim immediately. Thus, we may safely assume that the prover claims to be inside R. Next, the verifier node v sends a packet containing a nonce to the prover using RF; the prover immediately echoes the packet back to the verifier using ultrasound. The verifier node v can then calculate how long it should take to hear the echo, namely, the sum of the time it takes to reach ℓ using RF, plus the time it takes for a return packet to go from ℓ to v using ultrasound. Thus, the total elapsed time for the prover to hear the echoed nonce should be about $d(v, \ell)/c + d(v, \ell)/s$ seconds. The only thing v has to do is time this process: If the elapsed time from the initial transmission to reception of the echo packet is more than this amount, the ver-

[1] In practice, the region is a sphere, instead of circle. However, for simplicity, our discussion will be phrased in terms of circles in the plane. This simplification makes the protocol easier to understand and does not affect the validity of our results.

ifier node v rejects the prover's claim; otherwise, if the elapsed time is at most this expected amount, v accepts. This protocol is summarized in Figure 2.

Why does this work? If the prover is able to return the packet within some maximum amount of time, then the verifier is assured that the prover is within $d(v, \ell)$ meters of v. This means that ℓ is known to be inside a circle of radius $d(v, \ell)$ centered at v. Call this circle C; then we know $\ell \in C$. Since R is defined to be a circle of radius at least $d(v, \ell)$ centered at v, we have $C \subseteq R$, and hence $\ell \in R$. In short, we know that the prover must be inside R.

If the prover cannot return the nonce in sufficient time, it may be for one of two reasons. Either the prover is more than $d(v, \ell)$ meters away from v, or the prover has some processing delay between receiving the RF packet and returning the ultrasound packet. We will explore this latter issue in the following section.

What if the prover tries to cheat by delaying his response? This attack only increases the total elapsed time of the process, thereby making the verifier reject. Intuitively, the longer it takes to complete the protocol, the farther away the prover appears to be. It is not in the prover's interest to appear to be farther from v, because this will put the prover's apparent location outside of R, hence making v reject the prover's claim.

Can the prover cheat by starting the transmission of the response early? No, this attack is not possible. The nonce in the packet prevents the prover from sending a reply before it has received the outgoing RF packet. Hence, the speed of light and sound prevents the prover from pretending to be closer to v than he really is.

3.2 Processing Delay & Nonuniform Regions

In this section, we present a slightly more advanced protocol that addresses three additional issues: the fact that the prover has a nonzero processing delay, the fact that packets take nonzero time to transmit, and the fact that R might not be a circle. We base this protocol on the simple protocol presented in the previous section. For a more complete treatment of these considerations, please see the full version of this paper [14].

Processing delay So far we have assumed that a prover can immediately echo back the nonce it was sent. In reality, of course, there is some finite processing delay. Let us start with the configuration mentioned in Section 3.1: We have a single verifier located at the center of a circular region R. Suppose the prover can bound its processing delay to be at most Δ_p seconds and can make the verifier node aware of this maximum delay. Then, if the prover claims to be at ℓ, the verifier node can compute the time for a prover actually at ℓ to get the packet back: the time for the RF signal to travel from v to ℓ, a processing delay of at most Δ_p, and finally the time for the sound to travel from ℓ back to v. This creates a problem when the prover is near the edge of R; the processing delay creates enough uncertainty that we cannot tell if the prover is inside or outside R. The solution is not to accept location claims such that the region of uncertainty lies outside R. Thus, we define the term *Region of Acceptance* (ROA) to be the area in which the verifier node v is sure that it can correctly verify claims for a prover. Note that this region depends on Δ_p. We write $\mathsf{ROA}(v, \Delta_p)$ to indicate the region where location claims are permitted by v, if the claimed processing delay is Δ_p. See Figure 3 for an illustration.

An alternate way to view $\mathsf{ROA}(v, \Delta_p)$ is that it is the region for which the protocol is complete.

