Secure and Precise Location Verification Using Distance Bounding and Simultaneous Multilateration

Jerry T. Chiang Jason J. Haas Yih-Chun Hu
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
1406 W. Green Street, Urbana, IL 61801-2918
{chiang2, jjhaas2, yihchun} @illinois.edu *

ABSTRACT

Due to the widespread adoption of the Global Positioning System (GPS), many systems have been designed to use the location information of participants. When these systems confer rights (such as access rights) based on location, such claim must be securely verified in order to prevent attackers from gaining access to resources that should be restricted. Substantial effort has been made on secure location verification; however, previous work does not address the impact of collusion attacks where adversaries share their private keys nor do they address a possible jamming attack where attackers inject a high amount of noise to prevent successful challenge and response receptions. In this paper, we propose a *secure multilateration scheme* that provides maximal security achievable by any time-of-flight based system that does not employ other verification methods.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General-Security and protections, (e.g., firewalls)

General Terms

Security, Verification

Keywords

location verification, distance bounding, multilateration

1. INTRODUCTION

In mobile and wireless networks, user location can be used for many purposes, including routing, access control, and navigation. For example, location information of a user has been incorporated in many geographic based routing schemes in sensor networks [7, 19, 5] where power conservation is crucial. However, if an attacker can successfully falsify his location claim, the attacker may receive better service, or gain access to a network or resource that it could not otherwise obtain. Therefore, all location claims should be securely verified before being used by a system.

Several approaches to secure location verification attempt to ensure that the user is within some circular region. These techniques generally assume a verifier is at the center of that region, and use physical limitations to prevent attackers outside of that region from successfully verifying the location. These limitations vary from protocol to protocol. Brands and Chaum [1] first designed a distance bounding protocol that exploits the fact that attackers cannot send signals faster than the speed of light. In Sastry et al.'s scheme Echo [10], users send an ultrasonic pulse back to the verifier to prove that they (or an accomplice) are within some distance of the verifier. Vora et al. exploited the race condition between a verifier at the center of the region and rejectors forming a circle substantially away from the region [18]. We build our scheme on top of Brands and Chaum's distance bounding protocol, but extend it to verify *a specific location* within the convex hull formed by the verifiers.

Throughout the paper, we will refer to the entity that claims and wishes to prove its location as the *prover*, and the entity that verifies such location claim the *verifier*. To determine that a prover P is within distance r from verifier V, the verifier simply measures the round trip time. In particular, if the verifier sends a challenge to the prover, and the prover can correctly respond within time duration $2r/c$, then the prover (or an accomplice) must be within a circle centered at V with radius r.

Our extension is based on the concept of *multilateration*: if a prover P proves to verifiers $V_1, \ldots, V_n$ that he is within radius r_i from V_i, then given there is only one prover, he must be within the *intersection* of all n circles.

Several protocols have been built using distance bounding and multilateration. For example, Čapkun and Hubaux proposed a multilateration scheme that specifies a δ test and a *point-in-triangle* test [14]. Naïve multilateration schemes, however, can be easily compromised. For example, if the verifiers perform distance bounding independently at different times, then an attacker can move between the tests and falsely prove his location.

Sastry et al. thus suggested the insight that secure multilateration scheme must use *simultaneous* verification [10]. In this paper, we propose a simultaneous and intertwined verification algorithm, and prove that it achieves the maximal security achievable by any protocols based solely on time-of-flight information. Sastry et al. made the observation that

*This material is based upon work partially supported by USARO under Contract Number W-911-NF-0710287.

WiSec'09, March 16–19, 2009, Zurich, Switzerland.

if at least one colluding attacker is within claimed distance from each of the verifiers, simultaneous multilateration may not be secure. While it is true that our system cannot defend against all collusion attacks, we will prove several properties regarding the feasibility of collusion attacks against our system.

Moreover, we propose a novel self jamming scheme that requires the attackers to have substantially more resources than the verifying system in order to perform attacks. Other techniques such as the use of directional antennas are orthogonal to our scheme and can be integrated to provide further security.

The rest of this paper is organized as follows. Section 2 gives background on the problem and prior work, including a discussion of applications, commonly considered attacks, related work, attacker model, and our system assumptions. We describe our protocol in detail and prove its optimality in Section 3. The threat analysis is given in Section 4. We then propose a novel self-jamming scheme in Section 5. Finally, we state our conclusions in Section 6.

2. BACKGROUND

In this section, we motivate the problem of secure location verification, discuss three common attacks that can be performed by an attacker, overview the related work, and discuss our attacker model and system assumptions.

2.1 Motivation and Applications

Due to the widespread adoption of the Global Positioning System (GPS), many routing schemes have been designed to use the location information of participants [7, 19, 5]. Geographical routing is attractive especially to sensor networks where power consumption should be minimized. However, since GPS signals can be easily forged, a location claim should be securely verified first.

A securely verified location claim can also be used for physical access control. Suppose we have a wireless access point in a building that handles confidential data. By verifying the location of users before granting access, we can make sure such access point is used only by hosts in the building. A very similar system can be built for access control of the resources for ubiquitous computing, where priority and access are granted according to a user's location. In these cases, an attacker who can falsify his location information can acquire confidential data or gain unfair access advantage over honest users.

Location verification can also be used against relay attacks in many wireless systems. Cryptographically authenticated RFID tokens are widely deployed today for building access-control. If an attacker is equipped with a proxy reader, the attacker may be able to proxy the signal between the RFID reader and the tag, thereby obtaining building access [3]. Such attacks are hard to prevent using cryptographic protocols but can be easily detected and prevented using basic distance bounding [4].

2.2 Common Attacks

Wormhole Attack.

In a wormhole attack, the attacker attempts to intercept the signal and pass the signal through a different route to the destination. In a distance bounding protocol, if the alternate route takes longer time, the perceived distance between the prover and verifier is enlarged. This increase in perceived distance may cause the claim to be rejected. However, if the alternate route takes less time, then the perceived distance is reduced. Moreover, if the attacker then delays the signal by the correct amount of time before passing it to the destination, the perceived distance can be artificially enlarged to the correct claimed distance.

Wireless techniques that use EM waves convey information at speed of light. Thus, any wireless technology that uses non-EM wave is susceptible to the wormhole attack where an attacker converts between types of waves and uses a different wave with faster speed, such as EM wave. Audio waves such as ultrasound are susceptible to wormhole attacks also because the velocity of the wave differs greatly depending on the medium, and especially since the velocity of sound in air is slower than in denser media.

