Protecting Privacy and Securing the Gathering of Location Proofs – The Secure Location Verification Proof Gathering Protocol

Michelle Graham and David Gray

School of Computing, Dublin City University, Dublin, Republic of Ireland
{mgraham,dgray}@computing.dcu.ie

Abstract. As wireless networks become increasingly ubiquitous, the demand for a method of locating a device has increased dramatically. Location Based Services are now commonplace but there are few methods of verifying or guaranteeing a location provided by a user without some specialised hardware, especially in larger scale networks. We propose a system for the verification of location claims, using proof gathered from neighbouring devices. In this paper we introduce a protocol to protect this proof gathering process, protecting the privacy of all involved parties and securing it from intruders and malicious claiming devices. We present the protocol in stages, extending the security of this protocol to allow for flexibility within its application. The Secure Location Verification Proof Gathering Protocol (SLVPGP) has been designed to function within the area of Vehicular Networks, although its application could be extended to any device with wireless & cryptographic capabilities.

1 Introduction

Frequently in applications, one device is required to authenticate itself to another in some way. This interaction often takes the form of exchanging a private secret shared by the two, such as a password, to prove the device's identity. However with ubiquitous networks increasing in prevalence, the issue of context has become more important. Instead of a device being required to produce a password in order to gain access to some resource within the network, it is simply required to prove that its current location is in the area of that network. This *location verification problem* can be solved by technologies such as Active Badge [1] and RFID [2], however these are based within a limited area and require an infrastructure of sensors in order to function. This restricts their usability to small scale networks.

In order to address the issue on a larger scale, we have designed a location verification system for use in large scale networks, specifically Vehicular Adhoc Networks (VANETs), requiring no special physical infrastructure to be in place. We have selected VANETs as a suitable environment in which to apply this location verification scheme due to the protocol's requirement of tamper-resistent devices and public key cryptography.

A.U. Schmidt and S. Lian (Eds.): MobiSec 2009, LNICST 17, pp. 160–171, 2009.

We propose to verify the presence of a specific device in a particular area using a two step process. First, the claiming device distance bounds [3] with multiple neighbouring devices, to prove that it is within range of other devices known to be in the area. These devices then state whether or not the claiming device successfully proved themself to be within range of them. These proofs are supplied by the claiming device to a central verifying device, who combines them to determine whether the claiming device is in the area.

In the rest of this paper we outline a system for the verification of location claims (Section 3). We then put forth a basic protocol for the protection of the distance bounding exchange. We extend the protocol incrementally to protect the exchange from all tampering and fraudulent attempts (Section 4). Finally, we discuss our future work on this issue (Section 7).

2 Related Work

Calculating the location of a device is a broad area of research. One of the major approaches to solving this problem is location determination. This approach attempts to determine the location of a device without any prior information. GPS is the most widely used technology to locate a device. However readings can be forged and GPS signals can be spoofed [4]. An encrypted version of GPS exists, the Precise Positioning Service (PPS), but this technology is currently restricted.

A second approach to the calculation of a device's location is location verification. Rather than attempting to discover the device's location without apriori knowledge, a device claims to be in a certain region or area. The system then checks whether this claim is legitimate. This approach reduces the complexity of the problem by minimising the area in which the device could be. In [5], the authors introduce the concept of Integrity (I) regions to authenticate messages within wireless networks without the use of pre-existing keys. However, the I-region is very minimal, typically only extending to the limits of the user of a device's field of vision, removing its potential for use in larger scale systems.

In [6], Sastry et al proposed another approach to location verification suitable for larger systems. The Echo protocol employs ultrasound and radio frequency time of flight measurements to gauge the upper bound on the distance between 2 devices. However, a device can employ RF to send its response as far as a colluding device, which can then replay the reply back to the verifier node using ultrasound. As RF has a faster propagation time than ultrasound, the attack would not be detected.