What kind of processing delays do we expect to see? Our experiences with the Berkeley Mica 2 sensor network platform indicate that a millisecond delay is completely feasible and realistic [10]. Each device is a small, embedded device running an 8 megahertz, 8-bit Atmel 128 processor. Under most applications, the processor is idle most of the time to increase battery life, so system bus and processor contention is a minor contributor to delay. The operating system, TinyOS, is extremely small and simple, so the delay in the network stack are also minimized. We expect that most of the delay will arise in using a media access control protocol (MAC) to acquire the sending channel. By using an extremely simple MAC protocol that does not include any waiting (sending immediately only if the channel is free), we can reduce both the variance and magnitude of the MAC delay. Each millisecond of delay contributes $\approx$ 33 centimeters of uncertainty. If a more powerful node is used, say comparable to a wireless base station, the delay could likely be reduced by an order of magnitude.

Packet transmission time Another subtle point in considering the security of the Echo Protocol is that transmitting a packet is not instantaneous since a sender sends the first bit of the packet and some time later finishes sending the last bit of the packet. An adversary can leverage this time differential if it is near the edge of the ROA. Under certain circumstances, an adversary can be outside the ROA, yet probabilistically convince the prover that it is inside, a violation of the security condition. An adversary that anticipates the first k bits of the nonce will gain an advantage since it can overlap the sending of the outbound packet while still receiving the incoming packet. By using a nonce generated by a cryptographically strong pseudo-random number generator, the verifier limits the effectiveness of this attack. An adversary has a 2^{-k} chance in correctly guessing k bits, so it is not feasible to anticipate more than a few bytes. As detailed in the full version of this paper [14], one solution is to incorporate the packet transmission time into Δ_p by increasing it by the maximum packet transmission time; we call the packet transission time $\Delta_{\min}$.

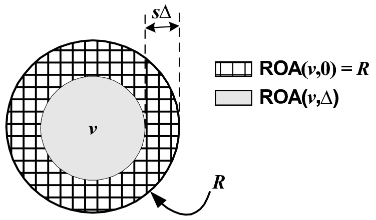

Figure 3: Diagram illustrating a single verifier at the center of a circular region R where there is an upper bound of Δ on the processing delay. The diagram illustrates the relationship between $\mathsf{ROA}(v, \Delta)$ and $\mathsf{ROA}(v, 0)$, the latter equal to R in this case.

Non-circular regions. Up until now, we have been assuming that R is a circle centered at v. However, that is not always a realistic assumption: perhaps we are interested in verifying location claims in a square room, for instance. We will now relax that assumption and assume that the verifier node is contained somewhere within an arbitrarily shaped region R. This causes a larger area to be *incomplete*, or non-verifiable, as shown in Figure 4. We will address incompleteness in the next section with our final iteration of the protocol.

Previously, $\mathsf{ROA}(v, 0)$ had been equivalent to R. But this will not work when R is not a circle centered at v. Since we are assuming that our communications equipment is omni-directional and that signals travel at the same speed in all directions, the ROA must be a circle. Furthermore, the ROA must be wholly contained within R. By definition, the ROA is the region where the verifier will accept a correctly functioning prover; if the ROA were not fully contained within R, the prover could accept a location claim for a prover outside of R, which would be unacceptable. Furthermore, we would like to maximize the area of the ROA since a larger ROA leads to a larger coverage. Thus, $\mathsf{ROA}(v, 0)$ should be the largest circle that fits within R; in other words, it should be the largest circle that is tangent to R and still contained within it.

We now extend the protocol to handle non-circular

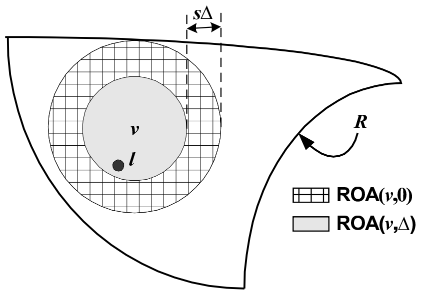

Figure 4: A single verifier v, inside a irregular region R. We are interested in proving that the prover is within R. The larger circle represents $\mathsf{ROA}(v, 0)$, the area in which v is useful for location verification proofs. This is the largest circle centered at v and wholly contained within R. The inner circle represents $\mathsf{ROA}(v, \Delta)$, the region in which v will accept location claims from a device that is able to bound its processing delay by Δ.

regions R where the verifier can bound its processing delay to be at most Δ_p. Recall that both the prover node and verifier node know R *a priori*. Using this, the verifier node can compute ahead of time the region $\mathsf{ROA}(v, 0)$.

