UNIVERSITY OF CALIFORNIA
RIVERSIDE

On Secure Localization Without Simultaneous Challenges

A Thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science

in

Computer Science

by

Smruti Parichha

December 2011

Thesis Committee:

Dr. Mart Molle, Chairperson
Dr. Michalis Faloutsos
Dr. Frank Vahid

The Thesis of Smruti Parichha is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I would like to extend my sincere thanks to my advisor Dr. Mart Molle, for his kind guidance and helpful advice over the years. This thesis would not have been possible without him. I have learned immensely through my continuous interaction with him. I would also like to thank my committee members, Dr. Michalis Faloutsos and Dr. Frank Vahid for taking the time to help and guide me, for their constructive feedback during presentations and the thesis defense. I sincerely thank Dr. Laxmi Narayan Bhuyan, who encouraged me to apply to graduate schools in USA, and for his help and guidance through the application process, and the staff of the CS department who have helped me whenever I needed them.

The networking group at the Department of Computer Science and Engineering, UC Riverside, are energetic and their enthusiasm is contagious. My thanks to Dr. Srikanth V. Krishnamurthy, Dr. Harsha Madhyastha and Dr Chinya. V. Ravishankar, from our group, who have helped me in one way or the other to reach my goals. My lab-mates have not only been great colleagues, but also great friends and well-wishers in this journey.

As I submit this thesis, I remember with gratitude, my teachers from high school and National Institute of Technology Rourkela (NITR). But for their efforts, I would not be in graduate school. I would like to thank friends who have been with me since childhood till date, and friends I met at Riverside who helped me feel at home here. Keeping in touch with them has helped me immensely throughout graduate school.

Lastly, I extend my thanks to my family with deep gratitude and love. They have stood by me in the toughest of times and cheered for me at every occasion of achievement. My greatest role models and supporters belong to my family. I feel extremely lucky to belong with them.

This thesis is dedicated to those who have constantly encouraged and inspired me to pursue knowledge through higher education.

ABSTRACT OF THE THESIS

On Secure Localization Without Simultaneous Challenges

by

Smruti Parichha

Master of Science, Graduate Program in Computer Science
University of California, Riverside, December 2011
Dr. Mart Molle, Chairperson

Time-based secure localization protocols allow a group of mutually trusted entities called *verifiers* to cooperatively determine the location of an untrusted and possibly malicious stranger entity called the *prover*. Many applications associate certain privileges with the true physical location of an entity, therefore there is an incentive for a prover to claim a more "valuable" location, different from its true location. A well known threat to time-based localization protocols is *distance fraud* where a malicious prover misrepresents its location by intentionally changing its response time across a series of bilateral dialogues with individual verifiers. To address this threat, secure localization protocols must use a technique called "simultaneous multilateration".

A recently introduced protocol used simultaneous challenges by multiple verifiers, over separate RF channels, to defend against distance fraud. The authors also claimed that simultaneous multilateration by using simultaneous challenges, is "***o****ptimal in the sense of achieving the maximal security* that can be provided by any time-based localization protocol. In the first part of this thesis, we show that the structure of this newly proposed protocol is unnecessary, and significantly more complex than existing protocols. We propose a new protocol named *Elliptical Multilateration (EM)* that does not use simultaneous challenges. Instead, our EM protocol achieves simultaneity in multilateration by using multiple passive receivers to observe the prover's response. Our EM protocol requires fewer resources, provides commensurate security against distance fraud, and is inherently more accurate.

The second part of this thesis focuses on the issues related to practical implementation of time-based localization protocols. Most existing works focus on cryptographic aspects of time-based secure localization protocols, and do not address the issues that arise when these time-based protocols are implemented on real systems. For example, many authors have designed protocols based on single-bit exchanges (which is non-conformant with standard networking protocols and hardware, and extremely difficult to implement), ignore inevitable measurement errors etc. In this thesis, we show that the magnitude of measurement errors depends largely on the structure of a protocol, and differ significantly across the known localization protocols. We investigate whether measurements can be made with sufficient accuracy to achieve localization on the order of a few meters.

To the best of our knowledge, this thesis is the first work that attempts to analyze measurement error in practical realization of different localization protocols. The factors influencing the measurement errors, which are highlighted in this thesis, are significant and cannot be ignored. We show that taking into account the significance of message structure and the factors influencing the measurement error, can lead to new protocols that are no worse in terms of security, need fewer message exchanges, and achieve better accuracy in comparison to existing time-based localization protocols.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In dynamic and self-organizing wireless networks, many applications require authentication of a node to include both its identity and its *physical location.* The ability of a node to determine its own position in the network (using GPS, anchor node beacons etc.) is termed *self-localization.* Due to the privileges associated with the physical location of nodes, there is high incentive for a dishonest node to claim a false, more valuable location $\hat{P}$, instead of its true location. Claiming a false location under self-localization is trivial. Therefore, even if nodes can self-localize, there is a need for protocols that enable other mutually trusted nodes in the network to securely verify the location claim of a node (*location verification*), or independently determine the node's location (*localization*) without trusting the node itself.

Localization protocols using the time-based ranging technique are known [4] to exhibit better security properties than those based on other techniques (Angle-of-Arrival (AoA), Received Signal Strength Indication (RSSI) etc.). Therefore, in this thesis, we focus on time-based localization protocols. In time-based secure localization/location verification protocols, a mutually trusted group of nodes (*verifiers*), use timed challenge-response dialogs to estimate their respective distances to the untrusted node (*prover*), whose location is to be determined/verified. The verifiers then mathematically combine their distance estimates to determine the prover's location. This process is called *multilateration* because multiple verifiers perform lateration (range based distance measurements) with the prover. In general, the prover has to engage in multiple rounds of challenge-response with different verifiers during a complete execution of any time-based localization protocol.

1.1 Motivation

Most of the existing literature about time-based localization or location verification protocols and secure distance measurement, have focused on cryptographic issues like "how does a verifier authenticate the identity of the entity that responded to its challenge?" or "Can we protect the identity or location privacy of the participants from eavesdroppers?" etc. In this thesis, we do not consider such cryptographic issues. Instead, we focus on two other issues:

1.1.1 Protocol Structure

A recently published work [5] proposed a time-based protocol for secure localization and claimed that their protocol is *optimal* in the sense of achieving the "*maximal security that can be provided by any protocols based solely on time-of-flight*". Their claim is independent of the cryptographic issues related to the content of the messages, and purely based on the pattern of message exchanges it employs. The protocol structure in [5] turns out to be significantly more complex than existing time-based secure localization protocols because of simultaneous challenges sent by all the verifiers in every round of the protocol. Therefore, we ask the question — “Are simultaneous challenges really necessary for secure localization?” More generally, “Is it possible that the additional complexity negatively affects the efficiency of the protocol in terms of the number of message exchanges required, resources consumed, and measurement accuracy?”

1.1.2 Implementation in Real Systems

In existing works on time-based localization protocols, most authors assume that each participating entity can control the departure time of its message(s) precisely, and also that they can timestamp the arrival/departure events for messages without any measurement error. Sometimes, the authors also make additional assumptions about messages and the physical channel over which they are exchanged, for example zero-length messages containing single data bits, zero response delay etc. Moreover, it is well known that for secure time-based protocols using a wireless channel, radio frequency (RF) must be used (as opposed to a slower medium like Ultrasound) to prevent the protocol from security breaches. Over an RF channel, a timing error of $3.3ns \equiv 1m$ error in distance measurement. Thus we ask the question – “Can we measure and/or control the protocol events with sufficient precision to achieve localization accuracy on the order of a few meters?” and “Does the structure and design of different protocols affect the achievable accuracy?”

1.2 Contributions

In this thesis, we first classify known time-based multilateration protocols based on how the challenge-response dialogs are executed in each. By doing this, we intend to highlight the differences in the structure and pattern of message exchanges during the challenge-response rounds in each. We also describe each protocol with a common mathematical notation, which we use later to compare and analyze the protocols.

Next, we answer the questions we posed in section 1.1 by introducing a new time-of-arrival protocol called "Elliptical Multilateration (EM)" for secure localization/ location verification. Our protocol does not require multiple simultaneous challenges in each round of challenge-response and uses only a single challenge in each round. Although it reduces the total number of messages exchanged, it provides better security and requires fewer resources than the protocol proposed in [5]. We describe our protocol and then compare its performance with existing protocols that we previously classified. The comparison is in terms of the number of challenge-response rounds required and messages generated when: (i) the prover is honest, (ii) the prover is malicious and sophisticated in terms of resources (multiple radios equipped with directional antennas), and (iii) the prover is malicious but limited in resources (single radio with an omnidirectional antenna). We show that our EM protocol requires equal or fewer message exchanges in each case. We also show that our EM protocol provides a level of security that is better or commensurate to that of other protocols, and is able to efficiently expose distance fraud attacks.

In the second part of this thesis, we show that (amount and magnitude) of measurement errors depend largely on the structure of message exchanges in a localization protocol, and differ significantly across known protocols for secure localization. Other works on measurement error in RF time-based protocols are related to specific testbeds, or are in the domain of precise time synchronization. To the best of our knowledge, this is the first work that attempts to analyze measurement error in practical realization of different localization protocols. The factors influencing the measurement errors, which are highlighted in this thesis, are significant and cannot be ignored. If we take these factors into account, we can design time-based secure localization protocols that are as secure, need fewer message exchanges, and achieve better accuracy in localization than existing ones.

Chapter 2

Background and Related Work

In this chapter, we classify existing time-based protocols for secure localization based on the structure and pattern of the challenge-response message exchanges between the verifier(s) and the prover. We highlight the differences in the patterns of the message exchanges because we will use these differences in chapter 5 to compare the errors introduced and the expected positioning accuracy when these protocols are implemented on real systems. We also describe all the protocols using a common notation, which will allow us to mathematically analyze and compare them later in this thesis.

2.1 System Model and Notation

We consider a wireless network where nodes may be mobile and are free to join and leave the network dynamically over time. However, during one execution of the protocol to verify a particular location claim, we assume that all participating nodes are at rest with respect to each other. The network consists of two kinds of nodes, under the usual assumptions for distance bounding and/or multilateration protocols in the literature.

Verifiers $\{v, w, ...\} \in V$ are mutually trusted nodes who can securely exchange information amongst themselves. Handling defective and/or malicious verifiers is beyond the scope of this thesis. We assume that all verifiers know each other's exact locations relative to some physical coordinate system. Moreover, all verifiers can timestamp message arrival/departure events with high precision, which also allows for synchronized clocks. The verifiers might establish shared keys with the prover for exchange of information not pertaining to location-based applications. The focus of this thesis will be on the secure localization aspect of authentication; trust related to other applications and the cryptographic aspects related to secure identity authentication will not be discussed.