In [7], Waters & Felten proposed the Proximity-Providing Protocol, which is based in wireless networks. Similar to the work done by Sastry et al, the PPP uses round trip times of packets to calculate the upper bound on distance between devices. It relies on a much more ubiquitous technology and is therefore far more practically applicable. However, this protocol is still susceptible to the terrorist fraud. A colluding device could participate in the distance bounding aspect of the protocol in place of the prover, shrinking the latency and making

the prover appear closer. We believe we have addressed this issue in our approach, discussed further in Section 4. Our approach also extends the concept behind the Proximity-Proving Protocol to remove the need for a trusted device to be used in the Location Manager role.

Many location verification approaches utilise distance bounding, however it is vulnerable to many circumventing attacks which render the results received incorrect. In [8], Clulow et al outline attacks on current distance bounding approaches, illustrating common flaws in existing protocols. In particular, they cite the use of a single nonce within the challenge-response exchange as vulnerable to a guessing attack. This issue is negated in the Secure Location Verification Proof Gathering Protocol (SLVPGP) through multiple iterations of single nonce challenge-response exchanges. Clulow et al also state that requiring participants to employ encryption during the timed phase of an exchange introduces an inaccuracy into the distance calculations. However, due to the approach to distance bounding employed by the SLVPGP, we believe that this issue does not apply.

3 A System for the Secure Verification of Location Claims

3.1 System Model and Assumptions

Before discussing the design of the SLVPGP, we review the environment in which it is employed and outline the assumptions made regarding participating devices.

1. *Location Claim* - A message containing the location at which a device making a claim purports to be
2. *Claimant* - An untrusted device which makes a location claim.
3. *Verifier* - A trusted entity that decides if a location claim is valid.
4. *Proof Provider* - A neighbouring device contacted by a Claimant to distance bound with, in order to provide proof of a location claim.
5. *External anonymity/confidentiality* - If a device is not involved in an SLVPGP run, they cannot learn location or identity information regarding participating devices even if within range of pertinent transmissions.
6. *Complete anonymity/confidentiality* - A device cannot learn identity or location information regarding another device, even if involved in the same SLVPGP run.

A Claimant wishes to prove its location to a Verifier within an untrusted 802.11 based environment [9]. Although the Claimant and Proof Providers are untrusted devices, we assume that they contain tamper-resistant modules in which all cryptographic keys are stored. This prevents devices from sharing keys in an attempt to sabotage the system's security. The Verifier provides a list of devices in the vicinity of the claimed location for use as Proof Providers. The Claimant distance bounds with each of these to ascertain that it is within range of that device. The results of these exchanges are supplied to the Verifier, which makes its decision based on this information.

The aim of our protocol is to secure the proof gathering process and protect it from intruders and malicious parties within the exchange. We define a secure protocol as one which satisfies the following three security properties. First, *anonymity of identity* - a device's identity should not be discoverable by any party other than the Verifier, except where expressly given to it. Second, *confidentiality of location* - the location of a device with a specific identity should not be discovereable by any party other than where expressly given to it. Finally, *authentication of information* - the origin of any given message must be known.

3.2 Threat Model

Honest nodes behave exactly as their roles dictate. They can communicate with the Verifier, along with anyone within radio range. They receive all messages broadcast within their area, but do not act on messages not intended for them. When a node participates in distance bounding, information regarding their location is leaked [10]. Some solutions to the issue of MAC address identification have been proposed [11,12,13], however this is beyond the scope of this research. The SLVPGP does not allow leaked information to be tied to a specific device's identity within the system, by protecting the identities of those involved.

Malicious nodes fall into one of three categories: malicious Claimants, malicious Proof Providers and malicious nodes external to the exchange, i.e. intruders. Malicious nodes cannot occupy the role of the Verifier as this is a trusted, un-compromisable entity. All malicious nodes can eavesdrop any message sent within the system. Malicious nodes may also manipulate received messages and retransmit them, but cannot prevent a node from receiving a specific message.