Recent advancements in quantum physics in the area of quantum entanglement seem to provide a method for faster-than-light communication. Labeled "spooky action at a distance" by Einstein, the qubits of a pair of entangled particles are shown to be correlated. However, Peres and Terno has recently shown that such entanglement cannot be used to instantaneously convey information in their *no-communication theorem* [8].

Since there are no known methods for communicating faster than speed of light, we design our protocol using EM waves in order to prevent wormhole attacks.

Jamming Attack.

Wireless communication is also susceptible to the jamming attack where a power-limited attacker injects into the communication channel high amount of noise to reduce the probability of successful message reception. The attacker is assumed to be power-limited since he must still abide the laws of physics. *Spread spectrum* techniques have been proposed to combat the jamming attack [17]. A common spread spectrum system is the *frequency-hopping code division multiple access* (FH-CDMA). In an FH-CDMA system, a wide frequency band is divided into many channels, and time is divided into time slots. Each user then chooses a channel from time slot to time slot according to a *hopping pattern*. Two users do not interfere with each other as long as they occupy different frequency channels in any given time slot. An attacker wishing to disrupt a particular user's service, without knowing the user's hopping pattern, must randomly choose a subset of channels on which to emit power. If the attacker only chooses a few channels on which to jam, it is likely that none of these channels are selected by the normal user, and the attacker consequently might not interfere with the user at all. If the attacker chooses to jam many frequency channels in the FH-CDMA system, since the attacker is power-limited, the interference in any frequency channel, in particular the channels used by the normal user, may be insignificant.

Collusion Attack.

In a collusion attack, multiple provers attack the system in a cooperative manner. Prior work considers colluding attackers that keep individual private keys secret. The system then differentiates each prover using such keys to prevent replay attacks and false identities. This limited collusion is referred to as the "terrorist attack."

2.3 Related Work

Location verification schemes can generally be divided into range-based and non-range-based categories. A range-based system uses physical information such as time-of-flight or signal strength while a non-range-based system uses only network topology to determine a user's location. A range-based system can generally achieve higher precision while a non-range-based system offers simplicity in design.

Brands and Chaum proposed the distance bounding protocol [1] where the verifier rapidly exchanges bits with a prover using radio wave. In the basic distance bounding protocol, the verifier and the prover first agree upon a bit stream, B. The verifier then generates a challenge stream, N, that is of the same length as B. For the i^{th} round of the protocol, the verifier sends bit N_i and starts timing. The prover, upon receiving N_i, responds with bit $N_i \oplus B_i$. If the prover can correctly respond within time t_i, the prover must reside within a sphere with radius $\frac{1}{2}c \max_i (t_i)$ around the verifier where c is the speed of light in vacuum. The scheme, however, can only be used to verify a prover is within a certain range from the verifier.

Sastry et al. later proposed the Echo protocol [10], a modification of the distance bounding protocol that uses ultrasound. Echo algorithm uses RF link for verifier-to-prover challenges and uses ultrasonic link for prover-to-verifier responses. Thus if the prover responds within time t_i, the prover is concluded to be at most ℓ away from the verifier where s is the speed of sound and ℓ satisfies

$$\max_i (t_i) = \frac{\ell}{c} + \frac{\ell}{s}$$

Bussard and Bagga modified the original distance bounding protocol to incorporate a user's secret key, i.e. her cryptographic identity, into the response such that given the response and challenge, it is easy to recover the secret key [2]. Their scheme prevents the terrorist fraud attack since an attacker that wishes to conceal his cryptographic identity cannot allow his colluders to be able to receive and decode challenges meant for himself, while providing them the ability to respond with correct answers.

Multilateration can be used to precisely verify a location claim. The original distance bounding protocol provided cryptographic methods to verify that the bit exchange was done by a particular user. However, we cannot use these methods to perform secure multilateration since colluding attackers that share cryptographic identities, such as attackers who share private keys among themselves, appear as one entity in the cryptographic sense.

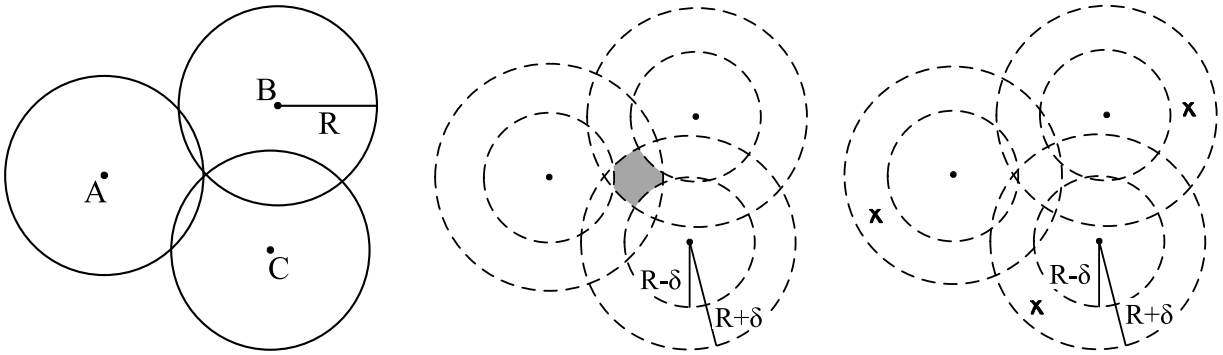
Čapkun and Hubaux proposed two tests for secure multilateration [14]. When the claimed distance between the prover and a verifier is r, and the perceived distance between the prover and a verifier is $\frac{1}{2}ct_i$ for the i^{th} round of the distance bounding protocol, the δ test limits the user's location ambiguity by rejecting the claim if $\left|r - \frac{1}{2}c \max_i (t_i)\right| \geq \delta$, for some predefined δ. The *point-in-triangle* test simply makes sure the prover is located within the triangle formed by three verifiers. While these tests are proven to be sufficient to prevent false location claims from a single attacker, they cannot successfully defend against colluding attackers.

For example, in Figure 1(a), the prover claims to be located at a distance R away from all three verifiers. However, due to processing delay and error in the location estimation, the prover may appear to be slightly farther away from each of the verifiers. The verifiers thus allow a time uncertainty up to δ/c, or a spatial uncertainty of δ. In Figure 1(b), the dashed lines denote $R \pm \delta$ region from each verifier. If the prover has no processing delay, he must reside in the intersection of the three $R \pm \delta$ regions in order for his location claim to be accepted. In figure 1(c), three colluding attackers who share their cryptographic identities can each pass the δ test with one verifier. The attackers, after passing the δ tests, will also pass the point-in-triangle test since the claimed location is indeed in the interior of the triangle formed by the three verifiers. The verifiers are then tricked into accepting the false location claim of the colluding attackers. Sastry et al. thus observed that multilateration must be done in a manner such that all verifications are performed simultaneously [10].