Prover p is an untrusted and possibly malicious node; p's goal is to convince the verifiers that its true physical location matches its claimed location $\hat{P}$ by engaging them in a location-verification protocol. The prover's *processing delay*, Δ_P, represents the minimum time between receiving a challenge and sending the appropriate response, and its value is known to the verifiers. A malicious prover might increase the value of Δ_P to mislead the verifier, however cannot reduce it. We do not assume that the prover has any knowledge of the information shared between the mutually cooperating verifiers.

Without loss of generality, we normalize time and space so that signals propagate at unit speed, i.e., one unit of distance per unit time. Thus, for any two points x and y, we can use $D(x, y)$ to represent both the distance and propagation delay between them. The nodes communicate by transmitting messages through an isotropic broadcast medium with no obstacles to block the direct line-of-sight paths, so $D(x, y) \equiv D(y, x)$ holds for all points x and y. In addition, the environment is anechoic so there are no multipath effects and each node receives a single copy of each message. Although various authors make different assumptions about the message structure – ranging from “a single bit” to “a data stream of arbitrary length”, there is a specific reference point within each message that is used for timing message arrival and departure events. We use the notation e_x^y to represent the discrete event that the reference point from a message sent by node x is now at the location of node y. A timestamp was denoted as a function of applying an observer entity o’s clock $C_o(.)$ to an event. In the rest of the thesis, we assume that the clocks of all the verifiers participating in time-based multilateration protocols are synchronized to a common time reference. Therefore, we drop the subscript o in $C_o(.)$ and denote the timestamps as $C(.)$ for simplicity while discussing localization protocols.

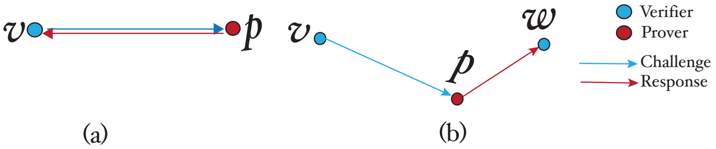


Figure 2.1: (a) a challenge-response echo executed between verifier v and prover p (b) a challenge-response relay consisting of a challenge sent from verifier v to prover p and a response sent from p to passive verifier w

Time-based secure localization protocols use multiple rounds of timed “challenge-response” message exchanges. Details of such exchanges will be described in section 2.2. A single round of challenge-response may be executed in one of two ways: A challenge-response pair that comprises of a challenge sent by verifier v to p and the corresponding response sent from p monitored by the same verifier v, will be referred to as a *“challenge-response echo”*. Such a two-way dialogue $v \rightarrow p \rightarrow v$ is illustrated in Fig. 2.1(a). The running time for a $v \rightarrow p \rightarrow v$ *echo* is

$$T_p(v, v) = D(v, p) + \Delta_P + D(p, v) \tag{2.1}$$

A challenge-response pair that comprises of a challenge sent from v to p and the corresponding response sent from p heard by a different verifier, say w, will be referred to as a "challenge-response relay" as shown in Fig.2.1(b). Let v be the verifier that sends the challenge in a particular round of challenge-response. When v executes a challenge-response echo $v \rightarrow p \rightarrow v$, the passive verifier w simultaneously monitors the challenge-response relay $v \rightarrow p \rightarrow w$. The total running time of the $v \rightarrow p \rightarrow w$ *relay* through p's true location is

$$T_p(v, w) = D(p, w) + \Delta_P + D(v, p) \tag{2.2}$$

2.2 Distance Bounding and In-Region Verification

Secure location verification using time-of-flight originated with the naïve *distance bounding* protocol proposed by Brands and Chaum [2]. Distance bounding therefore allows the verifier v to calculate an upper bound on its distance from prover p, and thus constrain its location to within a circular disk, with v at the center. Here the prover first commits a self-chosen bit string ξ to the verifier and the verifier generates a challenge nonce ζ containing the same number of bits. Thereafter, the verifier executes a rapid succession of challenge-response rounds with the prover.

Notice that the jth round includes the following discrete events as shown in Fig. 2.2: (i) e_v^v, the verifier sends ζ_j; (ii) e_v^p, the prover receives ζ_j, then stores it and computes $\chi_j = f(\zeta_j, \xi_j)$, where $f(\cdot)$ is a pre-arranged function such as XOR; (iii) e_p^p, the prover sends χ_j; and (iv) e_p^v, the verifier receives χ_j and then stores it. Since events (i) and (iv) occur at the verifier, v can use the difference in their respective timestamps as one sample of the

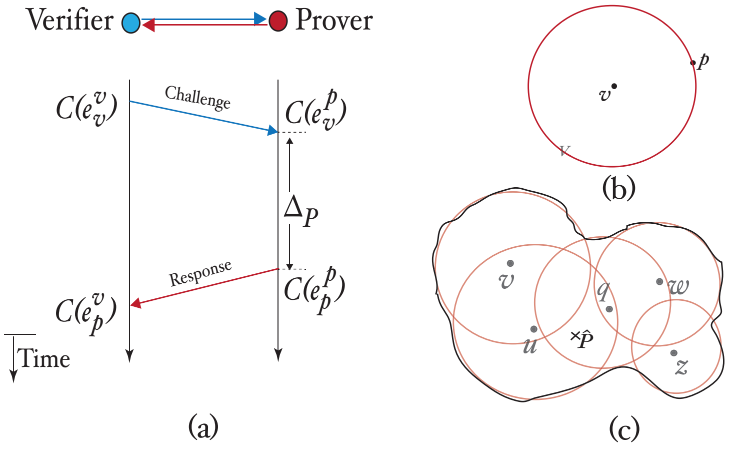


Figure 2.2: (a) space-time diagram of a single round of challenge-response (b) circular constraint generated on p's location when a single verifier v executes distance bounding (c) in-region verification with multiple verifiers

running time for a $v \to p \to v$ challenge-response echo:

$$T(v,v) \equiv C(e_p^v) - C(e_v^v) = D(v,p) + \Delta_P + D(p,v) \tag{2.3}$$

Since $T(v,v)$ includes (at least) p's known processing delay, Δ_P, v knows that

$$D(v,p) \leq (T(v,v) - \Delta_P)/2 \tag{2.4}$$

Following the last round, p signs the copy of ζ it has collected with the commitment and sends it to v for verification. The implicit assumption in this protocol that the prover cannot reply faster than the speed at which signals travel over medium of communication used. If it holds, the prover cannot pretend to be closer than it really is.

Sastry et al. [15] first formalized the problem of *in-region verification*, where p's presence in a region of arbitrary shape (instead of a circular region) needs to be verified.

They proposed the *echo protocol*) where the prover "immediately" echos the exact contents of a challenge nonce from the verifier. The echo protocol is executed with multiple verifiers, chosen such that the circular constraints generated by each overlap to cover the desired region where p's presence needs to be confirmed. This is illustrated in Fig. 2.2(c)

2.3 Secure Multilateration Protocols

Multilateration, which combines distance bounds generated by at least $(d + 1)$ verifiers, can be used to determine the location of the prover in d-dimensional space.

2.3.1 Single Input Single Output (SISO) Multilateration

The simplest case of multilateration is the *Verifiable Multilateration (VM)* protocol of Capkun and Hubaux [4]. In each round of VM, the prover p responds to a single challenge (input) generated by a single verifier $v \in V$. This verifier is also the sole receiver that monitors p's response (output), therefore it is termed as *Single Input Single Output (SISO).* In each round of SISO multilateration, a different verifier executes a challenge-response echo with the prover, and can constrain the prover's location to a circular disk around itself. The intersection of all the circular constraints generated determines the location of the prover p. However, multilateration through a sequence of challenge-response echoes is vulnerable to *distance fraud* attacks, where a malicious prover changes the value of Δ_P for each challenge-response echo with a different verifier [15].

VM introduces the *point-in-triangle* test to eliminate this threat. If location $\hat{P}$ is within the triangle formed by the locations of the three verifiers (more generally, the convex hull generated by the N participating verifiers), then any point $p \neq \hat{P}$ we must have

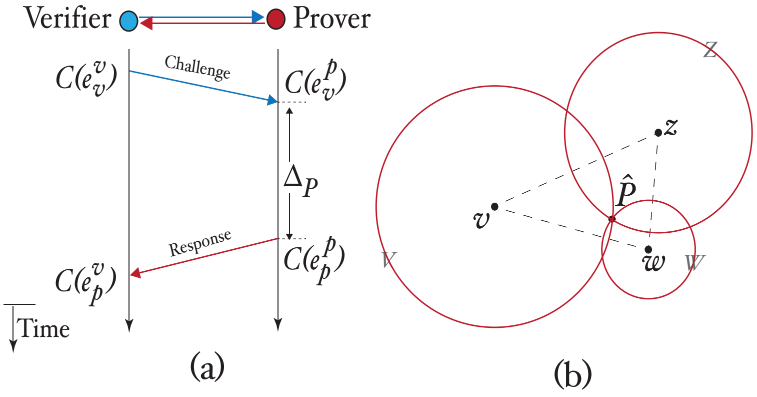


Figure 2.3: (a) space time diagram for an individual challenge-response echo (b) combining circular constraints generated by each verifier to localize the prover

$D(v,p) > D(v,\hat{P})$ for at least one $v \in V$. Thus, a single dishonest prover cannot cheat all verifiers unless it can generate the response to v's challenge in less time than Δ_P.

The *VM* protocol also introduced the δ*-test* for deciding whether or not verifier v should be satisfied by p's response. Let

$$\hat{T}(v,v) \equiv 2 \cdot D(v,\hat{P}) + \Delta_P \tag{2.5}$$

be the expected running time for a $v \rightarrow p \rightarrow v$ challenge-response echo when the prover's true location matches its claim of $\hat{P}$, and $T_p(v,v)$ from Eq. 2.3 be its observed value. Then v should accept $\hat{P}$ if

$$|\hat{T}(v,v) - T_p(v,v)| \leq \delta \tag{2.6}$$

If p satisifes the δ-test for all verifiers in V, then they conclude that p's location is somewhere within the mutual intersection of their respective (circular) distance bounds.

2.3.2 Single Input Multiple Output (SIMO) Multilateration

SISO multilateration requires the verifiers to administer the point-in-triangle and the δ tests to prevent a malicious prover from cheating across multiple rounds of challenge-response echoes. A better way to prevent distance fraud by cheating is to execute multilateration using multi-party dialogs instead of a simple bilateral challenge-response echoes in each round. In each round of SIMO multilateration, the prover p must respond to a single input (challenge generated by one of the verifiers), but multiple verifiers at different locations monitor the response in parallel (multiple output). Therefore, in each round of SIMO multilateration, a single verifier $v \in V$ executes a challenge-response echo with the prover, while multiple other verifiers $\{w, z, u, ...\} \in V$ monitor challenge-response relays.

Hyperbolic Multilateration

Hyperbolic multilateration has a long history of application to surveillance and navigation systems [16]. Here three or more synchronized receivers at known locations can jointly determine the location of another node based on the TDoA of its transmission(s). The same method also allows one receiver to determine its own location from the TDoA of synchronized transmissions originating from multiple locations, such as in the Global Positioning System.