We consider multiple attacks on the SLVPGP, most of which are attempted by malicious Claimants attempting to influence the result of their location verification. The first is the *guessing attack*. A malicious Claimant could attempt to guess the correct response to a challenge and send its response prior to receiving the challenge nonce. If successful, the malicious Claimant could prove itself to be within range of the Proof Provider. However, this is addressed through the use of multiple challenge-response iterations, reducing the probability of success. Another attack attempted by a malicious Claimant is the *terrorist fraud* [14]. A malicious Claimant has a proxy act as a man-in-the-middle between himself and the verifying device. This allows the malicious Claimant to convince a Proof Provider that he is closer than he truly is. Finally, in a *snooping attack*, an external intruder node may attempt to gain access to location or identification information through eavesdropping. These attacks are thwarted through the use of encryption on messages containing private information. The case of a malicious proof provider lying about the presence of an honest Claimant is dealt with by the Verifier through the use of trustworthiness levels.

3.3 The Role of the Verifier

The SLVPGP relies upon the Verifier to supply suitable Proof Providers for use in during a run of the protocol, and to determine the final verdict on a claim

using their proofs. Unlike the Claimant and its Proof Providers, the Verifier is a trusted device outside the system's environment, possessing the identities and public keys of all devices within the entire Location Verification system. The exact functionality of the Verifier lies beyond the scope of this paper, but we outline here the Verifier's basic processes required by the protocol.

The Claimant does not participate in the selection of Proof Providers for use in a location claim. If involved, it would gain the ability to manipulate the process. Instead, the Verifier geographically routes [15] a request seeking Proof Provider volunteers to the area in which the Claimant purports to be. Only devices in the area of the claimed location receive this request, and Proof Providers are only selected from the volunteers responding to this request. Therefore all Proof Providers used within the protocol are in the correct area.

The Verifier's principle task is to assess the possibility of a device's location claim, based on information gathered for it by the Claimant from Proof Providers. However, the reliability of one Proof Provider's verdict is different to that of another, and so each must be weighted according to that device's past behaviour. This process is achieved using the beta probability density function [16]. Therefore a device with a long record of honest behaviour will have a stronger vote than that of a device with an unreliable past.

3.4 The Use of Distance Bounding in the SLVPGP

In the SLVPGP, distance bounding is not used to calculate specific distances between the Claimant and its Proof Providers. Instead, it allows a Proof Provider to confirm that the Claimant is not attempting to execute a terrorist fraud on the system. In order for the Claimant to participate in distance bounding , they must either be in that area or collude with a device that is. The round trip time required for an exchange when the Claimant is perpetrating a terrorist fraud is drastically different to the time required for an honest exchange. It is this time discrepancy that is being tested when distance bounding is performed. In addition to this, employing distance bounding without requiring specific distances between devices, "off the shelf" wireless devices become feasible, allowing the protocol far greater scope for application.

In order for distance bounding to function as a method of detecting a terrorist fraud attack, the Claimant must be required to actively participate in the exchange. The employment of authentication forces the Claimant to receive the message and compose the corresponding reply, rather than employing a proxy. When dealing with authentication during distance bounding, the concept of using any form of encryption is usually dismissed. When using distance bounding to gauge physical distance, the time required to digitally sign something is thought to overpower the time required to send a message on a round trip. However, we require only that there is a distinction between an honest exchange and an exchange involving a proxy device. As the difference between honest and fraudulent round trip times are quite dramatic prior to the inclusion of digital signatures, we believe that the distinction will hold. We intend to investigate this further in the future (see section 7).

4 The Secure Location Verification Proof Gathering Protocol (SLVPGP)

4.1 Protocol Outline

In order to understand what security risks are present in the proof gathering process, we first model a protocol devoid of any security. We outline the protocol as a sequence of steps taken by the parties involved in the exchange: the Verifier (V), the Claimant (C) and N Proof Providers (B_i where $i \in \{1...N\}$).