Čapkun and Hubaux subsequently suggested two schemes against collusion attacks [15]. In the first scheme, the verifiers fingerprint all users; thereby making sure that different users cannot act as one single entity. In the second scheme, the users are all given tamper-proof hardware; thus the attackers cannot clone a node and subsequently collude. The first approach requires additional assumption on the system tolerance on the variability of fingerprints and also the attackers' sophistication in tuning his own fingerprints. The second approach, on the other hand, requires all users to have tamper-proof hardware and can be very expensive for a system with large number of users. This approach was independently proposed by Singelee and Preneel [11] to defend against colluding terrorist attackers. These two approaches are orthogonal to our approach and can nonetheless be incorporated to provided additional security features.

Vora and Nesterenko proposed a novel scheme that takes advantage of the broadcast nature of wireless communication [18]. Verifiers in this protocol can act in one of two ways: as a normal verifier that grants acceptance to a prover's claim, or as a rejector that rejects a prover's location claim. A prover proves his location by broadcasting a signal, and the time-difference-of-arrival is used between the verifier and the rejectors to determine whether the prover's claimed location is accepted. Figure 2 illustrates this protocol. The center verifier is the acceptor, the shaded circle is the acceptance zone, and the rejectors lie in a ring outside the acceptance zone. However, to remain secure against an adversary using a directional antenna, Vora's scheme requires an excessive number of verifiers. When an attacker is at distance k away from the ring of rejectors, and the attacker's directional antenna can achieve directionality of $2\pi/\beta$, the rejectors must be at most $2k \tan(\beta/2)$ apart in order to reject this attacker.

Čapkun et al. recently proposed a location verification protocol where some verifiers remain hidden and mobile [16]. Since the attackers are not aware of these hidden verifiers, a forged signal sent by the attackers will not reach these hidden verifiers at the correct time or incident angle depending on whether the underlying system uses time of flight measurement or directional antenna measurement. Such approach is secure if and only if the hidden mobile verifiers can actually be made completely hidden. Any verifier, in order to receive messages, must be built with local oscillators. Since the oscillators cannot be completely isolated in the real world, some power is leaked into the air and enables *device fingerprinting*. Rasmussen and Čapkun have demonstrated that radio fingerprinting is indeed feasible [9], hence



(a) Prover is R away from all three verifiers (b) Acceptance zone (c) Colluding attackers

Figure 1: Illustration of the δ test

the location of a hidden verifier might be detectable. While such possibility of compromise exists, it is still impractical with today's technology.

2.4 Attacker Model

In this paper, we only assume attackers cannot decode encrypted messages and are *power-limited*. That is, at any given time, an attacker can only output a finite amount of power. This assumption is realistic as attackers are still constrained by the laws of physics. Since we have only a very basic assumption on the attackers, we must carefully select our communication techniques. Namely, between ultrasound and RF wave, we should use the latter to prevent wormhole attacks. Čapkun and Hubaux gave an in-depth analysis and concluded that RF time-of-flight based systems exhibit the best security properties compared to other techniques [14].

When using RF wave communication with time-of-flight measurement, the prover cannot artificially reduce the measured distance because there is no known working communication technique that is faster than the speed of light in vacuum. Simply put, the round trip time-of-flight of a RF signal is at least $2\ell/c$ where ℓ is the physical distance between the prover and the verifier. We calculate the *perceived distance* as $\frac{1}{2}ct$ where t is the measured round trip time of flight; a prover cannot decrease his perceived distance from a verifier, but can enlarge his distance by delaying the response. Since the perceived distance can only be larger than the physical distance, the verifier can safely assume that the prover is located no farther away than the perceived distance.

The original distance bounding protocol and all subsequent protocols spent great effort to defend against terrorist collusion attackers. However, when the attackers share their private key information, these protocols can no longer distinguish between the attackers and might mistake several attackers as a single entity. In this paper we make no restriction on the information attackers share.

2.5 System Assumptions

Our protocol, while making very minimal assumptions about the attackers, assumes all verifiers are trustworthy, secure, and are able to weakly time synchronize among themselves. The granularity to which verifiers can synchronize time directly affects the accuracy of the location proof. For example, when two verifiers are unsynchronized by 1ns, the resulting uncertainty increases by as much as 0.3 meters.

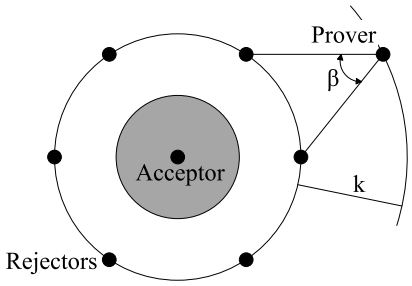


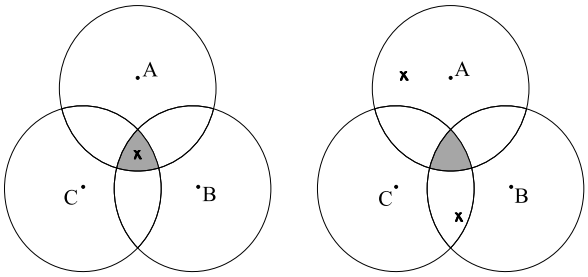
Figure 2: Illustration of protocol proposed by Vora and Nesterenko

This increase in uncertainty need not decrease security, but it may reduce the set of locations that can be effectively verified by the protocol. To effectively synchronize time, verifiers may synchronize over wires directly connecting them to each other. An alternative is to resynchronize wirelessly using the wired network to bootstrap: first, each pair of verifiers that are in wireless range of each other also are preconfigured with their distance (possibly measured using GPS or surveying equipment). Next, they use an asynchronous wired network to agree on a frequency to use for their transmissions. Then one verifier sends a message and the other verifier timestamps the receipt of this message. Because this uses the hardware the verifier already has for the challenge mechanism, these timestamps must already be very accurate, allowing for tightly synchronized clocks.

We assume that each verifier can communicate with every other verifier using a separate secure communication channel. These assumptions can be achieved by having physically secure verifiers communicate over secure wired links. We also assume that each prover shares a secret key with all verifiers. To distribute a key to a prover, we can use predistribution techniques such as the resurrecting duckling [12], public-key based techniques, or other key establishment techniques such as [13].

3. MULTILATERATION AND DISTANCE BOUNDING PROTOCOL

Since our protocol uses only radio waves, we will normalize our distance and time with respect to the speed of light



(a) Multilateration without colluding attackers

(b) Multilateration with colluding attackers

Figure 3: Multilateration

through the remainder of this paper. That is, we define a unit distance to be the distance traveled by radio wave over one unit time.