Hyperbolic multilateration is an example of SIMO multilateration, where a single verifier, say v, sends the challenge, but N verifiers independently record the arrival time for the prover's response. SIMO hyperbolic multilateration eliminates *distance fraud* attacks by removing Δ_P from the equations for determining p's location. For example, suppose verifiers v and w both use synchronized clocks to timestamp their respective arrival times for the same response from p, say $C(e_p^v)$ and $C(e_p^w)$. Even though neither verifier knows p's true location, both verifiers are timing the *same message* – which left p at time $C(e_p^p)$, and thereafter took $D(p,v)$ to reach v and $D(p,w)$ to reach w. Thus $C(e_p^v) - D(p,v) \equiv C(e_p^w) - D(p,w) = C(e_p^p)$, and hence

$$D(p,v) - D(p,w) = C(e_p^v) - C(e_p^w) \tag{2.7}$$

whose solution is lobe VW of the *hyperbola* with foci v and w. [1]

Localization with Witnesses

Saha and Molle [14, 13] proposed a SIMO hyperbolic localization protocol, dubbed *Localization with Witnesses*, that does *not* require the verifiers to have synchronized clocks. In their protocol, intended for localization in dynamic and self-organizing networks, a single verifier, say v, is chosen to engage in a packet-based $v \to p \to v$ challenge-response *echo* protocol with prover p over an RF broadcast channel. At the same time, multiple receive-

[1] A conic section with foci at a and b will be denoted as AB in this thesis.

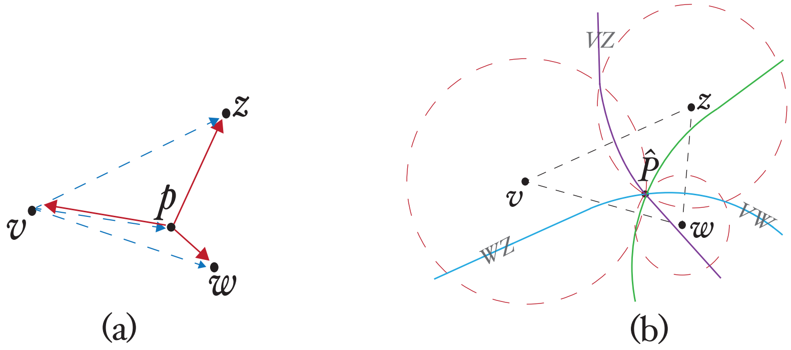


Figure 2.4: SIMO Hyperbolic Multilateration

only verifiers (called *witnesses*) each timestamp v's challenge and p's response to determine their respective *interarrival times*, say

$$A_w(v,p) = C(e_p^w) - C(e_v^w) \tag{2.8}$$

with respect to witness w. Since all verifiers know each other's exact positions, w can easily find the running time for a $v \to p \to w$ challenge-response *relay* through p's actual location:

$$T_p(v,w) = A_w(v,p) + D(v,w) \equiv D(v,p) + \Delta_P + D(p,w) \tag{2.9}$$

Since $D(v,p) + \Delta_P$ appears in the running times reported by every verifier, those terms must vanish when we form the *difference* in reported running times across pairs of verifiers, so it is impossible for p to manipulate the outcome by changing Δ_P. Indeed, the constraint

$$T_p(v,v) - T_p(v,w) \equiv D(p,v) - D(p,w) \tag{2.10}$$

is equivalent to Eq. (2.7) for general TDoA multilateration protocols. However, a key advantage of Eq. (2.10) is that the verifiers only report *timestamp differences* to one another, rather than actual timestamp values. Thus, it is sufficient for their individual clocks to be *syntonized* (i.e., running at the same rate with an unknown offset) rather than completely synchronized.[2]

[2] Moreover, v's physical radio has a crystal oscillator to control the output rate of its transmitter, while the physical radio for every witness $w \in V$ has a timing recovery mechanism to match the speed of its receiver to the exact data rate of the incoming data stream. Thus, if every verifier uses a clock derived from its own crystal oscillator for timestamping packets, then all witnesses can (re)calibrate their individual clocks in real time, during each challenge-response dialogue, from the recovered symbol rate to its own crystal oscillator. Thus, every witness w can report the value of $T_p(v,w)$ in units derived from the speed of v's transmit oscillator [11].

Localization With Hidden Base Stations

Capkun and Srivastava proposed a hyperbolic multilateration protocol for infrastructure-centric systems [3]. In their protocol, a single "public" base station sends a single challenge to prover, while the arrival time for prover's response is recorded by multiple covert verifiers. The position of the prover is then calculated by solving a least squares problem with the recorded arrival times as inputs.

It is interesting to note that both SIMO protocols – Localization with Witnesses and Localization with Hidden Base Stations, prevent malicious behavior by leveraging the prover's lack of knowledge of a subset of the participating verifiers. The passive witnesses in the former protocol serve the same purpose as the covert base stations in the latter. The difference is that Localization with Witnesses is designed for ad-hoc network settings, where an infrastructure of anchor nodes is absent, while the latter is specifically designed for networks that have an infrastructure where the positions of the base stations are static. In either case, the prover's partial knowledge of the number of verifiers that it responds to in each round, and no knowledge of the passive/hidden verifier locations, thwarts a wide range of security threats like collusion attacks, jamming and replay etc., because it is impossible to launch these attacks without complete and accurate knowledge of verifier locations.

2.3.3 Multiple Input Multiple Output (MIMO) Multilateration

A recently proposed multilateration protocol [5] requires the prover to respond to multiple simultaneous inputs (challenges), one from each participating verifier. The prover's response is also monitored in parallel by each verifier, and hence this method is named *Multiple Input Multiple Output (MIMO)*. The MIMO multilateration protocol proposed in [5] extends Capkun and Hubaux's VM such that in each round of the protocol, the prover simultaneously executes challenge-response echoes with each verifier, instead of sequentially executing a challenge-response echo with each verifier, one-at-a-time. Unlike SIMO multilateration, none of the verifiers time challenge-response relays.

The verifiers begin by communicating amongst themselves over a secure channel to generate an N-part challenge and agree on a common arrival time, τ, when all parts of the challenge must reach $\hat{P}$. Clearly the requirement $C(e_v^p) = C(e_w^p) = \cdots \equiv \tau$ can be satisfied by $C(e_v^v) = \tau - D(v, \hat{P})$ for all $v \in V$. By design, p cannot generate its response until it has received all N parts from the challenge, say by forming the bitwise XOR across all N parts of the challenge and a pre-arranged section of a shared secret key. Thus, if an honest prover claims its true location, then p will broadcast its response to all verifiers at time $\tau + \Delta_P$. Since each verifier $v \in V$ hears the response at time $C(e_p^v) = \tau + \Delta_P + D(p, v)$, its measurement of the running time for the protocol satisfies $T_p(v, v) \equiv C(e_p^v) - C(e_v^v) = \hat{T}(v, v)$ as required. Conversely, if a dishonest prover is at a different location, then because of the *point-in-triangle test*, there must be at least one verifier, say z, for which $D(z, p) > D(z, \hat{P})$ and p must fail the δ-test administered by z.

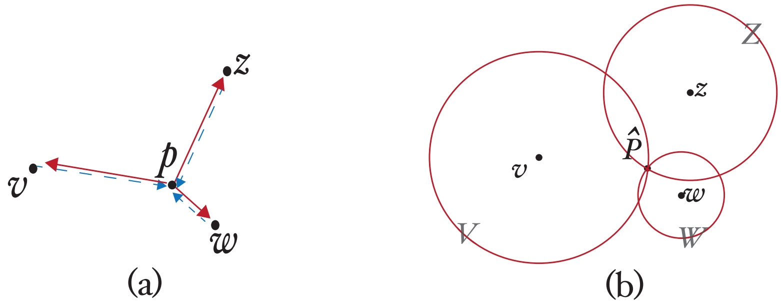


Figure 2.5: MIMO Multilateration

The concept of *self-jamming* by the verifiers to block an out-of-position prover from receiving any parts from the challenge was also introduced in [5]. In this case, every verifier jams the RF channel(s) assigned to the other verifiers at all times *except* when it is transmitting its own part of the challenge. Thus, a malicious prover p will be blocked from receiving v's challenge by the jamming signal originating from w unless both challenges arrive at p simultaneously, i.e., p is somewhere on the hyperbola VW. Moreover, even in that case, both challenges from v and w will still be jammed by another verifier, say z, unless p's location is also on hyperbolas VZ and WZ — which is only true if p is at $\hat{P}$.

2.4 Observations

In this chapter, we introduced the system model and common assumptions under which time-based secure localization protocols are designed. We showed that all time-based secure localization protocols incorporate variants of the challenge-response message structure originally introduced in the naive distance bounding protocol. Instead of simply verifying whether a prover is present within a region of interest, determining the prover's specific location requires multilateration. We categorized multilateration protocols into three classes: SISO, SIMO and MIMO based on the structure and patterns of the challenge-response message exchanges. SISO multilateration is vulnerable to distance fraud since the prover can manipulate the response time Δ_P individually for each verifier across different challenge-response rounds. The SISO protocol Verifiable Multilateration (VM) introduced the point-in-triangle and δ tests to address this threat. A better way to prevent distance fraud is to incorporate simultaneity in multilateration. This can be done either by having multiple verifiers simultaneously monitor challenge-response relays (SIMO) or have all the verifiers send their challenges simultaneously (MIMO). In the chapters following this, we will answer the question – how do these three categories of multilateration protocols compare in terms of number of message exchanges required, security, and the expected positioning accuracy when they are implemented on real systems ?

In section 3.1 of their work, Chiang et al. [5] claim that "multilateration must be performed *simultaneously* by modifying the distance bounding protocol so that the prover responds to simultaneous challenges from each verifier". However, we note that SIMO hyperbolic multilateration protocols based on Time Difference of Arrival (TDoA) methods *already* support simultaneous multilateration by using multiple receivers, *without* using simultaneous challenges. On the other hand, MIMO achieves simultaneity in multilateration with simultaneous challenges, but uses a greater number of messages and requires more hardware resources, as compared to SIMO protocols, to securely localize the prover under similar conditions. Therefore, we ask if simultaneous challenges, like in the MIMO protocol, are really required for secure localization?

In Theorem 3.2, of their work [5], it was also stated that their protocol is *optimal* in the sense of achieving the "*maximal security that can be provided by any protocols based solely on time-of-flight* [... because ...] 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". However, we note that rejecting $\hat{P}$ is only helpful if the prover is actually dishonest; otherwise we are simply increasing the false positive rate and lowering the effectiveness of the protocol. Therefore, we ask how much uncertainty will be reported by a protocol using simultaneous challenges in comparison to those using a single challenge for simultaneous multilateration in the *absence* of any malicious behavior.