The protocol then proceeds as follows: the prover first broadcasts its claimed location ℓ and processing delay Δ_p to the verifier. If $\ell \notin \mathsf{ROA}(v, \Delta_p)$, the verifier should immediately reject the location claim since it will not be able to definitively validate the claim. Otherwise, the verifier node broadcasts a nonce to the prover; the prover echoes the nonce back over ultrasound. The verifier can again time the communication: if it is no greater than the time for the signal to travel out and back and allowing for processing delay, the verifier accepts the location claim. Recall that Δ_p is an intrinsic property of the prover. So by sending Δ_p as the first step of the protocol, it can receive an early rejection if its delay is too large for its claimed location; thus, Δ_p is only useful for the prover. By lying about Δ_p, an adversary only affects early rejection and not the security of the protocol (see Section 4 for the complete proof of security). Intuitively, the total time that the verifier allows for a message to go out and come back is fixed and independent of Δ_p. So, when an adversary exaggerates Δ_p, it simply allows itself less time to respond. We expect that in practice if the prover were rejected early, the verifier would tell the prover $\mathsf{ROA}(v, \Delta_p)$ so the prover could move into a verifiable area.

3.3 Full Protocol Description: The Echo Protocol

In the final iteration of the protocol, we introduce multiple verifier nodes in an attempt to increase the coverage of R. Recall that if R is not a circle, no single node can provide 100% coverage. Consequently, multiple verifiers are needed. Intuitively, we will run the protocol presented in Section 3.2 after selecting one verifier from among the set of verifiers V.

The protocol is quite simple. See Figure 6 for the complete definition. First, a verifier is chosen so that the claimed location ℓ lies within that verifier's ROA. If no such verifier exists, execution is aborted, since the claim can not be verified. After choosing a verifier v to participate, v sends a packet to p using RF, which is echoed back to it using ultrasound. v can calculate how long it should take to hear the echo, namely, the sum of the time it takes to reach ℓ using RF, plus Δ_p, plus the time it takes for a return packet to go from ℓ to v using ultrasound. If the measured elapsed time exceeds this anticipated time, v rejects the location claim. The nonce in the packet prevents the prover from sending a reply before it has received the outgoing RF packet.

The extra verifier nodes serve to expand the region of acceptance within R. Thus, while $\mathsf{ROA}(v, \Delta_p)$ refers to the region that one particular verifier node can accept, we define $\mathsf{ROA}(\Delta_p)$ to be the region where at least one verifier node can prove location claims. It is

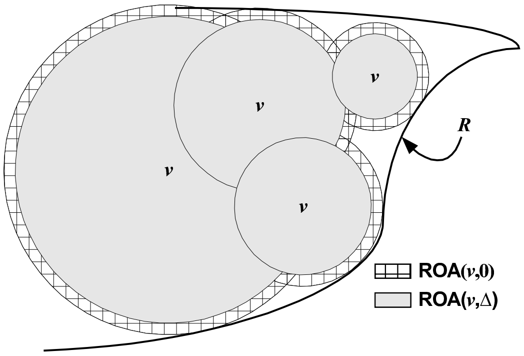

Figure 5: The relationship between $\mathsf{ROA}(v)$ (for a single verifier v) and the aggregate ROA. Each gray circle represents $\mathsf{ROA}(v,\Delta)$ for a particular verifier v. Taken collectively, the gray region represents $\mathsf{ROA}(\Delta)$, the aggregate region in which the set of verifiers can successfully verify the location of a prover that features a processing delay less than Δ. Note that $\mathsf{ROA}(\Delta)$ is wholly contained within R.

then clear that

$$\mathsf{ROA}(\Delta_p) \equiv \bigcup_{v \in V} \mathsf{ROA}(v, \Delta_p)$$

since the set of verifiers can accept a location proof if the claimed location is inside at least one verifier's region of acceptance.