Multilateration can be used to verify a prover's exact location. If a prover is simultaneously within distance r_i of verifier V_i, then the prover must reside in the intersection of these circles. That is, if C_i is a circle of radius r_i centered at verifier V_i, then given there is only one prover, he must reside in $\bigcap C_i$. Figure 3(a) shows that a prover, denoted by the cross mark, proves his location in the shaded region by multilateration.

However, if the verifiers naïvely perform bit exchanges independently, then colluding attackers can cheat easily. For example, in Figure 3(b), A, B, and C are verifiers who perform independent verifications. Two colluding attackers, denoted by cross marks in the figure, can trick the verifiers into thinking someone is in the shaded area. This is done by having the attacker closer to A exchange bits with A, and having the other attacker perform bit exchanges with B and C.

3.1 Our Scheme

We observe that multilateration must be performed *simultaneously* by modifying the distance bounding protocol so that the prover responds to simultaneous challenges from each verifier. If there are a total of N verifiers, then each verifier sends a separate challenge, and the prover responds only when he has heard all challenges. The prover will prove that he has heard all challenges by combining all N challenges using a mathematical function. The function should have the property that when given N inputs, it produces a deterministic output; however, when given $M < N$ inputs, all outputs are equally likely. For clarity in description, we choose the bitwise exclusive or (XOR) of all N received bits and a predetermined random bit of the secret key, which satisfies these desirable function properties. XOR'ing part of the secret key does not undermine the user's shared secret since the message specifying the uplink and downlink frequencies is encoded with the secret key, thus being able to decode and find out the frequencies is equivalent to already knowing the secret key. On the other hand, as in the protocol proposed by Bussard and Bagga, XOR'ing the secret key prevents the terrorist collusion attack since a terrorist attacker would not release this bit to his colluders and must process all challenges by himself. Sastry et al. made similar observations on the choice of mathematical functions but suggested using a hash function which is costly in time and is unlikely to provide the granularity required when working with radio waves. In particular, if the uncertainty in time used to compute a hash function is 1 μs, the equivalent uncertainty in distance is 300 m, and is generally prohibitive for verifying specific location claims.

Our protocol can use any modulation scheme and multiple access protocol so long as they provide the decoding speed required. However, for the simplicity of describing our protocol, we adapt frequency shift keying for bit transmission, and frequency division to allow multiple senders to send simultaneously. That is, a verifier V_i is allocated two frequencies, f_{i0} and f_{i1}. To transmit the bit 0, V_i transmits on frequency f_{i0}, and to transmit the bit 1, V_i transmits on frequency f_{i1}. If the prover detects a signal on f_{i0} and not on f_{i1} the prover decodes a 0 from V_i, and if the prover detects a signal on f_{i1} and not on f_{i0}, the prover decodes a 1 from V_i, otherwise the prover makes no decision.

The prover initiates the verification request by first submitting his location claim securely using his shared secret with the verifiers. The verifiers then use the shared secrets to inform the prover the $2N$ downlink frequencies, $f_{10} \ldots f_{N1}$, and the 2 uplink frequencies, f_{p0} and f_{p1}. The verifiers agree on the challenges to send and time synchronize using their own secure channel; each verifier thus can calculate the expected response. Each verifier then sends its challenge at a time so that all N bits *arrive at the claimed location simultaneously*. If the location claim is correct, the prover receives all signals simultaneously, calculates the response value, then broadcasts the response value using the correct uplink frequency, which is in turn received by all verifiers. This is equivalent to one round of bit exchange in the original distance bounding protocol [1]. To perform another round in our protocol, the verifiers will select a fresh set of $2N + 2$ frequencies to avoid replay and jamming attacks.

When the response from prover is received, each verifier first checks if the response bit is correct. If the response bit value is incorrect, the location claim is rejected. If the verifiers do not receive a response or if the verifiers receive an ambiguous response (i.e., receiving both 0 and 1), then without penalizing the prover, the verifiers will initiate the next round by selecting a fresh set of downlink and uplink frequencies. If the response bit value is correct, the verifier moves on to check the elapsed time between when the verifier sent his challenge and when the response was received. Since the response bit is only one bit in length, the prover has a 50% chance to reply correctly by simply guessing. As in previous distance bounding protocols, our protocol is run many times to diminish the probability of the prover guessing correctly every round.

The additional time complexity of our scheme is minuscule compared to the original distance bounding protocol proposed by Brands and Chaum: the verifiers do not perform any additional calculation compared to Brands and Chaum, and the provers only need to perform one fast calculation such as XOR.

After each verifier has calculated the circle in which the prover must reside, we can intersect these circles to find a region in which the prover must reside. Because of processing delay, multipath, and other phenomena, this region is unlikely to be a single point. We show the properties of the uncertainty region in the rest of this section.

3.2 Calculation of Uncertainty

As in previous work, we use the δ-test [14], but we modify it slightly for our environment. Each verifier calculates the round-trip time by taking the difference between when it sent the challenge and when it received the response. If the claimed location is at a distance of r_i from verifier V_i, and the correct response is not received until $2\ell_i$ after the challenge was sent, then we define the *uncertainty* to be $u_i = 2\ell_i - 2r_i$.

The amount of uncertainty we accept is given by a threshold δ_i, which can vary based on verifier, claimed location, the purpose of the location proof, and other factors. For example, if the prover wants to prove that he is in a room, then the threshold δ may be small when the claimed location is close to a wall, but large when the claimed location is at the center of the room. This is because if the prover is close to a wall, even a reasonable uncertainty may cause the verified region in which the prover resides contains areas outside the room; on the other hand, the same uncertainty might be small enough so that the verified region is completely contained in the room when the claimed location is at the center. Likewise, when the prover is close to a boundary and wants to prove that he is within that boundary, verifiers along the boundary may admit relatively large δ, but a verifier perpendicular to the boundary will require a small δ. This is because a small uncertainty observed by the verifier perpendicular to the boundary would could push the verified region outside of the convex hull; on the other hand, relatively large uncertainty observed by the verifiers along the boundary can be admitted while keeping the verified region inside the restricted area. Once each δ_i is determined, the location claim is accepted if $|u_i| \leq \delta_i \quad \forall V_i$.

3.3 Lower Bound on Uncertainty

In our protocol, the uncertainty in round trip time measured by any verifier is greater than or equal to that measured in the original distance bounding protocol by the same verifier, with equality only if the location claim is correct. This statement is proven in Lemma 3.1. That is, our protocol poses a more stringent burden of proof on the prover than the original distance bounding protocol.