Chapter 3

A New SIMO Protocol – Elliptical Multilateration

The most recent claim about securing localization through multilateration is that executing simultaneous multilateration with simultaneous challenges (MIMO multilateration) provides maximal security that can be provided by any time-based secure localization protocol. We claim that simultaneous multilateration with multiple passive receivers (SIMO multilateration) is a better method for secure localization, and simultaneous challenges are not necessary. In fact, a SIMO design for simultaneous multilateration not only guarantees no worse security than a MIMO design for the single prover case examined in this thesis, but is also inherently more accurate when implemented on real systems.

3.1 Elliptical Multilateration – Protocol Description

To examine whether simultaneous challenges are really necessary, we propose a new Time-of-Arrival (ToA) simultaneous multilateration protocol called *Elliptical Multilateration (EM)*. In our protocol, a single challenge is sent in each round of challenge-response, but simultaneity in multilateration is achieved by using multiple verifiers that act only as passive listeners, and simultaneously time the prover's response. The verifiers localize the prover by mathematically combining elliptical constraints generated on the prover's location, and hence the name "Elliptical Multilateration".

The prover initiates the protocol by claiming a location $\hat{P}$. A complete execution of EM consists of a few rounds of challenge and response. In each round, the verifier that sends the single challenge, is picked at random (without replacement) from the set of participating verifiers; this verifier will be termed as the lead verifier $\hat{v}$ for that round. The other verifiers act as passive listeners. When $\hat{v}$ executes a challenge-response echo $\hat{v} \to p \to \hat{v}$, a passive verifier, say w, simultaneously monitors the challenge-response relay $\hat{v} \to p \to w$. The arrival of the challenge from $\hat{v}$ is timestamped by w, followed by the arrival of the response from p. w records the interarrival time $A_w(\hat{v}, p)$ similar to Eq.2.8. w can then calculate the running time of the relay $\hat{v} \to p \to w$ though p's true location similar to Eq.2.9.

$$T_p(\hat{v}, w) = D(\hat{v}, p) + \Delta p + D(p, w) \tag{3.1}$$

w also calculates the expected running time of the relay $\hat{v} \to \hat{P} \to w$ through the claimed location $\hat{P}$:

$$\hat{T}(\hat{v}, w) = D(\hat{v}, \hat{P}) + \Delta p + D(\hat{P}, w) \tag{3.2}$$

If p's response time is no less than Δ_P, then $T_p(v, w) \leq \hat{T}(v, w)$ implies that $D(v, p) + D(p, w) \leq D(v, \hat{P}) + D(\hat{P}, w)$. In this case, w can conclude that p's actual location is on the elliptical disk VW: where the sum of the distances from any point to its foci at v and w is at most $D(v, \hat{P}) + D(w, \hat{P})$ (see Fig. 3.1(a)). Notice that every intersection point for circles with centers at v and w is also in the ellipse VW. If a third passive verifier, z, also monitors the same challenge-response dialogue, it will constrain p's location to another elliptical disk VZ (Fig. 3.1(b)) . By merging their individual results, the three verifiers can constrain p's location to the mutual intersection of circular disk centered at v and elliptical disks VW and VZ – which is the shaded convex region resembling a "propeller blade" that includes both $\hat{P}$ and verifier v as shown in Fig. 3.1(c). In each subsequent round, p's location can be constrained to a different convex region. The prover's location can thus be determined by the mutual intersection of all such regions as shown in Fig. 3.1(d).

Similar to VM and MIMO multilateration, EM also uses the point-in-triangle and δ tests to prevent distance fraud. Therefore, the prover location must first satisfy the point-in-triangle test before the δ test can be administered. The prover's location is accepted *iff* it successfully passes both tests. Similar to other protocols that use these two tests, prover locations that are outside of the convex hull formed by the verifiers cannot be determined/verified by our protocol. [1]

[1] Prover locations outside the convex hull formed by the verifiers can be handled by Time-Difference-of-Arrival (TDoA) based hyperbolic multilateration, which will not be discussed in this chapter. Although TDoA multilateration protocols have this advantage, they have other drawbacks, which will be discussed later.

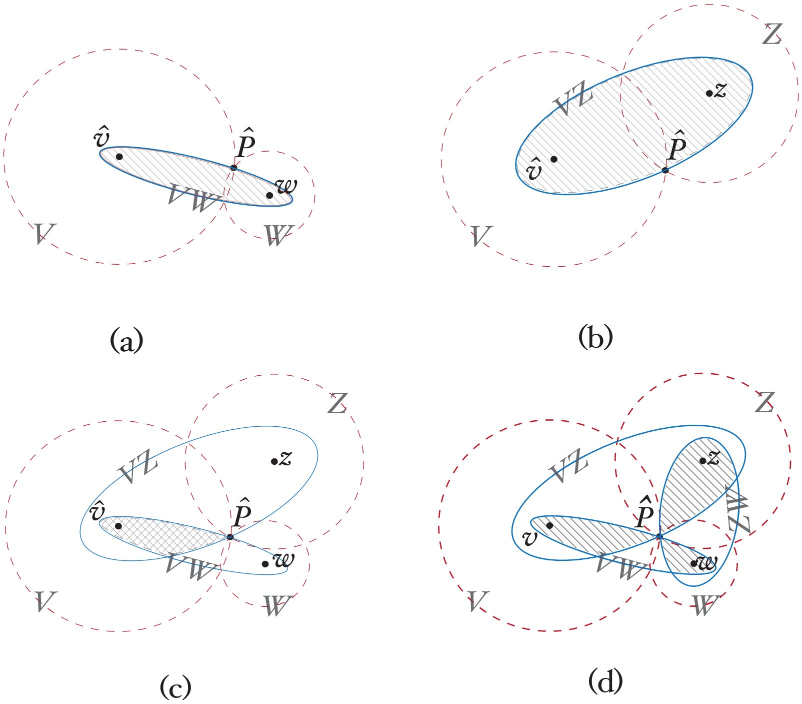


Figure 3.1: (a) elliptical constraint formed by two verifiers– the lead verifier and a passive verifier (b) elliptical constraint formed with another passive verifier in the same challenge-response round (c) in a single round of challenge response, the prover can be constrained to a “propeller blade” shaped shaded convex region (d) the three constraint regions generated by different lead verifiers

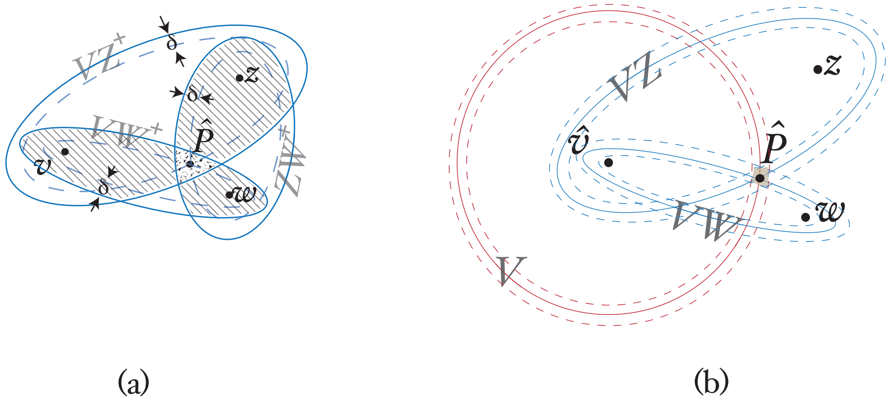
3.2 Security Analysis for Elliptical Multilateration

For the security analysis, we assume that the location $\hat{P}$ claimed by the prover may be incorrect. Although the prover cannot lower its response time below Δ_P, we assume that p is smart enough to delay its response by a suitable amount, to match the time the response would have taken to reach any verifier $x \in V$, if the prover were truly present at $\hat{P}$.

In section 2.3.1, we showed that the one-sided δ test in conjunction with the point-in-triangle test, suffices to securely localize the prover with Verifiable Multilateration (VM). In a SISO multilateration protocol, the two-sided δ test has no additional value than the one-sided δ test. This is because it still takes at least three challenge-response rounds to complete the localization process, and it does not provide any additional security against timing manipulations since the prover responds to only a single verifier at a time. To compare the effectiveness of the δ-test in the case of EM, we will first discuss the one-sided δ test, and then the two-sided *delta*-test for EM. Administering the two-sided δ test in the case of EM decreases the odds that a malicious prover can successfully cheat. With multiple receivers monitoring each challenge-response dialog and administering the two-sided $\delta - test$, at most two rounds of challenge-response dialogs are required to expose even a resourceful prover that can send separately timed responses to individual verifiers.

3.2.1 Case 1: Verifiers Administer the One-Sided δ Test

In the one-sided δ test, the prover's response is rejected if $T_p(\hat{v}, w) - \hat{T}(\hat{v}, w) > \delta$, which means early responses are accepted and late responses are rejected. Therefore, $T_p(\hat{v}, w)$ can exceed the expected value $\hat{T}(\hat{v}, w)$ by only a small amount, depending on the parameter δ. If the one-sided δ test is administered, we can add δ to every one-hop distance. The acceptable region around verifier v is now V^+. Similarly circles W and Z are also expanded to W^+ and Z^+, and the elliptical disks VZ, VW and ZW to VZ^+, VW^+ and ZW^+. The solution for p's location will now be the intersection of the expanded convex shaded regions. If δ is sufficiently small, this shift merely expands the solution to a small neighborhood around $\hat{P}$ with size $O(\delta)$. This is illustrated in Fig. 3.2(a). From the figure, we can also see that p can be located by the mutual intersection of two "propeller blade" shaped regions, which means that ***two challenges, sent sequentially by two different verifiers, are sufficient to locate*** p ***when the one-sided*** δ ***test is administered***.



(a) (b)

Figure 3.2: (a) constraints generated by the one sided δ test (b) constraints generated by the two-sided δ test when the prover is honest

3.2.2 Case 2: Verifiers Administer the Two-Sided δ Test

In the two-sided δ test, neither early nor late responses are accepted by the verifiers. Therefore, the prover's response is rejected if $|T_p(\hat{v}, w) - \hat{T}(\hat{v}, w)| > \delta$. If the prover is honest and sends a single response after the designated time Δ_P, it must be located at the intersection of the boundaries of the circular constraints generated in the timed echo with $\hat{v}$, and the elliptical constraints generated in the timed relays with the other verifiers. This is shown in Fig.3.2(b).

(i)Prover Possesses Independently Controlled Radios With Directional Antennas for Each Verifier

A dishonest prover, not present at $\hat{P}$, must delay its response to different verifiers by different amounts of time to pass the two-sided δ test with each of them. If the prover possesses multiple radios, each with a directional antenna targeting a specific verifier, it can time-shift the response separately for each verifier. Therefore if the prover is at any location such that it can add a delay to the response for each verifier and satisfy the timing requirements, it will pass the test with each of them.

If v sends the first challenge, then the prover can respond to all three verifiers v, z and w if it is within the "propeller blade" shaped shaded region containing v and $\hat{P}$ by individually delaying the response to each verifier by a suitable amount. For example, the response to z in this case must be delayed by a time equivalent to the distance $|(D(v,\hat{P}) + D(\hat{P},z)) - (D(v,p) + D(p,z))|$ to match the expected arrival time at z. Therefore, allowing the prover to possess independently controlled radios with directional antennas for each verifier, reduces the effectiveness of the two-sided δ-test to that of the one-sided δ-test.

