An Architecture and Protocol, an Access Control Model, and a Sighting Blurring Algorithm for Improving Users' Security in the context of Location Based Services Operating on the Internet

Cameron Ross Dunne, B.Sc., M.Sc.

A dissertation presented in partial fulfillment of the requirements for the award of

Doctor of Philosophy

to the

Dublin City University
School of Computing

Supervisor: Dr. David Gray

December 2008

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy (Ph.D.) is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____

Student Number: 96047135

Date : 12/12/2008

Acknowledgements

I would like to thank everybody who helped me in any way during the completion of this thesis.

In particular, I would like to thank Dr. David Gray for his supervision, assistance, and guidance. I would also like to thank Dr. Thibault Candebat for his ongoing support and encouragement.

I would like to thank Enterprise Ireland for its support with this research under grants IF/2002/336 and PC/2004/446.

Finally, I would like to give a special thanks to all my family and friends.

Contents

Abstract

A new type of service, known as a *Location Based Service* (LBS), is emerging that incorporates users' location information, and many of these LBSs operate over the Internet. However, the potential misuse of this location information is a serious concern. Therefore, the main goal of this thesis is to develop techniques, which increase users' security and privacy, for use with these LBSs.

The first technique that we propose is a three-party protocol that is used to mutually identify and authenticate users, LBSs, and a trusted middleware infrastructure that is responsible for managing the users' identity and location information. This protocol enables users to simultaneously identify and authenticate themselves to the infrastructure using real identities, and to the LBSs using pseudonyms. This protocol can be subsequently used to securely exchange messages containing location information.

The second technique that we propose is an access control model that enables users to create permissions that specify which users and LBSs are entitled to obtain location information about which other users, under what circumstances the location information is released to the users and LBSs, and the accuracy of any location information that is released to the users and LBSs.

The third technique that we propose is a blurring algorithm that performs spatial blurring on users' location information. It does not perform temporal blurring, because this reduces an LBS's ability to offer a useful service. Instead, our blurring algorithm introduces a new parameter that specifies the frequency with which location information is released for a particular user. This frequency parameter is a function of the size of the blurred location.

These three techniques can be used as part of an overall solution for providing users with increased security while using LBSs that operate over the Internet.

List of Figures

Chapter 1

Introduction

1.1 Introduction

Currently there are many services that may be invoked using the Internet. In general, these services, and the Internet, have no knowledge or concept of location. However, our usage of the Internet is changing and evolving. We no longer access it solely from desktop computers that are in fixed locations. More and more we access it using mobile devices. Using these mobile devices we are able to maintain our connection to the Internet while on the move. Another recent development is the ability to obtain the mobile device's current location. This can be obtained by either the mobile device itself, or by the network within which it is operating. This connected mobility together with the location awareness enables a new range of services, known as *Location Based Services* (LBSs), which take the mobile device's location into consideration when providing the service.

In this chapter we provide an introduction to mobile devices and positioning technologies. These are both significant enablers of LBSs. We then describe the problem statement that this thesis addresses, together with the main goals of this thesis. Finally, we will outline the unique contributions of this thesis.

1.2 Enablers

The two most significant enablers of LBSs are mobile devices and positioning technologies.

1.2.1 Mobile Devices

We consider that mobile devices in the context of this thesis are widely available personal communication devices that are not restricted to use in a single location. In particular, we consider that mobile devices are either mobile phones, smart phones, or *Personal Digital Assistants* (PDAs). However, as technological advancements are made the differences between these three types of mobile devices decrease, and it becomes increasingly difficult to distinguish between them.

Mobile devices are continuously experiencing significant technological advancements. For example, their size and mass has decreased dramatically, while their computational power, input and output capabilities, and communication bandwidth have all increased rapidly. The vast majority of mobile devices are now capable of connecting to the Internet and viewing web pages. Therefore, they are a significant enabler of LBSs, because they enable users to invoke LBSs regardless of where these users are located.

Current mobile phones are small, lightweight devices, which contain relatively large colour displays, and have integrated digital cameras. They provide users with all the functionality that is needed to make and receive both voice and video calls. They also enable users to avail of a wide range of multimedia services, such as text, picture, and email messaging. They often contain simple applications, such as task lists and calculators.

Smart phones are mobile devices that are built upon the capabilities of mobile phones. Therefore, they contain all the phone functionality found in mobile phones. However, the feature of smart phones that differentiates them from standard mobile phones is the fact that smart phones contain an accessible operating system that enables users to install their own applications. Since smart phones contain more functional operating systems, they generally require more powerful hardware than

standard mobile phones. Consequently, smart phones are often heavier and larger than mobile phones.

Originally PDAs were designed to replace the traditional paper based personal organisers. They normally contain an operating system, and they have diary, contacts list, task list, and notes applications. Current PDAs have the same capabilities as smart phones, but they normally have much larger screens. Users normally interact with PDAs by either touching their screens directly or by using a stylus on their screens. PDAs are the most powerful and feature rich mobile devices, and therefore they are the heaviest and largest mobile devices.

The ownership and usage of mobile devices has increased dramatically in recent years. By the middle of 2008 the mobile device penetration rate in Ireland was 121%, and the mobile device penetration rate in European was 118% [23]. Indeed, mobile devices have become an integral part of our lives!

1.2.2 Positioning Technologies

Positioning technologies make it possible to locate mobile devices, and there have been significant advancements in this area recently. They are the single most important enablers of LBSs, because without them it is almost impossible to create meaningful LBSs. The positioning technologies that are most suitable for use with LBSs are scalable, relatively cheap, and easily integrated with mobile devices. There are two broad categories of positioning technologies that have these properties. These are satellite based positioning technologies and mobile phone network based positioning technologies.

The basic concept of the satellite based positioning technologies is that a location in three-dimensional space can be precisely located by finding the intersection of four spheres. The centre of each sphere represents a satellite, and the radius of the sphere represents the distance to the positioning device. The most widely available global satellite based positioning system is the *Global Positioning System* (GPS) [59]. It was developed by the American Department of Defence to enable the military forces to accurately determine their position, their velocity, and the time, using a common reference system anywhere on or above the Earth. GPS was subsequently made

available for use by the general public, although it still remains under military control. The main disadvantage of GPS in the context of LBSs is that it only works well in environments where the positioning device has a clear view of the sky. Therefore, it does not work well in built-up urban areas, and it does not work inside buildings.

The mobile phone network based positioning technologies are based on mobile phone network infrastructures rather than satellite infrastructures. Mobile phone network operators throughout the world have established cellular networks, and these mobile phone networks cover vast areas of land. In particular, they normally cover areas of land that are populated by humans. Therefore, these networks offer the potential of being able to locate mobile devices that are operating within them. The network operators have recognised this potential since the development of the initial GSM networks. Since then, many different methods for locating mobile devices within GSM and UMTS networks have been developed [1]. For example, both the *Time of Arrival* method and the *Enhanced Observed Time Difference* method calculate the mobile device's location by measuring the times required for signals to travel between the mobile phone network infrastructure and the mobile device. These times are used to calculate distances, and the mobile device's location is obtained by calculating the intersection of these distances from the mobile phone network infrastructure. The main disadvantages of these methods are that they require upgrades to both the users' mobile devices and the mobile phone network infrastructure, and their accuracy varies depending on the mobile phone network topology.

1.3 Location Based Services

These new mobile devices, which can be combined with new positioning technologies, enable a completely new type of service to be created. This new type of service is known as an LBS. These LBSs can provide users with information that may be tailored to their current locations. Alternatively, these LBSs can tailor the information based on the location of some other person of interest to the user.

In practice there are many different forms of LBSs, therefore a more general definition of an LBS is:

> *An LBS is a service that takes into consideration the location of the user who invoked the service, and/or the location or locations of some other entity or entities, when it is providing the service.*

An interesting aspect of this definition is that the type of entity or entities that a user may be interested in locating is very broad. Possible examples of these entities include other people, animals, vehicles, objects, and places.

There are many existing examples of LBSs that demonstrate their potential. For example, one of the first LBSs developed was a tourist guide called *Cyberguide* [2]. This LBS enables users in Atlanta to view their current location, and to locate nearby *Points of Interest* (PoI). For each PoI, they can obtain related information, and leave comments for other users to view. They can also create customised tours of the city for themselves based on their preferences. Similar tourist guide LBSs have also been developed for Lancaster [22] and Adelaide [69].

A conference assistant is presented in [25] that forms the basis of an LBS. The user invokes this LBS when he/she first enters the conference venue. After registering his/her contact details, the user specifies his/her research interests. The user can also inform the LBS which other attendees are his/her colleagues. Throughout the conference, the LBS provides the user with location details, and background material, for presentations and demonstrations that may be of interest.

There are currently several companies offering vehicle tracking LBSs on a commercial basis. These LBSs are designed for individuals with valuable cars, or for company fleets. They allow the vehicles to be located in real-time using a web interface.

Similarly, there are several companies offering child tracking LBSs on a commercial basis. These LBSs enable parents to locate their children in real-time using a web interface. The children who are locatable must be in possession of a mobile device or some purpose built hardware that contains positioning technology.

1.4 Problem Statement and Goals

Security, and in particular privacy, continues to be the main concern in the area of eCommerce [32], and it is also critical for mCommerce, particularly in the context of LBSs. Users tend to be reluctant to provide personal location information, which we refer to as sightings, to third party LBSs [9], and they may require significant compensation for these sightings [24]. In fact, sightings are so critical to a user's privacy that they are specifically covered by the EU directive on the processing and storage of personal data [31]. Potential misuse of sightings raises some important issues. Sightings may fall into the hands of someone with malicious intentions, and this may place users in life threatening situations. Therefore, users must be able to trust network operators, LBSs, and other users not to misuse their sightings.

The existing LBSs that operate over the Internet are based on a trust model where there is complete trust between both the user and the network operator, and between the network operator and the LBS. Indeed, there are very few of these LBSs that operate over the Internet because it is difficult for them to obtain the trust of the network operator. Furthermore, once an LBS has obtained the trust of the network operator then the LBS can obtain accurate sightings of any user at any time. Therefore, the current trust model between network operators and LBSs is an *all-or-nothing* trust model. Consequently, the trust model between users and LBSs is also an all-or-nothing trust model.

A more desirable approach is to create a new trust model that reduces and limits the trust required between the network operator and the LBS. This in turn would reduce and limit the trust required between the user and the LBS. The effect of this proposed trust model is that it would become easier for LBSs to obtain sightings of users, since the users would have control over the release of their sightings. This would empower the rapid creation of Internet based LBSs. Indeed, it is possible that any existing website could become LBS enabled. For example, an existing website belonging to a bank could become an LBS by enabling users to locate their nearest automatic teller machines.

The goal of creating a new trust model that reduces and limits the amount of

trust that users must have in network operators, LBSs, and other users is extremely broad. It is also possible that it can never be fully realised. The most significant issue associated with achieving this goal is the secure management of identity information and location information. Therefore, the main goals of this thesis are to develop security techniques for use with LBSs. In particular, we will describe an architecture and protocol, an access control model, and a sighting blurring algorithm which all increase users' security.

1.4.1 Architecture and Protocol

One important security feature that enables users to reduce the amount of trust that they must place in LBSs is the ability to hide their identities, and hence access the LBSs in an anonymous manner. This will not hinder many LBSs from offering a useful service, such as LBSs that enable a user to locate the nearest petrol station. However, many LBSs may require some form of persistent identification for users, such as LBSs that enable a user to locate his/her friends. In these circumstances users can identify themselves to the LBSs using *pseudonyms* [65].

In order for a user to successfully invoke an LBS over the Internet several different entities will need to communicate with each other. This creates opportunities for a party with malicious intentions to either eavesdrop on the communications, or to interfere with the communications. The risks posed by these opportunities can be reduced by using security techniques to ensure that all entities are correctly identified and authenticated to each other, and that the confidentiality, integrity, and non-repudiation of all subsequent communications can be ensured.

Therefore, the first goal of this thesis is to develop an architecture and protocol that enables users with multiple identities to securely communicate with LBSs and other users.

1.4.2 Access Control Model

Another important security feature that enables users to reduce the amount of trust that they must place in both LBSs and other users is the ability to control the release

of their sightings by using an access control model. A suitable access control model should be based on a system where users create permissions that specify which LBSs and other users are entitled to obtain their sightings, under what circumstances these sightings are obtainable, and the maximum accuracy of these sightings. The access control model is then responsible for releasing users' sightings in accordance with their permissions.

Therefore, the second goal of this thesis is to develop an access control model that controls the release of users' sightings.

1.4.3 Sighting Blurring Algorithm

The final security feature that we consider that enables users to reduce the amount of trust that they must place in both LBSs and other users is the ability to reduce the accuracy of their sightings. We refer to this process as *sighting blurring*. Many LBSs will not be hindered from offering useful services by sighting blurring, because these LBSs will not require very accurate sightings. For example, an LBS that enables a user to locate all touristic attractions within 100,000m does not need the user's sightings with an accuracy greater than 1,000m.

Therefore, the third goal of this thesis is to develop a sighting blurring algorithm that reduces the accuracy of users' sightings.

1.5 Thesis Contributions

There are three major contributions of our work that are described in this thesis.

1.5.1 Architecture and Protocol

We have developed an architecture for users, LBSs, and an infrastructure, which is a trusted middleware entity that facilitates the operation of these LBSs over the Internet in a secure manner. In particular, the infrastructure provides common functionality regarding the users' identity information and sighting information.

We have also developed a protocol that uses this architecture, so that users, LBSs, and an infrastructure can achieve three-party mutual identification and au-

thentication. This protocol allows users to simultaneously identify and authenticate themselves to the infrastructure using a real identity, and to the LBS using a pseudonym. Indeed, each user can be identified and authenticated using a different pseudonym with each LBS. This is achieved without requiring the mobile device to have significantly greater resources, and without the need for additional messages to be exchanged. This usage of pseudonyms with LBSs provides users with increased privacy without necessarily reducing the usefulness of the LBSs. The confidentiality, integrity, and non-repudiation of all subsequent messages can be guaranteed, and we described how this can be used so that a user receives sighting information from an LBS, which in turn receives this sighting information from the infrastructure.

1.5.2 Access Control Model

We have developed an access control model that is implemented within the infrastructure within our architecture. Our access control model is based on a system where users create permissions that specify who is entitled to obtain their sightings, under what circumstances these sightings are obtainable, and the maximum accuracy of these sightings. Our access control model is then responsible for releasing users' sightings in accordance with their permissions. Our architecture assumes that a user, who is requesting sighting information about another user, communicates directly with the LBS that provides the relevant service. This LBS then communicates with the infrastructure, in order to obtain the necessary sightings. Therefore, both the user and the LBS are seeking sightings from the point-of-view of the infrastructure. This in effect creates two different *subjects* in the context of an access control model.

The main novelty of our access control model is that it enables users to specify two different types of permission. The first type of permission is used to specify which users are trusted, and therefore can obtain sighting information, and the second type of permission is used to specify which LBSs are trusted, and therefore can obtain sighting information. This has the effect of creating a *whitelist* for users, and a separate whitelist for LBSs. The access control model will only allow sightings to be released if it is presented with both a valid permission specified in terms of

users, and a valid permission specified in terms of LBSs.

1.5.3 Sighting Blurring Algorithm

We have developed a sighting blurring algorithm that offers users increased privacy by performing spatial blurring on the location components of their sightings. Instead of performing temporal blurring, our sighting blurring algorithm will not produce any new blurred sightings of the user until a specific duration has elapsed. This has the effect of limiting the frequency with which new sightings can be obtained for a user, and this frequency is a function of the area of the location component of the sighting that was produced by the spatial blurring.

There are three advantages of our sighting blurring algorithm compared to the sighting blurring algorithms presented in the related research. Firstly, our sighting blurring algorithm is designed to be resistant to mathematical attacks based on frequent sighting requests. Secondly, our sighting blurring algorithm generates blurred sightings with a consistent sighting accuracy. Thirdly, our sighting blurring algorithm is only dependent on the sightings of the users that are directly involved in a current LBS invocation, and it is independent of the sightings of all other users.

1.6 Thesis Outline

The remainder of this thesis is structured as follows:

- In Chapter 2 we describe the background concepts that are relevant to this thesis. In particular, we describe the cryptographic concepts and the charging concepts that we build upon in Chapter 4, and we describe the location concepts that we build upon in both Chapter 4 and Chapter 6.

- In Chapter 3 we describe research efforts that are related to LBSs in terms of architectures and protocols, access control models, and sighting blurring algorithms.

- In Chapter 4 we describe an architecture for users, an infrastructure, and LBSs, which facilitates the operation of these LBSs over the Internet. We

also describe a protocol that enables these three entities to achieve three-party mutual identification and authentication. In particular, this protocol allows users to simultaneously identify and authenticate themselves to the infrastructure using one identity, and to the LBS using another identity. This protocol then guarantees the confidentiality, integrity, and non-repudiation of all subsequent messages.

- In Chapter 5 we describe an access control model that is implemented within the infrastructure. This access control model is based on users who are targets creating permissions for users who are indirect requesters, and for LBSs that are proxy requesters. These permissions specify which requesters are entitled to obtain sightings of which targets, under what circumstances these sightings are released, and the maximum accuracy of these sightings.

- In Chapter 6 we describe the sighting blurring algorithm that we have developed, which is implemented within the infrastructure in our architecture. Our sighting blurring algorithm offers users increased privacy by decreasing the accuracy of their sightings based on the sighting accuracy allowed by the access control model described in Chapter 5.

- In Chapter 7 we form our conclusions to this thesis. In particular, we identify both the major contributions and the minor contributions of this thesis in terms of our architecture and protocol, our access control model, and our sighting blurring algorithm. We also identify several areas of the work described in this thesis that require further investigation. Finally, we identify the publications that have arisen as a result of the work described in this thesis.

The work described in this thesis complements the work described in the thesis written by my colleague Thibault Candebat [17].

Chapter 2

Background

2.1 Introduction

In this chapter we describe the background concepts that are relevant to this thesis. In particular, we describe the cryptographic concepts and the charging concepts that we build upon in Chapter 4, and we describe the location concepts that we build upon in both Chapter 4 and Chapter 6.

2.2 Cryptography

In this section we describe the notation, the authentication protocol, and the cryptography system that we build upon in Chapter 4.

2.2.1 Notation

In order to describe the cryptographic techniques and protocols, we introduce the following notation:

- A and B are two entities who wish to communicate with each other.
- M, M_1, M_2, ... M_n are messages. These can be combined together to form fewer larger messages, and then split apart to reform the original messages. This can be done in any standard and consistent manner.

- $A \rightarrow B : M$ denotes that A prepares the message M and then sends it to B, and that B receives M and then processes it.
- N_A is a nonce generated by A for use by anybody, and $N_{A,B}$ is a nonce generated by A for use by B.
- $T_{A,B}$ is a timestamp created by A for use by B.
- $K_{A,B}$ is a key used for symmetric cryptography between A and B.
- $\{M_1, M_2, M_3, \ldots M_n\}_{K_{A,B}}$ represents the combined message of M_1 to M_n encrypted using the symmetric key shared between A and B.
- K_A represents the asymmetric key pair belonging to A. This consists of the public key K_A^+, and the private key K_A^-.
- $\{M_1, M_2, M_3, \ldots M_n\}_{K_A^+}$ represents the combined message of M_1 to M_n encrypted using the public key of A.
- $\{|M_1, M_2, M_3, \ldots M_n|\}_{K_A^-}$ represents the combined message of M_1 to M_n signed without message recovery using the private key of A.

2.2.2 X.509 Authentication

The *X.509 Public Key Infrastructure* (PKI) [46] describes several different authentication protocols. These protocols provide identification and authentication between two entities, and they also guarantee the confidentiality, integrity, and non-repudiation of messages exchanged. This is achieved using asymmetric cryptography whereby the entities demonstrate that they possess the correct private keys.

The X.509 authentication protocol that is of most interest to us is the *Two-way Authentication* protocol. This is a two step authentication protocol that is based on A and B both sending a single message to each other. Two-way Authentication is used to establish the following:

- The identities of both A and B.

- The first message was actually generated by A, and the second message was actually generated by B. Therefore, both A and B are authenticated. This provides mutual authentication.
- The first message was actually intended for B, and the second message was actually intended for A.
- The integrity of both messages.
- The freshness of both messages.

Two-way Authentication consists of the following steps:

1. A generates $N_{A,B}$.
2. A sends the following message to B:

$$T_{A,B}, N_{A,B}, B, \{|T_{A,B}, N_{A,B}, B|\}_{K_A^-}$$

A can include an additional message (M_1) that enables B to create a shared session key that can be used for further communications. In this case A sends the following message to B:

$$T_{A,B}, N_{A,B}, B, \{M_1\}_{K_B^+}, \{|T_{A,B}, N_{A,B}, B, \{M_1\}_{K_B^+}|\}_{K_A^-}$$

A can also include an additional message (M_2) for B, which B is capable of verifying. In this case A sends the following message to B:

$$T_{A,B}, N_{A,B}, B, M_2, \{|T_{A,B}, N_{A,B}, B, M_2|\}_{K_A^-}$$

A can combine these messages to create a single message for B. In this case A sends the following message to B:

$$T_{A,B}, N_{A,B}, B, \{M_1\}_{K_B^+}, M_2, \{|T_{A,B}, N_{A,B}, B, \{M_1\}_{K_B^+}, M_2|\}_{K_A^-}$$

3. B obtains K_A^+, and asserts that it is valid based on the certificate belonging to A. B then verifies the integrity of the signed message.

4. B checks that it is the intended recipient.

5. B checks that $T_{A,B}$ is current.

6. B checks that the message has not been replayed by establishing the uniqueness of $N_{A,B}$.

7. If $\{M_1\}_{K_B^+}$ is present, then B decrypts it, and uses M_1 to generate $K_{A,B}$. $K_{A,B}$ is then used to symmetrically encrypt and decrypt the remainder of the session.

8. B generates $N_{B,A}$.

9. B sends the following message to A:

$$T_{B,A}, N_{B,A}, A, N_{A,B}, \{|T_{B,A}, N_{B,A}, A, N_{A,B}|\}_{K_B^-}$$

B can include an additional message (M_3) that enables A to create a shared session key that can be used for further communications. In this case B sends the following message to A:

$$T_{B,A}, N_{B,A}, A, N_{A,B}, \{M_3\}_{K_A^+}, \{|T_{B,A}, N_{B,A}, A, N_{A,B}, \{M_3\}_{K_A^+}|\}_{K_B^-}$$

B can also include an additional message (M_4) for A, which A is capable of verifying. In this case B sends the following message to A:

$$T_{B,A}, N_{B,A}, A, N_{A,B}, M_4, \{|T_{B,A}, N_{B,A}, A, N_{A,B}, M_4|\}_{K_B^-}$$

B can combine these messages to create a single message for A. In this case B sends the following message to A:

$$T_{B,A}, N_{B,A}, A, N_{A,B}, \{M_3\}_{K_A^+}, M_4, \{|T_{B,A}, N_{B,A}, A, N_{A,B}, \{M_3\}_{K_A^+}, M_4|\}_{K_B^-}$$

10. A obtains K_B^+, and asserts that it is valid based on the certificate belonging to B. A then verifies the integrity of the signed message.

11. A checks that it is the intended recipient.

12. A checks that $T_{B,A}$ is current.

13. A checks that the message has not been replayed by establishing the uniqueness of $N_{B,A}$.

14. A checks that the $N_{A,B}$ that was received is the same as the $N_{A,B}$ that was sent.

15. If $\{M_3\}_{K_A^+}$ is present, then A decrypts it, and uses M_3 to generate $K_{A,B}$. $K_{A,B}$ is then used to symmetrically encrypt and decrypt the remainder of the session.

Syntactic Abbreviations

We define syntactic abbreviations for each of the variations of the two messages in this protocol, and we will use these syntactic abbreviations in the remainder of this thesis. The first message ($\mathcal{M}$) that is sent from A to B is:

$$\begin{aligned}
\mathcal{M}(A, B, K_A^-) &\triangleq T_{A,B}, N_{A,B}, B, \\
&\quad \{|T_{A,B}, N_{A,B}, B|\}_{K_A^-} \\
\mathcal{M}(A, B, K_A^-, M_2) &\triangleq T_{A,B}, N_{A,B}, B, M_2, \\
&\quad \{|T_{A,B}, N_{A,B}, B, M_2|\}_{K_A^-} \\
\mathcal{M}(A, B, M_1, K_B^+, K_A^-) &\triangleq T_{A,B}, N_{A,B}, B, \{M_1\}_{K_B^+}, \\
&\quad \{|T_{A,B}, N_{A,B}, B, \{M_1\}_{K_B^+}|\}_{K_A^-} \\
\mathcal{M}(A, B, M_1, K_B^+, K_A^-, M_2) &\triangleq T_{A,B}, N_{A,B}, B, \{M_1\}_{K_B^+}, M_2, \\
&\quad \{|T_{A,B}, N_{A,B}, B, \{M_1\}_{K_B^+}, M_2|\}_{K_A^-}
\end{aligned}$$

The second message ($\mathcal{M}'$), which is sent from B to A in response to $\mathcal{M}$, is:

$$\begin{aligned}
\mathcal{M}'(B, A, K_B^-) &\triangleq T_{B,A}, N_{B,A}, A, N_{A,B}, \\
&\quad \{|T_{B,A}, N_{B,A}, A, N_{A,B}|\}_{K_B^-} \\
\mathcal{M}'(B, A, K_B^-, M_4) &\triangleq T_{B,A}, N_{B,A}, A, N_{A,B}, M_4, \\
&\quad \{|T_{B,A}, N_{B,A}, A, N_{A,B}, M_4|\}_{K_B^-} \\
\mathcal{M}'(B, A, M_3, K_A^+, K_B^-) &\triangleq T_{B,A}, N_{B,A}, A, N_{A,B}, \{M_3\}_{K_A^+}, \\
&\quad \{|T_{B,A}, N_{B,A}, A, N_{A,B}, \{M_3\}_{K_A^+}|\}_{K_B^-} \\
\mathcal{M}'(B, A, M_3, K_A^+, K_B^-, M_4) &\triangleq T_{B,A}, N_{B,A}, A, N_{A,B}, \{M_3\}_{K_A^+}, M_4, \\
&\quad \{|T_{B,A}, N_{B,A}, A, N_{A,B}, \{M_3\}_{K_A^+}, M_4|\}_{K_B^-}
\end{aligned}$$

2.2.3 Mediated Identity Based Cryptography

Candebat has developed a PKI in the context of LBSs that is based on a mediated identity based cryptography system [17, 18, 19]. The *Identity Based Encryption* (IBE) and the *Identity Based Signature* (IBS) aspects of the mediated identity based cryptography system are based on work described in [68] and [43]. The mediated aspects of the mediated identity based cryptography system are based on work described in [8] and [21]. The mediated identity based cryptography system provides asymmetric encryption, asymmetric decryption, signing, and verification services. Its main advantages compared to a traditional cryptography system such as X.509 are:

- The revocation of keys occurs instantly.
- There is no need for entities to retrieve and validate revocation information. This process is usually costly in terms of both bandwidth and processing power.
- All public keys are strings that represent the names of the entities that own them. This identity based cryptography simplifies the management of public keys significantly.

The architecture of the mediated identity based cryptography system requires an additional two entities:

- **The Security Mediator**

  The security mediator assists the entities in the system by partially decrypting ciphertexts and by partially signing plaintexts while the keys remain valid. The security mediator is not needed for encrypting plaintexts or verifying signatures. Since the security mediator is required to perform these actions between the two entities that are communicating with each other, it is designed to operate as a middleware component.

- **The Private Key Generator**

  The private key generator is responsible for generating each entity's private key. This private key consists of two private key shares. The security mediator uses one of these private key shares, and the entity uses the other private key share. For example, if the entity's name is A then its public key is K_A^+. The private key generator will generate the private key shares k_A^- and $k_{A,SM}^-$. k_A^- is then used by A, and $k_{A,SM}^-$ is used by the security mediator.

In order to describe the cryptographic processes of the mediated identity based cryptography system we will extend the notation that was defined previously by defining P as a plaintext, C as a ciphertext, and S as a signature. The encryption process, $\mathcal{E}$, consists of a single step involving the public key belonging to A. This process is defined as:

$$\mathcal{E}_{K_A^+}(P) = C$$

The decryption process, $\mathcal{D}$, consists of two steps. Firstly, the ciphertext is decrypted by the security mediator using its private key share for A. Secondly, the result of this decryption is again decrypted by A using its private key share. This process is defined as:

$$\mathcal{D}_{k_A^-}(\mathcal{D}_{k_{A,SM}^-}(C)) = P$$

The signing process, $\mathcal{S}$, also consists of two steps. Firstly, the plaintext is partially signed by A using its private key share. Secondly, the signature is completed by the security mediator using its private key share for A. This process is defined as:

$$\mathcal{S}_{k^-_{A,SM}}(\mathcal{S}_{k^-_A}(P)) = S$$

The verification process, $\mathcal{V}$, consists of evaluating the signature using the public key belonging to A. This will be true if the signature is valid, or false if the signature is invalid. This process is defined as:

$$\mathcal{V}_{K^+_A}(S) = True \text{ or } False$$

The mediated identity based cryptography system consists of three keys for each entity. For example, the keys associated with A are K^+_A, k^-_A and $k^-_{A,SM}$. However, the cryptographic processes have the same effect as using virtual keys in a traditional cryptography system. In particular, we introduce the virtual keys K^-_A and k^+_A. The keys associated with A are then K^+_A, K^-_A, k^+_A and k^-_A.

