Secure, Private Proofs of Location

Brent R. Waters Edward W. Felten
Secure Internet Programming Laboratory
Department of Computer Science
Princeton University
{bwaters,felten}@cs.princeton.edu

Abstract

We present the design of a system that can securely prove the location of a mobile device. In our system the device attempts to prove its location to a party known as the Verifier using a local network. We designed a protocol that securely measures the proximity of the device to the local network. We accomplish this by securely measuring the round-trip signal propagation latency. This technique protects the protocol from powerful attacks by an adversary. The protocol maintains the identity of the device and Verifier as private. We believe we are the first to design a location-proving system that offers both integrity and privacy. Additionally, we provide a solution to deciding which local networks are suitable for location proving. Finally, we show how our basic protocol can be adapted to securely prove the exact position of a tamper-resistant device even when the device is in the possession of an adversary.

1 Introduction

The proliferation of small mobile devices has sparked an interest in systems that can determine the location of a device with high precision. While researchers have achieved important results in this area, they have given less attention to the security of these location-determining schemes. Integrity and privacy are both important elements of a location-proving system's security. The integrity of a location system is important because often a user in control of a device will have incentive to falsify the report of its location. A system that maintains proper security should protect against an attack from such a user. Privacy is also critical to a location system. In many applications the location of a device is privileged information. A location-proving system should protect against unwanted parties learning this information.

In this paper we present a system that allows a device to securely prove its location to another party. We believe we are the first to consider a location-proving system that provides both integrity and privacy.

1.1 The Problem

We present a model in which a party known as the Verifier is interested in the location of a Device to which it does not have immediate access. The Verifier might trust the Device if it were manufactured to be tamper-resistant, but does not trust the environment surrounding the Device, including the user in possession of the Device.

This model is motivated by practical situations. For example, lenders of customer equipment will often find themselves in the position of the Verifier. Universities that lend laptops to their students might wish to have the laptops remain within the confines of the university campuses. Operators of electronic home arrest monitoring systems have a similar problem [7]. In these systems a tamper-resistant Device is attached to the ankle of person under house arrest. The Verifier wants to make sure that the Device (and thus the person) is at the house during certain hours of the day.

In many circumstances the Verifier will not have control of the networking infrastructure at the location of the Device. The Verifier will then need to turn to a third party that will help the Device prove its location.

One possibility is to use a large global system such as the Global Positioning System (GPS) or the cell phone network. While these systems have been useful for many applications their usefulness in proving location is limited. These large systems did not have location proving as an original design objective and to adapt them for such a purpose would be costly and complex. Additionally, the coverage of these systems does not reach many indoor areas where location proving might be desirable.

We turn our attention to small wireless networks where each small network can vouch for the presence of Devices in a small area that it covers. We call the access points of these networks Location Managers. One advantage of using small networks as Location Managers is that the coverage of location-proving networks can grow incrementally.

The use of several Location Managers presents challenges in the design of a location-proving system. A Verifier must decide who it will trust to be a Location Manager for a given area. The Location Manager must be trustworthy and have a networking infrastructure capable of facilitating location proofs in the area. The challenge of choosing a suitable Location Manager becomes difficult when the number of locations that a Device might potentially visit is large and the Verifier is unable to investigate each one individually.

A large privacy issue exists in this model. The Device will possibly be proving its location with several different Location Managers. As discussed the identities of the Device and Verifier are privileged information in many applications. It is important that the this information does not leak out to eavesdroppers or even the Location Managers facilitating the proofs.

Finally, we distinguish two notions of location that we refer to as proximity and position. A Device's proximity to a Location Manager is a measure of how close the Device is to the Location Manager. The Device's position refers to where the Device exists in space. For the remainder of the paper we will make explicit what notion of location we are referring to when necessary. If an adversary is presumed not to exist and the proximity of the Device to enough Location Managers is known then the position of a Device can be determined by triangulation In the case where an adversary is possibly present then standard triangulation techniques will not always work. However, a variation on standard triangulation can securely determine the Device's location as we see in Section 7.

1.2 Our Contributions

In this paper we present a protocol that allows a Device to prove its proximity to a Verifier via a Location Manager. The integrity goal of the system is that a Device cannot be shown to be closer to a Location Manager than it really is.