In the Echo protocol, the infrastructure chooses a single verifier node to participate in the protocol. A verifier v may participate if $\ell \in \mathsf{ROA}(v, \Delta_p)$, since by definition that is the region for which it can perform secure location verification proofs. Note that the claimed location ℓ may be inside $\mathsf{ROA}(v, \Delta_p)$ for many different verifier nodes v, hence more than one verifier node might be eligible for participation in the protocol. We only require one to be chosen, and we allow the verifiers to use any convenient leader election mechanism for choosing which particular verifier node will run the protocol. They may have a purely deterministic mechanism for electing verifiers, or they may use a dynamic algorithm in an attempt to conserve power, for example.

COMMUNICATION PHASE:

1. $p \xrightarrow{\text{radio}} \text{broadcast} : (\ell, \Delta_p)$.
   The prover broadcasts its claimed location ℓ and processing delay Δ_p.
2. $v : t_s \leftarrow \textbf{time}\ ()$.
   $v \xrightarrow{\text{radio}} p : N$.
   A single verifier v starts its timer and responds with a random nonce.
   We require $\ell \in \mathsf{ROA}(v, \Delta_p)$ and $\Delta_p \geq \Delta_{\min}$.
   If no such verifier exists or Δ_p is invalid, **abort**.
3. $p \xrightarrow{\text{sound}} v : N$.
   $v : t_f \leftarrow \textbf{time}\ ()$.
   The prover echoes the nonce over ultrasound.
   The verifier records the finish time.

VERIFIER COMPUTATION PHASE:

4. **if** sent nonce differs from received nonce
   return false
5. **if** $t_f - t_s > \frac{d(v,\ell)}{c} + \frac{d(v,\ell)}{s} + \Delta_p$
   return false
6. Otherwise, **return true**

Figure 6: Formal description of the Echo protocol, which can perform location verification in an arbitrary region R with multiple verifier nodes. We represent the prover node as p and the verifier node that runs the protocol as v. In step 2, $\Delta_{\min}$ represents the lower bound on the delay, which is incurred by transmitting the packet.

4 Security Analysis

As explained in Section 3, the Echo protocol relies on timing: the amount of time it takes to get a response from the prover bounds how far the prover can be from the verifier. We will now show that it is impossible for an adversary outside R to convince the verifier that it is in R.

Proof of security The heart of the argument is that an attacker would not be able to get the sound signal to the verifier in time. In order to confirm that the prover is at ℓ, all a particular verifier node v must do is verify that the incoming sound signal, which includes the outgoing nonce, is received within

$$t_{\text{max}} \leq \frac{d(v,\ell)}{c} + \frac{d(v,\ell)}{s} + \Delta_p \quad \text{seconds},$$

where $d(v,\ell)$ is the distance from the verifier to the claimed location, c is the speed of radio propagation (the speed of light, which may vary depending on the medium through which it passes), s is the speed of sound, and Δ_p is the prover's processing delay. As described in Section 3.2, Δ_p includes the packet transmission time. This is checked by the verifier in step two of the communication phase of the protocol. Recall that v agrees to run the protocol only if $\ell \in \mathsf{ROA}(v,\Delta_p)$, i.e., if the circle of radius $d(v,\ell)+\Delta_p \cdot s$ lies wholly within R. As long as Δ_p is positive (guaranteed by step two), we know that $\mathsf{ROA}(v,\Delta_p) \subset R$. As we saw in Section 3.2, Δ_p is used an an optimization and potentially a hint to the prover, so even if an adversary lies about Δ_p, security is assured.

By definition, the attacker A is outside R; thus we have

$$d(v,A) > d(v,\ell) + \Delta_p \cdot s.$$

Let Δt^{A} denote the elapsed time $(t_f - t_i)$ when the attacker finishes sending its response (message 3 of the Echo protocol). The attacker has only two choices: either guess at least some of the bits of N, or learn the entire nonce N from v. In the former case, the attacker's success probability can be made negligibly small by choosing N from a set of sufficient size. In the latter case, it will take at least $d(v,A)/c$ seconds after v first reveals N before A can receive N, because no signal can travel faster than the speed of light. Because v reveals N for the first time in message 2 of the protocol, $\Delta t^{\text{A}} \geq d(v,A)/c$ in this case. Now, since the attacker cannot finish transmitting its response before it has received the entire nonce, and because the attacker's response cannot travel faster than the speed of sound, the minimum time required for the attacker to hear N and get a response to v is