1. C→V: C, X_C
   The Claimant transmits its identity (C) and location (X_C) to the Verifier, requesting that its location claim be verified.
2. V→C: $B_1, B_2, ...B_N$
   The Verifier sends a list of Proof Providers to the Claimant.
3. C→B_i: $B_i, \mathcal{N}_i, \mathcal{N}_i^{'}$
   The Claimant broadcasts a message for each Proof Provider containing their identity and two nonces (long random integers), $\mathcal{N}_i$and $\mathcal{N}_i^{'}$.
4. The Claimant and Proof Providers create a chain of M hashes for each nonce. These are noted as $H_{i,k}$ and $H_{i,k}^{'}$ respectively, where $i \in \{1...N\}$ and $k \in \{1...M\}$.
5. Distance Bounding
   This stage is performed multiple times to lessen the effect of any network issues and to reduce the effectiveness of a guessing attack.
   (a) B_i starts its timer.
   (b) B_i →C: k, $H_{i,k}$, $\mathcal{N}_i^{''}$
   B_i sends a message to the Claimant containing a new random nonce $\mathcal{N}_i^{''}$, a randomly selected value from the hash chain of $\mathcal{N}_i$, $H_{i,k}$, and its position in the chain, k. The value of k decreases with each distance bounding iteration. The Claimant checks whether the kth value in the $\mathcal{N}_i$ hash chain matches $H_{i,k}$ and if so, continues to the next step.
   (c) C→B_i: $H_{i,k}^{'}$, $\mathcal{N}_i^{''}$
   The Claimant sends a message to B_i containing $\mathcal{N}_i^{''}$and the kth value in the $\mathcal{N}_i^{'}$ hash chain. B_i compares this with the kth value in its $\mathcal{N}_i^{'}$ hash chain. If the two values match and the received value of $\mathcal{N}_i^{''}$ matches its own sent value, B_i stops its timer and calculates the round trip latency (subtracting out its own internal processing time).
6. B_i →C: T_i, X_i, L_i
   B_i sends the Claimant its proof, comprised of its current location (X_i) and overall decision regarding the presence of the Claimant in its vicinity (L_i). A timestamp is also included to tie the proof to this specific point in time.
7. C→V: C, T_C, X_C, $\{T_1, X_1, L_1\}$, ..., $\{T_N, X_N, L_N\}$
   The Claimant compiles all the proofs gathered from its Proof Providers and forwards them to the Verifier, along with its identity C, current location X_C and a timestamp T_C to tie the proofs to that point in time.

4.2 Protocol Discussion

Due to the possibility of network issues such as delays and lost packets, the result of a location claim could be affected through the true results of a single distance bounding exchange being distorted. We have included multiple distance bounding exchanges within the protocol to compensate for this possibility. In order to avoid high data overheads, we employ hashing [17] to calculate multiple nonces from a single initial nonce. In this method, the kth nonce $H_{i,k}$ in a hash chain is calculated from $\mathcal{N}_i$ using the formula $H_{i,k} = \mathcal{H}^k(\mathcal{N}_i)$, i.e. the hash of the initial nonce is hashed and this third value is then hashed to produce a fourth value etc.

The hash chains produced from $\mathcal{N}_i$ and $\mathcal{N}_i^{'}$ allow the Claimant and Proof Provider B_i to validate that they are interacting with their expected device. $\mathcal{N}_i$ proves the message's origin is from a Proof Provider engaged in the Claimant's current location claim. The value from the hash chain derived from $\mathcal{N}_i^{'}$ verifies that the message received originated from the Claimant involved in that specific location claim. $\mathcal{N}_i^{''}$ is used to ensure that the Claimant cannot guess which nonce in the chain to send and transmitting his response before receiving the Proof Provider's message. As the Claimant's reply contains $\mathcal{N}_i^{''}$, it is forced to wait until it receives this value. This dramatically reduces the possibility of an early transmission attack successfully being accepted as a valid response.