We choose the round-trip latency of wireless communication between the Device and Location Manager to determine the proximity of the Device to the Location Manager. We use signal propagation latency as our basic metric in order to protect our system from what we call a "proxy" attack. An adversary can execute a proxy attack when the Device is away from a Location Manager. In this case the adversary operates a proxy that is nearby the Location Manager. The adversary will connect the antenna of the Device to the proxy. The proxy can then act as a repeater for the Device. The adversary in effect extends the antenna of the Device via the proxy to make it appear as though the Device was nearby the proxy (Figure 1). This attack will subvert many methods that are used for determining the proximity or position of the Device [1, 11, 13, 2, 12]. However, the use of a proxy will not be of any assistance if round-trip latency is the basic measure, as time for the signal to travel between the proxy and the Device will be accounted for in the total latency.

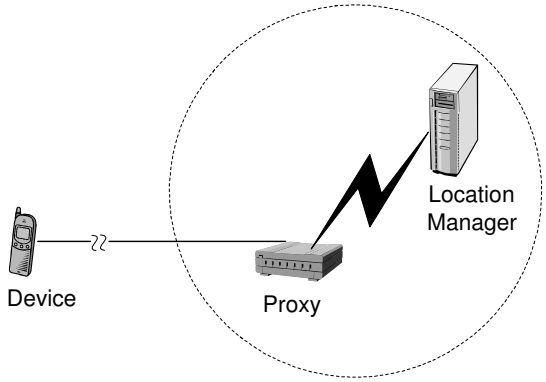

Figure 1. The Device is far away from the Location Manager, however an adversary controls a proxy box that is close to the Location Manager and is connected to the Device. The proxy relays data between the Device and Location Manager, making it appear as though the Device was in the Location Manager's range.

We designed a protocol that allows the Device to prove its proximity to a Location Manager. The protocol securely measures the round-trip signal propagation time. This method is secure against the described proxy attack. Additionally, our protocol keeps the identities of the Device and Verifier private from the Location Manager and from eavesdroppers.

Network operators of existing Wireless LANs appear to be good candidates for Location Managers as many Wireless LANs have already been deployed. Commercial Wireless LAN technology is currently unsuitable for estimating position based on round-trip latency. To take advantage of the wide-spread deployment of these Wireless LANs we offer an alternative definition of proximity based on network visibility. A Device is considered to be at a location if it can communicate with the Location Manager. A system based on this definition has the disadvantage that it is susceptible to the proxy attack described above. Whether this definition is sufficient is application dependent. A location-proving system based on the alternative definition offers a gain in ease of implementation and deployment at the cost of some security. We derive an additional protocol that meets this alternative definition. We implemented a location-proving system that is based on this alternative protocol.

Additionally, we propose a PKI (Public Key Infrastructure) that our system uses to authenticate Location Managers. Our PKI uses a hierarchy of locations with authorities of larger regions delegating responsibility to smaller authorities within the larger region. A chain of delegation can continue on down until it reaches the level of the Location Manager.

Finally, we discuss how our protocol for securely proving a Device's proximity to a Location Manager could be adapted to securely prove the position of the Device. We show how standard triangulation techniques can be modified to account for the presence of an adversary.

2 Related Work

2.1 Location Determining

The Global Positioning system is probably the most widely recognized location-determining system [1]. However, its use in proving the position of devices seems limited as false input of GPS signals can be generated by users in possession of the device [5, 6]. There exists a military segment of GPS known as the Precise Positioning Service (PPS). In PPS signals are encrypted so that they can not be forged by a user. This service has not been open to the commercial sector. Even if PPS were commercially available this would entail trusting every device that used this service to hold a global encryption secret. The use of a global secret seems very unlikely to work with a large deployment of devices.

RF Technologies, Inc.'s Local Positioning System (LPS) is used for tracking and locating office equipment [13]. A novel aspect of their work is that they use the round-trip latency of signal propagation to measure the distance from a tag placed on a mobile object to an antenna at a fixed position. A signal is sent from the antenna and the tag uniquely transforms the signal The antenna then reads the processed signal from the tag and records the latency. The processing time of the signal in the tag is fixed so the component of latency due to signal propagation can be isolated. The system uses multiple antennas to triangulate on the position of the tag. The designers achieved an accuracy of approximately 2m using a 40MHz clocking rate of the chip. This project demonstrates the feasibility of using radio round-trip latency to estimate the distance from a mobile object to a fixed base station. The project was not designed to be robust against malicious attacks.