In order to demonstrate the usage of these keys, we will explain the operation of our mediated identity based cryptography system in situations where A and B communicate with each other. The encryption and decryption processes are as follows:

1. A encrypts the message, M, using the public key belonging to B, to produce $\{M\}_{K^+_B}$.
2. A sends $\{M\}_{K^+_B}$ to the security mediator.
3. The security mediator checks the status of the keys belonging to B. If they have been revoked, then the security mediator generates an appropriate error message, and the procedure is terminated. Otherwise, the security mediator partially decrypts $\{M\}_{K^+_B}$ using $k^-_{B,SM}$. This in effect creates $\{M\}_{k^+_B}$.
4. The security mediator sends $\{M\}_{k^+_B}$ to B.

5. B decrypts $\{M\}_{k_B^+}$ using k_B^- to reveal M.

The signing and verification processes are as follows:

1. A partially signs M using k_A^- to produce $\{|M|\}_{k_A^-}$.
2. A sends $\{|M|\}_{k_A^-}$ to the security mediator.
3. The security mediator checks the status of the keys belonging to A. If they have been revoked, then the security mediator generates an appropriate error message, and the procedure is terminated. Otherwise, the security mediator completes the signing of $\{|M|\}_{k_A^-}$ using $k_{A,SM}^-$. This in effect creates $\{|M|\}_{K_A^-}$.
4. The security mediator sends $\{|M|\}_{K_A^-}$ to B.
5. B verifies $\{|M|\}_{K_A^-}$ using K_A^+.

It is important to note that neither an entity using the mediated identity based cryptography system, nor the security mediator, can recover a plaintext from a ciphertext, or generate a valid signature, on its own.

2.3 Location

In this section we describe the location representation types that we reference in Chapter 3, the two mathematical location representations that we use in Chapter 6, and the location protocols that we use in Chapter 4.

2.3.1 Location Representation Types

There are many different location representations that can be used to describe a single location. However, all of these location representations can be categorised into three different types of location representations:

- The *mathematical location representations* are all location representations that represent locations mathematically. For example, two mathematical location representations that are based on coordinate reference systems are described in Section 2.3.2.

- The *political location representations* are all location representations that represent locations using a name that was created by, and is meaningful to, humans. Many of these political location representations are standardised and are known publicly, such as street names, region names, or county names. However, it is possible to create private political location representations. An example of this occurs when a courier company divides a country into locations based on their nearest depots. Political representations of locations are often referred to as *symbolic locations*.

- The *logical location representations* are all location representations that represent locations using the relationship that the location has with a particular person. Therefore, it is impossible to determine the mathematical location or political location of a logical location without some additional information, such as the identity of the person being located. Examples of logical representations of locations are *"Home"* and *"Work"*.

Each of these types of location representations is commonly used in everyday circumstances.

2.3.2 Mathematical Location Representations

There are two mathematical location representations that are relevant to the work described in this thesis, and both of these mathematical location representations are based on coordinate reference systems.

WGS 84

Coordinate reference systems provide a means of describing a location in relation to the Earth [58]. For example, once such coordinate reference system describes locations in terms of their latitude, longitude, and altitude coordinates. The latitude is the angle that is created between the location and the plain that the equator lies on, the longitude is the angle that is created between the location and the plain that the Greenwich Meridian lies on, and the altitude is the height above sea level. This concept is shown in Figure 2.1.

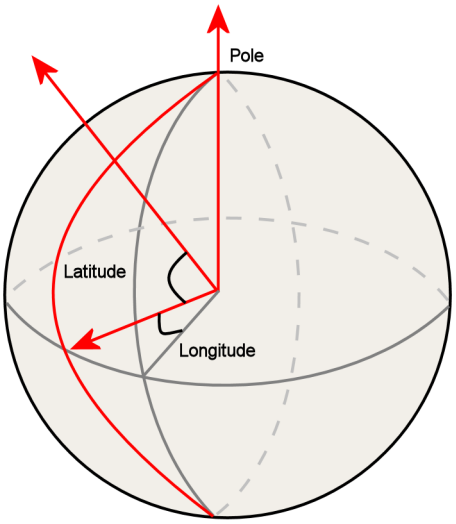

Figure 2.1: Example Coordinate Reference System

Regardless of which coordinate reference system is used, there are still problems relating the coordinate reference system to the Earth. For example, the sea level used to define altitude is constantly changing. In order to relate the coordinate reference system to the Earth, the shape of the Earth is modelled mathematically using an ellipsoid.

Therefore, to fully describe a location on, or near, the surface of the Earth the coordinates of the location, as well as the ellipsoid modelling the Earth, are needed. Together these two pieces of information are known as a *geodetic datum*.

The most relevant geodetic datum to the work described in this thesis is the *WGS 84* geodetic datum [54]. This was originally developed by the American Department of Defence with the design goal of providing a fairly high level of precision everywhere in the world, and it is the geodetic datum that is used by GPS.

Irish Grid

The *Irish Grid* is a coordinate reference system that is based on an imaginary surface that is centred over the centre of Ireland [57]. However, in order to keep the scales positive its true origin is transposed to a false origin southwest of Ireland. Locations in Ireland are then described in terms of their north and east distances from the

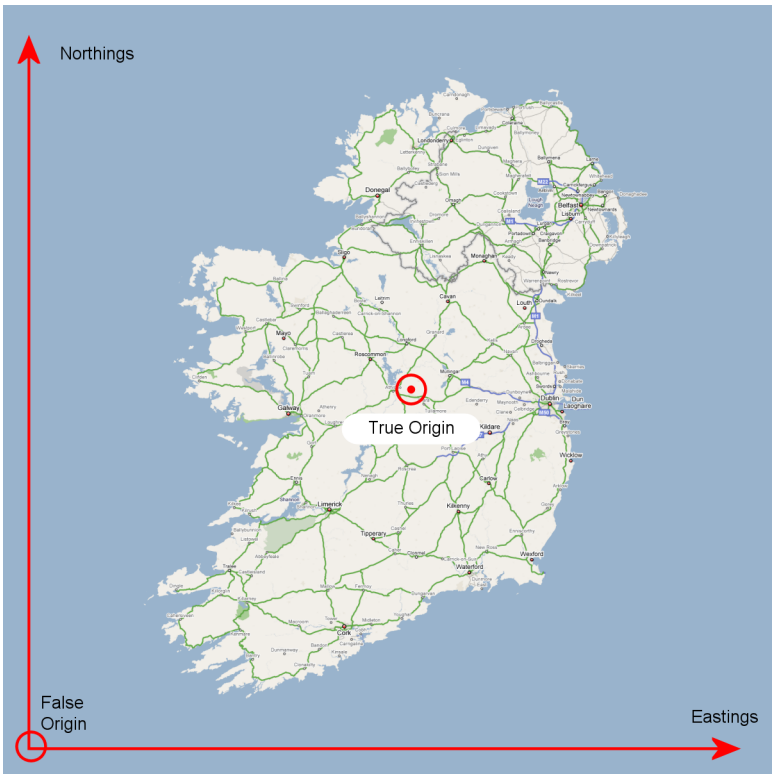

Figure 2.2: The Irish Grid (Map © Google Maps)

origin in metres. These distances are known as *Northings* and *Eastings*. The concept of the *Irish Grid* is shown in Figure 2.2.

2.3.3 Location Protocols

There are several different Internet based protocols that enable third parties to obtain location information about users' mobile devices from network operators. These protocols enable third parties to obtain this location information in a consistent manner regardless of the types of users' mobile devices, and the positioning technology used to locate them. The most popular of these protocols provide similar services, and there are no significant differences between them.

Mobile Location Protocol

The *Open Mobile Alliance* (OMA) is a consortium of companies, whose main aim is to develop and promote open standards within the mobile phone industry [55]. They

developed a standard, known as the *Mobile Location Protocol* (MLP), for exchanging location information about users [56]. The MLP consists of several services, and each service consists of a number of messages.

The most relevant service to the work described in this thesis is the *Standard Location Immediate Service*. This service is used to immediately obtain the location of one of more mobile devices. The third party initiates this service by sending a *Standard Location Immediate Request* message to the network operator, and the network operator returns a *Standard Location Immediate Answer* message, which contains the locations of all the requested mobile devices, to the third party.

Parlay

The *Parlay Group* is a consortium of companies, whose main aim is to develop and promote open standards within the telecommunications industry [60]. Its members come from many different areas, such as equipment manufacturers, network operators, and software developers. It focuses on standards that enable third parties to access and use network operators' resources in a secure, measurable, and chargeable manner. For example, these resources can be related to call control, SMS, MMS, charging, and location. Using these standards provides benefits to all parties involved. Equipment manufacturers are more likely to sell their hardware, because it supports an open standard. Network operators can generate extra revenues from their existing hardware. Software developers can develop applications regardless of the underlying hardware. Finally, users experience a better service because there are more applications available to them.

The Parlay Group has developed an open API called the *Parlay API*, and this API provides several methods that allow third parties to obtain location information about users' mobile devices [62].

However, in recent years the web services model has experienced widespread adoption. In order to benefit from this, the Parley Group developed a new set of standards that enable third parties to access and use the network resources using the web services model. These new standards are known as the *Parlay X Web Services*, and they are the most relevant standards to the work described in this thesis.

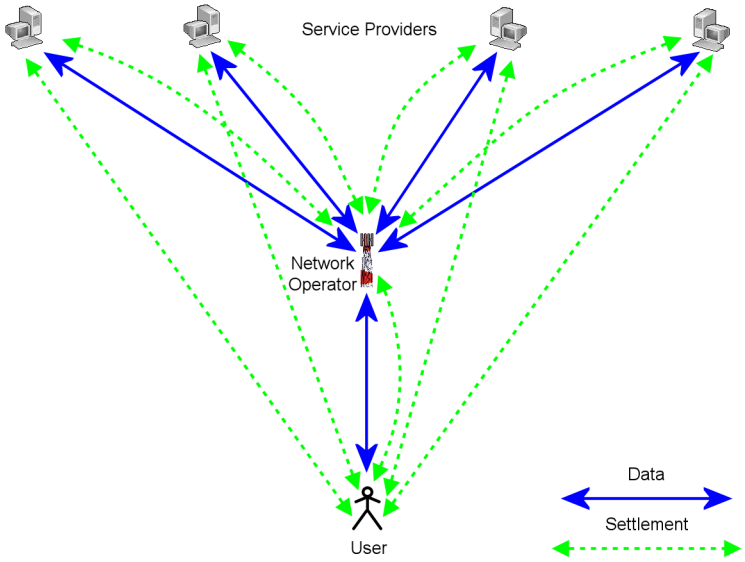

Figure 2.3: Traditional Settlement Model

The *TerminalLocation* service enables third parties to immediately obtain location information about users' mobile devices from network operators [64]. In particular, this service contains a *getLocation* method that is used to obtain the location of a single mobile device, and a *getLocationForGroup* method that is used to obtain the locations of a group of mobile devices.

2.4 Charging

In this section we describe the revenue sharing settlement model and charging protocols that we build upon in Chapter 4.

2.4.1 Revenue Sharing Settlement Model

Settlement is the process where one entity makes a payment to another entity for the charges resulting from the provision of a service. In the traditional settlement model, users had completely separate settlement arrangements with network operators and service providers, as shown in Figure 2.3.

However, there are several disadvantages with this approach. The user must establish a new settlement arrangement with every new service provider whose ser-

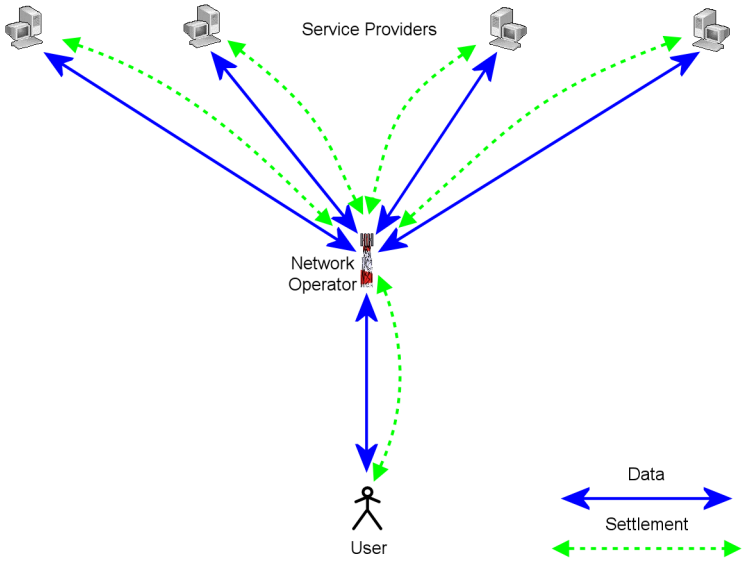

Figure 2.4: Revenue Sharing Settlement Model

vices he/she wishes to use, and this normally involves a significant amount of effort. A consequence of these settlement arrangements is that it is impossible for the user to remain anonymous to the service provider. Finally, as more users start using services, and more service providers become available, the total number of settlement arrangements between users and service providers grows rapidly.

An alternative settlement model, which overcomes these disadvantages, is the revenue sharing settlement model. This settlement model involves one party collecting revenue from the user, and sharing it with the other parties with which the user interacts, as shown in Figure 2.4. Normally, the parties that share revenue are offering services at different levels in the value chain. The result of only one party collecting revenue from the user is that he/she receives a single bill covering all the charges that have been incurred, regardless of how many parties were involved in providing the service. This is often referred to as *one stop billing*.

The relationship between the network operators and the service providers makes them very good candidates for using the revenue sharing settlement model. There are several reasons for this:

- The network operators already have established charging processes. Therefore,

they can extend these existing charging processes to support the new charging requirements of the service providers.

- The network operators already have established settlement arrangements with their customers. Therefore, no additional settlement arrangements need to be established in order to collect extra revenues from these users.
- Making payments is effortless from the users' points of view. This greatly increases the likelihood of users paying for using services.

This settlement model is already widely used within the telecommunications industry. A popular example of its use occurs between network operators and service providers that provide premium rate services such as technical support, competitions, and adult services.

2.4.2 Charging Protocols

There are several different Internet based protocols that enable third parties to exchange charging information, which relates to users, with network operators. Therefore, these protocols are suitable for implementing a revenue sharing settlement model. Third parties can use these protocols to exchange charging information in a consistent manner regardless of the types of commercial relationships that the users have with the network operators.

Parlay

The most relevant charging protocol to the work described in this thesis is the charging protocol developed by the Parlay Group. The Parlay API enables third parties to use many different charging services [61]. However, the Parlay X Web Services for charging are more relevant to the work described in this thesis [63].

The *Amount Charging* service is used to immediately apply charging details, which are expressed in monetary terms, to a user's account. The most relevant methods in this service are the *chargeAmount* method and the *refundAmount* method. Both of these methods require the user's identity, the amount to charge or refund, and a description of the charge or refund.

The *Volume Charging* service is used to immediately apply charging details, which are expressed in terms of a volume of some non-monetary unit, to a user's account. The most relevant methods are the *chargeVolume* method and the *refundVolume* method. Both of these methods require the user's identity, the volume to charge or refund, and a description of the charge or refund. There is also a *getAmount* method that is used to get the monetary value of a particular volume of some non-monetary unit for a particular user.

The *Reserve Amount Charging* service is used to do reservation charging in monetary terms. The *reserveAmount* method is used to reserve an amount of money in the user's account for use with this service. This method requires the user's identity, the amount to reserve, and a description of the reservation. The *reserveAdditionalAmount* method is used to either increase or decrease an existing reservation. The *chargeReservation* method is used to apply the charge to the user's account. Finally, the *releaseReservation* method is used when the user will not incur any further charges for this service.

The *Reserve Volume Charging* service is used to do reservation charging in terms of a volume of some non-monetary unit. The *reserveVolume* method is used to reserve a charge, which is expressed as a volume of some non-monetary unit, in the user's account for use with this service. This method requires the user's identity, the amount to reserve, and a description of the reservation. The *reserveAdditionalVolume* method is used to either increase or decrease an existing reservation. The *chargeReservation* method is used to apply the charge to the user's account. Finally, the *releaseReservation* method is used when the user will not incur any further charges for this service. There is also a *getAmount* method that is used to get the monetary value of a particular volume of some non-monetary unit for a particular user.

2.5 Conclusions

In this chapter we described the background concepts that are relevant to this thesis. In particular, we described the cryptographic concepts and the charging concepts

that we build upon in Chapter 4, and we described the location concepts that we build upon in both Chapter 4 and Chapter 6.

Chapter 3

Related Research

3.1 Introduction

In this chapter we describe research efforts that are related to LBSs in terms of architectures and protocols, access control models, and sighting blurring algorithms.

3.2 Architectures and Protocols

In this section we describe research efforts that are related to LBSs in terms of architectures and protocols.

3.2.1 Transcoding Middleware Architecture

The use of an architecture that contains a middleware entity between mobile devices and web servers has been proposed for improving web browsing. In particular, HTTP proxy servers [33] can be used to modify content in real-time for mobile devices. This process is known as *transcoding*. The architecture and message flow for transcoding services are shown in Figure 3.1.

The message flow for transcoding services consists of the following four steps:

1. The user enters the address of a web server into the web browser on his/her mobile device. The mobile device sends this address to the transcoding proxy server.

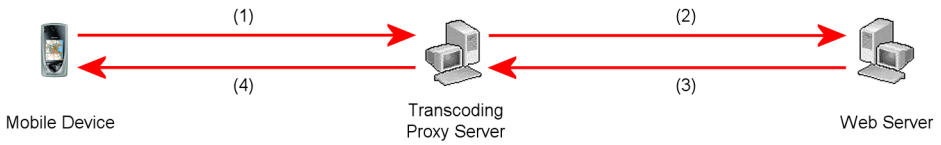

Figure 3.1: Transcoding Architecture and Message Flow

2. The transcoding proxy server contacts the web server, and requests the web content that was specified by the user.

3. The web server returns the specified web content to the transcoding proxy server.

4. The transcoding proxy server modifies the web content in some manner. The modified web content is then returned to the user's mobile device and displayed using his/her web browser.

There has been considerable research into different methods of modifying the web content as it passes through the transcoding proxy server. The authors of [15] propose a transcoding proxy server that improves the web browsing experience for the user by reducing latency. It does this by reducing the amount of data that the user's mobile device must send and receive. For example, the transcoding proxy server validates web server addresses, and redirects requests based on network load. It also performs additional processing such as prefetching web content in anticipation of the user requesting it. The transcoding proxy server is also capable of modifying the web content to create additional application value. For example, it can replace post codes within the web content with town names or city names.

A transcoding proxy server is proposed in [14] to facilitate web browsing using mobile devices and wireless networks. The transcoding proxy server can use one protocol to communicate with the mobile device, and a different protocol to communicate with the web server. This can facilitate more efficient network usage, and faster response times. The transcoding proxy server can filter and modify the HTML [70] content. For example, references to media types and scripting languages that are not supported by the mobile device can be removed. Images can be modified to

suit the mobile device's capabilities by down-scaling them, converting them to other file formats, and reducing their colour depth.

The authors of [51] propose a transcoding proxy server that modifies the web content based on community preferences. The idea behind this is that the transcoding proxy server learns about a user's interests based on his/her web browsing activities. The transcoding proxy server then tries to categorise the user based on a community of users with similar interests. When the user requests web content, the transcoding proxy server modifies the web content based on the community's interest in that web content. The transcoding proxy server contains a mechanism for allowing users to specify certain preferences, and it is always possible for the user to retrieve the original high quality content with some additional clicks.

3.2.2 LBS Middleware Architecture

The use of an architecture that contains a middleware entity has also been proposed many times in the context of LBSs.

The developers of the *GUIDE* project use several middleware entities between the user's mobile device and the web server [22]. These middleware entities place contextual information, such as location information, into web pages without the web servers needing to process any contextual information. To achieve this, several `GUIDE` tags are proposed. These tags have a similar structure to normal HTML tags, but represent contextual information. For example, there is a `GUIDE` tag that represents the user's current location. The web content developers can use these `GUIDE` tags within their HTML documents. When a user wishes to view a new web page his/her mobile device's web browser sends a request to the middleware, which in turn retrieves the web content from the relevant web server. The middleware then replaces all of the `GUIDE` tags with normal HTML that represents the contextual information, such as the user's current location. This new web content is then returned to the user's mobile device's web browser.

Brunato and Battiti propose a location based recommendation system, known as *PILGRIM*, which is based on a middleware entity [16]. This recommendation system tailors search results based on the user's current location, and the search

results are ranked using the observations of previous users. The information about the user's web browsing activity is captured using a modified HTTP proxy server, and this information is then used as an input into the recommendation system.

Gruteser and Grunwald propose a middleware entity called a *location server* in [37]. This middleware entity operates between the user's mobile device and the LBS. It is responsible for performing sighting blurring, and the sighting blurring algorithm is described in Section 3.4.4. Bellavista, Corradi, and Giannelli also propose a middleware entity for performing sighting blurring [10]. Their approach is described in Section 3.4.3.

Hengartner describes a middleware entity that operates between the user's mobile device and the LBS in [40]. Hengartner assumes that the user trusts the middleware entity, and that the user does not trust the LBS with his/her sightings. Therefore, *private information retrieval* is used by a *trusted computing* module within the LBS to ensure that it never has access to the user's sightings.

Myles, Friday, and Davies propose a middleware entity that operates between the user's mobile device and the LBS [52]. This middleware entity is responsible for releasing users' sightings to LBSs in accordance with the users' privacy policies. These privacy policies are specified using a modified version of an existing standard for privacy policies, called the *Platform for Privacy Preferences* [71]. This standard is designed to specify privacy policies in the context of websites, and it specifies what information websites will collect from users, and how this information will be used. The standard does not cover related issues, such as the secure transfer of the information, or the policy enforcement.

3.2.3 Identification and Authentication Protocols

Identification and authentication play a crucial role in providing users with increased security in the context of LBSs, and there are several research efforts in this area. Perhaps the simplest approach is to remove all identification information from messages that are exchanged with LBSs. The advantage of this approach where identification information is removed is that it offers the user very strong privacy. However, this is achieved by reducing the usefulness of certain LBSs that require some form

of persistent identification.

Therefore, another popular approach is to use temporary pseudonyms that change frequently, instead of using real identities. For example, the authors of [47] propose using this approach in the context of LBSs. In particular, their approach is designed for use within mobile phone networks.

The authors of [67] propose a system where sightings are recorded as tuples containing the location of a user, the time when he/she was located, and a unique random number that can be changed at any stage by the user. The user allows an LBS to start accessing his/her location information by providing the LBS with his/her unique random number. The user stops the LBS from accessing his/her location information by changing his/her unique random number. Therefore, these random numbers are in effect pseudonyms.

Finally, Beresford and Stajano propose using a middleware server between the user's mobile device and the LBS [11, 12]. This middleware server is responsible for removing information about the user's real identity. It is also inserts a random pseudonym on behalf of the user, and it can change this random pseudonym frequently. This ability is used to provide a form of sighting blurring, and this is described in Section 3.4.1.

In all of these examples the random pseudonyms change very frequently in order to prevent their owners being identified. Therefore, users are anonymous when they invoke LBSs, and hence there is no need for identification or authentication.

Escudero-Pascual and Maguire propose using a middleware entity to provide the user with increased privacy by hiding both the location of his/her mobile device, and his/her identity, from the LBS [30]. Cryptographic techniques are used to mutually identify and authenticate the entities. All subsequent messages can then be encrypted and signed. The main disadvantage of this proposal is that there is no support for users having multiple pseudonyms. Furthermore, all entities are identified and authenticated with each other in a pair-wise manner.

Hauser and Kabatnik propose an architecture where each request for sightings is created by either a user or an LBS [39]. These sighting requests are then sent to a server that is capable of sighting the users. Therefore, there are only ever two

entities participating in any sighting request (the user or the LBS and the server), and these two entities are mutually identified and authenticated to each other using cryptographic techniques. Hauser and Kabatnik propose using an identity based cryptography system, which enables users to create and sign authorisation certificates. The user or LBS who is requesting the sightings creates and signs a sighting request containing the identity of the user to be sighted, and the relevant authorisation certificate. The user whose sighting is requested is identified using a pseudonym. However, since these users are not participating in the sighting request protocol, there is no need for them to be able to perform cryptographic operations using their pseudonyms.

3.2.4 Conclusions

In this section we described research efforts based on transcoding proxy servers designed for use with mobile devices, as well as research efforts in the context of LBSs. All of these research efforts are based on an architecture that uses a middleware entity. Therefore, an architecture based on a middleware entity is a suitable architecture for facilitating the operation of LBSs over the Internet. Furthermore, such a middleware entity is suitable for providing common functionality regarding the users' identity information and sighting information.

We also described research efforts that provide identification and authentication in the context of LBSs. Some of these research efforts enable users to access LBSs anonymously or using temporary pseudonyms, and hence there is no need for identification or authentication. These research efforts enable users to reduce the amount of trust that they must place in LBSs, albeit at the expense of reducing the usefulness of certain LBSs that require some form of persistent identification.

We then described research efforts that enable users to have persistent pseudonyms. However, none of these research efforts enable users to perform cryptographic operations using their pseudonyms. Furthermore, many of the research efforts described in this section are based on architectures consisting of users, LBSs, and middleware entities. However, none of these research efforts propose any three-party identification and authentication protocols.

In Section 4.12 we will compare the related research described in this section with our architecture and protocol.

3.3 Access Control Models

In this section we describe research efforts that are related to LBSs in terms of access control models.

3.3.1 Single Subject Centralised Access Control Models

The authors of [45] propose a centralised access control model, which consists of permissions that incorporate contextual information about both the user who is the subject and the user who is the object. Therefore, their access control model assumes that there will only be one subject in each request for a user's sightings. Users create their own contexts based on their current contextual information. For example, a user might create a *Work* context to be used during the working hours of the weekdays, and a *Shopping* context to be used at the weekends if the user is in a shopping area of the city. Users then create permissions, which specify the users who are subjects that can request their sightings, based on these contexts. Thus, the permissions used by the access control model to determine the rights of a user who is a subject will vary automatically depending on the context of the user who is the object of the permissions. Users manage their contexts and permissions using a web-based interface.

Ray and Kumar propose extending an access control model based on *Mandatory Access Control* (MAC) so that it includes location information [66]. They have designed their access control model primarily for use by military applications. Their access control model requires that all locations are part of a hierarchy of political locations (see Section 2.3.1), and every location is associated with a security level. Their access control model is used to control which users, as subjects, have privileges to access which resources, as objects, based on the locations of both the users and the resources.

Atluri and Shin present an access control model for combining users' sightings

with their profiles and permissions [6, 7]. Their access control model is based on a traditional access control model that assumes that there will only be one user who is a subject in each request for a user's sightings. However, the main focus of this research effort is to develop a data structure to efficiently manage users' sightings, profiles, and permissions. In particular, this tree based data structure is capable of storing and querying all past, present, and future sightings.

Finally, the authors of [4] propose modifications to a traditional access control model that enable its policies to contain conditions that are expressed in terms of the sighting information of the user who is the subject. In order to achieve this, the authors propose several location based predicates that can be used within the policies. These location based predicates can be used to determine if the user who is the subject is in a certain area, if he/she is within a certain distance range of another user or an area, if his/her speed is within a certain speed range, and if he/she is in the same area as a certain range of other users. Each of these location based predicates returns a boolean result and confidence value for this result.

3.3.2 Single Subject Distributed Access Control Models

Hauser and Kabatnik consider that there will be too many users of LBSs for a centralised access control model to be successful [39]. Therefore, they propose a distributed access control model that is based on public key cryptography and certificates. Their access control model uses identity based cryptography where public keys are used to identify the users who own them. Users create authorisation certificates that contain both the identity of the user or LBS with which they want to share their sightings and the associated permission. Users then sign these authorisation certificates, and distribute them to the users and LBSs that are named within them. When a user or LBS wants to sight a user, he/she/it creates a sighting request that contains both the identity of the user to be sighted and the relevant authorisation certificate. This sighting request is then sent to a server that is capable of sighting users.

Zhong, Goldberg, and Hengartner have developed three different protocols to control the release of information about users' sightings [73]. In particular, these

three protocols allow a user to obtain information about sightings of another user if, and only if, both users are in close proximity to each other. The quantification of close proximity is agreed by both users when they invoke the protocol. Therefore, these three protocols have the effect of enforcing proximity based permissions within an access control model. All three protocols are based on a single user requesting information about the sightings of another single user, and a distributed architecture is used to ensure that no third parties have access to users' sightings. All three protocols are based on public key cryptography. The main differences between the three protocols are the number of message exchanges that they require, and the amount of sighting information that is revealed to the users. For example, one of the protocols enables both users to obtain only the distance between them if they are in close proximity to each other, whereas another of the protocols enables both users to obtain a sighting of the other user if they are in close proximity to each other.

3.3.3 Dual Subject Centralised Access Control Models

Leonhardt and Magee propose a centralised access control model that supports permissions containing more than one subject [50]. Each permission in this access control model is specified in terms of three parameters. The first parameter specifies the subject or subjects who can use the permission. If more than one subject is specified then the permission can only be used by all of the specified subjects simultaneously. These subjects can be either users or locations. The second parameter specifies the action that the permission entitles the subject or subjects to perform. The most relevant actions are the ability to obtain sightings of another user or users, and the ability to test for the presence of a user or users in a location or locations. The third parameter specifies the object or objects that the permission covers. If more than one object is specified then the action must be applied to all of these objects. These objects can be either users or locations. An example of a permission in this access control model that contains two subjects is a permission that entitles *Stefano* to obtain sightings of *Carlotta* if he is using the *FriendFinder* LBS. This is specified as a permission with two subjects (*Stefano* and *FriendFinder*), an action

to sight, and a single object (*Carlotta*). An example of a permission in this access control model that contains two objects is a permission that entitles *Stefano* to obtain sightings of *Carlotta* if she is in *UCD*. This is specified as a permission with a single subject (*Stefano*), an action to sight, and two objects (*Carlotta* and *UCD*).