The use of a randomly selected hash ($H_{i,k}$) increases the unpredictability of the Proof Provider's distance bounding challenge message for the Claimant. However, given the value $H_{i,k}$, an intruder could then calculate all values above it in the hash chain. For this reason the value of k decreases with each iteration of distance bounding, moving downwards within the chain. Although this reduces the reusability of the hash chain, it removes the risk of a security leak.

The Proof Providers measure the latency of each exchange, and these latencies are used to calculate the Proof Provider's final verdict, a boolean representing either a positive or negative result. However, the protocol as shown above upholds none of the security protocols outlined in section 3.1. Both the distance bounding exchange and the verdicts are susceptible to tampering, either by an intruder or a malicious Claimant. In addition to this, proofs can be fabricated by a malicious Claimant without any interaction with a Proof Provider. Finally, the protocol is vulnerable to the terrorist fraud as the distance bounding messages are not tied to the Claimant. In order to address these vulnerabilities, we have repeatedly extended the basic protocol, increasing security with each extension.

5 Extending the Protocol

As before, we outline these protocols as a series of steps taken by the Claimant (C), Proof Providers (B_i where $i \in \{1...N\}$) and Verifier (V). We assume that all parties involved have asymmetric key pairs associated with them. These key pairs will be noted as K_A^- and K_A^+ for the private and public keys respectively, where A is the owning party's identity.

Algorithm 1. SLVPGP Extension 1

1. C→V: $\{$—C,X_C—$\}_{K_C^-}$
2. V→C: $\{|B_1, B_2, ..., B_N|\}_{K_V^-}$
3. C→ B_i: B_i,$\{\mathcal{N}_i, \mathcal{N}_i^{'}\}_{K_{B_i}^+}$
4. The Claimant & Proof Providers create a chain of M hashes for each nonce.
5. Distance Bounding (executed multiple times)
   (a) B_i starts its timer.
   (b) B_i →C: k, $H_{i,k}$,$\mathcal{N}_i^{''}$
   (c) C→B_i: $H_{i,k}^{'}$, $\{|\mathcal{N}_i^{''}|\}_{K_C^-}$
6. B_i→C: $\{|T_i, X_i, L_i, C|\}_{K_{B_i}^-}$
7. C→V: $\{|T_C, X_C|\}_{K_C^-}$,$\{|T_1, X_1, L_1, C|\}_{K_{B_1}^-}$, ..., $\{|T_N, X_N, L_N, C|\}_{K_{B_N}^-}$

5.1 The SLVPGP: Extension 1

The protocol extension shown in algorithm 1 removes many of the vulnerabilities outlined regarding the basic protocol. Encrypting $\mathcal{N}_i$ and $\mathcal{N}_i^{'}$ prevents intruding devices from gaining knowledge of these values without collusion. The addition of a digitial signature to the echoing nonce removes the danger of a collusion attack going undetected & provides authentication. A malicious Claimant can no longer fabricate proof for the Verifier, nor can it undetectably alter the content of an existing proof. Similarly, an intruder cannot fraudulently participate in the exchange. Finally, the inclusion of the Claimant's identity within the signed proof message from the Proof Provider prevents malicious Claimants from using valid proofs pertaining to another Claimant as evidence of their location claim.

5.2 The SLVPGP: Extension 2

In algorithm 1, we extended the protocol to provide authentication and prevent many attacks such as the terrorist fraud. In this section we further extend the protocol (algorithm 2) to provide an increased level of anonymity and confidentiality. We achieve this through the addition of encryption.

The addition of encryption builds upon this framework to include both external anonymity and external confidentiality. While the Claimant and all involved Proof Providers have access to the location and identity information being sent over the run of the protocol, no other devices can discover it, whether they are within range of the transmissions or not.