$$\begin{aligned}
\Delta t_{\text{min}} &= \Delta t^{\text{A}} + \frac{d(v,A)}{s} \\
&\geq \frac{d(v,A)}{c} + \frac{d(v,A)}{s} \\
&> \frac{d(v,\ell)+\Delta_p \cdot s}{c} + \frac{d(v,\ell)+\Delta_p \cdot s}{s} \\
&\geq \frac{d(v,\ell)}{c} + \frac{d(v,\ell)}{s} + \frac{\Delta_p \cdot s}{c} + \frac{\Delta_p \cdot s}{s} \\
&\geq \frac{d(v,\ell)}{c} + \frac{d(v,\ell)}{s} + \Delta_p.
\end{aligned}$$

Consequently, the attacker's signal cannot reach the verifier before the deadline. Note that nowhere in our analysis did we rely on *which* verifier node was used. The only difference would be in the magnitude of the error terms and, therefore, in the chance that the location claim would even be accepted for verification. The attacker does not gain any advantage by selecting a different verifier from the one selected to participate.

Attacks One possible attack could exploit the difference in propagation speed of sound in different media. For example, the speed of sound in steel is 5032 m/s, nearly 15 times faster than in air; other materials exhibit similarly higher sound transmission speeds than air. If the verifier's estimation of s is slower than the actual one, then the proof above does not apply. If this is a valid threat model—say there is a lot of metal near the verification region that is capable of transmitting sound from the outside—then the verifier's estimation of s should be adjusted. This can be done once on a site-specific basis. An alternate defense would be to have other verifier nodes confirm the estimate of s based on when the sound signals are received.

More generally, we require that there be no way for an attacker to generate sound waves from afar without being subject to speed-of-sound delays. For instance, a remote attacker could call up some person in R over the telephone and convince the victim to put the call on speakerphone, then run the protocol. If the ultrasound reply can go over the telephone with sufficiently high fidelity, then the attacker might be able to spoof his location. The key is that the attacker has evaded the speed-of-sound limit on signal propagation by exploiting the ability to remotely actuate a loudspeaker located inside R. We expect such "remote actuation" attacks will be very difficult to mount in practice; in our example, band-limited phones would block ultrasound.

Variants We Rejected One might also consider the implications of other variants of the protocol, where the use of sound and radio for the outgoing and incoming signals is changed from (radio, sound) to (radio, radio), (sound, radio), or (sound, sound). If radio communication is used in both directions, then the error term $\Delta \cdot c$ would be very large (10^5 to 10^6 times as large as the sound case), and it is quite likely that the verifier would not accept location claims at all, since the error might exceed the size of R itself! Thus, at least one of the two directions should use sound.

Why did we reject (sound, radio)? There is a subtle attack. If sound is used in the outgoing direction, an attacker might be able to break security by using laser-based remote "bugging." The trick is to bounce a laser off a window within R and analyze the return signal to detect the vibration of the window, which would allow a sophisticated attacker outside R to "bug" a room within R from miles away without being subject to speed-of-sound delays on the propagation of the sonic signal. Thus, "remote bugging" attacks effectively speed up the transmission speed of the sound wave and thereby invalidate our security proof above. We thus reject (sound, radio) and (sound, sound) since both rely on transmitting sound in the outgoing direction.

The (radio, sound) protocol is more secure against such attacks, because "remote actuation" seems significantly more difficult than "remote bugging," and the security of the (radio, sound) protocol rests only on the difficulty of "remote actuation" and not on the hardness of "remote bugging." For this reason, the Echo protocol uses radio in the outgoing direction and reserves ultrasound for the return signal from the prover.

Privacy Implications We now look at privacy concerns related to the Echo Protocol. Clearly, the prover reveals information to the verifier – it's trying to convince the verifier that it is in the ROA, after all. The prover does have some control over what the verifier learns, however. A prover situated very close to the verifier can wait some time before replying with its nonce. This increases the verifier's uncertainty about the prover's location. A prover can employ this technique to ensure that the verifier only learns that the verifier is inside the ROA centered around it.