Hengartner and Steenkiste propose an architecture for an access control model with two types of queries [41, 42]. The first type of query is called a *user query*, and this is used to obtain a user's sightings. The second type of query is called a *room query*, and this is used to determine which users are at a particular location. There are two types of corresponding permissions. The first type of permission is a user permission, and this specifies who is allowed to obtain sightings of a particular user. The second type of permission is a room permission, and this specifies who is allowed to obtain sightings of users who are in a particular location. Both of these permissions can contain restrictions that are specified in terms of specific users, locations, and times. The concept of *service trust* is then introduced in order to allow users to specify which services they trust with their sightings. These trusted services are then allowed to receive sightings on behalf of users who are the subjects of the permissions. The trusted services always inherit the permissions of the users who are the subjects of these permissions. Both the users and the trusted services are capable of delegating their rights to other users and services to form delegation chains. Public key cryptography and certificates are used to implement these permissions and trusts. Although this access control model is implemented as a centralised access control model, the use of public key cryptography and certificates makes this access control model suitable for implementing as a dual subject distributed access control model.

3.3.4 Conclusions

In this section we described access control model research efforts in the context of LBSs. Both the single subject centralised access control models and the single subject distributed access control models assume that there will be only one subject involved in each request for a user's sightings. Therefore, these single subject access control models are not adequate for use in circumstances where sighting requests

can originate from users, LBSs, or combinations of both users and LBSs. The dual subject centralised access control models are designed to address this inadequacy. However, none of theses access control models recognise users and LBSs as distinct, but equally important, subjects.

Both the single subject centralised access control models and the dual subject centralised access control models require another entity that is responsible for storing and managing permissions. The main advantage of this approach is that the users and LBSs do not need to store and manage permissions. However, this approach relies on a single entity, and it may not be scalable. In contrast, the single subject distributed access control models do not require the assistance of another entity to store and manage permissions. The main advantage of this approach is that there is no need to centrally store and manage permissions. This facilitates both redundancy and scalability. The most significant disadvantage of this approach is that users and LBSs must store and manage the permissions. Finally, we are not aware of any dual subject distributed access control models.

In Section 5.6 we will compare the related research described in this section with our access control model and its requirements, which are described in Section 5.2.

3.4 Sighting Blurring Algorithms

In this section we describe research efforts that are related to LBSs in terms of sighting blurring algorithms.

3.4.1 Anonymity Based Blurring Algorithms

Beresford and Stajano propose an approach for protecting users' location information based on a technique used in anonymous communications [11, 12]. This approach works in scenarios where the user is identified using a random pseudonym, and this pseudonym changes regularly.

The basic idea is that special regions called *mix zones* are created. Normally users can be located very accurately. However, when they enter a mix zone their locations are reported as being in the mix zone, without specifying the exact location.

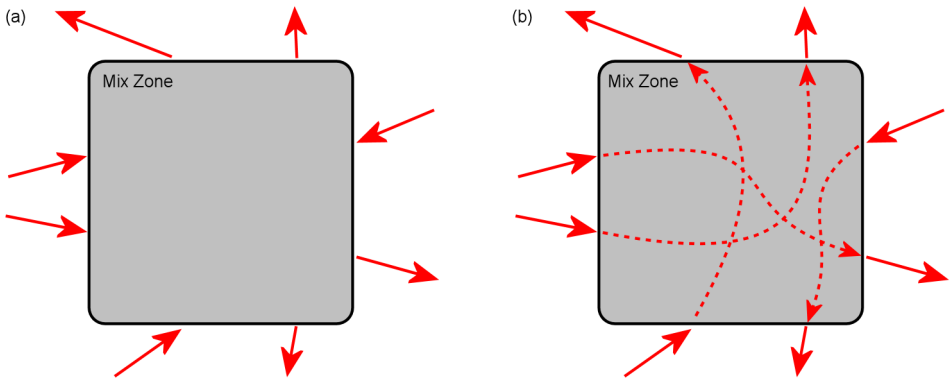

Figure 3.2: Mix Zone Example

Whenever the user enters a mix zone his/her pseudonym is changed. By ensuring that there are always at least two users in the mix zone, it is not possible to link the old pseudonym and the new pseudonym with certainty, and hence the user cannot be identified and tracked. The exact number of users that must be present in the mix zone is known as the *anonymity set size*.

An example of the mix zone concept is shown in Figure 3.2. The external view of the mix zone is shown in Figure 3.2 (a). It is not possible to link the four paths that are entering the mix zone and the four paths that are exiting the mix zone with certainty. The internal view of the mix zone with internal paths is shown in Figure 3.2 (b).

There are two significant disadvantages associated with this approach. Firstly, it is not possible for users to have persistent identifications. Secondly, the accuracy of the users' sightings varies depending on their locations.

3.4.2 Probability Based Blurring Algorithms

Several authors have proposed techniques that blur a user's sightings by introducing an element of uncertainty to his/her blurred sightings, and this uncertainty can be quantified as a probability.

Duckham and Kulik propose a sighting blurring algorithm that creates additional locations where the user might be located, such that the probability of the user being located in any of these additional locations is the same as the probability that the

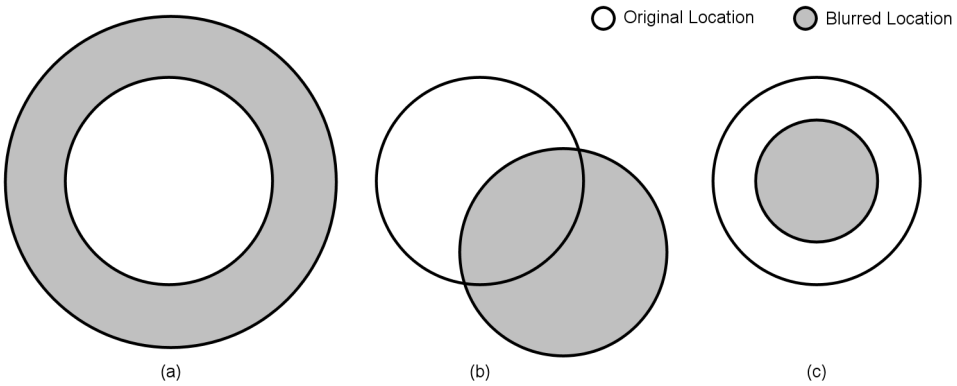

Figure 3.3: Probability Based Location Blurring Algorithms

user is in the location in which he/she is really located [26].

The authors of [5] propose three different sighting blurring algorithms that are based on probabilities. All three of these algorithms assume that the user is in a circular location. The first algorithm increases the radius of the circle containing the user, as shown in Figure 3.3 (a). The second algorithm shifts the centre of the circle containing the user, as shown in Figure 3.3 (b). The third algorithm reduces the radius of the circle containing the user, as shown in Figure 3.3 (c). All three of these algorithms can be combined together to offer the user increased privacy. The authors describe a way of enabling users to specify their privacy in terms of the first algorithm, even though the specified privacy can be achieved using a combination of all three algorithms.

Finally, the authors of [20] propose an algorithm that can generate a list of users who might be in or near a specific location, along with the probability that these users really are in or near this location.

The disadvantage of all probability based blurring algorithms is that they do not produce blurred sightings that are guaranteed to contain the user.

3.4.3 Political Location Blurring Algorithms

Political location blurring algorithms work by dividing a location into a hierarchy of political locations (see Section 2.3.1). For example, consider the hierarchy shown in Figure 3.4 where *University* is composed of several buildings including the *Com-*

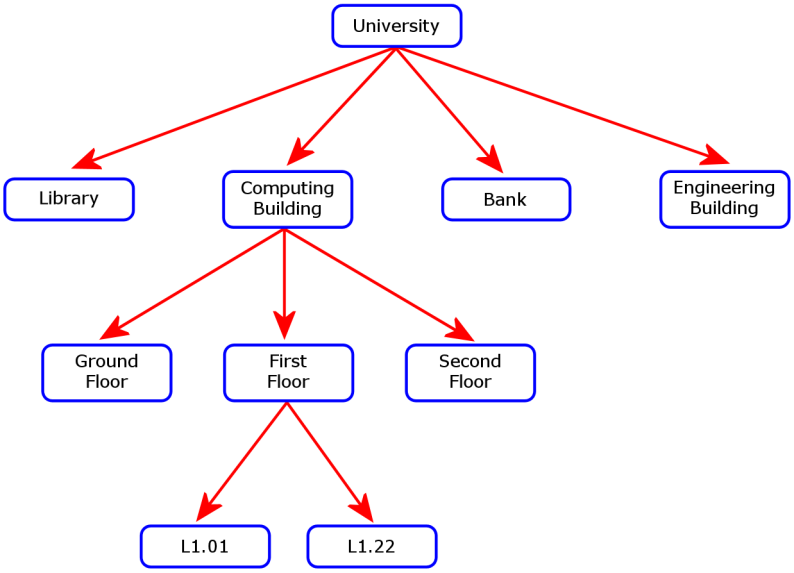

Figure 3.4: Political Location Hierarchy

puting Building, and the *Computing Building* is composed of three floors, and the *First Floor* is composed of two rooms. The accuracy of a location can be blurred by choosing one of its ancestors in the hierarchy. In the example, the location *L1.01* can be blurred by using *First Floor* or *Computing Building* depending on the required level of blurring.

Several authors have already proposed approaches that use political location blurring. Bellavista, Corradi, and Giannelli have developed a middleware proxy that performs location blurring of political locations in the context of Wi-Fi networks [10]. The access control model developed by Leonhardt and Magee supports location blurring of political locations [50]. Narayanan groups similar political locations, which are referred to as *states*, together to create *realms* [53]. In our example, *Library*, *Computer Building*, *Bank*, and *Engineering Building* are all states within the realm representing buildings in *University*. The accuracy of any location can then be blurred by choosing the appropriate realm.

The main disadvantage of using algorithms that blur political locations is that there is no guarantee of the accuracy of any blurred locations. For example, the *Bank* location in *University* is significantly smaller than the *Library* location. Also, it could be difficult to create a consistent location hierarchy that covers large locations

due to the effort involved, and the different types of underlying political locations.

3.4.4 Spatial and Temporal Blurring Algorithms

Spatial blurring algorithms work by increasing the spatial aspects of sightings, and temporal blurring algorithms work by increasing the temporal aspects of sightings. Both of these algorithms are often used together, since they both modify the sightings in a complimentary manner.

Gruteser and Grunwald propose a middleware entity for performing sighting blurring [37]. In this approach the users' sighting are described using the tuple $\langle (x_1, x_2), (y_1, y_2), (t_1, t_2) \rangle$, where $\langle (x_1, x_2), (y_1, y_2) \rangle$ represents the rectangle containing the user, and (t_1, t_2) represents the time interval when the user was within the rectangle. The authors propose a concept called *k-anonymity*, which is used to describe a user that is only identifiable as being one of k users. For example, if $k = 10$, then a user's sighting will always be reported as being the same as nine other users. Their *cloaking* algorithm can use a mix of spatial blurring and temporal blurring in order to achieve the k-anonymity. If the user is in a very densely populated area, then $|x_1 - x_2|$, $|y_1 - y_2|$, and $|t_1 - t_2|$ can become very small. However, when the user is in a very sparsely populated area then $|x_1 - x_2|$, $|y_1 - y_2|$, and $|t_1 - t_2|$ can become very large.

The authors extend their approach in [36] by introducing two concepts that prevent an attacker from tracking a user. The first concept is called *minutiae suppression*, and this blocks the release of a user's sightings when he/she is in a location that would divulge his/her identity. The second concept is called *path segmentation*, and this divides a user's path into several smaller paths that cannot be attributed to the same user because his/her pseudonym changes when one path ends and another begins. Hoh and Gruteser extend the concept of path segmentation with a concept called *path perturbation*, which intentionally modifies users sightings so that their paths are briefly indistinguishable from each other [44].

Gruteser and Liu extend the concepts of k-anonymity and minutiae suppression in [38] by developing three algorithms that limit the circumstances in which sightings of users are released. These algorithms require that all locations that can contain

users be classified as either sensitive or insensitive. The *base* algorithm is a direct application of minutiae suppression that only releases sightings of a user if he/she is in an insensitive area. The *bounded-rate* algorithm extends the base algorithm by introducing a threshold for releasing new sightings of a user when he/she is in an insensitive area. Finally, the *k-area* algorithm extends the base algorithm by only releasing new sightings of a user in an insensitive area if these sightings do not enable an attacker to determine which of k sensitive areas the user has visited.

Gedik and Liu extend k-anonymity in [34] by allowing each user to specify his/her own k-anonymity. Furthermore, they allow each user to specify the maximum amount that his/her spatial or temporal data will be blurred.

The main disadvantage of using blurring algorithms based on k-anonymity is that these algorithms require knowledge of current sightings of the user who is being located and every other user in his/her vicinity, in order to determine which users are the other $k - 1$ users. Another disadvantage of these blurring algorithms is that the sighting accuracy of the users' sightings can change dramatically due the proximity of other users.

3.4.5 Frequency Based Blurring Algorithms

Frequency based blurring algorithms are sighting blurring algorithms that control the frequency with which users' sightings are released. Thus, an attacker cannot gain any additional information by invoking the sighting blurring algorithm frequently.

The only frequency based blurring algorithm which we are aware of is the algorithm developed by Candebat [17]. In this algorithm a user's sightings are spatially blurred by increasing their location components in accordance with the user's privacy preferences. However, the algorithm will only release fresh sightings of a user if the user could be located anywhere within the released location with equal probability. This probability is based on calculating the maximum distance the user could have travelled since the last sighting of him/her, based on his/her current instantaneous speed.

One of the consequences of this algorithm is that it requires fresh input sightings of the user to calculate his/her current instantaneous speed, but the blurred sightings

that are created using these fresh input sightings might never be released by the algorithm. A disadvantage of this algorithm is that it is possible for an attacker to determine the user's current instantaneous speed based on the frequency with which sightings are released. Also, the algorithm will only ever release a single sighting for each blurred location, regardless of how long the user stays in that location. Therefore, these sightings might not be fresh.

3.4.6 Conclusions

In this section we described sighting blurring algorithm research efforts in the context of LBSs. Although these research efforts have many merits, they also have significant shortcomings. Some of the sighting blurring algorithms described in this section offer users increased privacy by making their sightings or paths anonymous. Although this is acceptable for some LBSs, there will be many other LBSs that may require some form of persistent identification for users.

Several of the sighting blurring algorithms produce blurred sightings with accuracies that vary depending on the circumstances of the invocation. It is possible that this inconsistency will hinder the creation of consistent LBSs, and this in turn will discourage the usage of these LBSs by users. Similarly, it is possible that that the probability based blurring algorithms will cause LBSs to provide services that are perceived to be inaccurate or misleading by users, and this will also discourage their usage.

The k-anonymity blurring algorithms require knowledge of current sightings of many users. However, this sighting knowledge may not be available due to limitations of the underlying positioning technology, the privacy preferences of the users who are not involved in the current invocation of the LBS, or the charges incurred for obtaining these sightings. Furthermore, the k-anonymity blurring algorithms are only capable of hiding a user amongst $k - 1$ other users of the algorithm. This would cause these algorithms to produce very blurred sightings if the take-up of the algorithm is low.

The frequency based blurring algorithm developed by Candebat is the most relevant to our sighting blurring algorithm. However, the charges incurred for obtaining

the sightings needed to calculate the users' current instantaneous speeds may cause this algorithm to be cost prohibitive.

In Section 6.5.3 we will compare the related research described in this section with our sighting blurring algorithm and its requirements, which are described in Section 6.3.

3.5 Conclusions

In this chapter we described research efforts that are related to LBSs in terms of architectures and protocols, access control models, and sighting blurring algorithms. Although these research efforts have many merits, they also have significant shortcomings. In the following three chapters we describe how we have addressed these shortcomings by developing our own architecture and protocol, access control model, and sighting blurring algorithm.

Chapter 4

Architecture and Protocol

4.1 Introduction

In this chapter we describe an architecture for users, an infrastructure, and LBSs, which facilitates the operation of these LBSs over the Internet. We also describe a protocol that enables these three entities to achieve three-party mutual identification and authentication. In particular, this protocol allows users to simultaneously identify and authenticate themselves to the infrastructure using one identity, and to the LBS using another identity. This protocol then guarantees the confidentiality, integrity, and non-repudiation of all subsequent messages.

4.2 Entities

In this section we describe the five entities within our architecture.

4.2.1 Locatables

A locatable entity is a mobile device that can be located. This may be because it contains either a satellite based positioning technology or a mobile phone network based positioning technology.

4.2.2 Network Operators

A network operator manages and maintains a network of locatables, and therefore it is capable of producing sightings of each locatable. The network operator provides an interface that enables other entities to obtain these sightings (see Section 2.3.3), albeit for a charge. Access to this interface is only provided to a limited number of entities, and the approval procedure is normally slow and requires significant human intervention.

The network operator supports a revenue sharing model (see Section 2.4.1), and it provides an interface that enables other entities to exchange charging details (see Section 2.4.2). Again, access to this interface is only provided to a limited number of entities, and the approval procedure is normally slow and requires significant human intervention.

The most common form of network operator in the context of publicly accessible LBSs is a mobile phone network operator.

4.2.3 LBSs

An LBS is a service that is operated by a party who is normally independent of the network operator.

4.2.4 The Infrastructure

The infrastructure entity is responsible for providing all of the common functionality that is required by LBSs. This factoring-out of the common functionality means that the developers of LBSs do not need to repeatedly re-implement similar functionality. The main focus of this functionality is on managing the users' identity and sighting information. In particular, the infrastructure is responsible for hosting an implementation of the access control model described in Chapter 5, and it is also responsible for performing sighting blurring as described in Chapter 6.

The infrastructure is capable of using the APIs that are supported by the network operators in order to obtain sightings of users, as well as to exchange charging details regarding the users.

4.2.5 Users

A user is an entity that subscribes to an infrastructure. This enables him/her to invoke LBSs using either his/her mobile device or a desktop computer. Users own locatables, and therefore we refer to locating users rather than locating locatables.

A user invokes an LBS by sending a request to it. The LBS then contacts the infrastructure and obtains the location information that it needs to provide the service to the user. The LBS normally processes this information further by applying application functionality and presentation functionality, and it then sends the results back to the user. For example, the LBS might generate maps containing the sightings, or provide directions to a location.

4.3 Architecture

The infrastructure is designed to operate as a *middleware* entity, whereby all communications between the users and the LBSs go via the infrastructure. However, users are unaware of this, because the infrastructure is a *transparent middleware*. Therefore, when a user establishes a connection with an LBS it is actually a virtual connection. In reality, the user has established a logical connection to the infrastructure, and the infrastructure has established a logical connection to the LBS. The infrastructure can behave as a *passive middleware* that does not modify the message sequences or message contents, or it can behave as an *active middleware* that modifies either the message sequences or message contents. An overview of this architecture containing these entities is shown in Figure 4.1.

4.4 Naming

Every user within our architecture is assigned a unique *account name* when he/she first registers with the infrastructure, and this account name is then used to identify the user to the infrastructure. Account names are only used between the user and the infrastructure, and inside the infrastructure. They consist of a unique value, and they can be linked to the user's personal details, and in particular, to details of

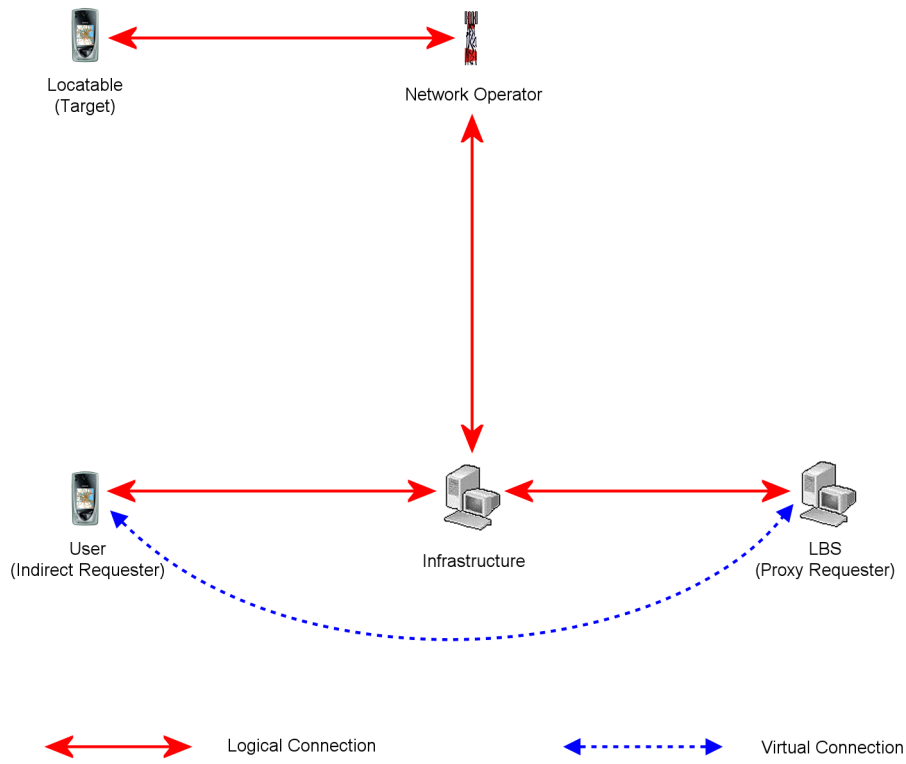

Figure 4.1: Architectural Diagram Showing Entities and Roles (in Parentheses)

his/her locatable.

Every user has one or more unique *public names*, and these are used to identify him/her to other users and to the LBSs. Public names can identify users by their real names, and therefore not all public names are pseudonyms [65]. Our architecture does not limit the number of public names that a user can possess, and with which LBSs these public names can be used. Every LBS has a single public name that is unique.

The relationship between a user and his/her account name is a *one-to-one* mapping, the relationship between a user's account name and his/her public names is a *one-to-many* mapping, and the relationship between an LBS and its public name is a one-to-one mapping. Every user knows his/her complete mapping, and the infrastructure knows the complete mapping for every user.

The infrastructure provides no external services that enable users or LBSs to link users with account names or public names. The only way that this linkage

can be discovered is if the user who is being linked describes the relationship using a different medium. These types of *out-of-band* communications are rarely used in the context of the Internet, because they force users to have some sort of real relationships with each other. However, in the context of LBSs many users will have real relationships with the users with whom they wish to share location information.

4.5 Roles

There are two different roles that are identified within our architecture:

- *Targets* are users who need to be sighted by the infrastructure as part of the delivery of LBSs. Therefore, it is their privacy that the access control model and sighting blurring algorithm are protecting.
- *Requesters* are users and LBSs who request sighting information about targets from the infrastructure. An *indirect requester* is a user who requests sighting information from an LBS, which then requests sightings from the infrastructure. Such an LBS is known as a *proxy requester*.

Both targets and requesters are identified using their public names, and a single mobile device can be both a target and a requester simultaneously. These roles are shown in Figure 4.1.

4.6 Sightings

A sighting for a target describes where it was, and when it was there. All sightings contain a location component that describes where the target was sighted, and a time interval that describes when the target was sighted. Therefore, a sighting states that the target was somewhere within the specified location for at least one instant during the specified time interval. The *sighting accuracy* of a sighting is a measure of the quality of the sighting such that a more accurate sighting has a smaller location and/or time interval. Sighting accuracies are always comparable. *Sighting blurring* is the process of taking an existing sighting and a desired sighting

accuracy, and creating a new sighting that has a sighting accuracy that is less than or equal to the desired sighting accuracy.

Sightings, sighting accuracies, and sighting blurring are described in detail in Chapter 6.

4.7 Permissions

We have developed an access control model that is implemented within the infrastructure within our architecture. Our access control model is based on a system where targets create permissions that have three purposes. Firstly, they specify which requester or requesters are entitled to obtain sightings of the target. Secondly, they specify under what circumstances these sightings are released to the requesters. Thirdly, they specify the maximum sighting accuracy of any sightings that are released to the requesters. A target can create multiple permissions for each public name, and all targets and requesters are identified within these permissions using public names. The infrastructure uses the access control model to release sightings of targets in accordance with their permissions.

Our access control model is specifically designed for use with our architecture, which consists of both indirect requesters and proxy requesters. Therefore, targets create two types of permission:

- *Indirect Access Permissions* (IAPs) that relate to indirect requesters.
- *Proxy Access Permissions* (PAPs) that relate to proxy requesters.

In order for the access control model to authorise the infrastructure to release a sighting, it must be presented with both an authorising IAP and an authorising PAP.

Our access control model is based on a distributed implementation where the infrastructure is not responsible for storing and selecting permissions. Therefore, targets sign the permissions that they create in order to ensure the authenticity and integrity of their permissions.

The access control model, PAPs, and IAPs are described in detail in Chapter 5.

4.8 Operation

Typically, our infrastructure will be used in the following simplified scenario:

1. A user, as an indirect requester, invokes an LBS, as a proxy requester, by sending a request for sighting information.
2. The proxy requester determines which users, as targets, need to be sighted in order to fulfil the indirect requester's request.
3. The proxy requester contacts the infrastructure and for each target it requests a sighting.
4. The infrastructure invokes the access control model using an IAP and PAP for each target in order to determine if the target is willing to allow the release of its sightings, and at which maximum sighting accuracy.
5. The infrastructure obtains a sighting from the network operator for each target, and incurs a charge based on the number of successful sightings.
6. The infrastructure blurs each sighting using a sighting blurring algorithm.
7. The infrastructure returns the targets' sightings to the proxy requester.
8. The proxy requester normally processes the sightings further, and combines them with additional information.
9. The proxy requester sends this processed sighting information to the indirect requester.

At this stage the indirect requester can invoke the proxy requester again, and the process repeats.

4.9 Adapted Mediated Identity Based Cryptography System

Asymmetric cryptography can be used with the X.509 PKI to provide identification and authentication between two entities, as well as to guarantee the confidentiality,

integrity, and non-repudiation of messages exchanged between them. However, there are several disadvantages of using the X.509 PKI with our architecture. Possibly the most significant disadvantage is that it is very problematic for the user to identify and authenticate himself/herself to the LBSs without revealing his/her account name. The most obvious approach to overcome this is for the user to maintain a key pair for each of his/her public names. However, this may not be very practical given that many users will use their mobile devices to access the LBSs.

In order to overcome the limitations of using the X.509 PKI we have used an adapted version of the mediated identity based cryptography system described in Section 2.2.3. The most significant adaptations that we made were:

- We include both the security mediator and the key generator within the infrastructure.
- We adapted the key generator so that the user has a single private key share that can be used with multiple different public keys. The security mediator has its own private key share for each of the user's public keys. Each private key corresponds to a user's account name, and the public key is the user's public name.

The main benefits of using this mediated identity based cryptography system instead of the X.509 PKI are:

- Users have a single private key share that can be used with many different public keys. This reduces the storage requirements of the user's mobile device, and removes the need for private key selection.
- The use of a single private key share with many different public keys enables a user and an LBS to use end-to-end encryption and signing, without the user revealing any of his/her other identities.
- There is no need for users to retrieve and validate revocation information, because this is managed by the infrastructure. This process is usually costly in terms of both bandwidth and processing power.

- All public keys are strings that represent the public names of their owners. This identity based cryptography simplifies the management of public keys significantly.

- If the private key share on the user's mobile device becomes compromised then he/she can be issued with a new private key share, and the security mediator can be issued with new private key shares for each of the user's public keys. However, the user's public names, and hence his/her public keys, remain unchanged.

In the adapted mediated identity based cryptography system, the infrastructure is responsible for generating the user's private keys. These include a private key for use with the user's account name, and a private key for each of the user's public names. Each private key consists of two private key shares. One of these is the user's private key share, and the other is the infrastructure's private key share. For example, consider the user whose account name is A, and whose public names are A_1, A_2, and A_3. The private keys belonging to A are shown in Table 4.1.

Key Name	*User's Private Key Share*	*Infrastructure's Private Key Share*
A	k_A^-	$k_{A,I}^-$
A_1	$k_{A_1}^-$	$k_{A_1,I}^-$
A_2	$k_{A_2}^-$	$k_{A_2,I}^-$
A_3	$k_{A_3}^-$	$k_{A_3,I}^-$

Table 4.1: Private Keys belonging to A

However, the infrastructure can generate each user's private keys in a way that enables the user's private key share for his/her public names to be the same as the user's private key share for his/her account name. Based on our example, the infrastructure will generate $k_{A_1,I}^-$, $k_{A_2,I}^-$, and $k_{A_3,I}^-$ so that that $k_{A_1}^-$, $k_{A_2}^-$, and $k_{A_3}^-$ are all equal to k_A^-.

Since this is an identity based cryptography system, the public keys belonging to each user are his/her public names. However, in order to simplify our notation, we will use our public key notation to show when a public name is being used as a public key. Based on our example, the public keys belonging to A are A, A_1, A_2,

and A_3. Therefore, we will use K_A^+, $K_{A_1}^+$, $K_{A_2}^+$, and $K_{A_3}^+$ to represent these public names being used as public keys.

4.10 Protocol

The protocol that we have designed to use our architecture consists of two phases. During the identification and authentication phase of our protocol the user, as an indirect requester, the infrastructure, and the LBS, as a proxy requester, all identify and authenticate each other. This is achieved using a combination of the X.509 Two-way Authentication protocol (see Section 2.2.2) and the adapted mediated identity based cryptography system, and our protocol establishes the same properties as the X.509 Two-way Authentication protocol. Furthermore, shared session keys are created that are subsequently used for symmetric cryptography in the remainder of the session. The advantage of using symmetric cryptography is that it is more efficient than asymmetric cryptography.

During the sighting request phase of our protocol the user, as an indirect requester, invokes the LBS, as a proxy requester, by making one or more sighting requests. This requires the indirect requester and proxy requester pair to request sightings for one or more users, as targets.