LEMMA 3.1. *Let P be a prover who seeks to prove his location claim. The uncertainty measured by verifier V_i in our protocol is at least the uncertainty measured by V_i in the original distance bounding protocol. That is, let the uncertainty of P measured by V_i in our protocol be u_i and let that measured by the original distance bounding protocol by Brands and Chaum be u_i^{DBP}. Then*

$$u_i \geq u_i^{DBP}$$

PROOF. The lemma is intuitively true since our protocol requires intertwined verification. However, we provide a quantitative measure that motivates later sections.

The uncertainty measured in both our protocol and the original distance bounding protocol can be broken into two terms, a processing delay, p_i, and any leftover uncertainty that results from location error, e_i, i.e.,

$$u_i = p_i + e_i$$

In our protocol, the processing delay is the time it takes from finishing collecting all N challenges to sending out the response, and is the same for all verifiers, i.e. $p_i = p$. Also, since the correct reply from the prover requires processing all N challenges, the prover must be able to process an individual challenge at least as quickly. (Otherwise we use the N-challenge processing algorithm to process the single challenge). That is, $p \geq p_i^{DBP} \quad \forall i$.

We then show that the measured location error is larger in our protocol than in the original distance bounding protocol. Let the actual distance and claimed distance between prover, P, and any verifier, V_i, be ℓ_i and r_i respectively. Then in the original distance bounding protocol, the i^{th} verifier expects the round trip time to be $2r_i$, but measures $2\ell_i$. Thus, the uncertainty caused by erroneous location in the original distance bounding protocol is $e_i^{DBP} = 2\ell_i - 2r_i$.

In our protocol, the N signals are sent at time $-r_i$ so they can reach the claimed location simultaneously at time 0. However, the prover is actually ℓ_i instead of the claimed r_i away. Thus, the prover cannot finish collecting all signals until time $\max_n(\ell_n - r_n)$. The prover then processes the signals and respond to all verifiers. The replies would take ℓ_i to reach verifier V_i at time $l_i + \max_n(l_n - r_n)$ for a total elapsed time of $l_i + \max_n(l_n - r_n) + r_i$. Thus the measured location error uncertainty is

$$\begin{aligned} e_i &= \ell_i + \max_n(\ell_n - r_n) + r_i - 2r_i \\ &= (\ell_i - r_i) + \max_n(\ell_n - r_n) \\ &\geq 2(\ell_i - r_i) \\ &= e_i^{DBP} \end{aligned}$$

Therefore,

$$u_i = (p_i + e_i) \geq \left(p_i^{DBP} + e_i^{DBP}\right) = u_i^{DBP}$$

□

The processing delay p_i is an implementation dependent variable and will not be considered for security analysis through the rest of the paper. That is, the security analysis provided in later sections assume $u_i = e_i = (\ell_i - r_i) + \max_n(\ell_n - r_n)$.

3.4 Optimality of Our Scheme

By observing the flow of information, we show that our protocol achieves the maximal security any location verification schemes based solely on time-of-flight can provide. Orthogonal techniques such as directional antennas or covert verifiers may be integrated to offer better security; however, we do not consider the usage of such techniques in this paper, and the security of such techniques should be considered separately.

THEOREM 3.2. *The protocol described in Section 3.1 provides maximal security that can be provided by any protocols based solely on time-of-flight.*

PROOF. In time-of-flight based protocols, a prover can only be caught falsifying a location claim if he cannot correctly respond within allowed elapsed time. That is, a prover can only be caught cheating if the uncertainty measured by a verifier is positive (we assume that attackers are sophisticated and can inject delay when uncertainty is negative). We thus prove our claim by showing that the uncertainty measured by any verifier in our system achieves an upper bound of the uncertainty measured by that same verifier in any systems based on time-of-flight alone. Consequently, if a set of provers can cheat a particular verifier in our system,

they would be able to cheat the same verifier in all other systems based on time-of-flight alone. In turn, if a set of provers can successfully attack our entire system, the same set of provers must also be able to successfully attack all other systems based on time-of-flight alone.

Let the collection of verifiers be $\mathbb{V}$, and the collection of transmitting verifiers be $\mathbb{V}_t \subseteq \mathbb{V}$. All verifiers, transmitting or silent, are assumed to have locations known to the public. Let there be a set of provers $\mathbb{P}$; any prover $P_k \in \mathbb{P}$ is ℓ_{ki} away from $V_i \in \mathbb{V}$. The set of provers seek to prove a location claim whose distance to verifier V_i is r_i. Finally, let verifier $V_i \in \mathbb{V}_t$ send out his challenge at time t_i.

Without loss of generality, we assume that the challenges generated by the set of transmitting verifiers $\mathbb{V}_t$ are intertwined. Meaning, the correct prover response is dependent on all challenges sent by transmitting verifiers. If only a subsets of challenges are intertwined, then we can isolate this subset of transmitters as the set of transmitting verifiers and consider the rest as silent verifiers. A set of verifiers can be regrouped into several sets with the above property. That is, any system that does not intertwine all its challenges can be viewed of as a set of systems, not necessarily mutually exclusive, each intertwining all its challenges.

We first observe that the challenge from V_i is expected to reach the claimed location at $t_i + r_i$. Since challenges are intertwined, the prover is expected to respond at

$$\max_{V_i \in \mathbb{V}_t} (t_i + r_i)$$

and the response arriving at V_i at time

$$E_i = \max_{V_n \in \mathbb{V}_t} (t_n + r_n) + r_i$$

The same challenge from V_i would reach P_k at $t_i + \ell_{ki}$. Hence, the earliest response from a prover can reach verifier V_i at time

$$\min_{P_k \in \mathbb{P}} \left(\max_{V_n \in \mathbb{V}_t} (t_n + \ell_{kn}) + \ell_{ki} \right)$$

The corresponding uncertainty ϕ_i, meaured by V_i, is simply the difference:

$$\phi_i = \min_{P_k \in \mathbb{P}} \left(\max_{V_n \in \mathbb{V}_t} (t_n + \ell_{kn}) + \ell_{ki} \right) - E_i$$

In our system, all verifiers send challenges that are intertwined, and the smallest uncertainty, u_i, measured by verifier V_i is shown in Lemma 3.1 to be

$$u_i = \min_{P_k \in \mathbb{P}} \left(\max_{V_n \in \mathbb{V}} (\ell_{kn} - r_n) + \ell_{ki} - r_i \right)$$

Adding and subtracting $\max_{V_n \in \mathbb{V}_t} (t_n + r_n)$, we get

$$u_i = \min_{P_k \in \mathbb{P}} \left(\max_{V_n \in \mathbb{V}} (\ell_{kn} - r_n) + \ell_{ki} + \max_{V_n \in \mathbb{V}_t} (t_n + r_n) \right) - E_i$$