There exist several other systems that are used to determine the location of a mobile device [11, 2, 12]. There are a variety of techniques used by these systems for location determining. However, all these systems assume a cooperative environment and do not defend against malicious use.

2.2 Location Proving

Gabber and Wool investigated the problem of proving the location of loaned customer equipment [5, 6]. They focused their attention on a location-proving issue confronting the Satellite TV industry. Satellite TV providers loan decoding equipment known as Set-Top Terminals (STTs) to residential customers on the condition that the STT stay at the customers' residence. The Satellite TV providers mandate this restriction to prevent the STT from being moved to a more commercial setting such as a bar. The providers would like to have a way for the loaned STT to be able to prove its

location to the providers. The authors thus consider a setting where the STT (or parts of it) can be trusted, but the environment around the STT, including the users, is not. The authors present and analyze three new methods for solving this problem. In the first method the secure STT is outfitted with a GPS device which it uses to determine its position. However, this method can be thwarted if the user generates fake GPS signals to the STT. In the second method the STT is outfitted with a cell phone device that can use the emergency 911 service to locate the device. This method falls prey to a proxy attack in which the attacker "extends" the antenna of the STT to appear to be at one location while it is truly at another. The authors note these attacks in their work. In their final method the STT contains a synchronized clock, and records at what time a specific signal from the satellite reaches the STT. Due to the nature of the speed of light a user cannot make the STT appear to be closer to the Satellite than it really is.

The final solution of Gabber and Wool differs from ours in that in that they use only one network for proving the location of the device. The network is tightly coupled with the party that wishes to verify the location of the device; indeed, the network and the device share an encryption key. If another party wished to use the location-proving service they would have to be trusted with the encryption secret. This becomes problematic if several parties of varying trustworthiness wanted to join the system. Additionally, their method requires that devices are able to receive a satellite signal. Reception of the satellite signal might not be feasible in indoor environments. Since the communication is unidirectional in their solution, the device must have a synchronized clock or access to some external synchronizing system. Finally, the authors do not discuss privacy as a design concern.

3 Proximity-Proving Protocol

3.1 Proximity-Proving Model

Before we describe our protocol for proving the proximity of a Device to a Location manager we review our model and assumptions. The Verifier is interested in learning the proximity of a trusted Device to a Location Manager. (This trust could be assured by the tamper-resistance of a Device.) The trusted Device is surrounded by an untrusted environment. The Device will receive as input from the untrusted environment the identity of a Location Manager. The Device will then attempt to prove its proximity to that particular Location Manager (Figure 2).

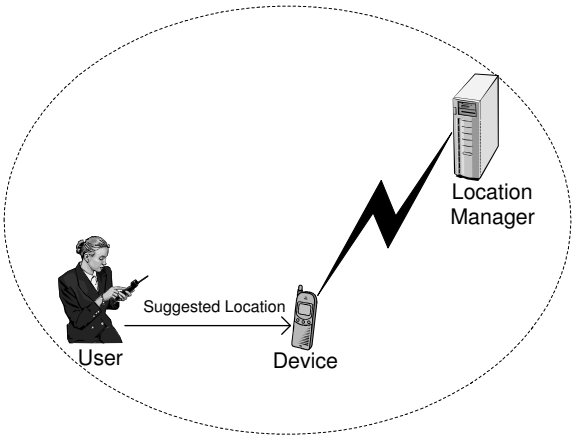

Figure 2. The Device receives a suggested location from the outside environment, in this case from a user. The Device then attempts to prove its location with a Location Manager associated with that suggestion.

We assume that the outside environment has a way of determining the location of a Device and the job of our protocol is to affirm a suggested location. Therefore, we do not specify how a reasonable suggested location could be given to the Device. In practice there are several possibilities for the source of input. For example, if a user was in control of a Device he might just know where he was and input his current location. Alternatively, one of the several location-determining systems [1, 11, 13, 2, 12] could be used to determine which location the Device was probably at and then the Device could attempt to verify the suggestion.

We define the integrity of the system as follows. No adversary should be able to make the Verifier believe the Device was closer a particular Location Manager than it really was. We treat all other forms of attack as denial of service attacks, which we discuss in Section 4.3.