In order to describe our protocol, we introduce the following entities:

- I is an instance of the infrastructure. Its private key is K_I^- and its public key is K_I^+.
- L_1 is an LBS. The private key associated with L_1 is $k_{L_1}^-$, and the virtual public key is $k_{L_1}^+$. The virtual private key associated with L_1 is $K_{L_1}^-$ and the public key is $K_{L_1}^+$.
- A is a user of I whose account name is A. The private key associated with A is k_A^-, and the virtual public key is k_A^+. A uses the public name A_1 with L_1. The virtual private key associated with A_1 is $K_{A_1}^-$ and the public key is $K_{A_1}^+$.
- B_1 and C_1 are the public names used with L_1 by two other users of I. Their

virtual private keys are $K^-_{B_1}$ and $K^-_{C_1}$ respectively, and their public keys are $K^+_{B_1}$ and $K^+_{C_1}$ respectively.

- Sightings are denoted as s_1, s_2, s_3, ..., and sighting accuracies are denoted as α_1, α_2, α_3, ...
- P^I_1, P^I_2, and P^I_3 are IAPs that allow A_1 to obtain sightings of A_1, B_1, and C_1 respectively. P^P_1, P^P_2, P^P_3 are PAPs that allow L_1 to obtain sightings of A_1, B_1, and C_1 respectively.

In the remainder of this section we will assume that A is using the public name A_1 to invoke a service offered by L_1 that requires sightings for A_1, B_1, and C_1. Therefore, A_1 is an indirect requester, L_1 is a proxy requester, and A_1, B_1, and C_1 are all targets. This is likely to occur if L_1 is an LBS that enables a user to view his/her friends' locations in relation to his/her own location.

4.10.1 Identification and Authentication

In the first two steps of the identification and authentication phase of our protocol the user and the infrastructure identify and authenticate each other as follows:

1. $A \rightarrow I : \mathcal{M}(A, I, M_1, K^+_I, k^-_A)$
2. $I \rightarrow A : \mathcal{M}'(I, A, M_2, k^+_A, K^-_I)$

The infrastructure and the LBS then identify and authenticate each other as follows:

3. $I \rightarrow L_1 : \mathcal{M}(I, L_1, M_3, k^+_{L_1}, K^-_I)$
4. $L_1 \rightarrow I : \mathcal{M}'(L_1, I, M_4, K^+_I, k^-_{L_1})$

This is effectively two different instances of the X.509 Two-way Authentication protocol occurring in series. The user and the LBS then identify and authenticate each other as follows:

5. $A \rightarrow L_1 : \mathcal{M}(A_1, L_1, M_5, K^+_{L_1}, K^-_{A_1})$

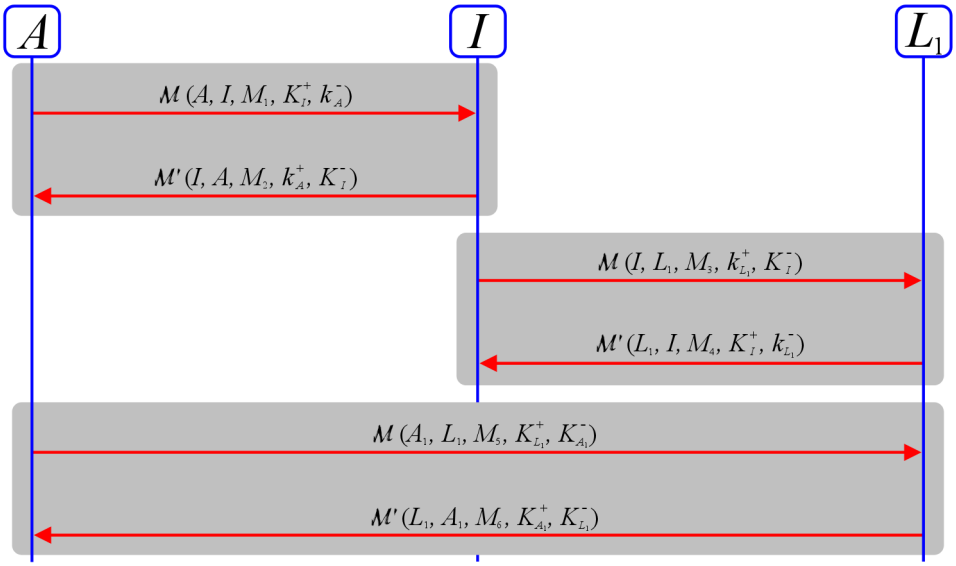

Figure 4.2: Identification and Authentication Sequence Diagram

6. $L_1 \rightarrow A : \mathcal{M}'(L_1, A_1, M_6, K^+_{A_1}, K^-_{L_1})$

After these six steps are completed successfully A and I are identified and authenticated to each other, and they have each generated $K_{A,I}$. Similarly, I and L_1 are identified and authenticated to each other, and they have each generated K_{I,L_1}. A is also identified and authenticated to L_1 as A_1, and L_1 is identified and authenticated to A as L_1. Both A and L_1 have each generated K_{A_1,L_1}. These three shared session keys can then be used to symmetrically encrypt and decrypt all subsequent communications. The complete sequence diagram for these three identification and authentication stages is shown in Figure 4.2.

However, due to the mediated nature of our cryptography system, A and L_1 are unable to use their virtual private keys to sign the complete messages in (5) and (6). Furthermore, they are unable to use their virtual private keys to decrypt the messages M_5 and M_6 that are used to generate the shared session keys in (5) and (6).

Therefore, A and L_1 need I to assist with these signings and decryptions, and this requires that the messages in (5) and (6) are sent via I. Therefore, (5), and (6) are described as follows:

5. (a) $A \rightarrow I : \mathcal{M}(A_1, L_1, M_5, K^+_{L_1}, k^-_A)$

   (b) I processes $\mathcal{M}(A_1, L_1, M_5, K^+_{L_1}, k^-_A)$ by completing the signature to create the signed message $\mathcal{M}(A_1, L_1, M_5, K^+_{L_1}, K^-_{A_1})$

   (c) I partially decrypts $\{M_5\}_{K^+_{L_1}}$, which is contained within $\mathcal{M}(A_1, L_1, M_5, K^+_{L_1}, K^-_{A_1})$, to produce $\{M_5\}_{k^+_{L_1}}$

   (d) $I \rightarrow L_1 : \mathcal{M}(A_1, L_1, M_5, K^+_{L_1}, K^-_{A_1}), \mathcal{M}(I, L_1, K^-_I, \{M_5\}_{k^+_{L_1}})$

6. (a) $L_1 \rightarrow I : \mathcal{M}'(L_1, A_1, M_6, K^+_{A_1}, k^-_{L_1})$

   (b) I processes $\mathcal{M}'(L_1, A_1, M_6, K^+_{A_1}, k^-_{L_1})$ by completing the signature to create the signed message $\mathcal{M}'(L_1, A_1, M_6, K^+_{A_1}, K^-_{L_1})$

   (c) I partially decrypts $\{M_6\}_{K^+_{A_1}}$, which is contained within $\mathcal{M}'(L_1, A_1, M_6, K^+_{A_1}, K^-_{L_1})$, to produce $\{M_6\}_{k^+_A}$

   (d) $I \rightarrow A : \mathcal{M}'(L_1, A_1, M_6, K^+_{A_1}, K^-_{L_1}), \mathcal{M}(I, A, K^-_I, \{M_6\}_{k^+_A})$

The reason that I sends both $\{M_5\}_{K^+_{L_1}}$ in $\mathcal{M}(A_1, L_1, M_5, K^+_{L_1}, K^-_{A_1})$ and $\{M_5\}_{k^+_{L_1}}$ in $\mathcal{M}(I, L_1, K^-_I, \{M_5\}_{k^+_{L_1}})$ to L_1 in (5)(d) is that L_1 needs $\{M_5\}_{K^+_{L_1}}$ in order to reconstruct the original message and then calculate its signature, and L_1 needs $\{M_5\}_{k^+_{L_1}}$ in order to successfully decrypt M_5. I sends $\{M_5\}_{k^+_{L_1}}$ in $\mathcal{M}(I, L_1, K^-_I, \{M_5\}_{k^+_{L_1}})$ to ensure its authenticity and integrity. Similarly, I sends both $\{M_6\}_{K^+_{A_1}}$ in $\mathcal{M}'(L_1, A_1, M_6, K^+_{A_1}, K^-_{L_1})$ and $\{M_6\}_{k^+_A}$ in $\mathcal{M}(I, A, K^-_I, \{M_6\}_{k^+_A})$ to A in (6)(d) for the same reasons.

The complete sequence diagram for the expanded version of these three identification and authentication stages is shown in Figure 4.3.

These three instances of the X.509 Two-way Authentication protocol combined with the adapted mediated identity based cryptography system generate eight different message exchanges. However, some of these message exchanges can be combined because they have a common sender and receiver. This produces a more optimised identification and authentication protocol.

The first step of our optimised identification and authentication protocol consists of A sending I a message containing the first message of A and I identifying

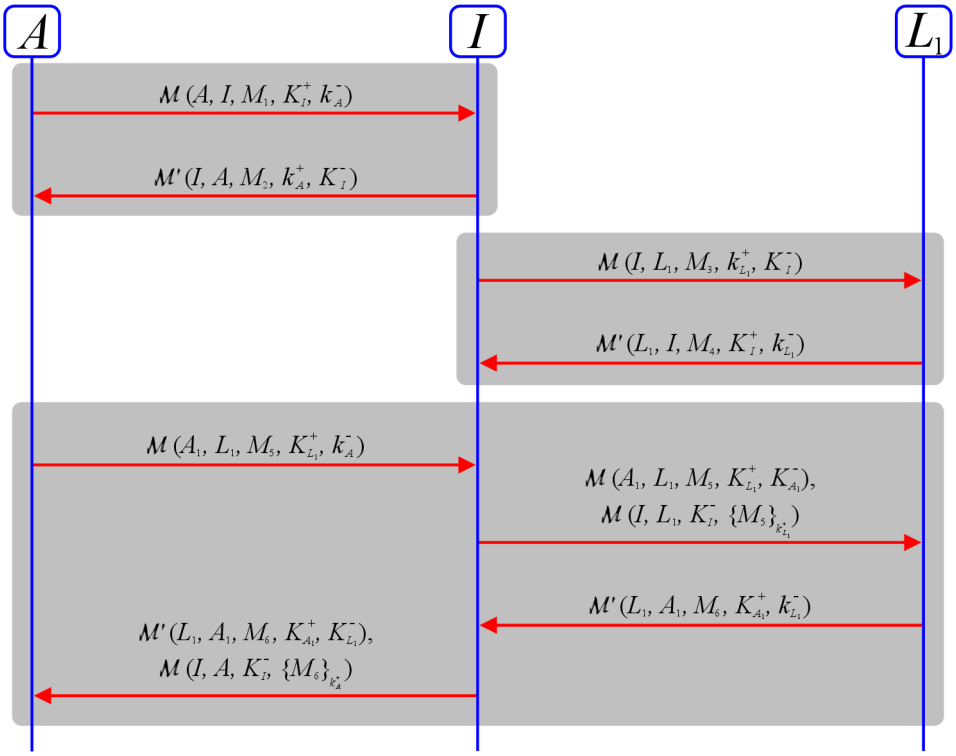

Figure 4.3: Expanded Identification and Authentication Sequence Diagram

and authenticating each other, and the first message of A and L_1 identifying and authenticating each other as A_1 and L_1:

1. $A \rightarrow I : \mathcal{M}(A, I, M_1, K_I^+, k_A^-), \mathcal{M}(A_1, L_1, M_5, K_{L_1}^+, k_A^-)$

The second step consists of I sending L_1 a message containing the first message of I and L_1 identifying and authenticating each other, and the first message of A and L_1 identifying and authenticating each other as A_1 and L_1:

2. $I \rightarrow L_1 : \mathcal{M}(I, L_1, M_3, k_{L_1}^+, K_I^-), \mathcal{M}(A_1, L_1, M_5, K_{L_1}^+, K_{A_1}^-), \mathcal{M}(I, L_1, K_I^-, \{M_5\}_{k_{L_1}^+})$

However, this step can be optimised by including $\{M_5\}_{k_{L_1}^+}$ in $\mathcal{M}(I, L_1, M_3, k_{L_1}^+, K_I^-)$. Therefore, the second step becomes:

2. $I \rightarrow L_1 : \mathcal{M}(I, L_1, M_3, k_{L_1}^+, K_I^-, \{M_5\}_{k_{L_1}^+}), \mathcal{M}(A_1, L_1, M_5, K_{L_1}^+, K_{A_1}^-)$

The third step consists of L_1 sending I a message containing the second message of I and L_1 identifying and authenticating each other, and the second message of A

and L_1 identifying and authenticating each other as A_1 and L_1:

3. $L_1 \rightarrow I : \mathcal{M}'(L_1, I, M_4, K_I^+, k_{L_1}^-), \mathcal{M}'(L_1, A_1, M_6, K_{A_1}^+, k_{L_1}^-)$

Finally, the fourth step consists of I sending A a message containing the second message of A and I identifying and authenticating each other, and the second message of A and L_1 identifying and authenticating each other as A_1 and L_1:

4. $I \rightarrow A : \mathcal{M}'(I, A, M_2, k_A^+, K_I^-), \mathcal{M}'(L_1, A_1, M_6, K_{A_1}^+, K_{L_1}^-), \mathcal{M}(I, A, K_I^-, \{M_6\}_{k_A^+})$

However, this step can be optimised by including $\{M_6\}_{k_A^+}$ in $\mathcal{M}'(I, A, M_2, k_A^+, K_I^-)$. Therefore, the fourth step becomes:

4. $I \rightarrow A : \mathcal{M}'(I, A, M_2, k_A^+, K_I^-, \{M_6\}_{k_A^+}), \mathcal{M}'(L_1, A_1, M_6, K_{A_1}^+, K_{L_1}^-)$

After these four steps are completed successfully A and I are identified and authenticated to each other, and they have each generated $K_{A,I}$. Similarly, I and L_1 are identified and authenticated to each other, and they have each generated K_{I,L_1}. A is also identified and authenticated to L_1 as A_1, and L_1 is identified and authenticated to A as L_1. Both A and L_1 have each generated K_{A_1,L_1}. These three shared session keys can then be used to symmetrically encrypt and decrypt all subsequent communications. The sequence diagram for this optimised identification and authentication protocol is shown in Figure 4.4.

4.10.2 Sighting Request

In the first step of the sighting request phase of our protocol the indirect requester creates an indirect sighting request, and sends it to the proxy requester. This indirect sighting request contains:

- The public name of the indirect requester.
- The public name of the proxy requester.
- The lifetime of the indirect request. This can be expressed using a start time and a stop time, or using a number that represents the maximum number of sighting requests that are allowed. Lifetimes are denoted using λ_1, λ_2, λ_3, ...

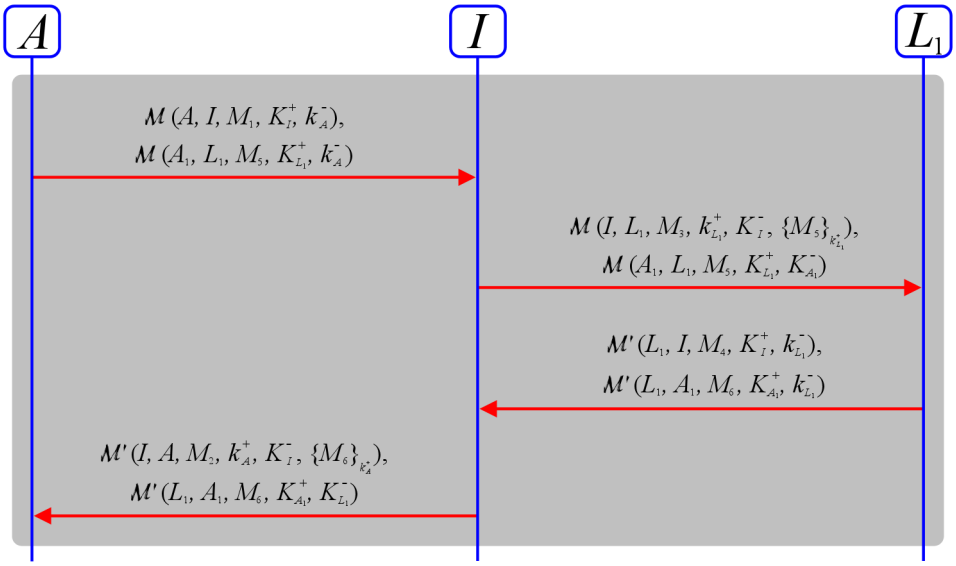

Figure 4.4: Optimised Identification and Authentication Sequence Diagram

- The public name of the target, the required sighting accuracy, and a valid IAP, for each target whose sighting the indirect requester wants the proxy requester to obtain.

We denote the first indirect request, R_1, which is created by A for L_1 as follows:

$$A_1, L_1, \lambda_1, ((A_1, \alpha_1, P_1^I), (B_1, \alpha_1, P_2^I), (C_1, \alpha_1, P_3^I))$$

A encrypts R_1 using K_{A_1,L_1} to ensure the authenticity, confidentiality, and integrity of the indirect sighting request to L_1. However, these properties of the indirect sighting request cannot be demonstrated to I, and a malicious LBS could create any indirect sighting requests. Furthermore, the non-repudiation of this indirect sighting request cannot be established. Therefore, A also signs R_1 using $K_{A_1}^-$. The names of the indirect requester and the proxy requester, and the lifetime, are included in the indirect sighting request to ensure that a malicious LBS cannot launch a replay attack. A then sends this message to L_1:

1. $A \rightarrow L_1 : \{R_1\}_{K_{A_1,L_1}}, \{|R_1|\}_{K_{A_1}^-}$

L_1 decrypts R_1, verifies its signature, and verifies that the indirect sighting request is valid. L_1 then creates a proxy sighting request, R_2, which contains a PAP for each target as follows:

$$R_1, \{|R_1|\}_{K^-_{A_1}}, ((A_1, P^P_1), (B_1, P^P_2), (C_1, P^P_3))$$

L_1 encrypts R_2 using K_{I,L_1} to ensure the authenticity, confidentiality, and integrity of the proxy sighting request to I. There is no need to consider a malicious LBS creating false proxy sighting requests, because the malicious LBS will be unable to forge the signed R_1 contained in R_2. However, the non-repudiation of this proxy sighting request cannot be established, so L_1 signs R_2 using $K^-_{L_1}$. L_1 then sends this message to I

2. $L_1 \rightarrow I : \{R_2\}_{K_{I,L_1}}, \{|R_2|\}_{K^-_{L_1}}$

I decrypts R_2, and verifies the signatures of both R_1 and R_2. I then verifies that the indirect sighting request is valid. For each target, I validates the IAP and PAP, and invokes the access control model to determine the maximum sighting accuracy allowed. I compares this with the requested sighting accuracy to determine which is lowest. I obtains a sighting from the network operator, and performs sighting blurring. I then creates a proxy sighting response, R_3, as follows:

$$((A_1, s_1), (B_1, s_2), (C_1, s_3))$$

I encrypts R_3 using K_{I,L_1} and signs R_3 using K^-_I to establish the same properties as the previous messages. I then sends this message to L_1:

3. $I \rightarrow L_1 : \{R_3\}_{K_{I,L_1}}, \{|R_3|\}_{K^-_I}$

L_1 decrypts R_3, and verifies its signature. L_1 then processes the sighting for each target in order to generate the indirect requester response, R_4, for A.

L_1 encrypts R_4 using K_{A_1,L_1} and signs R_4 using $K^-_{L_1}$, and sends this message to A:

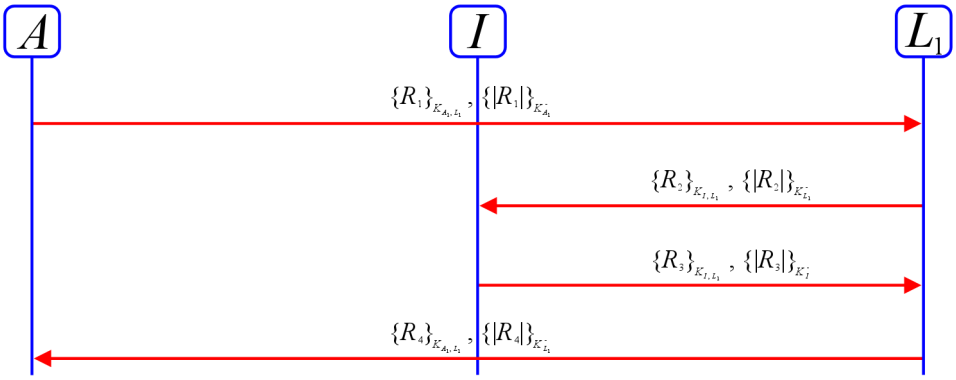

Figure 4.5: Sighting Request Sequence Diagram

4. $L_1 \rightarrow A : \{R_4\}_{K_{A_1,L_1}}, \{|R_4|\}_{K^-_{L_1}}$

A decrypts R_4, and verifies its signature. A now has the sighting information about A_1, B_1, and C_1 that was originally requested.

The sequence diagram for the sighting request phase of our protocol is shown in Figure 4.5.

As in the identification and authentication phase, A and L_1 are unable to use their virtual private keys to sign the messages in (1), (2), and (4), due to the mediated nature of our cryptography system. Therefore, I completes these signatures as it did in the identification and authentication phase.

Furthermore, in the remainder of this thesis we will not explicitly describe the involvement of the infrastructure in mediated operations using our cryptography system.

4.10.3 Charging

Our protocol enables a user, an LBS, and an infrastructure to identify and authenticate each other. It then enables the user to obtain sighting information about other users from the LBS, which in turn obtains this sighting information from the infrastructure. The authenticity, confidentiality, integrity, and non-repudiation of all the messages used to exchange this information are established because our protocol is based on the X.509 Two-way Authentication protocol, albeit using the adapted mediated identity based cryptography system. These properties provide

an opportunity to use our protocol for additional services within the context of our architecture.

The service that interests us the most is the ability to use our protocol to charge users in real-time for invoking LBSs, and then apportion these charges between the LBSs and the infrastructure based on a revenue sharing model. In our architecture we assume that the infrastructure is the entity that collects the revenue from the user on behalf of the LBS, and the LBS in turn shares some of this revenue with the infrastructure. This would enable the user to have a single commercial relationship that can be used to buy services from many different LBSs. The infrastructure is capable of revenue collection because all users must subscribe to the infrastructure before they invoke the LBSs.

There are many possible charging services that can be built upon our protocol. As an example, we will describe a simple charging service that enables users to create *vouchers* for LBSs based on a *push model*. This model enables users to offer payments to the LBSs, rather than the LBSs taking payments from the users. This is similar to traditional payment methods, such as cash or cheque, which are used in the real world. The advantage of this type of payment is that it gives the users complete control over how much they pay the LBSs. This is significant because it reduces the amount of trust that users must have in LBSs. This push model is in contrast to the *pull model*, where the LBSs remove the payment from the users' accounts. The pull model is used in most online purchases made with credit cards. The main disadvantage of this payment model is that it requires the users to trust the LBSs.

Our example charging service enables users, as indirect requesters, to create vouchers. Each of these vouchers contains:

- The public name of the indirect requester.
- A nonce that is used as a serial number.
- The public name of the proxy requester.
- The lifetime of the voucher expressed using a start time and a stop time.

- The value and currency of the voucher.

In our example, L_1 charges 1€ each time that it is invoked. Therefore, A creates a voucher, V_1, for L_1 as follows:

$$A_1, N_A, L_1, \lambda_2, 1€$$

A encrypts V_1 using $K_{A,I}$ to ensure its authenticity, confidentiality, and integrity. However, the non-repudiation of this voucher cannot be established. Therefore, A also signs V_1 using k_A^-. A then sends this message to I:

1. $A \rightarrow I : \{V_1\}_{K_{A,I}}, \{|V_1|\}_{k_A^-}$

I queries the network operator that A uses in order to determine if A has sufficient funds to make a payment of 1€. If A does have sufficient funds then I reserves 1€ from A's funds for the lifetime λ_2. I then completes the signature of $\{|V_1|\}_{k_A^-}$ to create $\{|V_1|\}_{K_{A_1}^-}$. Therefore, we are using the mediated nature of our cryptography system to ensure that a voucher that has been signed by the indirect requester using its public name has been authorised by the infrastructure.

I keeps a copy of this voucher and its signature for the lifetime λ_2, and I associates these with the reservation of A's funds. I also uses the copy of the voucher to redeem or revoke it. I then sends the voucher and its signature back to A:

2. $I \rightarrow A : \{V_1\}_{K_{A,I}}, \{|V_1|\}_{K_{A_1}^-}$

A now has a voucher that it can use as A_1 to pay L_1 a maximum amount of 1€. A can then send this voucher and its signature to L_1 as part of the indirect sighting request message in the sighting request phase of our protocol. L_1 then sends the voucher and its signature to I as part of the proxy sighting request message in order to redeem it. I then verifies the signature of the voucher, validates that it is being redeemed within its lifetime, and validates that it is being redeemed by the correct proxy requester on behalf of the correct indirect requester. I also validates that the voucher has not been redeemed or revoked. I then applies the charge of 1€ to A, and I refunds this amount to L_1.

Alternatively, I can deduct a charge from this amount before refunding L_1. For example, if I charges L_1 0.1€ for each successful sighting request, then I will deduct 0.3€ for sighting A_1, B_1, and C_1. In this case, I refunds L_1 with 0.7€. Therefore, our charging service is being used to implement a revenue sharing model between the indirect requesters, the proxy requesters, and the infrastructure.

4.11 Implementation

We have developed a partial implementation of the infrastructure entity, which we call *The Orient Platform*. We use HTTP for all communications between the indirect requester's mobile device and *The Orient Platform*, because there is already widespread support for HTTP on most mobile devices. The indirect requester's mobile device sends messages to *The Orient Platform* using HTTP POST variables, and receives the responses in the HTTP message body. We use an XML [72] protocol called *The Orient Protocol*, which we developed ourselves, for communications between *The Orient Platform* and the LBSs. This protocol consists of four identification and authentication messages corresponding to the messages outlined in Section 4.10.1, and four sighting request messages corresponding to the messages outlined in Section 4.10.2.

The architecture that we are using requires all location related messages between the indirect requester's mobile device and the LBS to go through *The Orient Platform*. However, the indirect requester's mobile device must still be able to exchange non-location related messages with both normal websites and with LBSs. There are several different approaches that can be used to enable this mixing of non-location related messages exchanged using HTTP and location related messages exchanged using *The Orient Protocol*.

The approach that we have chosen consists of the indirect requester's mobile device sending all of its HTTP requests to *The Orient Platform*, which then determines which requests are normal HTTP requests and which requests require *The Orient Protocol*. The normal HTTP requests are sent to the intended destination server. The corresponding HTTP responses are sent back to *The Orient Platform*,

and then to the indirect requester's mobile device. However, if *The Orient Platform* determines that the original HTTP request needs to be serviced using *The Orient Protocol*, then *The Orient Platform* sends a message using *The Orient Protocol* to the relevant LBS.

This approach is relatively easy to implement because it is possible to ensure that all of the indirect requester's mobile device's HTTP requests go to *The Orient Platform* by configuring the indirect requester's mobile device's HTTP proxy address so that it uses the address of *The Orient Platform*. We adopt a convention such as using a specific port number for all HTTP requests that require *The Orient Protocol*. This enables *The Orient Platform* to distinguish between HTTP based messages and *The Orient Protocol* messages simply by examining the port number in the URL [13]. Therefore, *The Orient Platform* is implemented as a non-transparent HTTP proxy server.

4.12 Comparison with Related Research

Our architecture, which consists of a middleware entity, is similar to the architecture described in the related research in Section 3.2. Indeed, both our infrastructure and the middleware entities described in the related research provide similar functionality. The advantage of our architecture is that it enables the infrastructure to provide all of the common functionality regarding the users' identity information and sighting information. Perhaps the most significant disadvantage is that the infrastructure becomes a single point of failure.

Our architecture enables users to have persistent pseudonyms that we refer to as public names, and users can perform cryptographic operations using these public names. The advantage of using public names is that it provides users with increased privacy while allowing them to have persistent identifications. Our architecture is similar to some of the architectures described in the related research in terms of enabling users to have persistent pseudonyms. However, it differs from these other architectures because it is the only architecture that enables users to perform cryptographic operations using these persistent pseudonyms.

Our protocol is specifically designed to use our architecture, and it enables users, LBSs, and an infrastructure to achieve three-party mutual identification and authentication. This has the advantage of providing greater security, although the disadvantage of this greater security is that it requires greater resources. Our protocol is similar to some of the protocols described in the related research in terms of supporting identification and authentication. However, it differs from these other protocols because it supports three-party identification and authentication, whereas the other protocols only support two-party identification and authentication. This difference is significant in the context of our architecture that consists of users, LBSs, and an infrastructure.

4.13 Conclusions

In this chapter we described an architecture for users, LBSs, and an infrastructure, which is a trusted middleware entity that facilitates the operation of these LBSs over the Internet in a secure manner. In particular, the infrastructure provides common functionality regarding the users' identity information and sighting information.

We have also described a protocol that is based on both the X.509 PKI and mediated identity based cryptography, and which uses this architecture, so that users, LBSs, and an infrastructure can achieve three-party mutual identification and authentication. This protocol allows users to simultaneously identify and authenticate themselves to the infrastructure using an account name, and to the LBS using a public name. Indeed, each user can be identified and authenticated using a different public name with each LBS. This is achieved without requiring the mobile device to have significantly greater resources, and without the need for additional messages to be exchanged. This usage of public names with LBSs provides users with increased privacy without necessarily reducing the usefulness of the LBSs. The confidentiality, integrity, and non-repudiation of all subsequent messages can be guaranteed, and we described how this can be used so that a user receives sighting information from an LBS, which in turn receives this sighting information from the infrastructure.