However, if a different verifier, say w sends the second challenge, the prover cannot cheat the verifiers in the same way unless it is located within the "propeller blade" shaped shaded region that contains w and $\hat{P}$, and thus fails verification in the second round. Theerfore, ***even if the prover can separately time shift its responses for different verifiers, it takes EM at most two challenges, sent sequentially by two different verifiers, to localize*** p.

(ii)Prover is Restricted to a Single Radio with an Omnidirectional Antenna

Restricting the prover to a single radio and omnidirectional antenna means p can send its response to all the verifiers only over a single transmission. As in the previous section, let us assume that verifier v sends the first challenge. Given that the claimed location $\hat{P}$ satisfies the point-in-triangle test with the set of three verifiers $\{v, w, z\}$, we can show that even if the prover is "strategically" located within the shaded region containing v and $\hat{P}$ as in section 3.2.2, it cannot respond on time to all the verifiers by delaying a single response transmission. In other words, if p is not truly located at $\hat{P}$, there must be at least

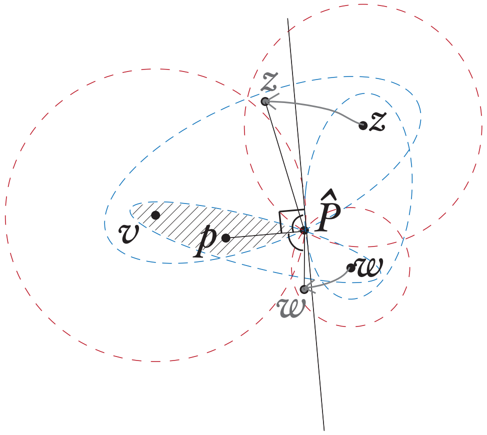


Figure 3.3: If a dishonest prover is located at p instead of the claimed location, the prover can respond on time to all three verifiers *iff* $\angle z\hat{P}p$ and $\angle w\hat{P}p$ are both less than 90°

one verifier, say $s \in \{v, w, z\}$ such that its distance from the true location p is greater than its distance from the claimed location $\hat{P}$, because of which the prover will fail verification because its response at s will arrive too late. To prove this, we consider the situation in figure 3.3, where p is the true location of the prover and $\hat{P}$ its the claimed location. If the prover's response can reach the verifier w on time then

$$D(p, w) \leq D(\hat{P}, w) \Rightarrow \angle p\hat{P}w \leq 90^\circ \tag{3.3}$$

Similarly, if the prover's response reaches z on time, we must also have

$$D(p, z) \leq D(\hat{P}, z) \Rightarrow \angle p\hat{P}z \leq 90^\circ \tag{3.4}$$

From (3.3) and (3.4) above, we have

$$\angle w\hat{P}z \leq 180^\circ \tag{3.5}$$

Therefore $\hat{P}$ must either lie on line wz or it must be on the side of the line opposite to the side that contains verifier v. Since this contradicts our previous assumption that $\hat{P}$ satisfies the point-in-triangle test with the set of three verifiers $\{v, w, z\}$, we proved that the prover cannot pass verification with three or more verifiers if it is located at p, different from the claimed location, if it possess a single radio with an omnidirectional antenna.

3.3 Comparison With Other Protocols

From the description of the elliptical multilateration, we observe that that EM is similar to VM and MIMO multilateration in many aspects. All of these three protocols are Time-of-Arrival (ToA) multilateration protocols. Each protocol requires the prover to satisfy the point-in-triangle test – in general, the claimed location must be within the convex hull formed by the participating verifiers. In each protocol, the geometrical constraints imposed by the verifiers after the challenge-response rounds are closed convex objects (i.e circles and ellipses) and each of them applies the δ test to verify if the prover is cheating. Due to these similarities between our SIMO EM, SISO VM and MIMO multilateration, we compare them in the following cases:

3.3.1 Honest Prover, $N (N \geq 3)$ Verifiers

An honest prover adheres to the protocol specifications and does not attempt to cheat the verifiers even if it may possess the hardware resources (multiple radios with directional antennas) that it can use to send separate responses to each verifier.

In this case, VM requires N rounds (one per verifier) to complete the localization process. During the execution of VM, a total of N challenges and N responses will be generated. EM requires two rounds irrespective of the number of verifiers, since the prover can always be localized by the mutual intersection of any two “propeller blade” shaped convex constraints generated on its location. During the localization process, EM generates two challenges and two responses. MIMO multilateration requires a single round to localize the prover by combining the N circular constraints generated by all the N verifiers. MIMO generates a total of N challenges and a single response.

If the number of verifiers N is large, the mutually trusted verifiers in VM and MIMO could agree among themselves to select a subset of (three or more) verifiers to run the protocol. Note that they could select the participating subset in advance rather than running an online algorithm, where they would have to share information after every round of challenge-response to decide if they could stop or need to continue protocol execution with additional rounds. When VM and MIMO both allow N verifiers to participate, then the total number of messages generated in MIMO is $N + 1$ over a single round, which is almost half of the $2N$ messages generated over N rounds in VM.

3.3.2 Malicious Prover with Multiple Independently-Controlled Radios

If the prover is malicious and possesses multiple-independently controlled radios, each with a directional antenna, it can send individually targeted responses to the verifiers. The two-sided δ test then has no more value than the one-sided δ test in this case. The point-in-triangle test requires that the prover’s claimed location be within the triangle formed by any set of three participating verifiers. If this test is satisfied, and the prover is not present at the claimed location, it must be at a larger distance than the expected distance from at least one of the verifiers, say z. Since p cannot lower its response delay below Δ_P, its response will reach late when z late and the distance fraud will be exposed.

In SISO VM, N challenge-response rounds are required to complete localization. The prover's cheating is caught in the round where z, the verifier whose distance to p is larger than the claimed distance, sends the challenge. This verifier may send its challenge in the first, second or third round depending on chance. Therefore, VM requires at least one and at most N rounds of challenge-response to expose distance fraud.

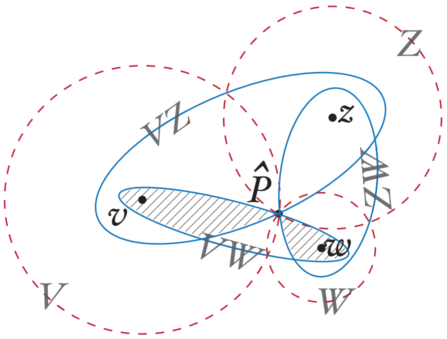


Figure 3.4: EM can expose a resourceful prover's fraud because it cannot be present in both the disjoint shaded regions at the same time

For SIMO EM (see Fig. 3.4), if v sends the challenge, and p is anywhere within the convex shaded region containing v, it can delay the responses appropriately to the three verifiers to satisfy each of them. However, when a second verifier, say w, sends the challenge, it has to be within the shaded convex region containing w to be able to pass verification with all three verifiers. The two shaded areas are disjoint except at the claimed location $\hat{P}$. Therefore, SIMO EM requires at least one, and at most two challenge-response rounds to expose p's fraud, depending on which verifier (v or w) sends the first challenge.

In MIMO multilateration, similar to SISO VM, the prover cannot claim a location other than its true location if it satisfies the point-in-triangle test because it must be further away from at least one of the verifiers, say z, than the expected distance. The difference is that the single shared response that is generated for all the verifiers will be delayed until the challenge from z arrives, therefore all the verifiers will detect cheating on the part of the prover, in a single round of challenge-response.

3.3.3 Malicious Prover with Single Radio, Omnidirectional Antenna

In the case where a malicious prover is restricted to a single radio with an omnidirectional antenna, the two-sided δ test has more power than the one-sided δ test in the case of SIMO EM.

SISO VM requires N challenge-response rounds whether or not the prover is restricted to a single radio with an omnidirectional antenna. Since the challenge-response rounds are always executed between the prover and a single verifier at a time, possessing multiple independently controllable radios that can target the verifiers individually is of no extra benefit to the prover.

Having multiple passive verifiers monitoring the response simultaneously in the case of SIMO EM, makes the two-sided δ test more powerful than the one-sided δ test, when the prover is resource-constrained. In section 3.2.2, we showed that a malicious prover present at any location other than $\hat{P}$ cannot respond to three or more verifiers on time, over a single response transmission, when the two-sided δ test is administered. Therefore, administering the two-sided δ test when the prover is resource constrained to a single radio with an omnidirectional antenna, reduces the number of rounds required (from two to one) in SIMO EM.

In the MIMO protocol, the prover must send its response over a single transmission if it is resource constrained. Since all the participating verifiers use this response for localization, the number of challenge-response rounds to detect distance fraud remains the same (one).

3.4 Summary

In conclusion, we summarize the discussions from this chapter in the tables below:

In Each Round of Challenge-Response, Number of	Protocol		
	SISO VM	**SIMO EM**	MIMO
Verifiers Sending Challenges	1	**1**	N
Verifiers Observing Response	1	**N**	N
Messages Generated	2	**2**	$N+1$

Table 3.1: Message Exchange Structure Across Different Multilateration Protocols

Case:	SISO VM	**SIMO EM**	MIMO
Honest Prover	N	**1**	1
Malicious Resourceful Prover	N	**2**	1
Malicious Resource-Constrained Prover	N	**1**	1

Table 3.2: Number of Challenge-Response Rounds Required to Complete Localization or Expose Distance Fraud

	SISO VM	**SIMO EM**	MIMO
Total Messages Exchanged	2N	**2 or 4**	N+1

Table 3.3: Total Messages Exchanged to Complete Localization or Expose Distance Fraud

Chapter 4

Precision Timing in Real Systems

Of the different methods that can be used to localize an entity in the localization protocol, time-based (ToA and TDoA) localization protocols are preferred to those based on other techniques like AoA and RSSI in adversarial settings. This is because time-based protocols exhibit better security properties [4] than other protocols.

Recall that while discussing the distance fraud attack, we assumed that the prover cannot lower its response delay below the known value Δ_P. However, if the prover can convince the verifiers that the propagation time of the response is greater than the true propagation time, it can still claim to be further away than it truly is, without lowering its response delay below Δ_P. The prover can achieve this by using a medium of propagation faster than the default medium used in the protocol. This would also make the localization protocol vulnerable to wormhole, replay and man-in-the-middle attacks. To prevent these, all message exchanges must be executed over radio frequency (RF) – the fastest known medium for wireless communication.

Using RF as the medium for message propagation can only increase security if timestamps can be generated with very high precision. Since the speed of RF is $3 \times 10^8 m/s$. an error of $3.3ns$ in timestamping equates to an error of $1m$ in distance measurement. Therefore, to achieve positioning accuracy in the order of a few meters, the entities participating in ToA multilateration must measure propagation times of the messages in the order of a nanosecond. In this chapter, we discuss the reasons why this is extremely hard to do in real systems.