3.2 Protocol Description

We outline the protocol as a sequence of steps taken by the following parties; the Verifier (V), Location

Manager (LM), and Device (D). We assume that the Verifier and Location Manager both have asymmetric encryption key pairs associated with them and the public keys for these pairs will be noted as K_V and K_{LM} respectively. The Device and Location Manager will also each have a signature key, the private keys will be noted as K_D^{Sign} and K_{LM}^{Sign} respectively. Additionally, all encryption and signature operations are assumed to be properly randomized.

1. $D \rightarrow LM : E_{K_{LM}}(\mathbf{start}, \mathbf{reply}, E_{K_V}(\mathbf{DevID}))$
   The Device sends an encrypted message to the location manager. The message contain two randomly generated nonces (**start** and **reply**). Each nonce is long enough so that the chance of an adversary randomly guessing them is negligible. The Device also sends its encrypted ID to the Location Manager. The ID is encrypted with the Verifier's public key so the Location Manager will not be able to read it. (Recall that encryption is properly randomized so the ciphertext will be unique each time.)
2. The Location Manager starts its timer.
3. $LM \rightarrow D : \mathbf{start}, \mathbf{echo}$
   The Location sends the nonce **start** and a new nonce **echo** to the Location Manager.
4. $D \rightarrow LM : \mathbf{reply}, \mathbf{echo}$
   The Location Manager sends the nonces **reply** and **echo** to the Device.
5. Upon receiving **reply** and **echo** back the Location Manager immediately stops its timer and records the round-trip latency.
6. $LM \rightarrow D :$
   $S_{K_{LM}^{Sign}}(\mathbf{latency}, \mathbf{current\ time}, E_{K_V}(\mathbf{DevID}))$
   The Location Manager signs a message containing the latency, current time, and the encrypted Device's ID. The Location Manager will subtract out his fixed internal processing delay from the latency measure. The Device will check that the encrypted ID matches what he gave in step 1 and that the current time is correct.
7. $D \rightarrow V :$
   $E_{K_V}(S_{K_D^{Sign}}(\mathbf{DevID}, \mathbf{LocID},$
   $S_{K_{LM}^{Sign}}(\mathbf{latency}, \mathbf{current\ time}, E_{K_V}(\mathbf{DevID}))))$

   The Device signs its ID, the Location Manager's ID, and the measured latency along with the signed message it received from the Location Manager the previous step. It then encrypts this with the Verifier's public key and sends the encrypted message to the Verifier. In the previous steps we assume that the Device and Location Manager have a direct wireless path to communicate on, but for the final step the encrypted message could be sent through any network.

3.3 Protocol Discussion

The Location Manager measures the latency from the time it sends the nonces **start** and **echo** to the Device to the time it receives the nonces **reply** and **echo**. It then signs this measurement and sends it back to the Device, which in turn will sign this and send it to the Verifier. The Verifier is interested in the round-trip propagation time of the signals, which can be used to calculate the distance from the Device to the Location Manager. The latency consists of the round-trip signal propagation time plus the internal processing time of both the Device and the Location Manager. Given the relatively short period of time that signal propagation takes it is important that the internal processing times are short and predictable. This constraint motivates our design choice to separate out the cryptographic parts of the protocol. The separation makes the processing for the timed part of our protocol as simple as possible. The Verifier presumably knows the characteristics of the Device that it was communicating with and the internal delay of the Location Manager would be public information along with its public encryption and signature verification keys. We stress that the internal processing times we refer to are associated with steps 2-5 of the protocol.

3.4 Protocol for Alternative Definition

The protocol as described above securely shows that the Location Manager is visible to the Device and proves the proximity of the Device to the Location Manager. We can remove the timed part of the protocol by removing steps 2-5 and only sending the encrypted ID in step 1. This will leave only a proof that a Device is visible to the Location Manager (possibly through a proxy). This modified protocol will meet our alternative definition of location based on network visibility.

4 Security Analysis

In this section we provide a security analysis of our protocols.

4.1 Integrity

Suppose that both the Device and the Location Manager behave honestly. Since the first message is encrypted, only the Device and Location Manager will know the nonces **start** and **reply** . When the Location Manager transmits the **start** nonce the RF signal will propagate to the Device. Once the nonce is transmitted there is nothing that an adversary can do to help it arrive at the Location Manager sooner, so the instant an adversary learns the **start** nonce that knowledge becomes useless to him. The same principle applies in the reverse direction. The integrity is derived the fact that no signal can travel faster than the original RF wave. If another, slower means, such as ultra sound, were used for communicating then the protocol would not be secure. Additionally, the Device will only use the signature from step 6 if it includes his encrypted ID. This ensures that the latency measurement is matched with the correct Device.