Since the minimization of u_i is done over the set of provers, it is independent of the maximization inside, which is independently performed over the set of verifiers, $\mathbb{V}$. Therefore, by changing the maximization to be performed over the subset of transmitting verifiers $\mathbb{V}_t$, the uncertainty is decreased:

$$u_i \geq \min_{P_k \in \mathbb{P}} \left(\max_{V_n \in \mathbb{V}_t} (\ell_{kn} - r_n) + \ell_{ki} + \max_{V_n \in \mathbb{V}_t} (t_n + r_n) \right) - E_i$$

Since the sum of the maxima is larger than the maximum of the sum, we collapse the two maximum terms inside:

$$\begin{aligned} u_i &\geq \min_{P_k \in \mathbb{P}} \left(\max_{V_n \in \mathbb{V}_t} (\ell_{kn} - r_n + t_n + r_n) + \ell_{ki} \right) - E_i \\ &= \min_{P_k \in \mathbb{P}} \left(\max_{V_n \in \mathbb{V}_t} (\ell_{kn} + t_n) + \ell_{ki} \right) - E_i \\ &= \phi_i \end{aligned}$$

□

4. THREAT ANALYSIS

4.1 Wormhole Attack

The speed of sound is known to change in different media. Thus if sound waves were used for simultaneous distance bounding, an adversary could set up a wormhole between two colluders. One method, for example, is to run a copper wire in between the end points of the wormhole, on which sound travels much faster. The speed of light, however, is known to be greatest in vacuum. Since the speed of light in air is very close to that in vacuum, wormhole attacks do not provide any practical advantages to an attacker in our RF system.

4.2 Jamming Attack

An attacker can often disrupt wireless services by injecting high level of noise into the frequency band. This jamming decreases the signal to noise ratio (SNR) and consequently reduces the probability of successful reception of the challenges and responses. We adopt the idea of *frequency-hopping code division multiple access* (FH-CDMA) to prevent this attack.

In an FH-CDMA system, a wide frequency band allocated to the system is divided into many channels. A user is assigned a *hopping pattern* according to which the user should occupy the channels from time slot to time slot. An attacker, without knowing a normal user's hopping pattern, must randomly choose a subset of channels to jam. That is, the attacker can inject noise into as few as one channel to as many as all the channels. However, any realistic attacker is power limited; that is, regardless of the number of channels an attacker chooses to jam, the attacker can only output limited power. If an attacker jams only limited number of channels, the attacker is likely to be able to disrupt all services on those channels; however, the normal user is not likely to use those channels since there are many channels from which to choose. If an attacker otherwise jams all the channels, then his total power output might not be enough to disrupt any service across the wide frequency band.

The initial setup messages are sent using FH-CDMA on a code shared between the prover and verifiers, such a code can be established using the shared secret. For example, we could first establish a total ordering using the shared secret and map the secret to a CDMA code. We then use the method proposed by Li et al. to secure the CDMA code with AES [6]. To avoid jamming the next time location claim is sent, the CDMA code can be secured with AES encryption and using cipher block chaining (CBC) in counter mode.

The actual challenges are then sent on $2N$ non-overlapping frequencies. As long as the number of total channels C is much greater than $2N$ (which should be possible because the bandwidth of each emitted signal is small), a jammer cannot

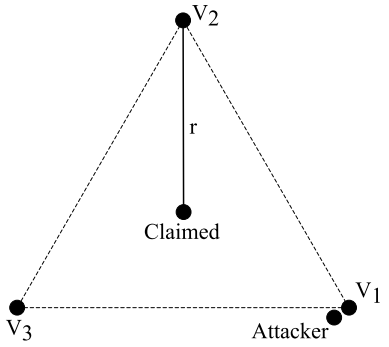


Figure 4: Illustration when an attacker is close to a verifier

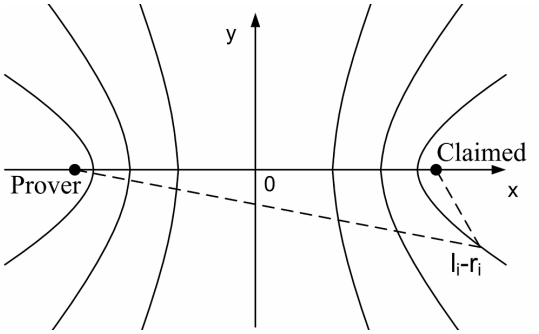


Figure 5: Hyperbolic contour of the difference in distances

a priori predict which channels to jam in order to disrupt the challenges. When the response is sent back, the jammer can disrupt the response only by transmitting on the specified channel that corresponds to the incorrect bit response. Therefore each channel chosen by the jammer with enough power has only $1/C$ chance to successfully disrupt the reply and N/C chance to successfully disrupt the challenge.

4.3 Replay Attack

In our scheme, the response bit is sent back using one of two channels. If the attacker knows either channel *a priori*, the attacker can simply sends on the channel he knows and probabilistically denies the verification of a normal user. In other words, if the attacker learned how to respond a 0 or 1, the system's availability suffers significantly.

One method an attacker can use is to record a normal user's response from round i, and replay such recording in round $i+1$. Our scheme thus uses fresh set of frequencies for each round. Moreover, since the communication between the prover and the verifiers is encoded using the shared secret key, the attackers cannot learn the chosen frequencies a priori. Thus, our scheme is resilient in the presence of replay attackers.

4.4 Collusion Attack

When a single prover makes a location claim, our intertwined verification ensures that she is where she claimed because she must perform all the verifications simultaneously, and her response would be late if she had falsified a location claim. However, while a single attacker cannot cheat all the verifiers, he may be able to cheat a subset of verifiers. For example, imagine we have three verifiers V_1, V_2, V_3 forming a regular triangle, an attacker sitting next to verifier V_1, and a claimed location at the geometric center of the triangle as shown in Figure 4. Let the distance between the verifiers and the center be r. All three verifiers would send challenges at time $-r$ and expect the correct response at time r. The attacker receives challenge from V_1 at time $-r$, and challenges from V_2, V_3 at time $\left(\sqrt{3}-1\right) r < r$. Thus, by delaying his response, the attacker is able to respond to verifier V_1 at time r.

Since one attacker may attack a subset of verifiers, a set of attackers may be able to collude and attack the entire system in a distributed manner. In the rest of this section, we give a mathematical model that allows us to completely characterize the feasibility of collusion attacks.