To the best of our knowledge, there are no experimental studies in which any of the time-based *secure* localization protocols discussed in this thesis have been implemented on real systems. However, we can get important insights into sources of measurement error by studying work in the domain of precision clock synchronization, specifically the IEEE 1588 Precision Time Protocol (1588 PTP), which is also a time-based protocol and has a similar precision timing requirement. First, we will discuss the similarities between a basic challenge-response round of a localization protocol and the second phase of message exchanges in 1588 PTP. Then, we cite results from the various implementations of 1588 PTP on real hardware over standard networking protocols.

4.1 Notation

Recall that in section 2.1, a timestamp for event e_x^y at the observer entity o's clock was denoted as $C_o(.)$. In our discussions of time-based localization protocols, we dropped the subscript o and denoted timestamps simply as $C(.)$ for simplicity because of the assumption that all the verifier clocks are synchronized to a common time reference. In the following discussions related to the 1588 PTP, this simplification does not hold since the two entities participating in the time synchronization protocol timestamp with their local clocks, which are initially not synchronized. The other assumptions about the medium being isotropic and anechoic and the normalization between time and space continue to hold in the discussions that follow.

4.2 Precision Timing Requirement in Time-Based Localization Protocols

Let us first consider a basic challenge-response round of a time-based localization protocol. We use v to denote a participating verifier, and p to denote the prover. A single round of challenge-response in a multilateration protocol includes the following discrete events (Refer Fig.4.1(a)): (i) e_v^v, the verifier sends the challenge; (ii) e_v^p, the prover receives the challenge, then stores it and computes the response, (iii) e_p^p, the prover sends the response; and (iv) e_p^v, the verifier receives the response and then stores it. The distance between the verifier and the prover is then computed by v from the timestamps as follows

$$D(v,p) \equiv \tau_{vp} = (C(e_p^v) - C(e_v^v) - \Delta_P)/2 \tag{4.1}$$

where Δ_P is the response delay of the prover whose value is publicly known and τ_{vp} is the one-way propagation delay of a reference point within the message, between v and p.

4.3 Precision Timing Requirement in IEEE 1588 PTP

The 1588 PTP [8] is a protocol that has been designed and standardized for synchronizing the clocks of nodes in a local area network. 1588 PTP allows a “slave” entity,

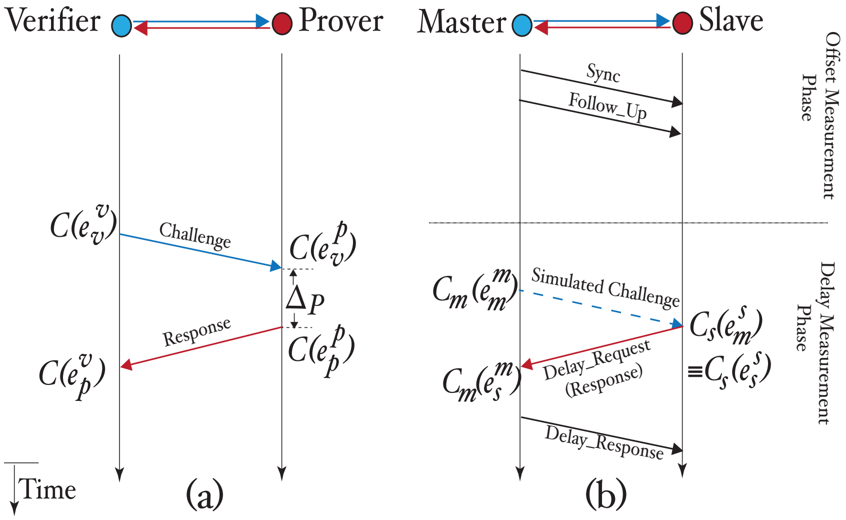


Figure 4.1: (a) a basic challenge-response round in a time-based secure localization protocol (b) message exchanges in the IEEE 1588 Precision Time Protocol

to synchronize its clock to a "master" entity (whose clock serves as the time reference). Fig 4.1(b) shows a space time diagram for the sequence of message exchanges between the"master" entity m and "slave" entity s. From the figure, we see that during the offset measurement phase, the first phase of PTP, the trusted "master" entity v sends periodic "Sync" and "Followup" messages to the slave s. The messages sent by the "master" are used by the slave to adjust its clock to satisfy $C_s(\cdot)||C_m(\cdot)$ with an offset lagging behind the master entity's clock by exactly $\tau_{ms} \equiv D(m, s)$, where $C_x(\cdot)||C_y(\cdot)$ denotes the case when the clocks of entities x and y are *syntonized* (i.e., they run at the same rate while maintaining some fixed offset, possibly not zero).

In the second phase of the protocol, the delay measurement phase, the slave entity s effectively *simulates* the basic challenge-response dialog from a time-based secure localization protocol in the second phase of the PTP. For this, consider the two events — e_s^s, when s sends a message corresponding to the "response" (Delay Request packet), and e_s^m, when the "response" reaches m. The "simulation" inserts an imaginary challenge ahead of the "response", defined by events e_m^m, when it left m and e_m^s, when it reached s. Because of first phase i.e the offset measurement phase, s knows that $C_s(e_m^s) = C_m(e_m^m)$ must hold. Moreover, because it is just a simulation, s sets $C_s(e_m^s) = C_s(e_s^s)$, and hence $\Delta_P \equiv 0$. Therefore, once m sends $C_m(e_s^m)$ as payload of the untimed Delay Response packet, the slave s (but *not* the master m) knows $C_m(e_m^m)$, Δ_P, $C_m(e_s^m)$, and can find τ_{ms} similar to Eq.(4.1).

The IEEE 1588 PTP needs precision timing to syntonize the master and slave entities' clocks. In a basic challenge-response dialog, the precision timing requirement arises from the need to adhere to the value of Δ_P. In either time-based protocol, *accuracy* is limited by the participants' abilities to measure event times. A detailed description of the similarities in message structure as well as timestamping support required by either protocol can be found in [11].

4.4 Difficulty of Controlling the Message Sending Time

A time-based protocol itself runs in the application layer of the network protocol stack. However, the message arrivals and departures occur in the network transceiver card, at the interface between the PHY hardware and the air medium (air-transceiver interface). Any kind of communication data experiences non-deterministic delays in passing through the software (network protocol stack) and hardware of an entity in the protocol before the actual transmission. Sending a message at the exact time decided by the protocol is not feasible due to delays across asynchronous interfaces – from host application to operating system to network interface controller and even to RF transceiver. A major contribution to these delays are OS scheduling delays in the higher layers of the network protocol stack.

In timing measurements conducted by Pasztor and Veitch [12], a test packet was sent from a sender node to a receiver node over a network. The monitoring components used were GPS synchronized measurement cards capable of capturing $100ns$ resolution timestamps. In one of the experiments they measured the difference in the targeted and measured packet inter-departure times. When the sender was running real-time Linux with the **send()** process as the only active user application, the maximum difference measured was $0.55ms$. However, when the sender runs Linux even with minor user activity, the maximum difference is much larger at $180ms$. According to [12], the real-time Linux sender almost eliminates scheduling error, therefore we can attribute the jump in the maximum difference to scheduling delays along the protocol stack .

Since these delays are unpredictable and randomly varying, it is not possible to estimate and subtract a fixed value to account for these delays. Techniques can be used to minimize the effect of these non-deterministic delays, for example, IEEE 1588 PTP uses "followup messages" (Fig. 4.1). When a beacon packet (Sync) is sent, a timestamp is captured at a lower layer closer to the actual time of departure, and sent to the application layer as feedback. The followup message contains this captured timestamp in the payload, which helps the "slave" entity to minimize the error introduced due to the discrepancy in the intended and actual sending times of the beacon packet.

4.5 Difficulty in Timestamping Message Arrivals and Departures Accurately

In addition to the difficulty in sending a message at the intended time, it is also difficult to capture an accurate timestamp for the arrival or departure of a message. In table 4.5, we tabulate the results from different implementations of 1588 PTP. We observe that the possibility of recording accurate timestamps increases only as we move the timestamping point down the protocol stack into the network interface card, and to the analog to digital sampling point, close to the “transceiver-medium interface”.

The DP83640 Precision PHYter [10], is an Ethernet transceiver specially designed to support the IEEE 1588 PTP for real-time industrial applications. In [7], it has been demonstrated that two entities can be time synchronized to under $10ns$ over a point-to-point connection, and to sub-nanosecond accuracy when the “Synchronous Ethernet ” mode enabled, using [10]. This commercially available transceiver contains a local PTP clock operating at $250MHz$, programmable to frequencies obtained by integral division of the base clock, and a counter which is incremented every $8ns$. The transceiver [10] is also capable of parsing the packets on-the-go, and triggering timestamps at the A/D sampling stage within the transceiver. These timestamps are then inserted into the payload of the packet

Work by	**Medium**	**Timestamping Point**	**Average Offset**
Kannisto et al. [9]	Ethernet	Device Driver	$\approx 1.8\mu s$
Kannisto et al. [9]	802.11b	Device Driver	$\approx 0.66\mu s$
Cooklev et al. [6]	802.11b	PHY-MAC interface	$\approx 0.2\mu s$
Kannisto et al. [9]	802.11b	PHY-MAC interface	$\approx 1.1ns$
D. Miller [7]	Ethernet	A/D Sampling and Reference Detection	$< 1ns$

Table 4.1: Comparing Different Implementations of IEEE 1588 PTP

itself before transmission of the packet onto the medium or before sending the contents to a higher layer after reception, so that a “followup” message is not required.

In the measurement setup, synchronization accuracy was measured with an oscilloscope to compare the delay between the output signal from the “master” entity and the corresponding synchronized output signal from the “slave” entity. The accuracy achieved has been cited in the last row of table 4.5.

4.6 Timestamps can be Processed Offline

Although secure localization protocols require very accurate timestamps, there is no requirement for real-time processing and the timestamps can be reported to the application after a (reasonably) small delay. This observation is important because it allows for designing protocols with a precision timing requirement in a way that only the timestamping functionality needs to be implemented close to the “air-PHY” interface.

Examples demonstrating this flexibility in design are the Aeroscout System [1] and that used in [17], where raw A/D time series collected upon the arrival of a message, at different entities, are sent to to a central server that processes them for offline timing alignment via cross correlation. Such an arrangement where a central entity performs complex timing and signal processing functions may not be feasible in non-infrastructure centric settings. An example of an alternative method to capture accurate timestamps in those cases is found in [11], where the authors have proposed a timestamping unit such that the timestamp is recorded within the transceiver hardware at the A/D sampling stage, but the values are buffered so that the application can read them later. This functionality requires only a few simple logic blocks – a clock generator, a timer and a few registers to buffer timestamps before they are passed on to the application layer, therefore it is simple to implement. The more complex features like mathematical algorithms can be implemented in the higher layers of the network protocol stack. Therefore, it suffices to implement only the delay-intolerant feature i.e the timestamping unit of the protocol within the transceiver hardware.