Recall the we have assumed the Device to be tamper-resistant, meaning that all processing steps of the Device in the protocol are done in tamper-resistant hardware. If the Device is compromised or was not tamper-resistant to begin with we would still like to have some level of security. Suppose the Device is compromised by an adversary. Then in order for the adversary to make the Verifier believe that the Device was a given distance from the Location Manager, the adversary must control a proxy within that distance. This follows from the fact that the **echo** nonce must be sent back to the Location Manager during the timed phase. If the adversary wishes to reply with the **echo** nonce soon enough he must control a proxy that is close enough to do that. Additionally, the encrypted Device ID is committed to the Location Manager in step 1 is included in the signature of step 7 so that an adversary controlling a compromised Device cannot hijack another proof.

If the Location Manager is compromised then a report of proximity could be falsified. However, this would not happen unless a proof of location with that Location Manager was initiated by the user or machine possessing the Device.

4.1.1 Integrity for Visibility Protocol Let's consider the integrity of a proof based upon the alternative definition of network visibility. Due to the signature in step 7, a Device cannot appear to be at a location unless it can communicate with the Location Manager. However, a powerful adversary could execute a proxy attack and effectively extend the visibility of the network to the Device. Such an attack could be executed by a variety of means. For example, an adversary could amplify the antenna of a Device to widen its Wireless range. Alternatively, the adversary could have two boxes at the Location Manager and the Device. It could then record and replay traffic between the two. The feasibility of such attacks depends on factors such as the physical security of the area around the Location Manager. In many circumstances the cost of a proxy attack will be prohibitive and the alternative version of the protocol will suffice.

4.2 Privacy

The Verifier might wish to keep its identity and the Device's identity private even to the Location Manager. To that end the Device does not identify itself to the Location Manager in any of the steps of the protocol. The final message is encrypted so that only the Verifier can read it. An eavesdropper could attempt to determine to whom the Device was sending the message in the final step by sniffing the network. However, there exist several systems that can be used to foil traffic analysis attacks [10, 4, 9]. Eavesdroppers can probably learn that a proof of Location is taking place, but they will not learn who the Device and Verifier are.

The identity of the Device could also be discovered by methods that work outside the basic framework of the protocol. For example, if every Device had a unique Medium Access Channel (MAC) identifier as common Ethernet network cards do then this could be used to identify a Device. Another possible leak of privacy occurs when a user in possession of the Device knows the Device's identity and gives it away through an out-of-band technique. Implementors of a real system should take care to avoid or at least identify these privacy pitfalls.

4.3 Denial of Service

Our system does not defend against Denial of Service attacks. An adversary could potentially block the Device from communicating to the outside world. Additionally, an adversary could place a buffer between the Device and the outside world to make it appear as though the Device was farther away from a Location Manager than it actually was. The severity of such Denial of Service attacks and the methods that should be used to deal with them will depend upon the application for which the system is being used.

4.4 Other Issues

The integrity of our system relies upon both the Device and the Location Manager being able to execute the timed steps of the protocol in a very predictable manner with low variability in the processing times Additionally, the Location Manager must be able to time this very precisely. A PC with a commercial Wireless LAN adapter currently will not be able to meet these performance requirements. However, specialized hardware could perform this task adequately. The Local Positioning System is able to determine distances within a few meters by measuring the round-trip latency for signal propagation [13].

If the participants of a location-proving system find the cost of deployment to be prohibitive they may choose a system based on our alternative definition of network visibility. For many applications the risk of a proxy attack might be worth the benefit of easier deployment.

5 Finding a Location Manager

In Section 3 we showed how a Device could securely prove its proximity to a particular Location Manger. The security of our system relies upon the Location Manger being trustworthy and being able to securely measure the proximity of a Device to a certain physical position. We now consider the problem of how the Verifier decides if a particular Location Manager can vouch for a particular physical area.

Each Location Manager can be described by a triplet of publicly available information, consisting of the Location Manager's physical location, a public encryption key, and a public signature-verification key. The Verifier must decide for every Location Manager whether to trust this information.

If the number of Location Managers that a Device could interact with is small then the task for the Verifier is relatively simple. The Verifier can independently research these few Location Managers and preprogram the Device with the appropriate entries.