Finally, we will build upon this architecture and protocol in Chapter 5 in order

to describe our access control model, and in Chapter 6 in order to describe our sighting blurring algorithm.

Chapter 5

Access Control Model

5.1 Introduction

In Chapter 4 we described an architecture for users, an infrastructure, and LBSs, which facilitates the operation of these LBSs over the Internet. We also described a protocol that enables these three entities to achieve three-party mutual identification and authentication. In particular, this protocol allows users to simultaneously identify and authenticate themselves to the infrastructure using one identity, and to the LBS using another identity. This protocol then guarantees the confidentiality, integrity, and non-repudiation of all subsequent messages.

In this chapter we describe an access control model that is implemented within the infrastructure. This access control model is based on users who are targets creating permissions for users who are indirect requesters, and for LBSs that are proxy requesters. These permissions specify which requesters are entitled to obtain sightings of which targets, under what circumstances these sightings are released, and the maximum accuracy of these sightings.

5.2 Requirements

We have identified the following requirements that our access control model must satisfy:

- **Requirement 1**

  The access control model must allow users to specify who is entitled to obtain their sightings in terms of users, LBSs, or both users and LBSs.

  Users may base their security specifications on the identities of the users, as indirect requesters, with whom they are willing to share their sightings. Indeed, Lederer, Mankoff, and Dey found that the indirect requester's identity is a significant determinant of users' security specifications [49]. This type of security specification can be represented using a permission that contains a single subject to represent the indirect requester. However, such a permission can be used with many different LBSs, as proxy requesters, and this might not be in accordance with the user's security specification.

  Alternatively, users may base their security specifications on the identities of the proxy requesters, with which they are willing to share their sightings. For example, Barkhuus and Dey found that the proxy requester's identity is a significant determinant of users' security specifications [9]. This type of security specification can be represented using a permission that contains a single subject to represent the proxy requester. However, such a permission can be used with many different indirect requesters, and this might not be in accordance with the user's security specification.

  However, there may be occasions when users will want to base their security specifications on the identities of both the indirect requesters and the proxy requesters simultaneously. This has the effect of allowing a user to either increase or decrease the level of trust that he/she has in the requesters if they cooperate with each other. For example, consider the simplified example where a user, *Stefano*, is willing to let another user, *Carlotta*, obtain his sightings with a low sighting accuracy using any LBS. Therefore, *Stefano* creates a permission in which *Carlotta* is the subject. *Carlotta* can use this permission with an LBS called *FriendFinder* that displays *Stefano's* location on a map, and *Stefano's* privacy will be respected because *FriendFinder* will

only be able to obtain his sightings with a low sighting accuracy. Now consider another LBS called *NearbyFriendFinder* that enables *Carlotta* to obtain *Stefano's* sightings with a very high sighting accuracy, but only if both *Stefano* and *Carlotta* are currently sighted very close to each other. *Carlotta* will be unable to use *NearbyFriendFinder* to obtain *Stefano's* sightings, because *NearbyFriendFinder* requires *Stefano's* sightings with a high sighting accuracy, and *Stefano* is only willing to let *Carlotta* obtain his sightings with a low sighting accuracy. However, since *NearbyFriendFinder* only releases *Stefano's* sightings in very limited circumstances, he is willing to let it obtain his sightings with a high sighting accuracy. *Stefano* cannot specify this permission in terms of *NearbyFriendFinder* alone, because *NearbyFriendFinder* would then be able to obtain *Stefano's* sightings with a high sighting accuracy at any time. Therefore, *Stefano's* security specification is based on both *Carlotta* and *NearbyFriendFinder*. This type of security specification can be represented using a permission that contains both a subject to represent the indirect requester and a subject to represent the proxy requester.

However, representing a security specification using a single permission that must contain both a subject to represent the indirect requester and a subject to represent the proxy requester is neither efficient nor scalable. Consider an example where *Stefano* wants to base his security specifications on m indirect requesters and n proxy requesters. In this example *Stefano* will need to create $m * n$ different permissions. *Stefano* must then create n new permissions for each new indirect requester that he wants to include in his security specifications. The creation of these new permissions for every proxy requester has the effect of allowing a user to specify that the indirect requester can obtain his/her sightings using any proxy requester that he/she already trusts. Similarly, *Stefano* must create m new permissions for each new proxy requester that he wants to include in his security specifications. The creation of these new permissions for every indirect requester has the effect of allowing a user to specify that the proxy requester can obtain his/her sightings using any indirect

requester that he/she already trusts.

Therefore, the access control model must allow users to independently express levels of trust in both indirect requesters and proxy requesters, but the access control model must not allow either the trusted indirect requesters or the trusted proxy requesters to request sightings unilaterally.

- **Requirement 2**

  The access control model must allow users to specify the circumstances in which their sightings are released.

  Users may base their security specifications on factors that are external to the access control model. For example, Anthony, Henderson, and Kotz found that users expressed different levels of willingness to share their sightings depending on their current locations [3]. Lederer, Mankoff, and Dey found that users expressed different levels of willingness to share their sightings depending on their current activities [49].

- **Requirement 3**

  The access control model must allow users to specify the maximum sighting accuracy of any sightings that are released.

  Typically, users will specify a greater sighting accuracy for indirect requesters and proxy requesters that are trusted. This sighting accuracy that is output from the access control model is then used to determine part of the input to our sighting blurring algorithm, as described in Chapter 6.

Our access control model, which supports these three requirements, enables users to create permissions that support a wide range of security specifications from very simple security specifications to very complex and personalised security specifications. This ability to support a wide range of security specifications will encourage the usage of LBSs by users, according to the findings of Anthony, Henderson, and Kotz in their empirical study of users' security requirements in the context of LBSs [3].

Since our access control model is required to allow users to create permissions regarding their own sightings, our access control model provides a form of *Discretionary Access Control* (DAC).

In Section 5.6 we will compare our access control model and its requirements with the related research described in Section 3.3.

5.3 Mathematical Model

In this section we describe the mathematical model upon which we build our access control model. In particular, we define the types, permissions, and access control algorithm that our access control model uses.

5.3.1 Types

There are three different types defined within the mathematical model of our access control model:

- We define $\mathcal{N}$ as the set of public names for users and LBSs. These public names are strings that have a consistent syntax. We define $\mathbb{P}(\mathcal{N})$ as the power set of $\mathcal{N}$.
- We define $\mathbb{B}$ as the set of boolean values. Therefore, $\mathbb{B} = \{\texttt{True}, \texttt{False}\}$.
- We define $\mathbb{A}$ as the totally ordered set of sighting accuracies. We denote the sighting containing the location consisting of the entire universe, with a duration of all time, as $\perp$. The sighting accuracy of $\perp$ is $\alpha_\perp$, and $\alpha_\perp$ is used to mean no sighting accuracy. Therefore, $\alpha_\perp$ is the least element in $\mathbb{A}$.

5.3.2 Permissions

Our access control model is based on targets creating permissions that specify which requesters are entitled to obtain sightings of which targets, under what circumstances these sightings are released, and the maximum accuracy of these sightings. A target can create multiple permissions for each public name, and all targets and requesters are identified within these permissions using public names. The infrastructure uses

our access control model to release sightings of targets in accordance with their permissions.

Our access control model is specifically designed for use with our architecture, which consists of both indirect requesters and proxy requesters. Therefore, our access control model enables users to specify two different types of permission. The first type of permission is used to specify which indirect requesters are trusted, and therefore can obtain sightings, and the second type of permission is used to specify which proxy requesters are trusted, and therefore can obtain sightings. This has the effect of creating a whitelist for users, and a separate whitelist for LBSs. The access control model will only allow sightings to be released if it is presented with both a valid permission specified in terms of users, and a valid permission specified in terms of LBSs.

Indirect Access Permission

An IAP is used by a target to specify which indirect requesters can indirectly access the infrastructure via a proxy requester to obtain its sightings, and at what accuracy these sightings can be obtained. Since an IAP only allows an indirect requester to indirectly access the infrastructure, this permission cannot be used on its own.

We define the set of IAP permissions as:

$$\mathcal{P}_{\texttt{IAP}} = \mathcal{N} \times \mathbb{P}(\mathcal{N}) \times \mathbb{P}(\mathcal{N}) \times \mathbb{B} \times \mathbb{A}$$

Given the IAP $\langle t, I, P, c, \alpha \rangle \in \mathcal{P}_{\texttt{IAP}}$ we have:

- t is the public name of the target that created this permission.
- I specifies which indirect requesters are allowed to use this permission to obtain sightings of t indirectly via a proxy requester in P.
- P specifies which proxy requesters are allowed to use this permission to act as a proxy in order to obtain sightings of t directly from the infrastructure on behalf of an indirect requester in I.

- c is used to determine if a sighting will be released based on parameters that are outside the scope of both the access control model and the sighting blurring algorithm. The access control model will never release a sighting accuracy if this value is `False`. Therefore, if t does not want to specify any condition, then it simply sets this value to `True`.

- α is the sighting accuracy at which t can be sighted.

Proxy Access Permission

A PAP is used by a target to specify which proxy requesters can directly access the infrastructure when operating as proxies for indirect requesters to obtain sightings of the target, and at what accuracy these sightings can be obtained. Since a PAP only allows a proxy requester to directly access the infrastructure when operating as a proxy for an indirect requester, this permission cannot be used on its own. Therefore, it is always used with an IAP.

We define the set of PAP permissions as:

$$\mathcal{P}_{\texttt{PAP}} = \mathcal{N} \times \mathbb{P}(\mathcal{N}) \times \mathbb{P}(\mathcal{N}) \times \mathbb{B} \times \mathbb{A} \times \mathbb{B}$$

Given the PAP $\langle t, P, I, c, \alpha, o \rangle \in \mathcal{P}_{\texttt{PAP}}$ we have:

- t is the public name of the target that created this permission.

- P specifies which proxy requesters are allowed to use this permission to act as a proxy in order to obtain sightings of t directly from the infrastructure on behalf of an indirect requester in I.

- I specifies which indirect requesters are allowed to use this permission to obtain sightings of t indirectly via a proxy requester in P.

- c is used to determine if a sighting will be released based on parameters that are outside the scope of both the access control model and the sighting blurring algorithm. The access control model will never release a sighting accuracy if

this value is `False`. Therefore, if t does not want to specify any condition, then it simply sets this value to `True`.

- α is the sighting accuracy at which t can be sighted.
- o is used to specify which of the sighting accuracies used in the IAP and the PAP has priority. If this value is `True`, then the access control model should use the sighting accuracy specified in this PAP. Otherwise the sighting accuracy specified in the IAP is used. This priority is assigned to the appropriate sighting accuracy regardless of which sighting accuracy is more accurate. In other words, the override parameter can be used to increase or decrease the sighting accuracy specified in the IAP. t sets this parameter based on its preferences, and t's reasoning for using this parameter is external to the access control model.

The set of all access permissions is defined as $\mathcal{P}_{\texttt{AP}} = \mathcal{P}_{\texttt{IAP}} \cup \mathcal{P}_{\texttt{PAP}}$.

5.3.3 Access Control Algorithm

Our access control model uses the access control algorithm described in Algorithm 1 when an indirect requester accesses the infrastructure via a proxy requester with an IAP and a PAP.

There are four inputs to the access control algorithm. These are the public name of the indirect requester, the public name of the proxy requester, and the IAP and PAP that they are supplying.

The output of the access control algorithm is always a sighting accuracy. However, in circumstances where the access control algorithm determines that the target is unwilling to allow the requesters to obtain a sighting then the sighting accuracy will be $\alpha_{\perp}$.

5.3.4 Examples

In order to demonstrate how our access control algorithm operates on IAPs and PAPs we will present some simple examples. These examples are based on the

Algorithm 1: The Access Control Algorithm

Input: i is the public name of the indirect requester.
Input: p is the public name of the proxy requester.
Input: The supplied IAP $\langle t_1, I_1, P_1, c_1, \alpha_1 \rangle \in \mathcal{P}_{\texttt{IAP}}$.
Input: The supplied PAP $\langle t_2, P_2, I_2, c_2, \alpha_2, o \rangle \in \mathcal{P}_{\texttt{PAP}}$.
Output: The allowed sighting accuracy.
if $t_1 \neq t_2$ **then**
    return $\alpha_\perp$
if $(i \notin I_1) \vee (i \notin I_2)$ **then**
    return $\alpha_\perp$
if $(p \notin P_2) \vee (p \notin P_1)$ **then**
    return $\alpha_\perp$
if $(\neg c_1) \vee (\neg c_2)$ **then**
    return $\alpha_\perp$
if o **then**
    return α_2
return α_1

following definitions of $\mathcal{N}$ and $\mathbb{A}$:

$$\mathcal{N} = \{A_1,\ B_1,\ C_1,\ D_1,\ E_1,\ F_1,\ G_1,\ H_1\}$$
$$\mathbb{A} = \{\alpha_\perp,\ \alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4\}$$

- **Example 1**

  The inputs to the access control algorithm are the indirect requester E_1, the proxy requester G_1, and the following IAP and PAP:

  $$\langle B_1,\ \{D_1,\ E_1\},\ \{G_1\},\ \texttt{True},\ \alpha_2 \rangle$$
  $$\langle E_1,\ \{G_1\},\ \{D_1,\ B_1\},\ \texttt{True},\ \alpha_4,\ \texttt{True} \rangle$$

  The access control algorithm will return a sighting accuracy of $\alpha_\perp$ because the target in each permission is different.

- **Example 2**

  The inputs to the access control algorithm are the indirect requester F_1, the

proxy requester G_1, and the following IAP and PAP:

$$\langle B_1,\ \{D_1,\ E_1\},\ \{G_1\},\ \texttt{True},\ \alpha_2\rangle$$
$$\langle B_1,\ \{G_1\},\ \{D_1,\ E_1,\ F_1\},\ \texttt{True},\ \alpha_4,\ \texttt{True}\rangle$$

The access control algorithm will return a sighting accuracy of $\alpha_\perp$ because $F_1 \notin \{D_1,\ E_1\}$ in the IAP.

- **Example 3**

  The inputs to the access control algorithm are the indirect requester E_1, the proxy requester G_1, and the following IAP and PAP:

$$\langle B_1,\ \{D_1,\ E_1\},\ \{G_1\},\ \texttt{True},\ \alpha_2\rangle$$
$$\langle B_1,\ \{G_1\},\ \{D_1,\ E_1\},\ \texttt{True},\ \alpha_4,\ \texttt{True}\rangle$$

  The access control algorithm will return a sighting accuracy of α_4. If the override parameter in the PAP was `False`, then the access control algorithm would have returned a sighting accuracy of α_2. In this case, specifying a sighting accuracy of α_4 in the PAP is unnecessary, so it can be replaced with $\alpha_\perp$.

5.4 Notation

In this section we describe an abstract syntax for a notation that expresses permissions in our access control model.

5.4.1 Boolean Expressions

Our access control model is based on permissions that contain sets of requesters. There are two significant disadvantages associated with this approach where the sets of requesters are stated explicitly within the permissions. Firstly, targets must revoke old permissions and create new permissions every time that they want to

add or remove a requester from an existing permission. Secondly, every time that the access control algorithm is invoked it must enumerate entire sets of requesters. As the sets of requesters in permissions grow larger, this approach becomes more cumbersome.

The disadvantages of this approach can be overcome by specifying these sets of requesters using set comprehension. However, calculating the set of requesters satisfying some predicate would be inefficient.

Fortunately, our access control algorithm only needs to perform a membership test in order to determine if a given requester is a member of a given set of requesters. This enables us to represent sets as predicates over requester names. Therefore, we will use boolean expressions, which contain a free variable representing the name of the requester to be tested, in our notation instead of predicates.

5.4.2 Abstract Syntax

The use of boolean expressions allows us to define the abstract syntax of IAPs and PAPs as follows:

$$IAP ::= \langle Name,\ B_{\texttt{Exp}},\ B_{\texttt{Exp}},\ B_{\texttt{Exp}},\ Accuracy \rangle$$
$$PAP ::= \langle Name,\ B_{\texttt{Exp}},\ B_{\texttt{Exp}},\ B_{\texttt{Exp}},\ Accuracy,\ B_{\texttt{Exp}} \rangle$$

A $Name$ is a syntactic representation of a value from $\mathcal{N}$, a $B_{\texttt{Exp}}$ is a boolean expression, and an $Accuracy$ is a valid sighting accuracy from $\mathbb{A}$. The concept of a boolean expression is well understood, and therefore we only provide a partial

definition of its abstract syntax:

$$
\begin{aligned}
B_{\texttt{Exp}} ::= & \ \texttt{True} \\
| & \ \texttt{False} \\
| & \ \neg B_{\texttt{Exp}} \\
| & \ B_{\texttt{Exp}} \wedge B_{\texttt{Exp}} \\
| & \ B_{\texttt{Exp}} \vee B_{\texttt{Exp}} \\
| & \ User \in \{Userlist\} \\
| & \ User \in \{User.Attribute\} \\
| & \ User.Attribute \\
| & \ User.Attribute = Value \\
| & \ \ldots
\end{aligned}
$$

A $User$ is either an explicitly named user, or a variable that represents a user. There are three different variables that targets can use in the boolean expressions within the permissions that they create. The target itself is represented using `#t`, the indirect requester is represented using `#i`, and the proxy requester is represented using `#p`. `System` is a special user that represents the system that is hosting the access control model. These variables are needed because the target does not necessarily know who these entities are when it is creating the permission. They are replaced with the actual values by the access control model immediately before the access control algorithm is invoked. The abstract syntax of the partial definition of

a $User$ is defined as follows:

$$
\begin{aligned}
User ::= & \; Name \\
| & \; \#\texttt{t} \\
| & \; \#\texttt{i} \\
| & \; \#\texttt{p} \\
| & \; \texttt{System} \\
| & \; User.Attribute \\
| & \; \ldots
\end{aligned}
$$

An *Attribute* is the name of an attribute of a $User$. There are many different attribute names, and these will depend on the underlying implementation of the access control model. $Userlist$ and $Value$ both have implicit definitions of their abstract syntax, and therefore we have not explicitly defined them.

5.4.3 Modified Access Control Algorithm

The access control algorithm that we previously defined in Algorithm 1 needs to modified to allow boolean expressions to be used in place of the sets. This modified access control algorithm is described in Algorithm 2.

5.4.4 Examples

In order to demonstrate how our modified access control algorithm operates on IAPs and PAPs that contain boolean expressions we will present some more examples. These examples are based on the following example definitions of $Name$ and *Accuracy*:

$$
\begin{aligned}
Name ::= & \; \texttt{Stefano} \mid \texttt{Carlotta} \mid \texttt{Maria} \mid \texttt{Ilaria} \mid \texttt{Alexia} \\
| & \; \texttt{FriendFinder} \mid \texttt{ColleagueFinder} \\
Accuracy ::= & \; \alpha_{\perp} \mid \texttt{1} \mid \texttt{2} \mid \texttt{3} \mid \texttt{4}
\end{aligned}
$$

Algorithm 2: The Modified Access Control Algorithm

Input: i is the public name of the indirect requester.
Input: p is the public name of the proxy requester.
Input: The supplied IAP $\langle t_1, i_1, p_1, c_1, \alpha_1 \rangle$.
Input: The supplied PAP $\langle t_2, p_2, i_2, c_2, \alpha_2, o \rangle$.
Output: The allowed sighting accuracy.
if $t_1 \neq t_2$ **then**
    return $\alpha_\perp$
if $(\neg i_1) \vee (\neg i_2)$ **then**
    return $\alpha_\perp$
if $(\neg p_2) \vee (\neg p_1)$ **then**
    return $\alpha_\perp$
if $(\neg c_1) \vee (\neg c_2)$ **then**
    return $\alpha_\perp$
if o **then**
    return α_2
return α_1

The example values in *Accuracy* can be used by our sighting blurring algorithm, which is described in Chapter 6.

The boolean expressions in our examples also use the following attributes:

- `Friends` is a set attribute that represents a $User$'s friends. In these examples `Maria.Friends` is {`Ilaria, Alexia`}.
- `isUser` is a boolean attribute that is `True` if a $User$ is a user, and `False` if a $User$ is an LBS.
- `IMStatus` is a string attribute that represents a $User$'s instant messenger status.
- `Day` is a time attribute that represents the current day. It is an attribute of the `System` user.
- `CurrentLocation` is an attribute that represents a $User$'s current location, and `HomeLocation` is an attribute that represents a $User$'s home location.

The following examples are based on two requesters indirectly accessing the infrastructure with an IAP and a PAP.

- **Example 4**

The inputs to the modified access control algorithm are the indirect requester `Ilaria`, the proxy requester `ColleagueFinder`, and the following IAP and PAP:

$\langle$`Maria`, #`i` $\in$ `Maria.Friends`, #`p` $\in$ {`FriendFinder`, `ColleagueFinder`}, `True`, `3`$\rangle$

$\langle$`Maria`, #`p` $\in$ {`FriendFinder`}, #`i` $\in$ `Maria.Friends`, `True`, $\alpha_\perp$, `False`$\rangle$

The modified access control algorithm will return a sighting accuracy of $\alpha_\perp$ because #`p` is not a member of {`FriendFinder`} in the PAP.

- **Example 5**

  If the proxy requester in Example 4 is `FriendFinder`, then the modified access control algorithm will return a sighting accuracy of 3.

  Additionally, if the indirect requester is `Stefano`, the modified access control algorithm will return a sighting accuracy of $\alpha_\perp$ because #`i` is not a member of `Maria.Friends` in both the IAP and the PAP. However, if `Maria` adds `Stefano` to her `Friends` attribute, then the modified access control algorithm will return a sighting accuracy of 3. This is significant, because the returned sighting accuracy changes without either the IAP or PAP permissions changing.

- **Example 6**

  The inputs to the modified access control algorithm are the indirect requester `Ilaria`, the proxy requester `FriendFinder`, and the following IAP and PAP:

  $\langle$`Stefano`, #`i.isUser`, #`p` $\in$ {`FriendFinder`}, `True`, `2`$\rangle$

  $\langle$`Stefano`, #`p` $\in$ {`FriendFinder`}, #`i` $\in$ {`Ilaria`, `Maria`, `Alexia`}, $\neg$((#`System.Day` = "`Saturday`") $\vee$ (#`System.Day` = "`Sunday`")), `3`, `True`$\rangle$

  The modified access control algorithm will return a sighting accuracy of 3 if

it is invoked on a weekday, or $\alpha_\perp$ if it is invoked on either a Saturday or a Sunday.

- **Example 7**

  If `Stefano` wanted the indirect requester `Ilaria`, using the proxy requester `FriendFinder`, to receive a sighting with a sighting accuracy of 4 on a weekday, or a sighting accuracy of 2 on either a Saturday or a Sunday, then he would create the following PAP in addition to the IAP in Example 6:

  $$\langle \texttt{Stefano},\ \#\texttt{p} \in \{\texttt{FriendFinder}\},\ \#\texttt{i} \in \{\texttt{Ilaria},\ \texttt{Maria},\ \texttt{Alexia}\},\ \texttt{True},\ 4,$$
  $$\neg((\#\texttt{System.Day} = \text{“}\texttt{Saturday}\text{”}) \vee (\#\texttt{System.Day} = \text{“}\texttt{Sunday}\text{”}))\rangle$$

  This example shows how a boolean expression can be used to specify which sighting accuracy has priority.

- **Example 8**

  The inputs to the modified access control algorithm are the indirect requester `Ilaria`, the proxy requester `FriendFinder`, and the following IAP and PAP:

  $$\langle \texttt{Maria},\ \#\texttt{i} \in \{\texttt{Ilaria},\ \texttt{Alexia}\},\ \neg\#\texttt{p.isUser},\ \texttt{True},\ 2\rangle$$
  $$\langle \texttt{Maria},\ \#\texttt{p} \in \{\texttt{FriendFinder}\},\ \#\texttt{i.isUser},\ \texttt{True},\ \alpha_\perp,\ \texttt{False}\rangle$$

  The modified access control algorithm will return a sighting accuracy of 2. This IAP effectively allows `Ilaria` to obtain sightings for `Maria`, with a sighting accuracy of 2, using any LBS that `Maria` trusts. However, if `Maria` has a higher trust in sighting requests jointly from `Ilaria` and `FriendFinder`, then `Maria` creates the following IAP that can be used with the previous PAP:

  $$\langle \texttt{Maria},\ \#\texttt{i} \in \{\texttt{Ilaria},\ \texttt{Alexia}\},\ \#\texttt{p} \in \{\texttt{FriendFinder}\},\ \texttt{True},\ 3\rangle$$

- **Example 9**

The inputs to the modified access control algorithm are the indirect requester `Maria`, the proxy requester `FriendFinder`, and the following IAP and PAP:

$\langle$`Stefano`, $\#$`i` $\in \{$`Ilaria`, `Maria`, `Alexia`$\}$, $\neg\#$`p.isUser`, `True`, `1`$\rangle$

$\langle$`Stefano`, $\neg\#$`p.isUser`, $\#$`i` $\in \{$`Ilaria`, `Maria`, `Alexia`$\}$, `True`, $\alpha_\perp$, `False`$\rangle$

The modified access control algorithm will return a sighting accuracy of 1. This IAP and PAP have the effect of allowing `Ilaria, Maria` and `Alexia` to indirectly retrieve `Stefano`'s sightings with a sighting accuracy of 1 using any LBS as the proxy requester. Therefore, this PAP enables `Stefano` to allow `Ilaria, Maria` and `Alexia` to delegate the sighting request rights that he gave them to any LBS.

5.4.5 Semantics

The access control model must enforce certain semantic rules at run-time, and if these rules are violated then it will return a sighting accuracy of $\alpha_\perp$.

Consider again a system based on the previous definitions, where the inputs to the modified access control algorithm are the indirect requester `Ilaria`, the proxy requester `FriendFinder`, and the following IAP and PAP:

$\langle$`Maria`, $\#$`i` $\in$ `Maria.Friends`, $\neg\#$`p.isUser`, `Alexia.IMStatus` = "`Online`", `3`$\rangle$

$\langle$`Maria`, $\#$`p` $\in \{$`FriendFinder`$\}$, $\#$`i` $\in$ `Maria.Friends`, `True`, $\alpha_\perp$, `False`$\rangle$

The modified access control algorithm will return a sighting accuracy of 3 if `Alexia`'s instant messenger status is "Online", and otherwise it will return a sighting accuracy of $\alpha_\perp$.

If `Ilaria` successfully uses this IAP to obtain a sighting accuracy of 3 for `Maria`, then `Ilaria` can determine that `Alexia`'s instant messenger status was "Online". If `Maria` can determine that `Ilaria` successfully uses this IAP to obtain a sighting accuracy of 3, then `Maria` also knows that `Alexia`'s instant messenger status was "Online". However, `Alexia` has not been part of this invocation of the access control

model, and she might be unwilling to share her instant messenger status with `Ilaria` and `Maria`.

Therefore, only the target's attributes and the indirect requester's attributes can be accessed within an IAP, and only the target's attributes and the proxy requester's attributes can be accessed within a PAP. (The `System` user is an exception, because its attributes can be contained in any permission.) The access control model returns a sighting accuracy of $\alpha_\perp$ if it is presented with a permission that contains attributes for a user who is neither the target nor the appropriate requester.

The use of the targets' and requesters' attributes within permissions can reveal additional information. For example, consider that the inputs to the modified access control algorithm are the indirect requester `Ilaria`, the proxy requester `FriendFinder`, and the following IAP and PAP:

$$\langle \texttt{Maria},\ \#\texttt{i} \in \texttt{Maria.Friends},\ \#\texttt{p} \in \{\texttt{FriendFinder},\ \texttt{ColleagueFinder}\},$$
$$\#\texttt{i.IMStatus} = \text{“}\texttt{Online}\text{”},\ 2\rangle$$
$$\langle \texttt{Maria},\ \#\texttt{p} \in \{\texttt{FriendFinder}\},\ \#\texttt{i} \in \texttt{Maria.Friends},\ \texttt{True},\ \alpha_\perp,\ \texttt{False}\rangle$$

The modified access control algorithm will return a sighting accuracy of 2 if `Ilaria`'s instant messenger status is "Online", and otherwise it will return a sighting accuracy of $\alpha_\perp$.

Again, if `Ilaria` successfully uses this IAP to obtain a sighting accuracy of 2 for `Maria`, then `Maria` can determine that `Ilaria`'s instant messenger status was "Online". Initially, this ability of a target to obtain information about the requesters appears to be a breach of the requesters' security. This becomes more apparent if we consider an attribute such as `atHome` that returns `True` if a $User$ is in his/her house and `False` otherwise.

However, there is no loss of security associated with this ability. This is because each requester can inspect the permission to determine which of its attributes will be accessed, and hence shared with the target. If it is not willing to share these attributes with the target then it does not use the permission. Similarly, the target will not create any permissions containing attributes that reveal additional information

to the requesters.

5.5 Implementation

Our access control model can be implemented as either a centralised system or a distributed system. In the centralised implementation the infrastructure is responsible for storing and selecting permissions. The main advantage of this implementation is that the requesters do not need to store and manage permissions. However, there are also significant disadvantages associated with this implementation. In particular, the infrastructure must determine which combinations of IAPs and PAPs to use. This could be difficult, because it is likely that there will be many suitable combinations for any indirect requester and proxy requester pair.

Therefore, we decided to focus on a distributed implementation of our access control model, where the requesters are responsible for storing and selecting permissions. When an indirect requester and a proxy requester pair require a sighting of a target, the indirect requester selects an IAP and the proxy requester selects a PAP. Both of these permissions are then supplied to the infrastructure.

The main advantage of this implementation is that there is no need to centrally store and manage permissions, and this facilitates scalability. Also, requesters can choose the permissions that they want to supply to the infrastructure, which gives them complete control over permission selection. The most significant disadvantage of this implementation is that users must manage the permissions themselves, and these permissions may need to be stored on mobile devices that have limited resources.