4.7 Conclusions

The discussions in this chapter lead to three important conclusions:

(i) It is not possible for a sending entity to precisely control the time of departure of a message and there exists an inevitable discrepancy between the intended and actual sending time of the packets.

(ii) As we move the timestamping point lower in the network protocol stack, i.e. start from the application layer and move closer to the PHY layer, the accuracy of measurements improves. This is because fewer non-deterministic delays creep into the measurements.

(iii) It suffices to move only the time-sensitive functionality i.e. the timestamping unit close to the "transceiver-medium" interface and implement the other parts of a time-based protocol in the higher layers or offline. This offers promise for designing systems that can capture very precise timestamps without adding too much complexity to the transceiver itself. From the implementation results tabulated, we can also observe that the required precision can be achieved by clocking the timestamping units with inexpensive crystal oscillators found in off-the-shelf hardware.

Chapter 5

Uncertainty in Measurements

In chapter 2, we classified multilateration protocols according to the structure of their challenge-response dialogs. In this chapter, we will analytically compare the effect of these structural differences on measurement error in practical realization of these protocols.

For any localization protocol, $\hat{T}(v, w)$ is the ideal expected value for the time elapsed between verifier v sending the challenge, and verifier w receiving the response, when the prover is truly present at location $\hat{P}$ and uses the agreed upon, publicly known response delay Δ_P. Conversely, $T_p(v, w)$ is the actual value obtained from timestamps recorded during the challenge-response dialog as measured by verifier w.

In [5], Chiang et al. define the *uncertainty*, U, as the difference between these two values. For example, when a single verifier v engages p in a SISO challenge-response echo protocol,then

$$U = \hat{T}(v, v) - T_p(v, v) \tag{5.1}$$

Chiang et al. [5] further assumed that *all* uncertainty is caused by malicious activity on the part of the prover, and hence a protocol that generates a larger value of U is always better (at rejecting the location claim of a malicious prover).

The δ test that is administered for all three classes of protocols is an application of statistical hypothesis testing, where the verifiers accept the null hypothesis (that the prover is **not** cheating) if $|U| \leq \delta$, and the alternative hypothesis (that the prover is cheating) is $|U| > \delta$. Unfortunately, not all of the uncertainty in a ToA multilateration protocol is caused by malicious behavior. As we will explain below, there are some unavoidable errors in detection and timestamping of message arrival and departure events. Therefore there is always a discrepancy between the actual times at which these events occur and the values of the corresponding timestamps. Therefore, these multilateration protocols might report a false positive (type I error) if $|U > \delta|$ even though the prover is honest, or a false negative (type II error) if $|U| \leq \delta$ when the prover is dishonest.

To show that all the uncertainty is not due to malicious behavior on the part of the prover, we analyze all three classes of protocols under the assumption that the prover is honest and claims its true location to a group of N verifiers.

Let the discrepancy between the time when a node, say x, intends to send a message to node y, and the actual time of transmission be σ_x^y. Such a discrepancy is also experienced in the time at which a node x actually receives a message from node y, and the timestamp it records for the event. We define ρ_x^y as the timestamping error at node x's receiver in timestamping a message that was sent by node y. The error in timestamping the arrival of a message is smaller than the error in timestamping a message transmission, therefore the values for ρ_x^y are expected to be smaller than those of σ_x^y. The error introduced due to these terms in the final distance estimates is significantly high and cannot be ignored.

5.1 Uncertainty in Time-of-Arrival (ToA) Multilateration

If v sends a message to p, and the discrepancies between sending and receiving times and the corresponding timestamps are taken into account, then the measured one-way propagation time from v to p will be

$$\begin{aligned} T_p(v,v) &= C(e_p^v) - C(e_v^v) \\ &= \sigma_v^p + D(v,p) + \rho_p^v + \Delta_P + \sigma_p^v + D(p,v) + \rho_v^p \end{aligned} \tag{5.2}$$

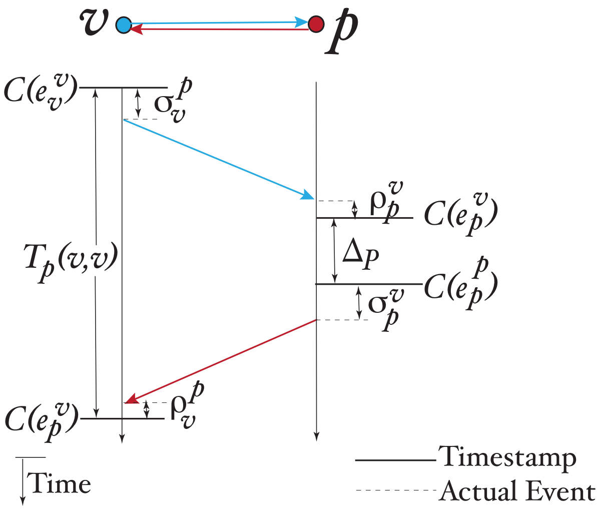


Figure 5.1: Space-time diagram showing the error terms in a single round of challenge-response between a verifier v and prover p

Fig. 5.2 shows a space-time diagram for execution of Capkun et al.'s SISO Verifiable Multilateration (VM). For the *VM* protocol, each verifier $v \in V$ executes a separate

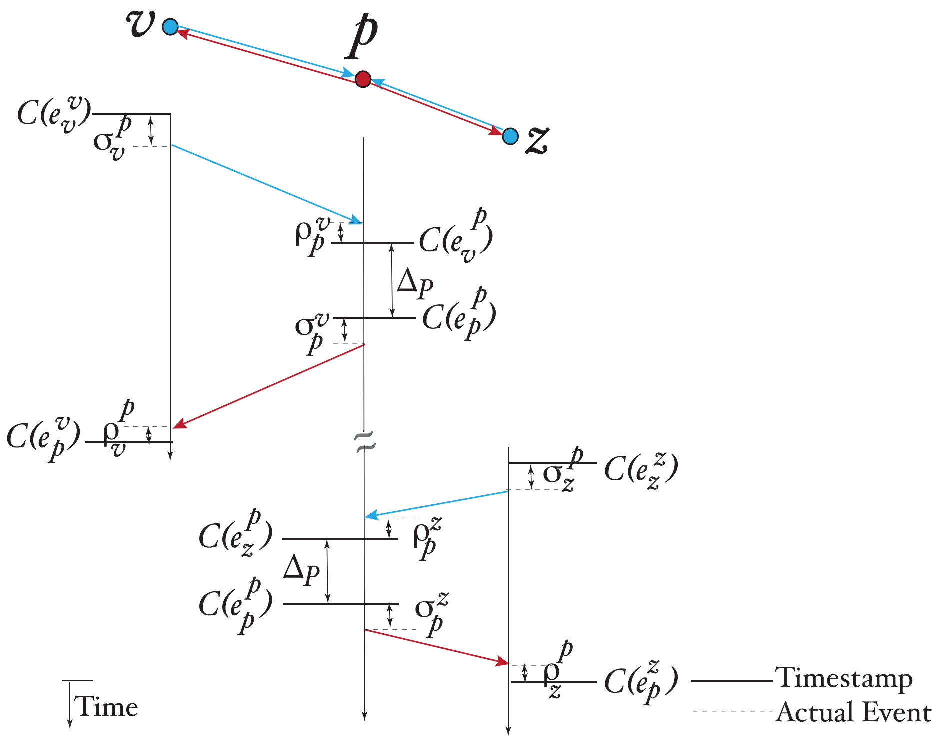


Figure 5.2: Space-time diagram showing the error terms in SISO multilateration with two verifiers

SISO challenge-response echo protocol with p, to obtain:

$$\begin{aligned}
U^{(v,v)} &\equiv T_p(v,v) - \hat{T}(v,v) \\
&= C(e_p^v) - C(e_v^v) - (D(v,p) + \Delta_P + D(p,v)) \\
&= \sigma_v^p + D(v,p) + \rho_p^v + \Delta_P + \sigma_p^v + D(p,v) + \rho_v^p - (D(v,p) + \Delta_P + D(p,v)) \\
&= \sigma_v^p + \rho_p^v + \sigma_p^v + \rho_v^p \qquad (5.3)
\end{aligned}$$

After all N rounds have been completed, the verifiers obtain

$$U_{VM} \equiv \max_{v \in V}\{U^{(v,v)}\} = \max_{v \in V}\{\sigma_v^p + \rho_p^v + \sigma_p^v + \rho_v^p\} \qquad (5.4)$$

Fig. 5.3 shows the space-time diagram for a single round of challenge-response of MIMO multilateration when discrepancies in intended and actual sending time during practical realization are taken into consideration. Compare this figure to Fig. 2.5 (a) to observe that the challenges arrive staggered in time at the prover, contrary to the assumption in the theoretical description of the protocol. The figure below also shows that the prover in Chiang et al.'s MIMO protocol cannot generate its response until it receives the *last challenge*, from verifier z say, at time $\tau + \sigma_z^p + \rho_p^z$. Since each verifier v uses the ideal transmission

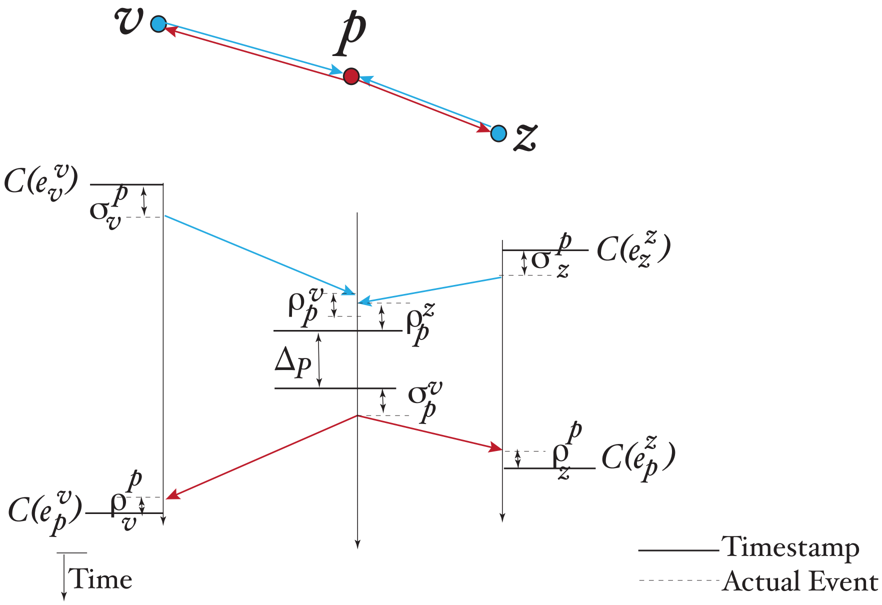


Figure 5.3: Space-time diagram showing the error terms in MIMO multilateration with two verifiers. Notice that the challenges from different verifiers do not reach the prover simultaneously.