This approach becomes intractable, however, when there are a large number of Location Managers that a Device might interact with. To solve this problem we propose a location-based PKI that our system can use to authenticate Location Managers. Our proposed PKI is organized around a hierarchy of locations. Authorities of larger regions will delegate responsibility of subregions until the delegation reaches the level of a Location Manager. The delegation is determined by physical boundaries. For example, a root authority for the United States could delegate its responsibility for New York State to another authority, which in turn could delegate its authority for New York City to a third authority. This third authority could then authenticate a Location Manager for Penn Station. The method of delegating authority along physical boundaries results in a tree of trust (Figure 3).

The technique described above can be implemented efficiently with certificate chains. The Verifier and Device will be loaded with just the root certificates. A Location Manager's certificate chain is validated by following the signature path from the Location Manager up to the root. This system works in a manner similar to DNSSEC [3]. However, DNSSEC has a virtual name space while this system is based upon physical locations.

6 Implementation

We implemented a location-proving system based on the alternative network-visibility based protocol. We implemented the system in Java using the Bouncy Castle Crypto API [8]. The software consisted of three programs representing the Location Manager, Device, and Verifier. We included a GUI for the operator of the Device. We use X.509 certificate chains for Location Manager authentication. Every Device and Verifier was loaded with root certificates and the each Location Managers has a complete chains for authentication of

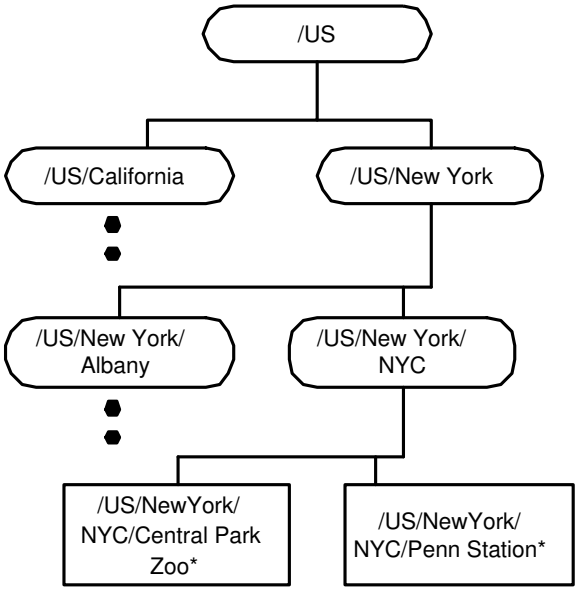

Figure 3. A partial view of a PKI based on physical boundaries. Trust is derived from parent nodes. The Location Managers are at the bottom; each is marked with a *.

his public information.

The implementation was tested on our department's Wireless LAN. We were able to prove that our location within the building based on the definition of network visibility.

7 Securely Determining the Position of a Device

We considered location proving as a problem of proving a Device's proximity to a Location Manager. For some applications it is desirable to securely prove a Device's exact physical position. In this section we examine how the protocol we described could be used to meet this goal.

Suppose a Device uses our protocol from Section 3 to prove its proximity to a Location Manager. An adversary is unable to make the Device appear closer to a Location Manager than it really is, but it could perhaps make it appear further away by inserting a buffer to delay communication between the Device and Location Manager. Let δ be the two-sided precision error in our system. That is without an adversary present the real proximity of the device is plus or minus δ of the reported proximity. Now suppose that a Device proves its proximity to a Location Manager and d is the measured distance. If the Device is confined to a plane we can then infer that the device is contained within a circle of radius $d+\delta$ (Figure 4). If the Device executes simultaneous proofs with multiple Location Managers then its position can be narrowed down to the intersection of the areas contained by these circles. If we operate in three dimensional space the Device is within a sphere of the same radius.

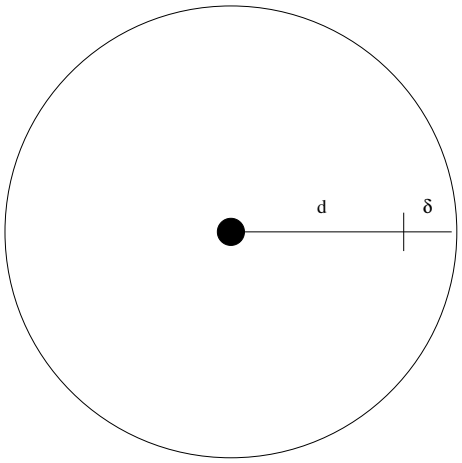

Figure 4. The protocol reports a distance of d. With the presence of an adversary the Device could be anywhere within the distance $d+\delta$ of the Location Manager.