From Lemma 3.1, we observe that the uncertainty measured by verifier V_i is

$$u_i = (\ell_i - r_i) + \max_n (\ell_n - r_n)$$

where ℓ_i is the distance from the prover to verifier V_i and r_i is the distance from the claimed location to verifier V_i. An attacker can cheat verifier V_i if $u_i \leq 0$. In the example given above,

$$\max_n (\ell_n - r_n) = \max_n \ell_n - r = (\sqrt{3}-1)r$$

and

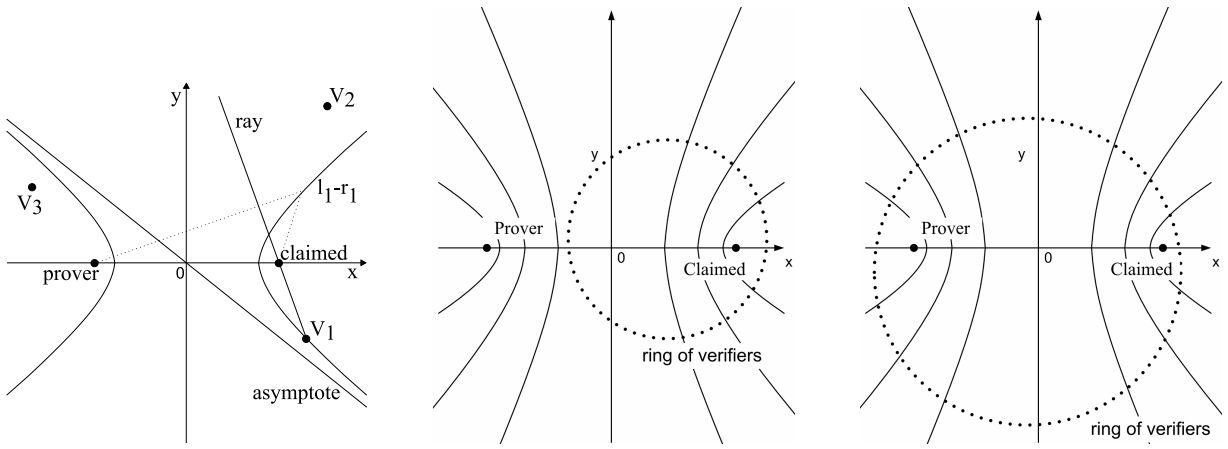
$$\ell_1 - r_1 = -r_1 = -r$$

for a sum that is less than zero.

To completely characterize our system, one needs to analyze the quantity $\ell_i - r_i$, the difference in distances from attacker to verifier V_i and from the claimed location to V_i. It is known that a hyperbola with foci f_1 and f_2 has the property that the differences in distances from any points on the hyperbola to the foci have the same magnitude. Therefore, if we let the attacker and the claimed location be the foci, the contour of the quantity $\ell_i - r_i$ is simply a collection of hyperbolas. We analyze two special cases: the first scenario has three verifiers forming a triangle, and the second scenario has infinitely many verifiers densely distributed on the boundary of a convex space. In both cases, the claimed location is assumed to be inside the convex hull. These two cases present the two extremes in number of verifiers used in the system.

To analyze both cases, we first orient the prover and the claimed location so that the prover is at $-d$ and the claimed location is at $+d$ on the x-axis as shown in Figure 5. We will refer to the contour that is perpendicular to the x-axis as the y-axis, this contour presents the collection of points that are equidistant from the prover and from the claimed location. Each hyperbola is made up of two contours that are symmetric about the y-axis. The two contours have same magnitude but opposite signs. In particular, the contour to the right of the y-axis (closer to the claimed location) represents a positive value of $\ell_i - r_i = a > 0$, and similarly, the contour to the left of the y-axis (closer to the prover) represents a negative value of $\ell_i - r_i = -a < 0$.

The first scenario is illustrated in Figure 6(a). Since the claimed location is in the interior of the verifier triangle, one of the verifiers, V_1, must be located to the right of the claimed location, on the contour $\ell_1 - r_1$. Furthermore,



(a) Three verifiers

(b) Infinitely many verifiers, attacker outside the convex space

(c) Infinitely many verifiers, attacker inside the convex space

Figure 6: Illustrations of the scenarios studied in Section 4.4

$\ell_1 - r_1 > 0$ since it lies on a contour closer to the claimed location than the prover. Since the contour plot is symmetric also about the x-axis, without loss of generality, we let the verifier V_1 have a negative y value. We then draw a ray from verifier V_1 through the claimed location, we also draw the asymptote of this particular hyperbolic contour running through quadrants II and IV, as shown in Figure 6(a). Again since the verifier triangle contains the claimed location, the remaining two verifiers must lie on separate sides of the ray. The ray never intersects the asymptote since the ray has a steeper slope. We call the verifier on the prover side of the ray V_3, and the other verifier V_2.

Two hyperbolic contours of same magnitude but opposite signs lie on two sides of the asymptote. Since the ray does not intersect the asymptote, $\ell_2 - r_2 > -(\ell_1 - r_1)$; consequently,

$$\max_n (\ell_n - r_n) + (\ell_2 - r_2) > 0$$

That is, an attacker can cheat at most one verifier, in this case V_3, by himself. In other words, if there are three verifiers, then regardless of where the attacker is, he needs at least two other colluding attackers to cheat our system with three verifiers.

In the second scenario we have infinitely many verifiers densely distributed on the *boundary of a convex space*. Since the verifiers are densely distributed, there must be at least one verifier located on the ray from claimed location to positive infinity, call it V_{pos} and observe $\ell_{pos} - r_{pos} = 2d$. If the prover is located outside the convex space, as shown in Figure 6(b), then no verifiers are located on the $-2d$ contour, which is a ray from the prover to negative infinity. Consequently, no verifier can be cheated by the attacker. If the prover is otherwise located inside the convex space, as shown in Figure 6(c), then only the verifiers located on the ray from the prover to negative infinity can be cheated. Since an attacker is needed for every orientation to cheat the system, the attacker needs the same order of colluders as there are verifiers in the system.

As all convex hulls can be partitioned into triangles, if the claimed location is inside the convex hull, it must be inside at least one triangle. We thus establish a very loose bound that attackers need at least two other colluders to cheat against any system with three or more verifiers.

COROLLARY 4.1. *Our system is secure against terrorist collusion attack.*

PROOF. In the terrorist attack scenario, the adversaries keep their individual private keys secret. Therefore, one particular attacker waits for the rest of attackers to route the challenges to him, process the challenges, and then reply. Because of triangle inequality, the attacker can respond faster if he himself receives the challenges and processes them than to have other attackers route the challenges. Thus a terrorist collusion attacker in our system performs no better than a single attacker, who we have shown is unable to attack the verifiers. □

Let N verifiers form a regular N-gon, and restrict the colluding attackers to be outside the N-gon, then through simulation, roughly N attackers are needed to attack our protocol. I.e., one attacker is needed to attack each verifier.