The distributed nature of these permissions raises some important security issues, and in particular, it must be possible to verify the authenticity and integrity of permissions. Therefore, we propose that users sign the permissions that they create, using our adapted meditated identity based cryptography system (see Section 4.9). Users also need to be able to revoke the permissions that they create. Therefore, each permission has a commencement date and an expiry date, and revoked permissions are identified by the infrastructure until they expire.

When an indirect requester and a proxy requester require a sighting of a target, the indirect requester selects an IAP and the proxy requester selects a PAP. Both of these permissions are then supplied to the infrastructure, and the infrastructure performs the following steps:

1. It uses our meditated identity based cryptography system to test the validity of both permissions by verifying their signatures using the public key of the user who created the permissions.

2. It verifies that the current date is between the commencement date and the expiry date of both permissions, and it ensures that neither permission has been revoked.

3. It parses both permissions and ensures that they are both syntactically and semantically correct.

4. It invokes the access control model to evaluate the IAP and PAP combination in order to determine the appropriate sighting accuracy for the target.

Although we are using our adapted meditated identity based cryptography system, it is possible to use any PKI with our access control model.

5.6 Comparison with Related Research

In this section we describe how our access control model satisfies the requirements that we identified in Section 5.2. We also compare it with the related research described in Section 3.3 in terms of our requirements. In particular, we identify its advantages and disadvantages, and we identify the similarities and differences between it and the related research.

- **Requirement 1**

  The access control model must allow users to specify who is entitled to obtain their sightings in terms of users, LBSs, or both users and LBSs.

Our access control model satisfies this requirement by allowing users to specify who is entitled to obtain their sightings in terms of users or LBSs. Therefore, it is similar to the single subject centralised access control models and the single subject distributed access control models in this regard.

Our access control model fully satisfies this requirement by also allowing users to specify who is entitled to obtain their sightings in terms of both users and LBSs. This aspect of our access control model is different to both the single subject centralised access control models and the single subject distributed access control models. Furthermore, it allows users to independently express levels of trust in both users and LBSs. However, it requires that a user and an LBS cooperate with each other in order to obtain a user's sightings. Therefore, our access control model is similar to the dual subject centralised access control models in this regard. However, it differs from the dual subject centralised access control models because none of them recognise users and LBSs as distinct, but equally important, subjects.

The advantage of our access control model allowing users to specify who is entitled to obtain their sightings in terms of users, LBSs, or both users and LBSs, is that it provides users with more control over the release of their sightings. Perhaps the most significant disadvantage of this extra control is that it introduces additional complexity to the users.

- **Requirement 2**

  The access control model must allow users to specify the circumstances in which their sightings are released.

  Our access control model satisfies this requirement because both the IAPs and the PAPs include a condition that enables the user who creates the permission to specify the circumstances in which his/her sightings are released. This condition can be based on any parameters that are outside the scope of the access control model, and which are supported by the underlying implementation of the access control model.

Our access control model is similar to several of the single subject centralised access control models and dual subject centralised access control models described in the related research because they also satisfy this requirement by allowing users to specify the circumstances in which their sightings are released.

- **Requirement 3**

  The access control model must allow users to specify the maximum sighting accuracy of any sightings that are released.

  Our access control model satisfies this requirement because both the IAPs and the PAPs allow the user who creates the permission to specify the maximum sighting accuracy of any of his/her sightings that are released. Our access control model does not make any assumptions about what type of sighting blurring algorithm will subsequently be used before the user's sightings are released. Therefore, it treats the sighting accuracies in an algorithm neutral manner. However, in Chapter 6 we will describe the sighting accuracies that we have defined for use by our sighting blurring algorithm.

  Our access control model is similar to many of the access control models described in the related research that satisfy this requirement because they also allow users to specify the sighting accuracy of any sightings that are released. However, these access control models generally describe sighting accuracies that are specifically designed for a single type of sighting blurring algorithm.

  The advantage of our access control model allowing users to specify the maximum sighting accuracy of any sightings that are released, is that it provides users with more control over the release of their sightings. Perhaps the most significant disadvantage of this extra control is that it introduces additional complexity to the users.

Some of the access control models described in the related research support permission delegation. Our access control model is similar to these access control models in this regard, because it allows users to create permissions that can be delegated.

The object in every permission within our access control model is always a user. This differs from some of the access control models described in the related research that allow the object in the permissions to be a location.

Finally, our access control model can be implemented as either a centralised access control model or as a distributed access control model. However, we decided to focus on the distributed implementation of our access control model, and our implementation has many similarities with the single subject distributed access control models described in the related research that use public key cryptography and certificates. The main advantage of the distributed implementation of our access control model is that there is no need to centrally store and manage permissions. This facilitates both redundancy and scalability. The most significant disadvantage of the distributed implementation of our access control model is that users and LBSs must store and manage the permissions.

5.7 Conclusions

In this chapter we described an access control model that is implemented within the infrastructure within our architecture. Our access control model is based on users who are targets creating permissions for users who are indirect requesters, and for LBSs that are proxy requesters. There are two types of permission within our access control model - IAPs that are used by indirect requesters, and PAPs that are used by proxy requesters. These permissions specify which requesters are entitled to obtain sightings of which targets, under what circumstances these sightings are released, and the maximum accuracy of these sightings. Our access control model is then responsible for releasing users' sightings in accordance with their permissions. When an indirect requester and a proxy requester pair request a sighting of a target, they must supply the infrastructure with an IAP and a PAP.

The main novelty of our access control model is that it enables users to specify two different types of permission - IAPs and PAPs. This has the effect of creating a whitelist for users, and a separate whitelist for LBSs. The access control model will only allow sightings to be released if it is presented with both a valid permission

specified in terms of users, and a valid permission specified in terms of LBSs.

Finally, we will use the sighting accuracy output from our access control model to determine the sighting accuracy part of the input to our sighting blurring algorithm in Chapter 6.

Chapter 6

Sighting Blurring Algorithm

6.1 Introduction

In Chapter 4 we described an architecture for users, an infrastructure, and LBSs, which facilitates the operation of these LBSs over the Internet. We also described a protocol that enables these three entities to achieve three-party mutual identification and authentication. In particular, this protocol allows users to simultaneously identify and authenticate themselves to the infrastructure using one identity, and to the LBS using another identity. This protocol then guarantees the confidentiality, integrity, and non-repudiation of all subsequent messages.

In Chapter 5 we described an access control model that is implemented within the infrastructure. This access control model is based on users who are targets creating permissions for users who are indirect requesters, and for LBSs that are proxy requesters. These permissions specify which requesters are entitled to obtain sightings of which targets, under what circumstances these sightings are released, and the maximum accuracy of these sightings.

In this chapter we describe the sighting blurring algorithm that we have developed, which is implemented within the infrastructure in our architecture. Our sighting blurring algorithm offers users increased privacy by decreasing the accuracy of their sightings based on the maximum sighting accuracy allowed by the access control model described in Chapter 5.

6.2 Mathematical Model

In this section we define the concepts of sightings, sighting accuracy, sighting blurring, and location translation, in a mathematical context. These concepts will then be used extensively in the remainder of this chapter in order to describe our sighting blurring algorithm.

6.2.1 Sightings

A sighting describes where a target was, and when it was there. Therefore, each sighting contains a location that describes where the user was sighted, and a time interval that describes when the user was sighted.

In order to define a location, we will first define a position $p \in \mathcal{P}$ as a tuple in a coordinate space. A location $l \in \mathcal{L}$ is then defined as an area such that $p \in l$ is true if the position p is within the area of the location l. A location is defined as a set of polygons. However, we consider that locations approximate squares for the sake of simplicity.

We define a subset relation on locations as:

$$l \subseteq l' \Leftrightarrow \forall p \in l \cdot p \in l'$$

The size of the area making up a location l is defined by the real number $|l|$. If $l \subseteq l'$, then $|l| \leq |l'|$.

In order to define a time interval, we will first define a time $t \in \mathcal{T}$ as an absolute time relative to some epoch. This can be represented as an integer value, such as the number of milliseconds from a given time and date. A time interval $i \in \mathcal{I}$ is a range defined by a start time and an end time. $t \in i$ is true if the time t occurs within the time interval i.

We define a subset relation on time intervals as:

$$i \subseteq i' \Leftrightarrow \forall t \in i \cdot t \in i'$$

The duration of a time interval i is defined by the real number $|i|$. If $i \subseteq i'$ then

$|i| \leq |i'|$.

A sighting $s \in \mathcal{S}$ is a pair $\langle l, i \rangle \in \mathcal{L} \times \mathcal{I}$, where $|l| > 0$ and $|i| > 0$. It states that a target was located somewhere within l at some time during the time interval i. This definition requires the location component l of sighting s to be a non-empty area. Therefore, a sighting can never represent a single position. In reality this is not an issue, because the underlying positioning technology does not return points. However, we can always approximate a single position p by selecting a location l with a sufficiently small area such that $p \in l$. Similarly, a time interval within a sighting can never represent a single point in time.

We define a sighting $s_1 = \langle l_1, i_1 \rangle$ to approximate another sighting $s_2 = \langle l_2, i_2 \rangle$ if $l_2 \subseteq l_1$ and $i_2 \subseteq i_1$. We denote this as $s_1 \sqsubseteq s_2$, and we note that $\sqsubseteq$ forms a partial order. We denote the sighting containing the location consisting of the entire universe, with a duration of all time, as $\perp$. Clearly $\perp \sqsubseteq s$ for all sightings $s \in \mathcal{S}$.

6.2.2 Sighting Accuracy

Consider a sighting s, with two approximations s_1 and s_2. This is denoted as $s_1 \sqsubseteq s$ and $s_2 \sqsubseteq s$. Assume that both s_1 and s_2 have the same time interval, but overlapping locations. Therefore, $s_1 \not\sqsubseteq s_2$ and $s_2 \not\sqsubseteq s_1$. In this case we cannot claim that either sighting approximates the other. Therefore, we have no way to compare s_1 and s_2.

In order to allow a useful comparison of sightings that are not comparable using $\sqsubseteq$ we introduce a metric to compute the quality of a sighting as a single numeric value. We call this metric a sighting accuracy function.

A function $f \in \mathcal{S} \rightarrow \mathbb{R}$ is a sighting accuracy function if it has the following properties:

- The sighting accuracy function applied to the location consisting of the entire universe with a duration of all time will produce a sighting quality of zero. This is denoted as:

$$f(\perp) = 0$$

- The sighting accuracy function applied to any possible sightings will always produce sighting qualities that are greater than or equal to zero. This is denoted as:

$$\forall s \in \mathcal{S} \cdot f(s) \geq 0$$

- Given any two sightings, if one sighting approximates the other sighting, then the sighting quality of the first sighting will be less than or equal to the sighting quality of the second sighting. This is denoted as:

$$\forall s_1, s_2 \in \mathcal{S} \cdot s_1 \sqsubseteq s_2 \Rightarrow f(s_1) \leq f(s_2)$$

As an example of a sighting accuracy function, we can define $f_1 \in \mathcal{S} \rightarrow \mathbb{R}$ as follows:

$$f_1(\perp) = 0$$
$$f_1(\langle l, i \rangle) = \frac{1}{|l| \times |i|}$$

We let $\mathcal{A}$ be the set of functions of type $\mathcal{S} \rightarrow \mathbb{R}$ that satisfy these properties.

6.2.3 Sighting Blurring

Sighting blurring is the process of taking an existing sighting, and a desired sighting accuracy that is less than the sighting accuracy of the existing sighting, and creating a new sighting that has a sighting accuracy that is less than or equal to the desired sighting accuracy.

Many LBSs will not be hindered from offering useful services by sighting blurring, because these LBSs will not require very accurate sightings. For example, an LBS that enables a user to locate all touristic attractions within 100,000m does not need the user's sightings with an accuracy greater than 1,000m.

Given a sighting accuracy function $f \in \mathcal{A}$, then a function $g \in \mathcal{S} \times \mathbb{R} \rightarrow \mathcal{S}$ is a sighting blurring function if it has the following properties:

- The sighting produced by a sighting blurring function will always approximate the original sighting that was used by the sighting blurring function. This is denoted as:

$$g(s, \alpha) \sqsubseteq s$$

- The sighting accuracy of the sighting produced by a sighting blurring function will always be less than or equal to the original sighting accuracy that was used by the sighting blurring function. This is denoted as:

$$f(g(s, \alpha)) \leq \alpha$$

We define $\mathcal{B}$ as the set of all functions of type $\mathcal{S} \times \mathbb{R} \rightarrow \mathcal{S}$ that satisfy these properties.

Given the sightings $s_1 = \langle l_1, i_1 \rangle$ and $s_2 = \langle l_2, i_2 \rangle$, and the sighting blurring function $g(s_1, \alpha) = s_2$, then we define g as a sighting blurring function that performs spatial blurring if, and only if, $l_1 \subset l_2$. Similarly, we define g as a sighting blurring function that performs temporal blurring if, and only if, $i_1 \subset i_2$.

For all functions $f \in \mathcal{A}$, the function g defined by $g(s, \alpha) = \perp$ is a sighting blurring function that uses a combination of spatial blurring and temporal blurring. However, this sighting blurring function is not very useful because the LBS will not be able to offer a meaningful service to the user. Therefore, it is vital that the sighting blurring function provides the correct amount of blurring and at the same time produces useful approximations.

In practice a user, as the indirect requester, and an LBS, as the proxy requester, will require sightings with a minimum sighting accuracy, α_{min}, and a user, as a target, will be prepared to allow the release of sightings with a maximum sighting accuracy, α_{max}. In order for the indirect requester to obtain a useful service from the proxy requester we require that $\alpha_{min} \leq \alpha_{max}$.

However, the infrastructure can use any sighting accuracy, α_{actual}, with the sight-

ing blurring algorithm to blur s as long as:

$$\alpha_{min} \leq f(g(s, \alpha_{actual})) \leq \alpha_{max}$$

This enables the infrastructure to provide the target with additional privacy.

6.2.4 Location Translation

Location translation is the process of converting a location expressed using one representation into the same location expressed using a different representation. The functions used to perform location translations are always either *one-to-one* functions or *many-to-one* functions.

It is important to note that both the original location representation and the translated location representation can be any type of location representation (see Section 2.3.1). It is also important to note that location translation functions can change the accuracy of the location information. In particular, the one-to-one location translation functions will not change the accuracy of the location information, whereas the many-to-one location translation functions will decrease the accuracy of the location information.

The most relevant types of location translation functions are *mathematical-to-mathematical*, *political-to-mathematical*, and *mathematical-to-political* location translation functions. Political-to-mathematical location translation functions are often known as *geocoding* functions, and mathematical-to-political location translation functions are often known as *reverse geocoding* functions. Mathematical-to-mathematical location translation functions are one-to-one location translation functions where there is no change in the location accuracy. Both political-to-mathematical location translation functions and mathematical-to-political location translation functions can be either one-to-one, in which case there is no change in the location accuracy, or many-to-one, in which case there is a decrease in the location accuracy.

6.3 Requirements

We have identified the following requirements that our sighting blurring algorithm must satisfy:

- **Requirement 1**

  The sighting blurring algorithm will generate sightings with exactly the requested sighting accuracy.

  In particular, the sighting blurring algorithm must not generate sightings that have a greater sighting accuracy than the requested sighting accuracy because this could compromise the target's privacy. Similarly, the sighting blurring algorithm must not generate sightings that have a lower sighting accuracy than the requested sighting accuracy because this could reduce the usefulness of the LBS. Therefore, the requested sighting accuracy must be greater than or equal to the minimum sighting accuracy that is required by the indirect requester and proxy requester pair, and less than or equal to the maximum sighting accuracy that is allowed by the target.

- **Requirement 2**

  The sighting blurring algorithm will generate sightings consisting of locations and time intervals that contain the targets.

  It is necessary that the generated blurred sightings are always correct, because otherwise the proxy requester would be providing a service that would be perceived as being incorrect by the indirect requester. This could undermine the acceptance and take-up of LBSs.

- **Requirement 3**

  The only sightings required as input by the sighting blurring algorithm will be the current sightings of the targets.

  In particular, the sighting blurring algorithm must be independent of the sightings of any other users. Therefore, the sighting blurring algorithm cannot be influenced by factors such as the proximity of other users to the target. This

is critical because the infrastructure may not be able to obtain sightings of other users due to limitations of the underlying technology of the locatables in the network operators' networks. Even if the network operator could supply sightings of these other users, the infrastructure will incur a charge for each sighting. This would make the operation of the sighting blurring algorithm costly. Additionally, the infrastructure may not be allowed to obtain the sightings of the users who are not involved in the current invocation of the LBS due to either the users' privacy preferences or the legal requirements that govern the use of their sightings.

- **Requirement 4**

  The sighting blurring algorithm will blur sightings so that the target could be located anywhere within the location at any instant within the time interval with equal probability.

  The privacy offered to the target would be compromised if this requirement were not satisfied.

- **Requirement 5**

  The sighting blurring algorithm must be capable of blurring sightings that are part of a stream of sightings of a particular target, and knowledge of previous blurred sightings should not affect the sighting accuracy of any future sightings.

  The privacy offered to the target would be compromised if this requirement were not satisfied.

- **Requirement 6**

  The sighting blurring algorithm must be capable of generating blurred sightings with a consistent sighting accuracy.

  This will facilitate the creation of consistent LBSs, which in turn will encourage the usage of these LBSs by users.

In Section 6.5.3 we will compare our sighting blurring algorithm and its requirements with the related research described in Section 3.4.

6.4 Sighting Blurring Algorithm

In this section we describe the concepts, requirements, and restrictions of our sighting blurring algorithm. We also provide a detailed description of our sighting blurring algorithm, which we have developed to address the requirements described in Section 6.3.

6.4.1 Concepts

There are several concepts that we use as building blocks in the description of our sighting blurring algorithm.

Magnitudes

A *magnitude* is a number that describes a quantity of length. Magnitudes are denoted as m_1, m_2, m_3, ..., and they form a sequence $\mathcal{M}$. This is denoted as follows:

$$\mathcal{M} = \langle m_1, m_2, m_3, \ldots, m_n \rangle$$

There are two properties that all magnitudes satisfy:

- Every magnitude in the sequence is greater than the following magnitude. That is, $m_j > m_{j+1}$ for all $j \in \{1, \ldots, n-1\}$.
- Every magnitude in the sequence is an integer multiple of the following magnitude. That is, $m_{j+1} | m_j$ for all $j \in \{1, \ldots, n-1\}$.

Typical values for these magnitudes that are likely to be useful from a user's perspective might be 100,000m, 10,000m, 1,000m, 100m, and 10m.

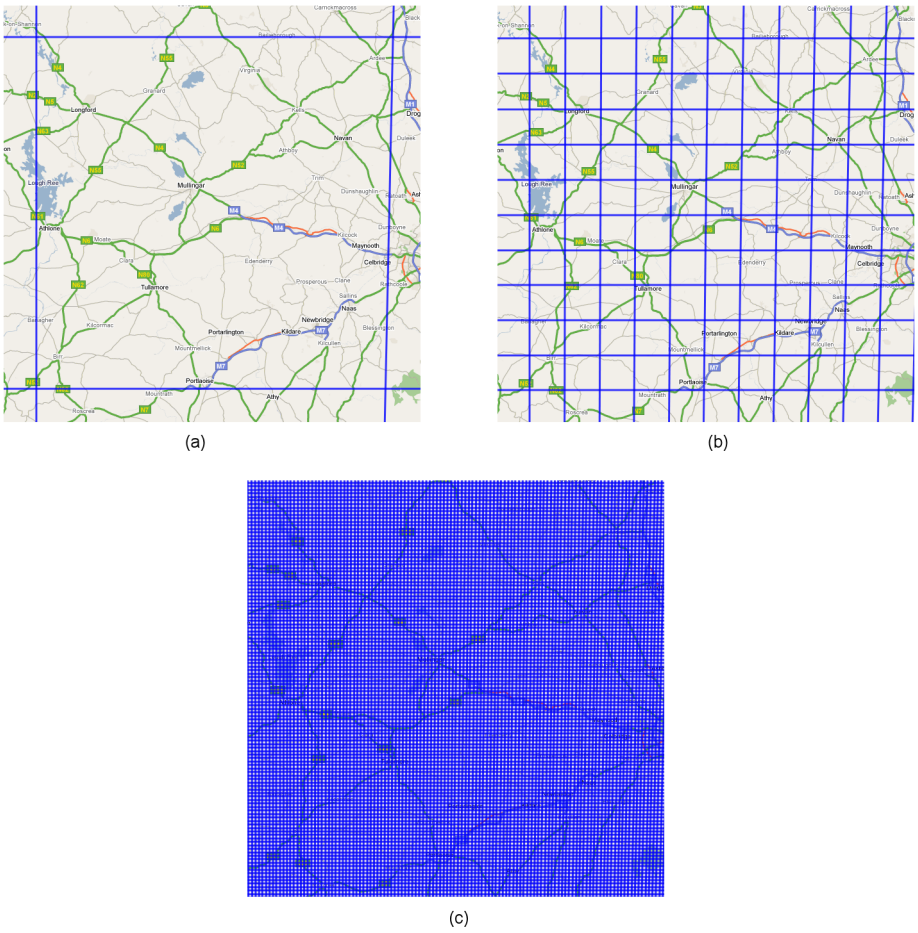

Figure 6.1: Grid Examples (Maps © Google Maps)

Grids

For each magnitude, the area that the sighting blurring algorithm covers is divided into a grid of non-overlapping squares with a common origin. Each square in this grid has a side length that is equal to the magnitude, and each of these squares is in a fixed location. The reason for specifying that the grid squares are non-overlapping and in fixed locations is to prevent an attacker from calculating the intersection of two or more squares.

An example of three different grids covering the same area is shown in Figure 6.1. The square in Figure 6.1 (a) has a side length of 100,000m, and this is ten times greater than the side length of each square in Figure 6.1 (b). Similarly, each square

in Figure 6.1 (b) has a side length that is ten times greater than the side length of each square in Figure 6.1 (c).

Locations

A location within any grid is simply a square. Every square in every grid can be specified by the coordinates of its southwest corner, and the magnitude of the square side. Therefore, a location can be specified as $(x \times m_j, y \times m_j, m_j)$. Alternatively, if $\mathcal{M}$ is known, then a location can be specified as (x, y, j).

Durations

A duration is a number that describes an amount of time, rather than an absolute time. Durations are denoted as d_1, d_2, d_3, ..., and they form a sequence $\mathcal{D}$. This is denoted as follows:

$$\mathcal{D} = \langle d_1, d_2, d_3, \ldots, d_n \rangle$$

There are several properties that all duration sequences satisfy. These are:

- Every duration in the sequence is greater than the following duration. That is, $d_j > d_{j+1}$ for all $j \in \{1, \ldots, n-1\}$.
- For every magnitude in $\mathcal{M}$ there is a corresponding duration in $\mathcal{D}$. That is, $|\mathcal{M}| = |\mathcal{D}|$.
- Each duration is chosen to enable a target who is in a grid square of magnitude m_j to move to any location inside an adjacent square in the duration d_j. Therefore, each d_j is calculated by dividing this distance by the average speed of a typical target.

These durations are pre-determined average values rather than dynamic values for a specific target for two reasons. Firstly, our sighting blurring algorithm is not aware of that specific target's current speed and direction due to Requirement 3.

Secondly, the requester may gain additional information pertaining to that target's current speed and direction by observing how its durations are changing.

It is worth noting that m_j/d_j is not constant for all $j \in \{1, \ldots, n\}$. This is because the average speed of a typical target will vary depending on the distance that it travels and the duration. For example, if the average speed is calculated over a relatively small distance and duration, then the average speed will be quite slow. However, as the distance and duration increase it is more likely that the average speed of the target increases also. This is because targets tend to travel faster when they have greater distances to travel.

Therefore, a time interval can be specified as (t, d_j). Alternatively, if $\mathcal{D}$ is known, then a time interval can be specified as (t, j).

Sightings

We previously stated that a sighting consisted of a location and a time interval. We denoted this as $s = \langle l, i \rangle$. We have also stated that a location can be specified within the context of our sighting blurring algorithm as (x, y, j). We can also specify the time interval within the context of our sighting blurring algorithm as (t, j). Therefore, we can redefine a sighting as $s = \langle (x \times m_j, y \times m_j), t, m_j, d_j \rangle$, or if both $\mathcal{M}$ and $\mathcal{D}$ are known, then a sighting can be specified as $s = \langle (x, y), t, j \rangle$.

Sighting Accuracies

In the context of our sighting blurring algorithm a magnitude is used to specify the side length of the grid square that represents the target's location, and the duration is used to specify for how long the sighting is valid, and hence to determine how frequently new sightings are created for the target.

Therefore, the sighting accuracy is based on (m_j, d_j), and f_1 can be redefined as follows:

$$f_1(\bot) = 0$$
$$f_1(\langle (x \times m_j, y \times m_j), t, m_j, d_j \rangle) = \frac{1}{m_j^2 \times d_j}$$

If both $\mathcal{M}$ and $\mathcal{D}$ are known, then we can define a new sighting accuracy function $f_2 \in \mathcal{A}$ as follows:

$$f_2(\perp) = 0$$
$$f_2(\langle (x, y), t, j \rangle) = j$$

6.4.2 Sighting Blurring Algorithm

In this section we provide both a description and a formal definition of our sighting blurring algorithm.

Overview

The basis for our sighting blurring algorithm is that it offers targets increased privacy by performing spatial blurring on the location components of their sightings. In particular, it blurs a target's input sighting for an indirect requester and proxy requester pair by producing a new sighting with a larger location component. It does not perform temporal blurring, because this reduces an LBS's ability to offer a useful service. Instead, our sighting blurring algorithm will not produce any new blurred sightings of the target for the same indirect requester and proxy requester pair until a specific duration has elapsed. This has the effect of limiting the frequency with which new sightings can be obtained for a target by an indirect requester and proxy requester pair, and this frequency is a function of the area of the location component of the sighting that was produced by the spatial blurring. Therefore, our sighting blurring algorithm is a combination of a spatial blurring algorithm and a frequency based blurring algorithm.

Inputs and Output

Our sighting blurring algorithm requires the following five inputs:

- The public name of the indirect requester.
- The public name of the proxy requester.

- The public name of the target.
- The input sighting of the target, which is denoted as $s_{input} = \langle (x_{input}, y_{input}), t_{input}, input \rangle$. Therefore, its sighting accuracy is calculated as $f_2(s_{input}) = input$, where $input \in \{1, \ldots, n\}$.
- The desired sighting accuracy of the output sighting. This is denoted as $output$, where $output \in \{1, \ldots, n\}$. The sighting accuracy of the input sighting must be greater than or equal to the desired sighting accuracy of the output sighting. That is, $input \geq output$.

The output of our sighting blurring algorithm is a blurred sighting, which is denoted as $s_{output} = \langle (x_{output}, y_{output}), t_{output}, output \rangle$. Therefore, its sighting accuracy is calculated as $f_2(s_{output}) = output$.

Description

The first time that our sighting blurring algorithm is requested to blur a sighting of the target using the desired sighting accuracy of $output$, for each indirect requester and proxy requester pair, it operates as follows:

1. It chooses the grid that has a side length of m_{output}.
2. From this grid it chooses the square that fully contains the input location $(x_{input}, y_{input}, input)$. This square is used to identify the blurred location, and it is denoted as $(x_{output}, y_{output}, output)$. The blurred location will always fully contain the input location, due to the way that magnitudes and grids are defined.
3. It creates t_{output} so that it is equal to t_{input}.
4. The blurred sighting $s_{output} = \langle (x_{output}, y_{output}), t_{output}, output \rangle$ is generated, stored, and returned.

If the same indirect requester and proxy requester pair require the sighting blurring algorithm to blur a new input sighting $s_{input'} = \langle (x_{input'}, y_{input'}), t_{input'}, input \rangle$,

for the same target with the same desired sighting accuracy of *output*, then it will operate as follows:

1. It compares the current time, $t_{current}$, with the t_{output} and *output* of the previous blurred sighting s_{output}.

2. If less time than d_{output} has elapsed since t_{output}, then s_{output} is returned again.

3. If more time than d_{output} has elapsed since t_{output}, then $s_{output'} = \langle (x_{output'}, y_{output'}), t_{output'}, output \rangle$ is generated, stored in place of s_{output}, and returned.

The reason for examining the current time and the previous blurred sighting before deciding which output sighting to release is that it prevents an attacker from obtaining several sightings in quick succession, and then determining the instant that the target moves from a square into an adjacent square. Clearly, the target's location at that moment is on the border of the two squares.

Algorithm

Our sighting blurring algorithm is described formally in the *blurSighting* method, which is described in Algorithm 3. This method requires the use of a data structure with suitable methods, which is capable of storing sightings for a fixed duration. Note that for the sake of simplicity we have included s_{input} as an input parameter to the *blurSighting* method, even though s_{input} might not be required. A more efficient implementation would use *input* as an input parameter instead of s_{input}, and another method within *blurSighting* that is capable of obtaining s_{input} immediately before it is required.

The *store* method is used to store a sighting, which has a specified sighting accuracy of a target that has been requested by an indirect requester and proxy requester pair, in the data structure. This sighting will be stored for a fixed duration, before it is discarded by another method in a separate thread. The *store* method is described in Algorithm 4.