5.3 The SLVPGP: Extension 3

Though the previous extension increases the protocol's level of security, honest participants could still record information on the parties involved, then use this information maliciously at some point in the future. For this reason, we wish to provide complete anonymity and confidentiality (algorithm 3).

Algorithm 2. SLVPGP Extension 2

1. C→V: $\{\{\text{—C},X_C\text{—}\}_{K_C^-}\}_{K_V^+}$
2. V→C: $\{\{\text{—}B_1, B_2, ..., B_N\text{—}\}_{K_V^-}\}_{K_C^+}$
3. C→ B_i:$\{B_i, \text{C}, \mathcal{N}_i, \mathcal{N}_i^{'}\}_{K_{B_i}^+}$
4. The Claimant & Proof Providers create a chain of M hashes for each nonce.
5. Distance Bounding (see notation in 1).
6. B_i →C: $\{\{|T_i, X_i, L_i, C|\}_{K_{B_i}^-}\}_{K_C^+}$
7. C→V:$\{\{|X_C, T_C|\}_{K_C^-}\}_{K_V^+}$, $\{\{|T_1, X_1, L_1, C|\}_{K_{B_1}^-}\}_{K_V^+}$, ..., $\{\{|T_N, X_N, L_N, C|\}_{K_{B_N}^-}\}_{K_V^+}$

Algorithm 3. SLVPGP Extension 3

1. C→V: $\{\{\text{—C},X_C\text{—}\}_{K_C^-}\}_{K_V^+}$
2. V→C: $\{\text{—}\ \{B_1, \mathcal{N}_1, \mathcal{N}_1^{'}, \mathcal{N}_1^{''}\}_{K_{B_1}^+}, \{\mathcal{N}_1, \mathcal{N}_1^{'}, \mathcal{N}_1^{''}\}_{K_C^+}, ...,$ $\{B_N, \mathcal{N}_N, \mathcal{N}_N^{'}, \mathcal{N}_N^{''}\}_{K_{B_N}^+}, \{\mathcal{N}_N, \mathcal{N}_N^{'}, \mathcal{N}_N^{''}\}_{K_C^+}\ \text{—}\}_{K_V^-}$
3. The Claimant decrypts and stores each nonce received from the Verifier.
4. C→ B_i: $\{B_i, \mathcal{N}_i, \mathcal{N}_i^{'}, \mathcal{N}_i^{''}\}_{K_{B_i}^+}, \{C\}_{K_V^+}$
5. The Claimant and Proof Providers create a chain of M hashes for each nonce.
6. Distance Bounding (see notation in 1).
7. B_i →C: $\mathcal{N}_i^{''}, \left\{|\{|\mathcal{N}_i^{''}|\}_{K_C^-}, \mathcal{N}_i^{''}|\right\}_{K_{B_i}^-}$, $\{\{|T_i, L_i, X_i, \{C\}_{K_V^+}|\}_{K_{B_i}^-}\}_{K_V^+}$
8. C→V: $\{\{|X_C, T_C|\}_{K_C^-}\}_{K_V^+}$, $\left\{|\{|\mathcal{N}_1^{''}|\}_{K_C^-}, \mathcal{N}_1^{''}|\right\}_{K_{B_1}^-}$, $\{\{|T_1, L_1, X_1, \{C\}_{K_V^+}|\}_{K_{B_1}^-}\}_{K_V^+}$, ..., $\left\{|\{|\mathcal{N}_N^{''}|\}_{K_C^-}, \mathcal{N}_N^{''}|\right\}_{K_{B_N}^-}$, $\{\{|T_N, L_N, X_N, \{C\}_{K_V^+}|\}_{K_{B_N}^-}\}_{K_V^+}$

However, the protocol is now vulnerable to a *denial of service attack*, where the Claimant is supplied with fraudulent proofs. This is because the Claimant is unable to differentiate between valid and fraudulent proof messages, as they are now encrypted with an unknown key. In order to solve this issue, the Claimant requires a method of matching a received proof with an unknown Proof Provider. $\mathcal{N}_i^{''}$ is used to verify that all proof messages received during an exchange were created by legitimate Proof Providers, removing an intruder's ability to undetectably insert illegitimate proofs into the exchange.