But what about an outside observer? A shareholder snooping at a company's headquarters might expect a press release if ten vice presidents each authenticate their location to the corporate boardroom. An adversary watching the interactions between the prover (vice president) and the room's infrastructure could infer the prover's location. In fact, since in most cases the verifier is non-malicious, an adversary could infer a prover's location just by knowing which verifier the prover interacts with. However, we note that it is easy enough to obtain the prover's location via other means: for example, an adversary can passively triangulate a node's ordinary wireless communications and determine its location independent of any location protocol [17].

If an adversary can find a prover passively using triangulation, why can't the verifier use triangulation? An adversary can inexpensively break the security of a system that uses triangulation. By appropriately sending different signal strengths to each verifier using directional antennas, an adversary can foil a triangulation system to create a ghost image at any location. This highlights a difference in the goals of the adversary and verifier: an adversary still "wins" if it can successfully triangulate only 1% of the time. However, the verifier must only accept claims in accordance with the security condition, so it cannot use an unreliable approach.

5 Related Work

Other Approaches A number of authors have proposed using time-of-flight measurements and the speed of light to securely gain location information about untrusted parties. Brands and Chaum proposed a time-bounded challenge-response protocol [4] as a defense against man-in-the-middle attacks on cryptographic identification schemes. Hu, et al., proposed using temporal packet leashes for wireless networks to defend against similar attacks [11]. However, a major limitation of these schemes is that both the prover and verifier send RF signals, requiring a much more accurate timing system at the verifier as well as tight real-time processing guarantees on both the prover and verifier for accurate readings. For these reasons, we believe our algorithm is better suited to mobile devices than those previous proposals.

In independent and concurrent work, Waters and Felten present a scheme that uses round-trip time-of-flight of RF signals to achieve goals similar to ours [21]. Their architecture is similar to ours, in that they, too, suggest focusing on secure location verification rather than on secure location determination. However, their reliance on RF seems likely to limit deployment, like the previous proposals mentioned above. Additionally, by using tamper-resistant trusted devices, they are able to defend against stronger adversaries. If their verifier accepts, they can successfully show that the trusted device is at the specified location. In comparison, we can show that the device or a collaborator has a presence at the specified location.

Vora and Nesterenko use a novel technique for the secure location verification problem that doesn't rely on time of flight[19]. The intuition behind their idea is simple: nodes nearby the prover's claimed location should hear a prover's broadcast, while those further away should not hear it. They make use of "rejector nodes" to monitor radio signals from a malicious verifier node that is outside the ROA. When a verifier claiming to be inside the ROA broadcasts and a rejector node hears the signal, the system does not accept the verifier's claim. Their scheme is, however, vulnerable to adversaries with directional antennas and requires very careful node placement to handle non-trivial radio falloff models.

Location & Localization The idea of using time-of-flight to estimate distance is not a new one: it dates back to the birth of radar systems, which often use time-difference-of-arrival (TDOA) to determine the range to detected objects. Ultrasonic time-of-flight ranging can even be found in nature, where it is used by bats.

Coarse-grained location authentication has been used in the television industry to prevent cloning of set-top boxes [8]. Gabber and Wool propose four coarse-grained techniques, relying on extensive telecommunications infrastructure such as satellites, paging and cellular networks. Their techniques rely on tamper-resistant hardware.

Location-limited channels provide a communication mechanism that is restricted to a short range and provides both endpoints a mechanism to guarantee the authenticity of each participant [16]. Balfanz, et al., have proposed using location-limited channels for location-based access control [3], and many others have also proposed use of limited-range radio broadcasts as a way to verify proximity [5, 6, 12]. However, there are no strong security guarantees that the communication range will always be limited as desired: an adversary with more powerful equipment may be able to participate in the protocols even if they are substantially further away than non-malicious parties.

Finally, there are many techniques to help localize devices [1, 2, 15, 13, 9, 20], GPS being one of the most widely deployed. However, none of those works addresses security, and in fact GPS signals can be spoofed [18, §3.2.2]. Nonetheless, we have noted that combining a (possibly insecure) localization mechanism with our secure location verification technique yields a secure localization algorithm. Thus, insecure localization protocols should be seen as complementary to our work on secure location verification.