To completely characterize our system using the hyperbolic contours, we can make the x-axis the real axis and the y-axis the imaginary axis. The complex arcsine and sine functions can then be used to analytically calculate the quantity $\ell_i - r_i$. In particular,

$$\ell_i - r_i = d \cdot \sin\left(\mathsf{Re}\left\{\arcsin\left(\frac{L(V_i)}{d}\right)\right\}\right)$$

where $L(V_i)$ is the complex location of verifier V_i. The uncertainty, $\max_k (\ell_k - r_k) + (\ell_i - r_i)$, can then be characterized for any verifier, prover, and claimed location setup.

5. SELF JAMMING SCHEME

In this section we propose a novel self jamming scheme. The self jamming scheme compromises the ability of our protocol to defend against jammers; however, the self jamming scheme improves the security of our system by requiring colluding attackers to be significantly more resourceful in order to carry out the attack. Simply put, there is a tradeoff between availability and integrity.

In the self jamming scheme, each verifier still sends his challenge for a short period of time so that all challenges reach the claimed location simultaneously. However, outside of this transmission period, the verifiers transmit with full power on all chosen challenge channels until the last challenge is transmitted. In essence, before all challenges are transmitted, the verifiers act as jammers when not sending challenges.

If the prover is seeking to attack the system by claiming incorrect location, the challenges will not reach the prover simultaneously. Moreover, when the signal of a single verifier V_i reaches the attacker, the attacker is at the same time being jammed by the other verifiers. We thus observe that every false claiming attacker must be equipped with N directional antennas, where N is the number of verifiers, and point one antenna at each of the verifiers in order to reject jamming from other verifiers and receive the challenge from the one verifier of interest. Since at least three, and as many as N, colluders are required to attack our system, the colluding party requires at least $3N$ and as many as N^2 directional antennas total. A prover correctly sending her location only needs one omnidirectional antenna to receive all challenges. The system itself, likewise, needs only one omnidirectional antenna per verifier, for a total of N antennas. Therefore, the self jamming scheme requires the colluding attackers to have 3 to N times as many directional antennas as the number of omnidirectional antennas the verifiers require.

6. CONCLUSIONS

The distance bounding protocol provides a strong result in verifying that a prover is within a certain distance from a verifier. In order to verify location information that is more precise, the multilateration technique can be used. We proposed a multilateration protocol and prove that it achieves the maximal security that any system based on time-of-flight alone can provide.

While we showed that time-of-flight based multilateration scheme can still be attacked by colluding attackers, we have provided a method to fully characterize the feasibility of such attacks against our system. Furthermore, we proposed a novel self-jamming scheme that requires the attackers to have substantially more resources than our system in order to attack it.

7. ACKNOWLEDGMENT

We would like to extend our appreciation to our shepherd, Professor Radha Poovendran, and the anonymous reviewers, for their valuable feedback.

8. REFERENCES

[1] Stefan Brands and David Chaum. Distance-bounding protocols (extended abstract). In *Theory and Application of Cryptographic Techniques*, pages 344–359, 1993.

[2] Laurent Bussard and Walid Bagga. Distance-bounding proof of knowledge to avoid real-time attacks. In *SEC '05: 20th IFIP International Information Security Conference*, Makuhari-Messe, Chiba, Japan, May 2005.

[3] Gerhard P. Hancke. Practical attacks on proximity identification systems (short paper). In *SP '06: Proceedings of the 2006 IEEE Symposium on Security and Privacy*, pages 328–333, Washington, DC, USA, 2006. IEEE Computer Society.

[4] G.P. Hancke and M.G. Kuhn. An RFID distance bounding protocol. *First International Conference on Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005.*, pages 67–73, 05-09 Sept. 2005.

[5] R. Jain, A. Puri, and R. Sengupta. Geographical routing using partial information for wireless ad hoc networks. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, 8(1):48–57, Feb 2001.

[6] Tontong Li, Jian Ren, Qi Ling, and Anil Jain. Physical layer built-in security analysis and enhancement of cdma systems. In *Proceedings of the Military Communications Conference, 2005. MILCOM 2005. IEEE*, pages 956–962, October 2005.

[7] Julio C. Navas and Tomasz Imielinski. GeoCast–geographic addressing and routing. In *MobiCom '97: Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*, pages 66–76, New York, NY, USA, 1997. ACM.

[8] Asher Peres and Daniel Terno. Quantum information and relativity theory. *Reviews of Modern Physics*, 76:93 – 123, Jan. 2004.

[9] Kasper Bonne Rasmussen and Srdjan Čapkun. Implications of radio fingerprinting on the security of sensor networks. In *SecureComm 2007: Proceedings of the 3rd international conference on security and privacy in communication networks*, pages 29–37, 2007.

[10] Naveen Sastry, Umesh Shankar, and David Wagner. Secure verification of location claims. In *Proceedings of the ACM Workshop on Wireless Security (WiSe 2003)*, pages 1–10, September 2003.

[11] D. Singelee and B. Preneel. Location verification using secure distance bounding protocols. In *IEEE International Conference on Mobile Adhoc and Sensor Systems, 2005 (MASS 2005).*

[12] Frank Stajano. The resurrecting duckling - what next? In *Revised Papers from the 8th International Workshop on Security Protocols*, pages 204–214, London, UK, 2001. Springer-Verlag.

[13] Mario Strasser, Christina Pöpper, Srdjan Čapkun, and Mario Čagalj. Jamming-resistant key establishment using uncoordinated frequency hopping. In *SP '08: Proceedings of the 2008 IEEE Symposium on Security and Privacy*, Washington, DC, USA, 2008. IEEE Computer Society.

[14] S. Čapkun and J.-P. Hubaux. Secure positioning of wireless devices with application to sensor networks. *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, 3:1917–1928 vol. 3, 13-17 March 2005.

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

[16] S. Čapkun, M. Srivastava, and M. Čagalj. Secure localization with hidden and mobile base stations. In *IEEE Conference on Computer Communications (INFOCOM)*. IEEE, apr 2006.

[17] A. J. Viterbi. *CDMA Principles of Spread Spectrum Communication*. Addison-Wesley, 1995.

[18] Adnan Vora and Mikhail Nesterenko. Secure location verification using radio broadcast. *IEEE Trans. Dependable Secur. Comput.*, 3(4):377–385, 2006.

[19] Yan Yu, Ramesh Govindan, and Deborah Estrin. Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks. Technical report, UCLA Computer Science Department Technical Report, 2001.