The Verifier encrypts the message containing the nonces, to prevent intruders from learning their values. However, as the Claimant also requires a copy of these values, the Verifier is forced to create and encrypt two messages. Broadcasting allows the Claimant to communicate with its Proof Providers without knowledge of their identities and vice versa, allowing complete anonymity and confidentiality. Therefore, we believe we have designed a secure protocol, satisfying our outlined security properties.

6 Overall Analysis

6.1 Cost Analysis

Each extension to the SLVPGP increases the security provided to those participating. However, these improvements in security also require extra data to be transmitted and extra time to be taken to compute encryptions or digital signatures. We define the *costs* of an extension to mean this extra data and/or time required.

The costs related to the first extension stem from the addition of digital signatures to protocol messages. As signatures must be calculated and transmitted, this generates both time and data transmission costs. However, these costs are minimal in relation to those incurred by the second and third extensions. The addition of encryption to these levels increases both the data and time costs required. The third extension is the most costly, however, due to the creation and transmission of duplicate messages to allow for complete anonymity and confidentiality.

Each of the three extensions to the SLVPGP are fully secure against the terrorist fraud, in addition to maintaining the integrity of the proof messages supplied to the Verifier by the Claimant. With the exponential increase in data overheads incurred within the third extension, complete anonymity and confidentiality come at quite a high cost. The second extension to the SLVPGP is the most widely applicable, providing anonymity and confidentiality against all devices external to an exchange without incurring extreme costs.

6.2 Security Analysis

We believe the protocol presented above is secure and satisfies the outlined security properties. Formal verification of the protocol hierarchy has been carried out to confirm this using the model checker FDR [18], along with Casper [19] to automatically generate CSP [20] descriptions. When writing in a language such as CSP, the density and complexity of the code often result in overlooked errors and logic flaws. Using Casper allows us to automate the production of the mathematical model description of the protocol, reducing the liklihood of errors occuring.

A script was created to model each extension, stipulating only the individual security properties claimed above. After each extension's script was checked and passed, the intruder's knowledge was amended to include the secure keys of individual participants. This allowed for the modelling of malicious participants. Even with security intentionally compromised in this manner, the security properties for each extension are upheld, excluding those related to the compromised device. This confirms that the intended security properties are supported and the protocol does not appear to have any weaknesses vulnerable to attack by intruders.

7 Future Work

We intend to analyse the effect of malicious and colluding Proof Providers on the system through simulations of the Verifier's calculations using sample proofs. In addition to this, we intend to analyse the number of proof providers required by the system for a proof to be successful, in both honest and hostile environments.

Another area of focus is the simulation of distance bounding including digital signatures. At present, our system assumes that distance bounding is capable of detecting a terrorist fraud through differenciating between a reasonable time for an honest exchange and the time required for a proxy exchange to complete. However, we intend to confirm this assumption through the simulation of both honest and proxy distance bounding exchanges encorporating digital signatures.

8 Conclusion

In this paper, we discussed the demand for a method of locating a device within a wireless network. We then focused on the problem of location verification and outlined our system for verifying a device's location claim. This allows a Claimant to prove its location to a central verifying agent through gathering proof of its proximity to local devices. Unlike previously existing approaches however, the system does not require trusted agents to act as Proof Providers. Instead, we make use of unknown and untrusted devices local to the Claimant, in possession of a tamper-resistant module, to protect all cryptographic keys. As the system is designed for use in an untrusted environment and the participants in the exchange are also untrusted, we noted that a Claimant merely distance bounding with local devices would not provide valid proofs for use within the system and that a method of protecting the integrity of information passed within it is needed. In order to secure this process from attack by both intruders and malicious participating devices, we have presented the Secure Location Verification Proof Gathering Protocol. We repeatedly extended this protocol to provide increasing levels of security, culminating in the provision of authentication, complete anonymity and confidentiality within the final protocol. This incremental approach to the protocol's design allows for flexibility in terms of cost vs benefit.