Suppose that there was no error in the proximity measurement. If the Verifier can assume that there is no adversary then triangulation can be used to determine the exact position of a Device from knowing its proximity to three non-co-linear Location Managers. However, if the Verifier is suspicious of the presence of an adversary the area of uncertainty can be very large (Figure 5).

7.1 Toward a Secure Positioning System

A secure positioning system should have two properties. The first is that the Device will always be inside the area resulting from the proof. The second requirement is that if there is no adversary present then the area that the Verifier believes the Device to be in should be small. Also, the maximum distance

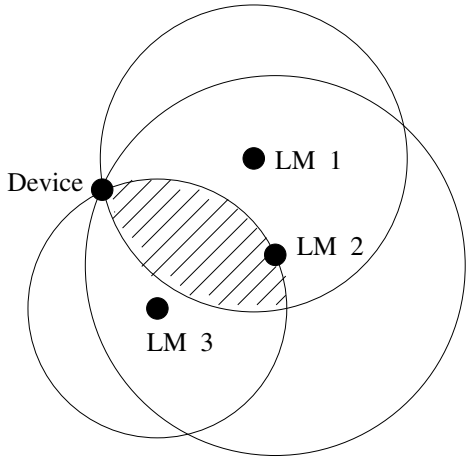

Figure 5. In this figure a Device conducts three simultaneous proofs of proximity with three Location Managers. If there was known to be no adversary then the Device must be positioned at the intersection of the three circles. However, since an adversary could make the Device appear farther from a Location Manager than it really is, the Verifier can only ascertain that the Device is in the intersection of the areas bounded by the circles (the marked area). In this example, the area of uncertainty is too large to be of much practical use.

from where the Device is to where the Verifier believes it could be should be short.

The first objective can be met quite simply by using our protocol. Each proof of proximity with a Location Manager is treated with maximum suspicion by the Verifier. Upon receiving a proof of proximity d it will believe that the Device is within a circle of radius $d+\delta$ centered around it. The Device can securely be placed inside the intersection of multiple circles constructed from simultaneous proofs of proximity with multiple Location Managers.

We show that such a system can be built to prove the position of a Device when the Device is surrounded by Location Managers. Suppose that δ, the error in measuring proximity, is 0. Also suppose that the Device is inside a triangle with a Location Managers at each corner and that it conducts a simultaneous proof of proximity to with each one.

The Verifier, upon receiving the proofs of proximity, will then be able to determine the position precisely. Let T be the true position of a Device and F be a different position that is inside the triangle. An adversary will be unable to make the Device appear to be at position F. This is due to the fact that F is closer to at least one corner of the triangle than T is for any $T \neq F$. Recall that if our protocol is used then a Device cannot appear closer to a Location Manager than it really is.

This argument provides intuition that a secure positioning system can be designed to position a Device that is surrounded by Location Managers.

Of course in any real system the error in determining proximity is greater than 0. In this case, if the Device is near an edge of the triangle then the maximum distance from where the Device is to where it could possibly be if there was an adversary can become large. However, if the Device is near the center of this triangle the maximum distance becomes small (Figures 7 and 6).

We can now design a secure positioning system based upon a grid of Location Managers. In this system a Device can prove its position to a high degree of accuracy by finding a group of Location Managers that form a triangle for which it is near the center. The Device will then perform simultaneous proofs of proximity with each of these Location Managers. Although we showed how a triangle could be used as the basic shape, other configurations could be used as well. What is important is that the Location Managers surround the Device .

We saw how the proof of position was more accurate when the Device was near the center of the triangle formed by the Location Manager. There are also some triangles and shapes that work better for this than others. For example an equilateral triangle works better than one with a large, obtuse angle.

While we have given our arguments in two dimensional space for simplicity, they can be extended naturally to three dimensional space.