Many authors have commented on the value of location-based access control [5, 6, 7, 12, 3].

6 Conclusion

We introduced the in-region verification problem. Then, we designed a provably secure, lightweight protocol to address it, named the Echo protocol. The Echo protocol does not require cryptography, time synchronization, or any prior agreement between the prover and verifier, making it suitable for low-cost devices such as those in sensor networks. It is robust against a malicious adversary with unbounded computing power; the security rests on physical properties of sound and RF signal propagation. We expect the Echo protocol to be a useful contribution in contexts where physical presence is used for access control.

References

[1] GPS Documentation. `https://www.peterson.af.mil/GPS_Support/gps_documentation.htm`.

[2] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *INFOCOM (2)*, pages 775–784, 2000.

[3] Dirk Balfanz, D.K. Smetters, Paul Stewart, and H. Chi Wong. Talking to Strangers: Authentication in Ad-Hoc Wireless Networks. In *Network and Distributed System Security Symposium Conference Proceedings*, 2002.

[4] Stefan Brands and David Chaum. Distance-Bounding Protocols. In *EUROCRYPT '93*, volume 765 of *LNCS*.

[5] Deborah Caswell and Philippe Debaty. Creating Web Representations for Places. In *2nd International Symposium on Handheld and Ubiquitous Computing*, pages 114–126, 2000.

[6] Mark D. Corner and Brian D. Noble. Zero-Interaction Authentication. In *MOBICOM '02*. ACM Press, 2002.

[7] Dorothy E. Denning and Peter F. MacDoran. Location-Based Authentication: Grounding Cyberspace for Better Security. In *Computer Fraud & Security*. Elsevier Science Ltd., February 1996.

[8] Eran Gabber and Avishai Wool. How to Prove Where You Are: Tracking the Location of Customer Equipment. In *Proceedings of the 5th ACM conference on Computer and Communications Security*, pages 142–149, 1998.

[9] Lewis Girod, Vladimir Bychkovskiy, Jeremy Elson, and Deborah Estrin. Locating Tiny Sensors in Time and Space: A Case Study. In *ICCD*, 2002.

[10] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for network sensors. In *ASPLOS*, 2002.

[11] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks. In *INFOCOM*, 2003.

[12] Tim Kindberg, Kan Zhang, and Narendar Shankar. Context Authentication Using Constrained Channels. In *Fourth IEEE Workshop on Mobile Computing Systems and Applications*, 2002.

[13] A.M. Ladd, K.E. Bekris, G. Marceau, A. Rudys, D.S. Wallach, and L.E. Kavraki. Robotics-Based Location Sensing for Wireless Ethernet. In *Eigth Annual International Conference on Mobile Computing and Networks (MobiCOM 2002)*, 2002.

[14] Naveen Sastry and Umesh Shankar and David Wagner. Secure verification of location claims. In *ACM Workshop on Wireless Security (WISE-2003)*, 2003.

[15] Nissanka B. Priyantha, Allen K. L. Miu, Hari Balakrishnan, and Seth J. Teller. The cricket compass for context-aware mobile applications. In *Mobile Computing and Networking*, pages 1–14, 2001.

[16] Frank Stajano and Ross Anderson. The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks. In *7th Security Protocols Workshop*, volume 1796 of *Lecture Notes in Computer Science*, pages 172–92, 1999.

[17] Ping Tao, Algis Rudys, Andrew Ladd, and Dan Wallach. Wireless LAN location sensing for security application. In *ACM Workshop on Wireless Security (WISE-2003)*, 2003.

[18] John A. Volpe. Vulnerability Assessment of the Transportation Infrastructure Relying on the Global Positioning System, August 2001.

[19] A. Vora and M. Nesterenko. Secure location verification using radio broadcast. `http://www.cs.kent.edu/~mikhail/Research/location.ps`, submitted to 4th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks.

[20] A. Ward, A. Jones, and A. Hopper. A New Location Technique for the Active Office. *IEEE Personal Communications*, 4(5):42–47, October 1997.

[21] Brent Waters and Ed Felten. Proving the Location of Tamper Resistent Devices. `http://www.cs.princeton.edu/~bwaters/research/location_proving.ps`.