References

1. Want, R., Hopper, A., Falcao, V., Gibbons, J.: The active badge location system. ACM Trans. Inf. Syst. 10(1), 91–102 (1992)
2. Sarma, S.E., Weis, S.A., Engels, D.W.: RFID systems and security and privacy implications. In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 454–469. Springer, Heidelberg (2003)
3. Brands, S., Chaum, D.: Distance-bounding protocols (extended abstract). In: Theory and Application of Cryptographic Techniques, pp. 344–359 (1993)
4. Gabber, E., Wool, A.: How to prove where you are: racking the location of customer equipment. In: CCS 1998: Proceedings of the 5th ACM conference on Computer and communications security, pp. 142–149. ACM, New York (1998)

5. Čapkun, S., Čagalj, M.: Integrity regions: authentication through presence in wireless networks. In: WiSe 2006: Proceedings of the 5th ACM workshop on Wireless security, pp. 1–10. ACM, New York (2006)
6. Sastry, N., Shankar, U., Wagner, D.: Secure verification of location claims. Technical report (2003)
7. Waters, B., Felten, E.: Secure, private proofs of location. Technical report, Princeton University (2003)
8. Clulow, J., Hancke, G.P., Kuhn, M.G., Moore, T.: So near and yet so far: Distance-bounding attacks in wireless networks. In: Security and Privacy in Ad-Hoc and Sensor Networks, pp. 83–97. Springer, Heidelberg (2006)
9. Association, I.S.: Ieee standards for information technology – specific requirements – part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. Published online (1999)
10. Rasmussen, K.B., Čapkun, S.: Location privacy of distance bounding protocols. In: CCS 2008: Proceedings of the 15th ACM conference on Computer and communications security, pp. 149–160. ACM Press, New York (2008)
11. Lindqvist, J., Takkinen, L.: Privacy management for secure mobility. In: WPES 2006: Proceedings of the 5th ACM workshop on Privacy in electronic society, pp. 63–69. ACM, New York (2006)
12. Huang, L., Matsuura, K., Yamane, H., Sezaki, K.: Enhancing wireless location privacy using silent period. In: 2005 IEEE Wireless Communications and Networking Conference, pp. 1187–1192 (2005)
13. Gruteser, M., Grunwald, D.: Enhancing location privacy in wireless lan through disposable interface identifiers: a quantitative analysis. Mobile Networks and Applications 10, 315–325 (2005)
14. Desmedt, Y.: Major security problems with the 'unforgeable' (feige)-fiat-shamir proofs of identity and how to overcome them. In: Proceedings of SECURICOM 1988, Sixth Worldwide Congress on Computer and Communications Security and Protection, pp. 147–159 (1988)
15. Mauve, M., Widmer, A., Hartenstein, H.: A survey on position-based routing in mobile ad hoc networks. IEEE Network 15, 30–39 (2001)
16. Josang, A., Ismail, R.: The beta reputation system. In: e-Reality: Constructing the Economy (2002)
17. Haller, N.M.: The S/KEY one-time password system. In: Proceedings of the Symposium on Network and Distributed System Security, pp. 151–157 (1994)
18. Roscoe, A.W.: Modelling and verifying key-exchange protocols using csp and fdr. In: Computer Security Foundations Workshop, p. 98 (1995)
19. Lowe, G.: Casper: A compiler for the analysis of security protocols. In: Computer Security Foundations Workshop IEEE, p. 18 (1997)
20. Brookes, S.D., Hoare, C.A.R., Roscoe, A.W.: A theory of communicating sequential processes. J. ACM 31, 560–599 (1984)