The *retrieve* method is used to retrieve a sighting, which has a specified sighting accuracy of a target that has been requested by an indirect requester and proxy

Algorithm 3: The *blurSighting* Method

Input: B_1 is the public name of the target.
Input: A_1 is the public name of the indirect requester.
Input: L_1 is the public name of the proxy requester.
Input: $s_{input} = \langle (x_{input}, y_{input}), t_{input}, input \rangle$ is the input sighting of the target.
Input: *output* is the desired sighting accuracy of the output sighting.
Output: s_{output} is the output sighting.
Data: Access to $\mathcal{M}$.

```
// If the output sighting accuracy is greater than the input
   sighting accuracy, then terminate the method
```
if $output > input$ **then**
    $s_{output} = \perp$
    return s_{output}
end
$s_{output} = retrieve(B_1, A_1, L_1, output)$
```
// If there is no stored blurred sighting, then create a new one
```
if $s_{output} == \perp$ **then**
    `// Round both the` x `and` y `coordinates down`
    $x_{output} = ((x_{input} \times m_{input}) - ((x_{input} \times m_{input}) \bmod m_{output}))/m_{output}$
    $y_{output} = ((y_{input} \times m_{input}) - ((y_{input} \times m_{input}) \bmod m_{output}))/m_{output}$
    $t_{output} = t_{input}$
    $s_{output} = \langle (x_{output}, y_{output}), t_{output}, output \rangle$
    $store(B_1, A_1, L_1, s_{output})$
    return s_{output}
end
```
// Otherwise return the stored blurred sighting
```
else
    return s_{output}
end

requester pair, from the data structure. If a suitable sighting does not exist then $\perp$ is returned. This method is described in Algorithm 5.

6.4.3 Input Sighting Pre-Processing

Our sighting blurring algorithm assumes that the underlying positioning technology is capable of producing raw sightings that are accurate with a high confidence level. Therefore, it is easy to produce the input sightings for our sighting blurring algorithm using these raw sightings.

Algorithm 4: The *store* Method

Input: B_1 is the public name of the target.
Input: A_1 is the public name of the indirect requester.
Input: L_1 is the public name of the proxy requester.
Input: $s = \langle (x, y), t, \alpha \rangle$ is the sighting to store.
Data: Access to $\mathcal{D}$.

// Write $\langle B_1, A_1, L_1, \alpha, s \rangle$ to thread-safe persistent storage
$write\langle B_1, A_1, L_1, \alpha, s \rangle$

// This method assumes that there is another method in a separate thread that removes sightings from the persistent storage when they expire
// A sighting expires when $currentTime() > (t + d_\alpha)$, where $currentTime()$ is a method that gets the current time from the system

return

Algorithm 5: The *retrieve* Method

Input: B_1 is the public name of the target.
Input: A_1 is the public name of the indirect requester.
Input: L_1 is the public name of the proxy requester.
Input: α is the sighting accuracy.
Output: s is the stored sighting.

// Return the sighting if it exists in the thread-safe persistent storage
if $\langle B_1, A_1, L_1, \alpha, s \rangle$ *exists* **then**
    return s
end
else
    return $\perp$
end

However, in certain circumstances we must do some pre-processing on the raw sightings that are used to produce the input sightings for our sighting blurring algorithm. In particular, the raw sightings may come from a wide variety of network operators, which might all use different underlying positioning technologies. Therefore, the raw sightings may be heterogeneous. In order for our sighting blurring algorithm to work, we require that these raw sightings be homogenised.

As an example, consider a raw sighting with a location that is represented using a circle. This circle is defined using a point for its centre, and it has a radius of

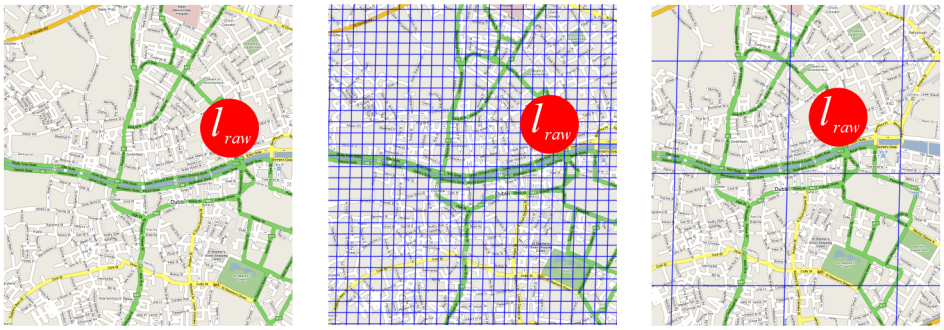

Figure 6.2: Raw Sighting Example (Maps © Google Maps)

250m. This circle is denoted as l_{raw}, and it is shown in Figure 6.2 (a). The problem with l_{raw} is that it does not fit into any single square in the example grid shown in Figure 6.2 (b), because each square in this grid has a side length of 100m. Therefore, the homogenisation process will choose a square from the grid with the smallest side length so that the raw location is fully contained within a single square. This is the example grid shown in Figure 6.2 (c) where the side length of each square is 1,000m.

This homogenisation process causes the sighting accuracy of the input sightings to decrease. In certain circumstances this decrease might be significant enough to prevent the sighting blurring algorithm from being invoked since the input sighting accuracy will be less than the desired output sighting accuracy. In this case the requesters are provided with a generic error message so that they cannot determine that the sighting blurring algorithm was not invoked due to the location component of the target's raw sighting. Therefore, this homogenisation process is not ideal, and we have identified this as an area for further investigation in Section 7.5.3.

By homogenising the raw sightings like this, we are able to guarantee that the input sightings have locations that really do contain the targets. Therefore, our sighting blurring algorithm always produces blurred sightings that are correct.

6.4.4 Design Restrictions

Our sighting blurring algorithm is designed to satisfy our requirements within the context of our architecture. As part of this design, we have made two restrictions

to its operation.

Firstly, our sighting blurring algorithm covers two-dimensional areas rather than three-dimensional spaces. Our reason for not using three-dimensional spaces is that targets rarely travel freely in the vertical direction, and their vertical locations are normally on, or very close to, the local ground level. Therefore, although blurring the vertical locations would give mathematically smaller sighting accuracies, it would not provide the targets with greater sighting privacy in reality.

Secondly, our sighting blurring algorithm is only capable of producing sightings with a limited number of different sighting accuracies. In practice we do not expect this to be a limitation if the sighting accuracies are chosen to cover a wide range.

6.5 Analysis

In this section we provide a theoretical analysis of our sighting blurring algorithm. First, we describe the scenarios in which our sighting blurring algorithm can be invoked. We then introduce an attack model, and we describe how our sighting blurring algorithm mitigates against these attacks. Finally, we compare our sighting blurring algorithm with the related research identified in Section 3.4.

6.5.1 Usage Scenarios

There are two different scenarios in which our sighting blurring algorithm can be invoked by an indirect requester and proxy requester pair for a particular target using a particular sighting accuracy. In the first scenario our sighting blurring algorithm is invoked infrequently, so that the time between invocations is greater than d_{output}. This scenario represents a series of once-off invocations, and this is likely to suit LBSs that provide an infrequent service relative to d_{output}.

Consider the first request to invoke our sighting blurring algorithm, r_1, as shown in Figure 6.3. The returned blurred sighting, s_1, states that the target was within the grid square specified by $(x_1 \times m_j, y_1 \times m_j)$ and m_{output}, at some time during the time interval defined by t_{output} and d_{output}. If s_1 was output at t_1, then t_1 will occur shortly after t_{output} and long before the time interval defined by t_{output} and

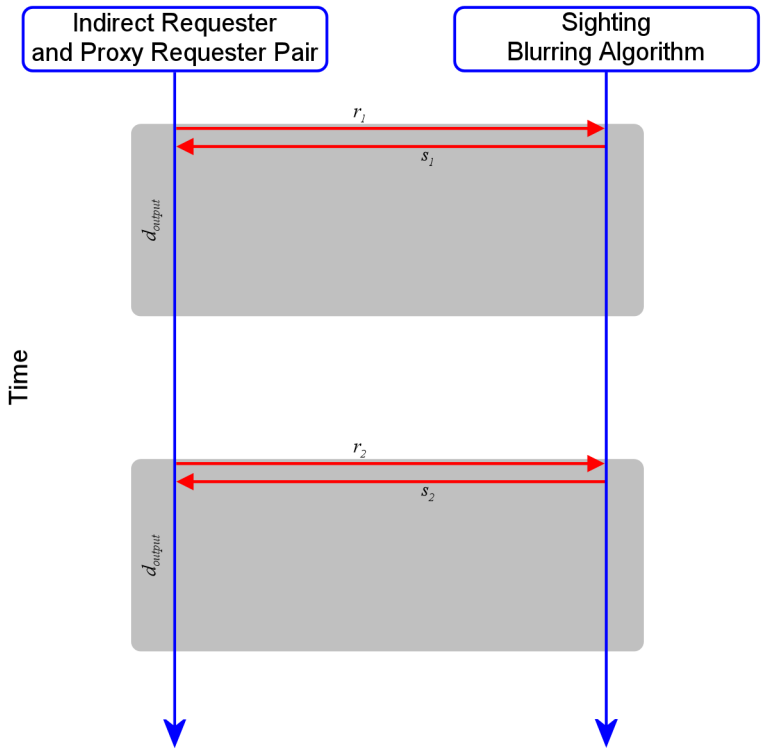

Figure 6.3: Sequence Diagram Showing an Indirect Requester and Proxy Requester Pair Invoking the Sighting Blurring Algorithm Infrequently

d_{output} passes. Therefore, the real time interval during which the target was sighted is between t_{output} and t_1, and hence s_1 is very fresh. This effectively means that no temporal blurring was used to produce s_1. The same scenario is repeated again for r_2 and s_2 in Figure 6.3.

In the second scenario the algorithm is invoked frequently, so that the time between invocations is less than d_{output}. Therefore, this scenario constitutes tracking the target.

Consider the request for invocation, r_3, as shown in Figure 6.4. The corresponding blurred sighting, s_3, is very fresh. However, when r_4 and r_5 are made within the duration d_{output} after s_3, the sighting blurring algorithm will return s_3 each time. Now the real time interval during which the target was sighted has increased, and hence s_3 is increasingly less fresh. This is true for all additional invocations within the duration d_{output} of the last blurred sighting.

The next request, r_6, is made after the duration d_{output} after s_3. Therefore, the

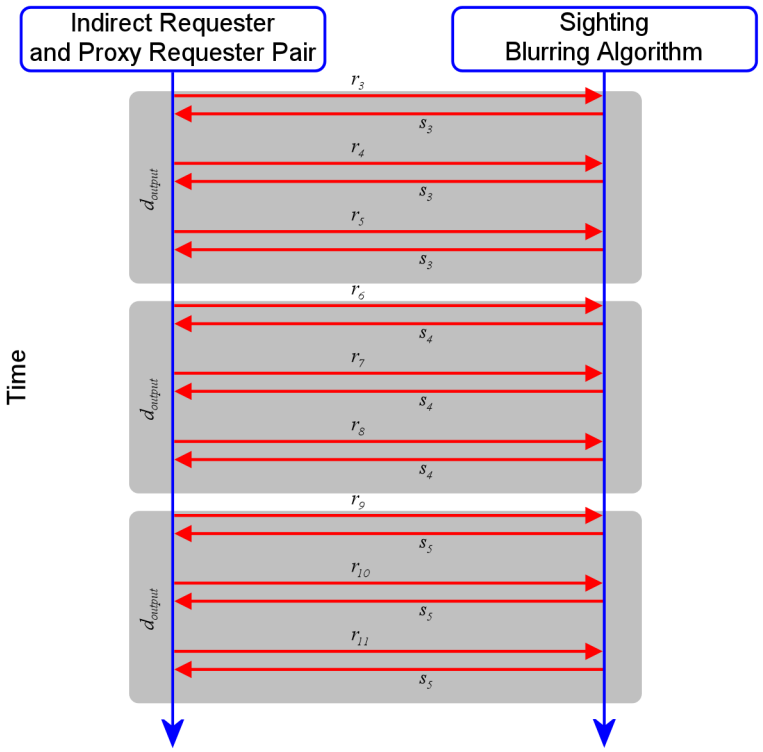

Figure 6.4: Sequence Diagram Showing an Indirect Requester and Proxy Requester Pair Invoking the Sighting Blurring Algorithm Frequently

sighting blurring algorithm will return a new sighting, s_4, which will be returned in response to all additional requests made within the duration d_{output} after s_4. This same scenario is repeated again for r_9, r_{10}, and r_{11} in Figure 6.4.

The combination of these two scenarios has the effect of making additional invocations within the duration d_{output} of the last blurred sighting meaningless, and therefore there is no incentive for the indirect requester and proxy requester pair to make these requests. In fact, there may be a disincentive if they are charged for each invocation. Therefore, our sighting blurring algorithm can be considered to be more meaningful to indirect requester and proxy requester pairs that infrequently invoke it relative to d_{output}.

We specify the frequency with which new sightings are released in terms of t_{output} and d_{output}, rather than using predetermined times and d_{output}, because this ensures that infrequent requests relative to d_{output} receive fresher sightings.

6.5.2 Attack Model

Our attack modes assumes that there will not be any external attackers, because we assume that they do not have access to the sightings due to a combination of the data security techniques described in Chapter 4 and the access control model described in Chapter 5. Therefore, we consider that an attacker is either an indirect requester or proxy requester, and hence the target is willing to be sighted by the attacker to some extent. We assume that the attacker cannot collude with other indirect requesters or proxy requesters. The attacker is able to obtain a real-time stream of blurred sightings of a particular target, and it can store these sightings. The goal of the attacker is to compute sightings that are more accurate than the sightings that it is allowed to obtain, and these computed sightings may occur in the past, the present, or the future.

Within our attack model we have identified three different types of attack that can be used by an attacker.

Mathematical Attacks

Mathematical attacks are based on the geometrical properties of the targets' sightings. Perhaps the most obvious such attack is an *intersection attack*. This occurs when the attacker is able to calculate the intersection of the location components of several blurred sightings within a short time duration. Consider a sighting blurring algorithm that blurs a target's sightings by creating a sighting consisting of a randomly located larger square that fully contains the target's location. If a different larger square is created each time the sighting blurring algorithm is invoked, then an attacker can invoke the sighting blurring algorithm several times in quick succession, and then calculate the intersection of these squares. This creates a sighting of the target that has an increased sighting accuracy. This type of intersection attack is also possible when the sighting blurring algorithm creates randomly sized blurred locations. This concept is shown in Figure 6.5, where the target's location is shown as l, and the intersection of three blurred locations is shaded.

Our sighting blurring algorithm is resistant to this type of attack, because it

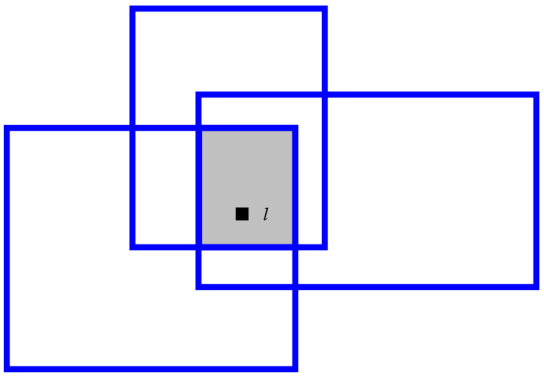

Figure 6.5: Intersection Attack on the Location l

requires that all squares within any grid are non-overlapping, and in fixed locations.

However, the use of these grids with non-overlapping and fixed squares creates the opportunity for another type of attack, which we refer to as a *border attack*. Consider a sighting blurring algorithm that returns the target's location as a square in such a grid. An attacker can invoke this sighting blurring algorithm very frequently. If the target remains within a single square within the grid, then the attacker will not be able to calculate a more accurate sighting of the target. However, the instant that the target moves from one square to another square within the grid, the attacker will know that the target is on the border of these two squares. Therefore, the sighting accuracy of the target's sighting is significantly increased.

Our sighting blurring algorithm is resistant to this type of attack, because it limits the frequency with which an attacker can obtain fresh sightings. This frequency is chosen so that on average the target could be located anywhere within the new location when the attacker obtains a sighting containing this new location.

Cartographical Attacks

Cartographical attacks use information relating to the physical features of a location to create a sighting with a greater sighting accuracy. This type of information is commonly available, and it is typically found on maps.

Consider the map shown in Figure 6.6 (a) that shows a railway running along the coastline in a rural location with a sparse road network. Figure 6.6 (b) shows

Figure 6.6: Cartographical Attack Example (Maps © Google Maps)

the same location being superimposed by a grid of squares where each square has a side length of 1,000m, and Figure 6.6 (c) shows the location components of a target's sightings. If an attacker observes a target travelling in these locations, then it is most likely that the target is travelling on a train. This is because there are no roads within many of these locations, and some of these locations contain significant amounts of water. Therefore, the attacker can calculate a significantly reduced location component for these sightings, as well as accurately predicting the target's future sightings.

Our sighting blurring algorithm is vulnerable to this type of attack, because it is not aware of any physical features within its grids. However, the target can avoid this type of attack by specifying a sighting accuracy in its permissions that corresponds to a grid with sufficiently large squares.

Sociological Attacks

Sociological attacks are based on the attacker having some knowledge of how society, or a group within society, behaves. This enables the attacker to infer how a target is likely to behave.

A common form of this type of attack, which we refer to as a *home attack*, occurs when an attacker tries to locate a target's home. There are several different heuristics that an attacker can use. For example, the target's home is likely to be in the location where the target is at 3:00:00 every morning, or the target's home is likely to be in the location where the target spends the most time. This type of attack can be undertaken manually, or automatically using reverse geocoding services and electronic *white pages* [48]. A variation of this attack can be used by the attacker to locate a target's principle place of occupation.

Our sighting blurring algorithm is capable of partially resisting this type of attack, because the attacker will be able to determine the square within the grid that contains the target's home. However, the attacker will be unable to determine the location of the target's home within this square if it contains more than one home. Therefore, it is important that the target specifies a sighting accuracy in its permissions that corresponds to a grid with sufficiently large squares relative to the urban density around its home.

Another sociological attack, which we refer to as a *return journey attack*, can occur after an attacker has determined the square within the grid that contains the target's home. This type of attack is based on the assumption that the target will return home using the same route that it used to leave its home.

Again, our sighting blurring algorithm is capable of partially resisting this type of attack, because the attacker will be able to predict the squares within the grid that contain the target's return journey. However, the attacker will be unable to determine the target's sightings with a greater sighting accuracy.

6.5.3 Comparison with Related Research

In this section we describe how our sighting blurring algorithm satisfies the requirements that we identified in Section 6.3. We also compare it with the related research described in Section 3.4 in terms of our requirements. In particular, we identify its advantages and disadvantages, and we identify the similarities and differences between it and the related research.

- **Requirement 1**

  The sighting blurring algorithm will generate sightings with exactly the requested sighting accuracy.

  Our sighting blurring algorithm satisfies this requirement because it is capable of generating blurred sightings that have the exact sighting accuracy that is requested, as long as this requested sighting accuracy has a corresponding value in the sequence of magnitudes. Furthermore, our sighting blurring algorithm quantifies sighting accuracy in terms of the area of the spatially blurred sightings and the durations for which they are valid.

  The advantage of our sighting blurring algorithm generating sightings with exactly the requested sighting accuracy is that it facilitates the creation of consistent LBSs, and this in turn will encourage the usage of these LBSs by users. The disadvantage of it satisfying this requirement is that it is unable to generate more accurate sightings in certain circumstances when the targets' security would not be decreased.

  Our sighting blurring algorithm is similar to all of the sighting blurring algorithms in the related research because they all generate blurred sightings. However, our sighting blurring algorithm differs from all of the sighting blurring algorithms in the related research because they all quantify sighting accuracies in terms of numbers of other users, probabilities, or dynamically varying areas and durations.

- **Requirement 2**

The sighting blurring algorithm will generate sightings consisting of locations and time intervals that contain the targets.

Our sighting blurring algorithm always produces sightings consisting of locations and time intervals that are guaranteed to contain the target, although the freshness of these sightings decreases if it is invoked frequently relative to the durations of the sightings.

The advantage of our sighting blurring algorithm satisfying this requirement is that the LBSs always provide services that are perceived as being accurate and correct by the users, and this in turn will encourage the usage of these LBSs by the users.

Our sighting blurring algorithm differs from the probability based blurring algorithms in the related research because they do not satisfy this requirement. Furthermore, it differs from some of the spatial and temporal blurring algorithms that do not support this requirement, because they will not generate sightings of a target if it is in certain locations.

- **Requirement 3**

  The only sightings required as input by the sighting blurring algorithm will be the current sightings of the targets.

  Our sighting blurring algorithm satisfies this requirement because it is only dependent on the current sighting of the target, and it is independent of the sightings of any other users.

  The advantage of our sighting blurring algorithm satisfying this requirement is that it cannot be influenced by factors such as the proximity of other users to the target. This is critical because this sighting knowledge may not be available due to limitations of the underlying positioning technology, the privacy preferences of the users who are not involved in the current invocation of the LBS, or the charges incurred for obtaining these sightings. The disadvantage of our sighting blurring algorithm satisfying this requirement is that it is unable to generate more accurate sightings in certain circumstances when the

targets' security would not be decreased.

Our sighting blurring algorithm is similar to the probability based blurring algorithms and the political location blurring algorithms with regard to this requirement. However, it differs from the anonymity based blurring algorithms and the spatial and temporal blurring algorithms with regard to this requirement, because they all require knowledge of current sightings of the target and other users. In particular, the spatial and temporal blurring algorithms that are based on k-anonymity require knowledge of current sightings of the target and every other user, in order to determine which users are the other $k-1$ users.

Indeed, an attacker could exploit this property if it is close to the spatially blurred location of a stationary target. Consider an attacker that receives a spatially blurred location of a target that has a k-anonymity of 3, as shown in Figure 6.7 (a). If the attacker enters the spatially blurred location and invokes the sighting blurring algorithm it will receive a smaller spatially blurred location because it has now replaced one of the k users. The attacker can keep repeating this until moving closer would exclude a second user from the k users. This is shown in Figure 6.7 (b). The attacker can then repeat the process travelling along the other axis, as shown in Figure 6.7 (c). The attacker can then use the intersection of these two smaller spatially blurred locations to generate an even smaller spatially blurred location, as shown by the shaded location in Figure 6.7 (c). Indeed, in this simple example the attacker can accurately calculate the target's location within this blurred location because the target has a k-anonymity of 3.

- **Requirement 4**

  The sighting blurring algorithm will blur sightings so that the target could be located anywhere within the location at any instant within the time interval with equal probability.

  Our sighting blurring algorithm is designed to reduce the frequency with which new sightings are released so that the target could have reached any location

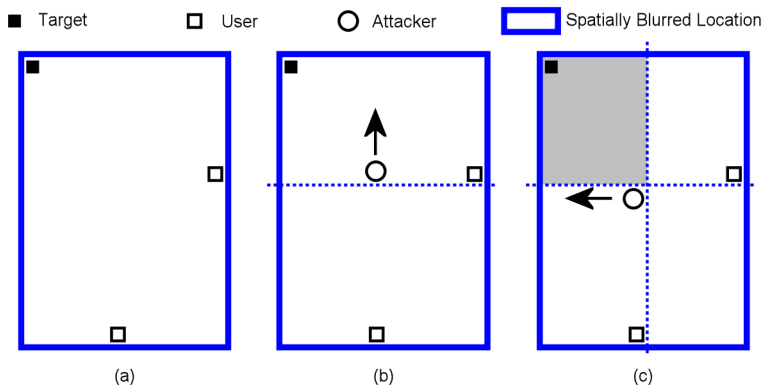

Figure 6.7: K-Anonymity Attack Example

within an adjacent location between sighting releases. The duration required to ensure this is based on the average speed of an average user. This differs from the frequency based blurring algorithm developed by Candebat, which uses the current instantaneous speed of the target. Our sighting blurring algorithm also differs from many of the other sighting blurring algorithms described in the related research with regard to this requirement, because they do not use time intervals.

The advantage of our sighting blurring algorithm satisfying this requirement is that it is resistant to mathematical attacks.

- **Requirement 5**

  The sighting blurring algorithm must be capable of blurring sightings that are part of a stream of sightings of a particular target, and knowledge of previous blurred sightings should not affect the sighting accuracy of any future sightings.

  Our sighting blurring algorithm is capable of being invoked frequently to blur sightings that are part of a stream of sightings for a particular target.

  The advantage of our sighting blurring algorithm satisfying this requirement is that it prevents an attacker from mounting a mathematical attack by invoking the sighting blurring algorithm frequently in order to calculate a new sighting with a greater sighting accuracy.

Our sighting blurring algorithm differs from the sighting blurring algorithms in the related research with regard to this requirement, because they are all vulnerable to this type of attack.

- **Requirement 6**

  The sighting blurring algorithm must be capable of generating blurred sightings with a consistent sighting accuracy.

  Our sighting blurring algorithm generates sightings with a consistent sighting accuracy, which is measured in terms of the area of the spatially blurred sightings and the durations for which they are valid.

  The advantage of our sighting blurring algorithm satisfying this requirement is that it facilitates the creation of consistent LBSs, and this in turn will encourage the usage of these LBSs by users. The disadvantage of it satisfying this requirement is that it is unable to generate more accurate sightings in certain circumstances when the targets' security would not be decreased.

  Our sighting blurring algorithm is similar to the probability based blurring algorithms with regard to this requirement. However, it differs from the anonymity based blurring algorithms, the political location blurring algorithms, and the spatial and temporal blurring algorithms with regard to this requirement, because they all generate sightings with variable sighting accuracies that are measured in terms of the area of the spatially blurred sightings.

Finally, our sighting blurring algorithm is similar to the anonymity based blurring algorithms with regard to supporting anonymous users. However, it differs from the anonymity based blurring algorithms because it does not require anonymity. Therefore, it is similar to many of the other sighting blurring algorithms with regard to supporting identified users. Although this was not an original requirement for our sighting blurring algorithm, it is an advantage that both anonymous users and identified users can be supported because it provides users with greater choice, and it facilitates LBSs that require some form of persistent identification.

6.6 Evaluation

In this section we describe the practical evaluation of our sighting blurring algorithm. First, we describe how we collected our sample sightings, and we provide an analysis of these sample sightings. We then describe the testbed that we used to implement and evaluate our sighting blurring algorithm. Finally, we evaluate our sighting blurring algorithm using the sample sightings and the testbed, and we present our results.

6.6.1 Sample Sightings

Ideally, we would have liked to evaluate our sighting blurring algorithm using real sightings of real users in real-time. However, this was not realistic due to the costs involved of reporting sightings in real-time, the inconvenience and complexity caused to users by requiring them to carry bulkier and more complex locatables, and the uncertainty and unpredictability of the input sightings.

Instead, we asked ten family members and friends to provide us with their sightings, and we supplied them with locatables based on GPS. A typical session during which the participants used the locatables lasted for approximately two or three days. They were shown how to switch their locatables on and off, and it was suggested to them that they switch off their locatables when they were in the same location for more than a few minutes.

After we collected each session, we displayed it using *Google Maps* [35], and we showed it to the participant. We asked him/her to describe the various journeys that occurred during the session. In particular, we enquired about the mode of transport used for each journey, and the answers we received were: on foot, bicycle, car, tram, city bus, and intercity bus.

These sessions provided us with more than 280,000 unique sightings, and approximately 100 journeys. We estimate by visual inspection that the location in the recorded sighting was always within 100m of the real location, and that it was often within 10m of the real location.

We found that the overall average journey distance was 27,108m with a stan-

dard deviation of 37,137m, the overall average journey duration was 3,972s with a standard deviation of 3,349s, and the overall average journey speed was 6.22m/s with a standard deviation of 3.93m/s. The fact that these standard deviations are relatively large is not surprising due to the wide range of modes of transport, and hence a wide range of journey types.

In order to provide a more meaningful overview of the journeys for which we have sightings, we re-evaluated them based on the mode of transport used. The average journey distance and standard deviation for each mode of transport is shown in Figure 6.8. The standard deviations for journeys where the mode of transport was on foot, tram, or city bus are all relatively small since these modes of transport are normally used for specific types of journeys. The standard deviations for journeys where the mode of transport was either bicycle or car are relatively large. In the case of journeys made by bicycle this is because some of the journeys were short commutes, while other journeys were part of training sessions for cycling races. In the case of journeys made by car this is because cars are equally suited to both short and long journeys. The standard deviation for journeys where the mode of transport was intercity bus is almost zero because all of these journeys were made between a single pair of locations.

The average journey duration and standard deviation for each mode of transport is shown in Figure 6.9. The standard deviations for journeys where the mode of transport was either bicycle or car are relatively large, and this is expected based on the standard deviations shown in Figure 6.8. The standard deviation for journeys where the mode of transport was on foot is also relatively large. This is probably because travelling on foot is relatively slow compared to the other modes of transport, and therefore a variation in the average journey duration does not cause a significant variation in the average journey distance. The standard deviations for journeys where the mode of transport was tram, city bus, or intercity bus are all relatively small since these modes of transport are normally used for specific types of journeys.