time for its own challenge to mark the beginning of the time-of-flight calculation, we have

$$\begin{aligned}
U_{MIMO} &\equiv \max_{v\in V}\{T_p(v,v) - \hat{T}(v,v)\} \\
&= \max_{v\in V}\{C(e_p^v) - C(e_v^v) - (D(v,p) + \Delta_P + D(p,v))\} \\
&= \max_{v\in V}\{\sigma_v^p + D(v,p) + \rho_p^v + (\sigma_z^p + \rho_p^z - \sigma_v^p - \rho_p^v) + \Delta_P \\
&\quad +\sigma_p^v + D(p,v) + \rho_v^p - (D(v,p) + \Delta_P + D(p,v))\} \\
&= \max_{z\in V}\{\sigma_z^p + \rho_p^z\} + \sigma_p^v + \max_{v\in V}\{\rho_v^p\} \\
&\geq U_{VM} \qquad (5.5)
\end{aligned}$$

Under SIMO multilateration, the uncertainty measured by a witness, say $w \neq v$, is given by

$$U^{(v,w)} \equiv T_p(v,w) - \hat{T}(v,w) \qquad (5.6)$$

$\hat{T}(v,w)$ should be the sum of, $A_w(v,p)$, the true value of interarrival time of the challenge from v and the response from p, and $D(v,w)$, similar to Eq. **??**. Therefore,

$$\begin{aligned}
\hat{T}(v,w) &= A_w(v,p) + D(v,w) \\
&= (D(v,p) + \Delta_P + D(p,w) - D(v,w)) + D(v,w) \qquad (5.7)
\end{aligned}$$

$T_p(v,w)$ is calculated from the measured values of $A_w(v,p)$ and the known value of $D(v,p)$

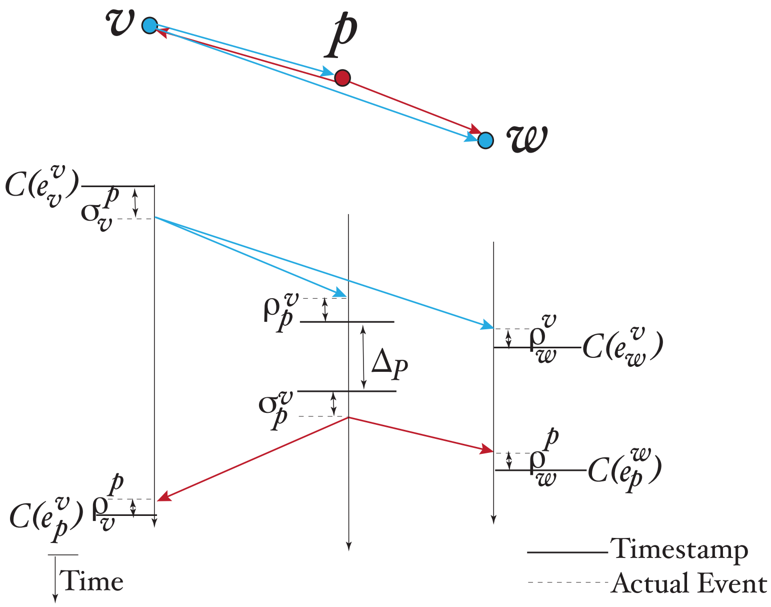


Figure 5.4: Space-time diagram showing the error added in during the calculation of the interarrival time $A_w(v, p)$ and its the effect on the uncertainty

$$
\begin{aligned}
T_p(v, w) &= (C(e_p^w) - C(c_v^v)) + D(v, w) \\
&= (\sigma_v^p + D(v, p) + \rho_p^v + \Delta_P + \sigma_p^v + D(p, w) + \rho_w^p \\
&\quad -\sigma_v^p - D(v, w) - \rho_w^v) + D(v, w) \qquad (5.8)
\end{aligned}
$$

Substituting Eqs. 5.7 and 5.8 in Eq. 5.6, we get

$$
U^{(v,w)} \equiv T_p(v, w) - \hat{T}(v, w) = -\rho_w^v + \rho_p^v + \sigma_p^v + \rho_w^p \qquad (5.9)
$$

Combining Eqs. (5.3) and (5.9) gives

$$\begin{aligned} U_{EM} &= \max_{w \in V}\{U^{(v,w)}\} \\ &= \rho_p^v + \sigma_p^v + \max\{\sigma_v^p + \rho_v^p, \max_{w \in V \neq v}\{\rho_w^p - \rho_w^v\}\} \\ &\leq U_{MIMO} \end{aligned} \tag{5.10}$$

From the equations above, we can conclude that even when the prover is honest and there is no malicious activity, the uncertainty in MIMO multilateration is higher than both SISO and SIMO multilateration. Therefore, under the similar conditions, a MIMO protocol must choose a larger value of δ as compared to a SISO or SIMO protocol to avoid rejecting a true location claim, thus the chances of reporting a false negative are higher in the case of a MIMO protocol as compared to the SISO or SIMO variants.

It is also worth noting that in a MIMO protocol, if one of the verifiers, say w, sends its challenge at an incorrect start time ($\neq \tau - D(w,p)$), all other verifiers will calculate a large value for U, and the prover's location might not be accepted even in the absence of malicious activity. Conversely, under SIMO multilateration, the result of localization is not effected by the time at which the challenge is sent by the single verifier. All other verifiers will base their calculations based on the time the challenge was sent, and not a scheduled start time.

Therefore MIMO multilateration, which uses more messages than the SIMO variant, not only increases the complexity of the protocol without enhancing the security in localization, but also is inherently less accurate.

Chapter 6

Conclusions

In this thesis, we classified known time-based localization protocols and highlighted the underlying differences in the structure and pattern of the message exchanges. Our first contribution was to introduce a simplified "SIMO" protocol, in which we reduced the number of inputs to a single challenge sent by a randomly-chosen verifier. Unlike earlier multiple-receiver localization protocols, which used a Time-Difference-of-Arrival formulation to localize p to the intersection of hyperbolas, our SIMO multilateration protocol follows a more conventional Time-of-Arrival formulation. However, because verifier w must now time a message *relay* from $v \to p \to w$, rather than a message *echo* $v \to p \to v$, the result of executing our protocol is to localize p to the intersection of *ellipses*, rather than circles surrounding the verifiers.

We also examined the performance of the *secure multilateration scheme* proposed by Chiang et al. [5] under the distance fraud attack with a single prover. We found that simultaneous challenges are unnecessary for secure multilateration. Rather simultaneous challenges as in MIMO multilateration require additional resources, while not providing any additional security than comparable SIMO protocols, and are also inherently less accurate in localizing the prover. We showed that the effect of measurement errors is unavoidable, significant and varies across different protocols, depending on their design. Amongst Time-of-Arrival protocols, our EM protocol performs best in terms of the number of message exchanges required, security in the presence of a resourceful prover, and expected accuracy when implemented on real systems. Therefore we advocate a SIMO design for Time-of-Arrival localization protocols.

Comparing both SIMO protocols – our new ToA Elliptical Multilateration (EM), and the traditional TDoA Hyperbolic Multilateration (HM), we found that each has its own merits and demerits. ToA EM is limited to localizing provers within the convex hull formed by the verifiers. TDoA HM doesn't have this limitation. It can be used to localize provers even outside of the convex hull formed by participating verifiers. Both protocols are equally effective against distance fraud launched by a prover restricted to a single radio and omnidirectional antenna . However, when the prover is more resourceful and may possess multiple radios equipped with directional antennas, TDoA HM may fail to catch cheating on the part of the prover. ToA EM scores in this case, because it can detect cheating even when the prover is resourceful in terms of hardware.

Many time-based localization protocols have been studied extensively, however none of the previous works have focused on the effect of measurement errors that are introduced when such protocols are implemented in real systems. The emphasis has always been on the structure of the message exchanges, or on the cryptographic aspects of these protocols. To the best of our knowledge, this thesis is the first to discuss the sources of measurement errors when these protocols are implemented on real systems. Accounting for issues that effect the accuracy while designing new time-based localization protocols can lead to better protocols, which will be superior to the existing ones not only in theory, but also in implementation.

Bibliography

[1] AeroScout. http://www.aeroscout.com, retrieved April 2011.

[2] S. Brands and D. Chaum. Distance-bounding protocols. In *Advances in Cryptology – EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pages 344–359. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1994.

[3] S. Capkun, M. Cagali, and M. Srivastava. Secure Localization with Hidden and Mobile Base Stations. In *IEEE INFOCOM*, April 2006.

[4] S. Capkun and J. P. Hubaux. Secure Positioning of Wireless Devices with Application to Sensor Networks. In *IEEE INFOCOM*, March 2005.

[5] J. T. Chiang, J. J. Haas, and Y. Hu. Secure and Precise Location Verification Using Simultaneous Distance Bounding and Simulataneous Multilateration. In *second ACM Conference on Wireless Network Security*, March 2009.

[6] T. Cooklev, J. C. Eidson, and A. Pakdaman. An implementation of IEEE 1588 Over IEEE 802.11b for Synchronization of Wireless Local Area Network Nodes. *IEEE Trans. on Instrumentation and Measurement*, 56:1632–1639, 2007.

[7] DP83640 Synchronous Ethernet Mode: Achieveing Sub-Nanosecond Accuracy in PTP Applications. In *Application Note 1730*. National Semiconductor, September 2007.

[8] IEEE. *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. Number 1588-2008. Piscataway, NJ, Jan 2008.

[9] J. Kannisto, T. Vanhatupa, M. Hannikainen, and T.D Hamalainen. Software and Hardware Prototypes of the IEEE 1588 Precision Time Protocol on Wireless LAN. In *14th IEEE Workshop on Local and Metropolitan Area Networks*, September 2005.

[10] Precision PHYter. http://www.national.com/pf/DP/DP83630.html, retrieved April 2011.

[11] S. Parichha and M. Molle. Localization and Clock Synchronization Need Similar Hardware Support in Wireless LANs. In *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, September 2008.

[12] A. Pasztor and D. Veitch. High Precision Active Probing for Internet Measurement. In *INET, The Internet Society*, 2001.

[13] A. Saha and M. Molle. Localization with Witnesses. In *Proc. 1st International Conference on New Technologies, Mobility and Security (NTMS 2007)*, May 2007.

[14] Arun Kumar Saha. *Cross Layer Techniques to Secure Peer-to-Peer Protocols for Location, Adjacency, and Identity Verification.* PhD thesis, USA, 2006. Adviser-Molle, Mart.

[15] N. Sastry, U. Shankar, and D. Wagner. Secure Verification of Location Claims. In *ACM workshop on Wireless Security (WiSe 2003)*, pages 1–10. ACM Press, 2003.

[16] Multilateration. In *Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/wiki/Multilateration, retrieved December 2010.

[17] R. Yamasaki, A. Ogino, T. Tamaki, T. Uta, N. Matsuzawa, and T. Kato. TDoA Location System for IEEE 802.11b WLAN. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2005.