7.2 Discussion

In the previous subsection we showed how the primitive of proving proximity can be extended to securely prove the position of Devices. We assert that an algorithm for triangulation is feasible in a system where

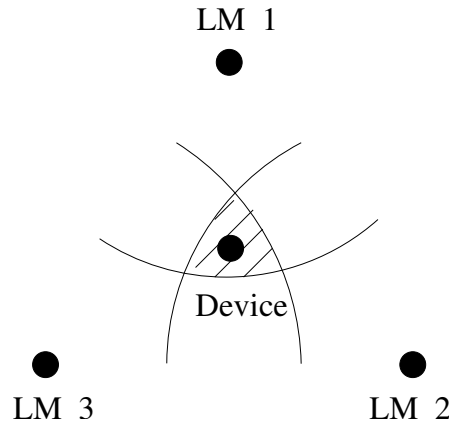

Figure 6. In this example, there is some uncertainty in the proximity measurement. The Device is close to the center of the triangle . The shaded error represents the area where the Verifier knows the Device to reside in. This represents the maximum distance from where the Device is to where the Verifier knows it could be. Since this distance is not much more than the error in the proximity measurement, the Device is able to prove its position with high precision.

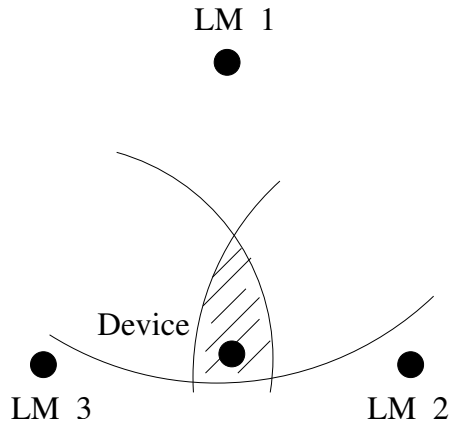

Figure 7. In this example, there is some uncertainty in the proximity measurement. The Device is close to the edge of the triangle that is formed by Location Managers 2 and 3. The shaded error represents the area where the Verifier knows the Device resides in. The distance from the Device to the top corner of this area is longer than what is desirable. This represents the maximum distance from where the Device is to where the Verifier believes it could be.

the position of the Device can be bounded within a small area.

A deployment of a real system will have additional complications. The possible layout of Location Managers might be affected by factors such as physical boundaries, property ownership, etc. The physical position of the Location Managers themselves will have to be precisely known. Additionally, there will have to be some coordination among Location Managers to make sure that simultaneous proofs of proximity do indeed take place at the same time.

While these issues will make the design and deployment of a real system challenging, we believe that a secure method for proving proximity can be used as a building block for a secure positioning system.

8 Conclusion

We demonstrated how a tamper-resistant Device can prove its proximity to another party. Our solution emphasized both the integrity of the proof and the privacy of the participants. In our solution the Device enlists the help of a third party known as the Location Manager. We designed a protocol with which the Device can securely demonstrate its location in terms of its proximity to the Location Manager. We offered an alternative protocol for location proving that trades off some security for ease of implementation. Additionally, we proposed a PKI based upon a physical hierarchy with which Location Managers are authenticated. Finally, we showed how the techniques we used for proving proximity to a Location Manager could be adapted to prove the position of a Device.

References

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

[2] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *In the Proceedings of IEEE INFOCOM*, volume 2, pages 775–784, March 2000.

[3] D. Eastlake. Domain name system security extensions. Request for Comments 2535, Internet Engineering Task Force, March 1999.

[4] Gabber, Gibbons, Matias, and Mayer. How to make personalized web browsing simple, secure, and anonymous. In *FC: International Conference on Financial Cryptography*. LNCS, Springer-Verlag, 1997.

[5] Eran Gabber and Avishai Wool. How to prove where you are: Tracking the location of customer equipment. In *ACM Conference on Computer and Communications Security*, pages 142–149, 1998.

[6] Eran Gabber and Avishai Wool. On location-restricted services. *IEEE Network*, November/December 1999.

[7] BI Inc. Web site at http://www.bi.com.

[8] Legion of the Bouncy Castle. Web site at http://www.bouncycastle.org.

[9] Michael K. Reiter and Aviel D. Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[10] The Anonymizer. Web site at http://www.anonymizer.com.

[11] Roy Want, Andy Hopper, Veronica Falcao, and Jon Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992.

[12] Anday Ward, Alan Jones, and Andy Hopper. A new location technique for the active office, 1997.

[13] Jay Werb and Colin Lanzl. Designing a positioning system for finding things and people indoors. *IEEE Spectrum*, 35(9):71–78, September 1998.