Finally, the average journey speed and standard deviation for each mode of transport is shown in Figure 6.10. The standard deviation for journeys where the

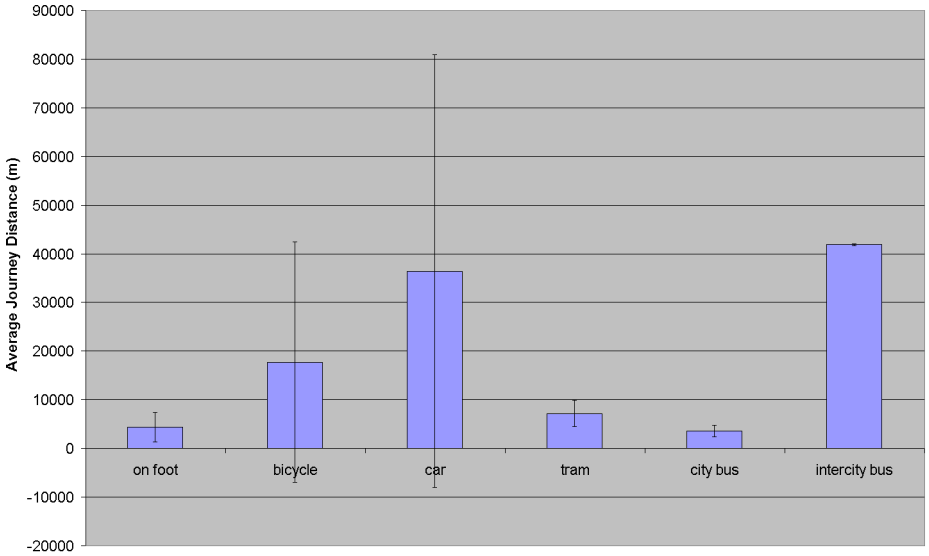

Figure 6.8: The Average Journey Distance and Standard Deviation for each Mode of Transport

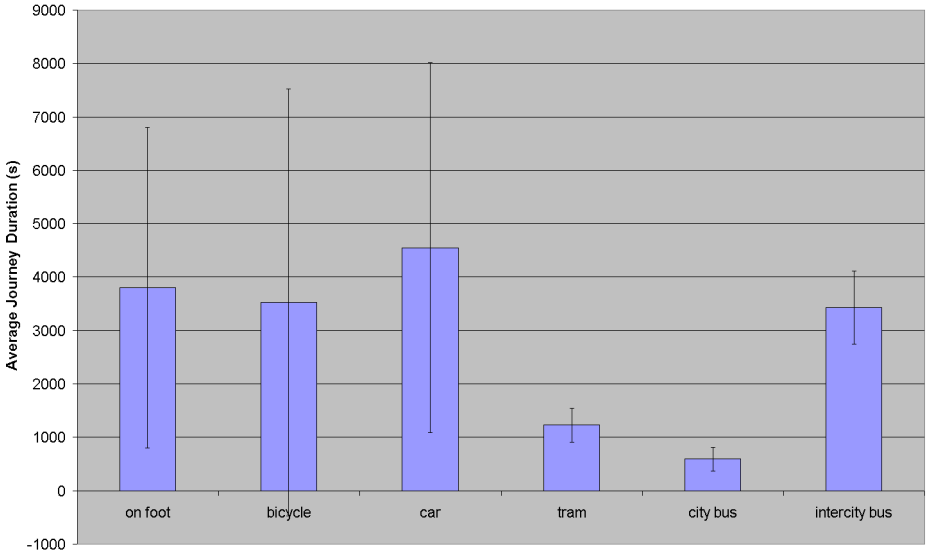

Figure 6.9: The Average Journey Duration and Standard Deviation for each Mode of Transport

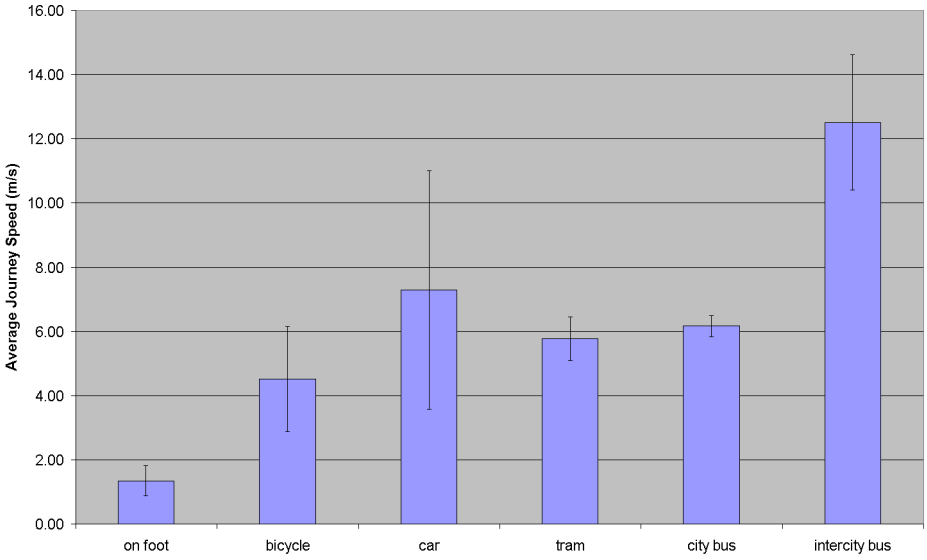

Figure 6.10: The Average Journey Speed and Standard Deviation for each Mode of Transport

mode of transport was car is the largest. This is due to the fact that cars are used in both urban environments that suffer from heavy traffic congestion, and extra-urban environments where traffic can flow freely without being impeded. The standard deviation for journeys where the mode of transport was bicycle is also relatively large. Again, this is due to the fact that bicycles were used for both slow commutes and fast training sessions. The relatively small standard deviations for journeys where the mode of transport was on foot, tram, city bus, or intercity bus suggest that these modes of transport are somewhat immune to variations in their environments.

6.6.2 Testbed

We have developed a prototype implementation, which we use as a testbed, in order to evaluate our sighting blurring algorithm with respect to our sample sightings.

Sighting Server

We have developed a *sighting server* that simulates a network operator, which can be queried using a location protocol (see Section 2.3.3) in order to obtain a sighting

of a target. Our sighting server uses the sample sightings, and it is capable of interpolating these sightings for the time intervals when the locatable was switched off.

Each sighting request to the sighting server contains a time parameter that the sighting server uses instead of the current time. This enables us to replay our sample sightings in real-time.

Magnitudes and Grids

Our testbed uses the following sequence of magnitudes, where each magnitude is specified in metres:

$$\mathcal{M} = \langle 100000, 10000, 1000, 100 \rangle$$

Therefore, we are using four different grids within our testbed.

Locations

The locatables used by the participants all recorded locations using the *WGS 84* format (see Section 2.3.2). This format is not optimum for performing distance based calculations within relatively small areas, so instead we decided to use the *Irish Grid* (see Section 2.3.2) to represent locations. This format is particularly suitable for our evaluation because it is a coordinate reference system that uses the metre as its unit of length. Therefore, we used a mathematical-to-mathematical location translation to convert all of the recorded locations from the *WGS 84* format to the *Irish Grid* format.

Durations

Calculating appropriate durations is critical for the successful operation of our sighting blurring algorithm. We have already stated that each duration is calculated by dividing the average distance required to move to any location in an adjacent square by the average journey speed of a typical target.

Calculating the average distance required to move to any location in an adjacent square requires us to calculate the maximum distance required to move to any location in an adjacent square. We assume that the squares in the grid have a side length of m_j, and that the average location in the initial square is its centre. There are four adjacent squares where the minimum distance required to move beyond them is $1.5m_j$, and there are four diagonally adjacent squares where the maximum distance required to move beyond them is $1.5\sqrt{2}m_j$. Therefore, we calculate the average distance required to move to any location in an adjacent square as follows:

$$\begin{aligned} \textit{Average Distance} &= \frac{4(1.5m_j) + 4(1.5\sqrt{2}m_j)}{8} \\ &= \frac{6m_j + 6\sqrt{2}m_j}{8} \\ &= \frac{6 + 6\sqrt{2}}{8} m_j \\ &\approxeq 1.8m_j \end{aligned}$$

Calculating the average journey speed of a typical target is more difficult. If the average journey speed is too slow, then the duration will be too large and the sighting blurring algorithm will unnecessarily reduce the frequency with which it produces blurred sightings. Conversely, if the average journey speed is too fast, then the duration will be too small and the sighting blurring algorithm will not provide the target with sufficient privacy.

We could use the overall average journey speed of 6.22m/s that we obtained from our sample sightings. However, this is likely to give poor performance due to its relatively large standard deviation.

We could use the target's current journey speed, or the target's average journey speed based on its mode of transport. However, our sighting blurring algorithm is not aware of either the target's current journey speed, or the target's mode of transport.

Instead, we re-evaluated our sightings based on the average journey distance. We calculated the average journey speed and standard deviation for each range of

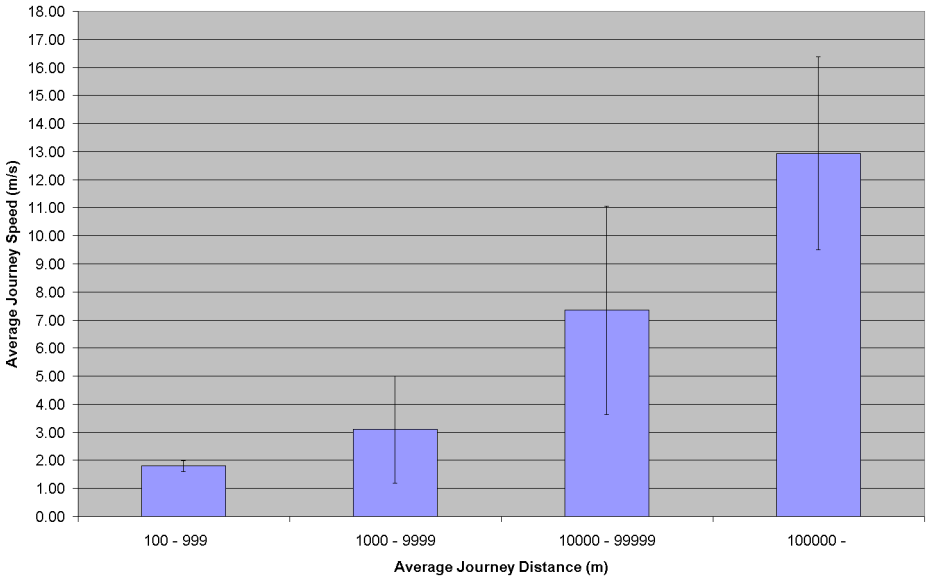

Figure 6.11: The Average Journey Speed and Standard Deviation for each Range Defined by our Magnitudes

average journey distances defined by our magnitudes, and the results are shown in Figure 6.11.

We then calculated each duration by dividing $1.8m_j$ by these average journey speeds. Therefore our testbed uses the following sequence of durations, where each duration is specified in seconds:

$$\mathcal{D} = \langle 13916, 2450, 581, 100 \rangle$$

These durations are approximately 4 hours, 40 minutes, 10 minutes, and 2 minutes. The maximum frequencies with which new sightings are released based on these durations are 6 times per day, 35 times per day, 148 times per day, and 862 times per day.

Algorithm

We have implemented our sighting blurring algorithm as described in Algorithm 3. For the purpose of evaluation, we added an additional input parameter, $t_{simulated}$, which represents a time to use instead of the current time. This enabled us to replay

our sample sightings in real-time.

The input sighting of the target is $s_{input} = \langle (x_{input}, y_{input}), t_{input}, 4 \rangle$, and its sighting accuracy is calculated as $f_2(s_{input}) = 4$. The desired sighting accuracy of the output sighting is *output*, where $output \in \{1, 2, 3, 4\}$.

6.6.3 Example Attacks

We evaluated our sighting blurring algorithm by emulating an attacker that frequently requests a target's sightings with a constant sighting accuracy over a 24 hour period starting at 00:00:00 and ending at 23:59:59. Since our sighting blurring algorithm is resistant to mathematical attacks, as described in Section 6.5.2, we focused on testing it against cartographical attacks and sociological attacks.

Example 1

This example is based on a target that lives in the centre of Dublin. The target walked to DCU in the morning, and returned to the city centre in the evening using a city bus. The route walked in the morning was different to the route used by the city bus in the evening. The target's raw locations are shown in Figure 6.12 (a).

Figure 6.12 (b) shows the location components of the target's blurred sightings using a sighting accuracy of 4, and Figure 6.12 (c) shows the location components of the target's blurred sightings using a sighting accuracy of 3. For both of these sighting accuracies, the mean distance between successive distinct blurred locations is approximately $1.4m_{output}$ and the standard deviation is approximately $0.8m_{output}$.

The blurred location of the target's home and place of work can be correctly identified in both of these cases using sociological attacks (the blurred location containing the target at 03:00:00 represented approximately 52% of the target's blurred locations, and the blurred location containing the target at 15:00:00 represented approximately 45% of the target's blurred locations).

The sighting accuracy of 4 does not offer the target significant protection from cartographical attacks because there is generally only one road within any of the blurred locations. Furthermore, the blurred location that represents the target's place of work is fully contained within the university campus. An attacker can also

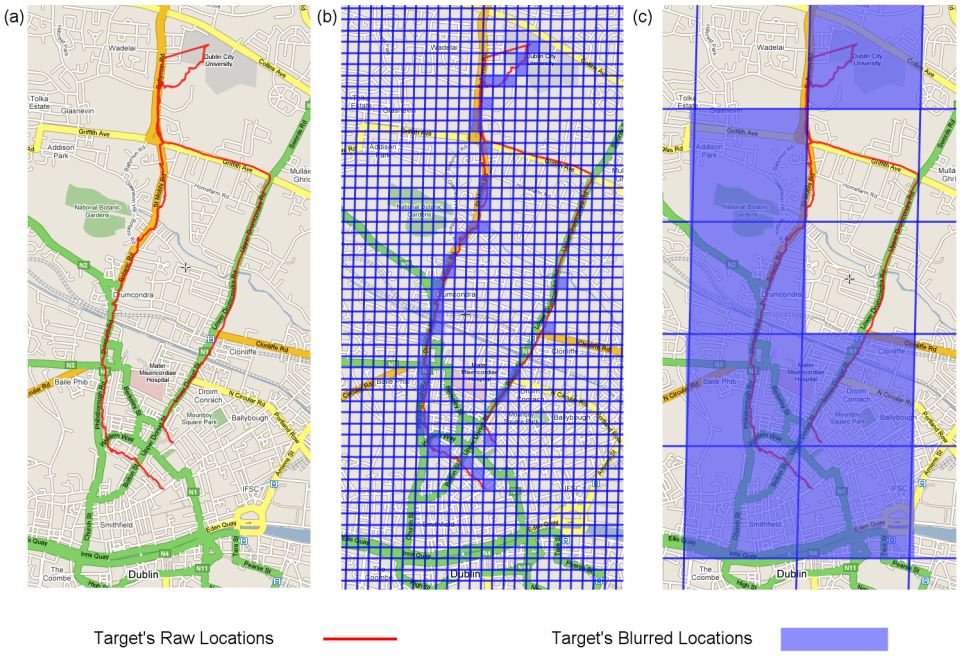

Figure 6.12: Sighting Blurring Algorithm Example 1 (Maps © Google Maps)

observe that the target's speed was significantly faster in the evening compared to the morning.

The risk posed by cartographical attacks in this example is significantly reduced by using a sighting accuracy of 3 because there are more viable roads and buildings that could contain the target within any blurred location. In both cases, the target's home cannot be identified within the blurred location because it contains an area of high-density housing.

Example 2

The example shown in Figure 6.13 (a) is based on a target that is a researcher in DCU, and the target lives in the student residences that are on the university campus. The target drove to UCD for a collaboration meeting in the morning, and drove home in the evening. The route used on both occasions had small variations in the city centre due to the use of one-way streets. Figure 6.13 (b) shows the location components of the target's blurred sightings using a sighting accuracy of 3. In this example, the mean distance between successive distinct blurred locations is

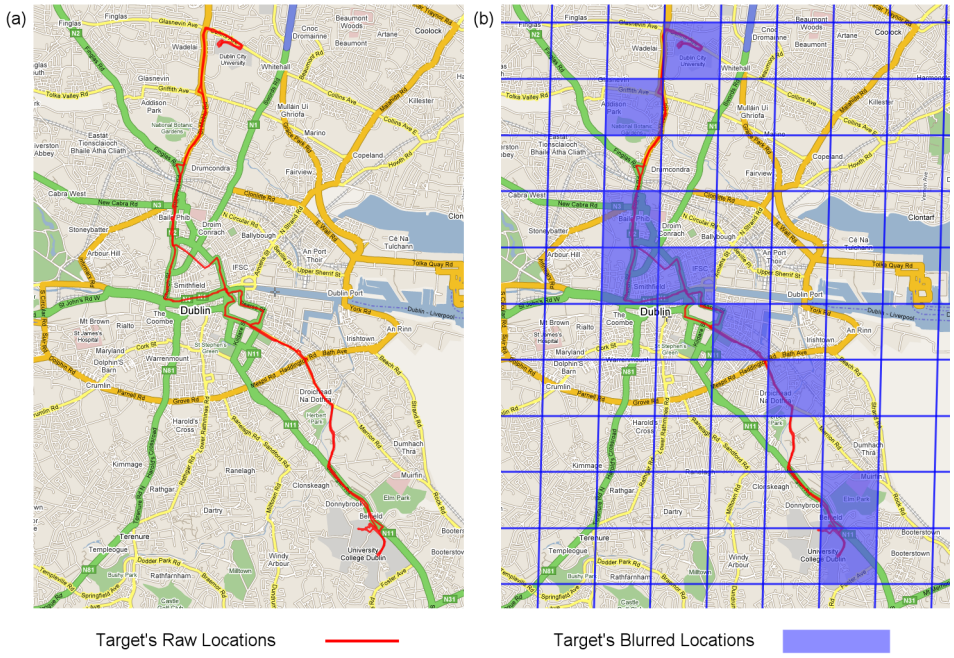

Figure 6.13: Sighting Blurring Algorithm Example 2 (Maps © Google Maps)

approximately $2.4m_{output}$ and the standard deviation is approximately $1.1m_{output}$.

The blurred location of the target's home can be correctly identified using sociological attacks (the blurred location containing the target at 03:00:00 represented approximately 50% of the target's blurred locations). However, it is not possible to identify the target's home as the student residences using a cartographical attack, because there are private residences within the same blurred location. The blurred location where the target was located during working hours can be correctly identified using sociological attacks (the blurred location containing the target at 15:00:00 represented approximately 45% of the target's blurred locations). However, this is not the target's main place of work, and this fact can be observed over a period of five working days. Indeed, on any of the other working days in this week the blurred location containing the target's home represented approximately 96% of the target's blurred locations.

Example 3

The example shown in Figure 6.14 (a) is based on a target that lives in a rural environment that is approximately 50,000m southwest of Dublin. The target drove to the city centre, collected some passengers, and then drove them to the airport. The target then drove home.

Figure 6.14 (b) shows the location components of the target's blurred sightings using a sighting accuracy of 3. Since the target was travelling mostly on motorways, and the sightings were recorded on a Saturday, the target was travelling significantly faster than average. Therefore, the mean distance between successive distinct blurred locations is approximately $5.8m_{output}$ and the standard deviation is approximately $3.5m_{output}$. However, it would be relatively easy to use a cartographical attack on any blurred location that is outside of the city, since many of these blurred locations contain only a single road.

Figure 6.14 (c) shows the location components of the target's blurred sightings using a sighting accuracy of 2. In this case, the mean distance between successive distinct blurred locations is approximately $1.9m_{output}$ and the standard deviation is approximately $1.1m_{output}$. The risk posed by cartographical attacks in this example is significantly reduced because there are more viable roads that could contain the target within any blurred location.

In both of these cases, the blurred location of the target's home can be correctly identified using sociological attacks (the blurred location containing the target at 03:00:00 represented approximately 75% of the target's blurred locations).

Cartographical attacks can be used with the blurred location of the target's home that has a sighting accuracy of 3, because there are less than 10 homes within this blurred location. However, the target's home cannot be located within the blurred location that has a sighting accuracy of 2, because there are several towns and many rural homes within this blurred location.

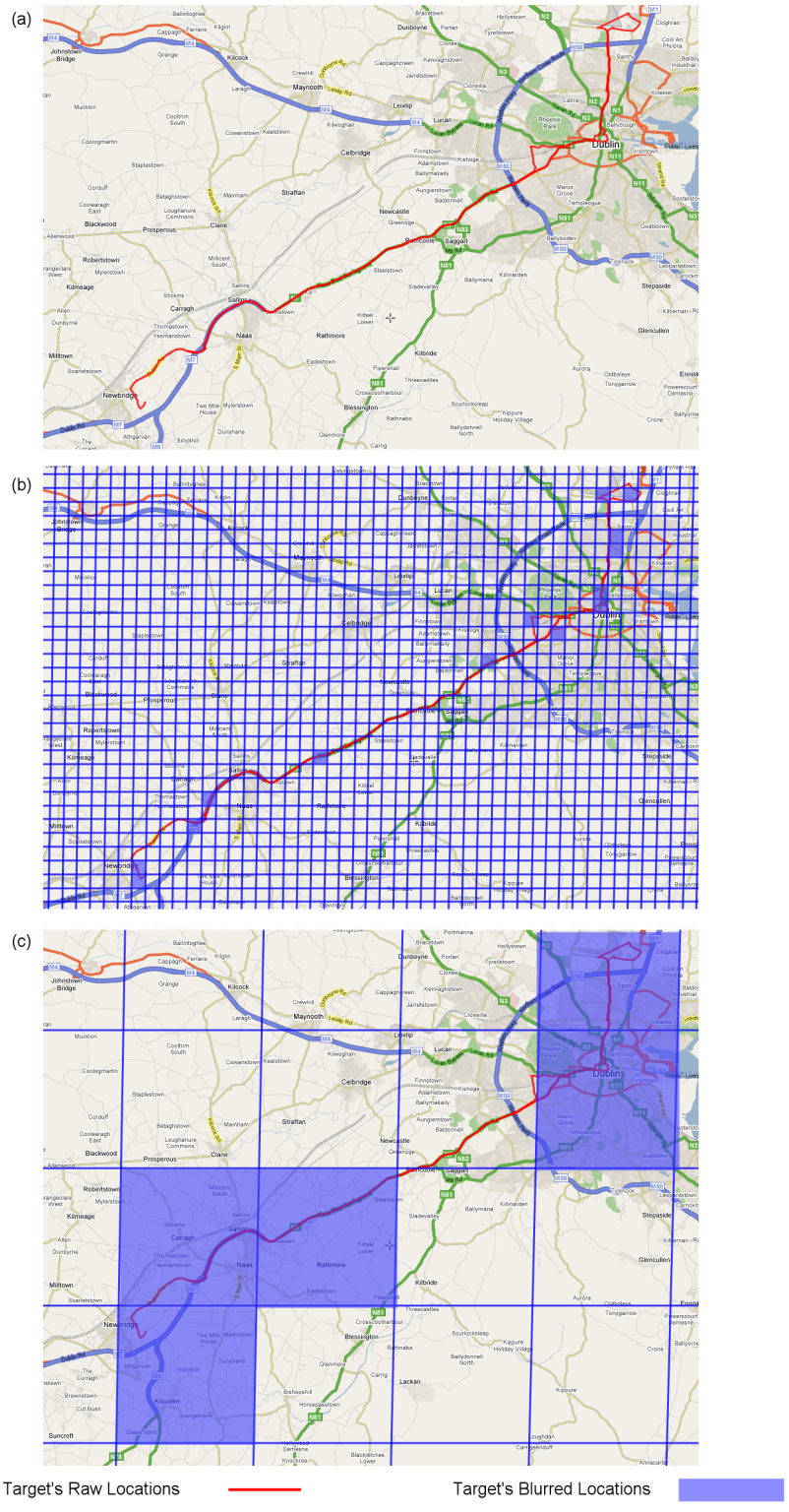

Figure 6.14: Sighting Blurring Algorithm Example 3 (Maps © Google Maps)

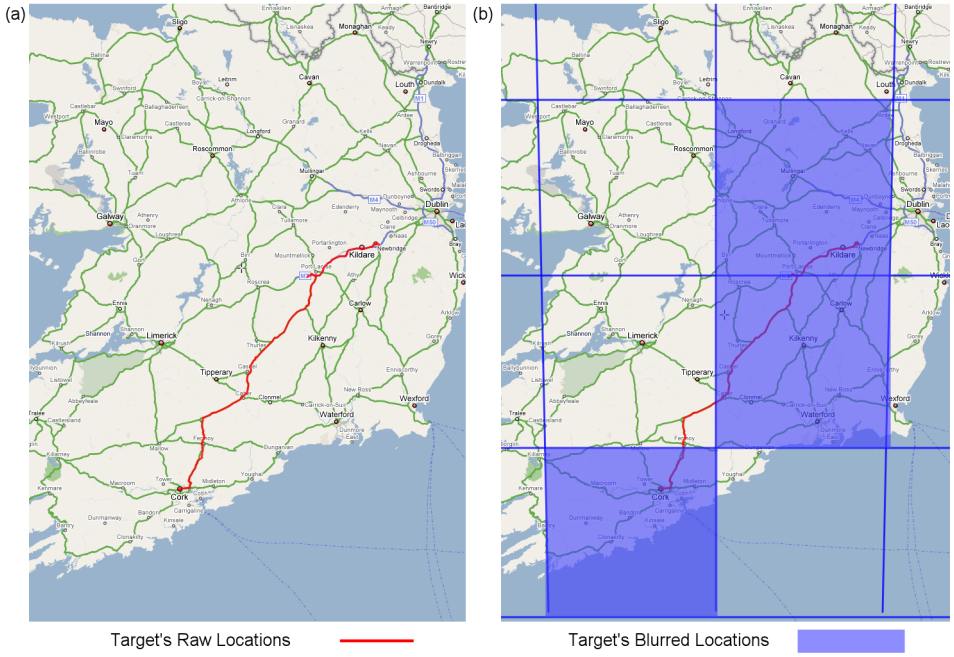

Figure 6.15: Sighting Blurring Algorithm Example 4 (Maps © Google Maps)

Example 4

This example is based on the target from the previous example driving to Cork city centre, and staying there for a weekend. The target's raw locations are shown in Figure 6.15 (a).

Figure 6.15 (b) shows the location components of the target's blurred sightings using a sighting accuracy of 1. In this example, the mean distance between successive distinct blurred locations is approximately $1.2m_{output}$ and the standard deviation is approximately $0.3m_{output}$. The risk posed by cartographical attacks in this example is very low because the blurred locations are very large.

The blurred location of the target's home can be correctly identified using sociological attacks (the blurred location containing the target at 03:00:00 represented approximately 43% of the target's blurred locations). This sociological attack identifies the blurred location containing the city as the location of the target's home on the following days. However, the blurred location containing the target's home can be distinguished from the blurred location that the target visited by observing the target's blurred locations over a period of a week.

6.6.4 Results

Our sighting blurring algorithm satisfies the requirements described in Section 6.3 due to its design, and therefore it is resistant to mathematical attacks. It provides requesters who infrequently invoke it relative to d_{output} with fresh sightings containing a blurred location component, as described in Section 6.5.1. Conversely, it prevents requesters obtaining sightings of a target with a greater sighting accuracy by frequently invoking it relative to d_{output}.

The sample sightings used in our example attacks are representative of our sample sightings, and the behaviour of our sighting blurring algorithm in the example attacks is representative of its behaviour in attacks against any of our sample sightings. Furthermore, the example attacks show that our sighting blurring algorithm is resistant to cartographical attacks and sociological attacks if grids with sufficiently large locations are used.

The durations used by our sighting blurring algorithm with our sample sightings are appropriate for the specified magnitudes, because the mean distance between successive distinct blurred locations is generally less than $2m_{output}$ and the standard deviation is generally less than m_{output}.

6.7 Conclusions

In this chapter we described the sighting blurring algorithm that we have developed, which is implemented within the infrastructure in our architecture. Our sighting blurring algorithm offers targets increased privacy by performing spatial blurring on the location components of their sightings. Instead of performing temporal blurring, our sighting blurring algorithm will not produce any new blurred sightings of the target until a specific duration has elapsed. This has the effect of limiting the frequency with which new sightings can be obtained for a target, and this frequency is a function of the area of the location component of the sighting that was produced by the spatial blurring.

There are three advantages of our sighting blurring algorithm compared to the sighting blurring algorithms presented in the related research. Firstly, our sighting

blurring algorithm is designed to be resistant to mathematical attacks based on frequent sighting requests. Secondly, our sighting blurring algorithm generates blurred sightings with a consistent sighting accuracy. Thirdly, our sighting blurring algorithm is only dependent on the sightings of the targets that are directly involved in a current LBS invocation, and it is independent of the sightings of all other users.

In conclusion, our sighting blurring algorithm can be used as part of an overall solution for providing users with increased privacy while using LBSs.

Chapter 7

Conclusions

7.1 Introduction

In this chapter we form our conclusions to this thesis. In particular, we identify both the major contributions and the minor contributions of this thesis in terms of our architecture and protocol, our access control model, and our sighting blurring algorithm. We also identify several areas of the work described in this thesis that require further investigation. Finally, we identify the publications that have arisen as a result of the work described in this thesis.

7.2 Thesis Summary

This thesis can be summarised as follows:

- In Chapter 1 we provided an introduction to mobile devices, positioning technologies, and LBSs. We then described the problem statement that this thesis addresses, together with the main goals of this thesis. Finally, we outlined the unique contributions of this thesis.
- In Chapter 2 we described the background concepts that are relevant to this thesis. In particular, we described the cryptographic concepts and the charging concepts that we built upon in Chapter 4, and we described the location concepts that we built upon in both Chapter 4 and Chapter 6.

- In Chapter 3 we described research efforts that are related to LBSs in terms of architectures and protocols, access control models, and sighting blurring algorithms.

- In Chapter 4 we described an architecture for users, an infrastructure, and LBSs, which facilitates the operation of these LBSs over the Internet. We also described a protocol that enables these three entities to achieve three-party mutual identification and authentication. In particular, this protocol allows users to simultaneously identify and authenticate themselves to the infrastructure using one identity, and to the LBS using another identity. This protocol then guarantees the confidentiality, integrity, and non-repudiation of all subsequent messages.

- In Chapter 5 we described an access control model that is implemented within the infrastructure. This access control model is based on users who are targets creating permissions for users who are indirect requesters, and for LBSs that are proxy requesters. These permissions specify which requesters are entitled to obtain sightings of which targets, under what circumstances these sightings are released, and the maximum accuracy of these sightings.

- In Chapter 6 we described the sighting blurring algorithm that we have developed, which is implemented within the infrastructure in our architecture. Our sighting blurring algorithm offers users increased privacy by decreasing the accuracy of their sightings based on the sighting accuracy allowed by the access control model described in Chapter 5.

7.3 Major Contributions

In this section we describe the three major contributions of our work that are described in this thesis. These contributions are a refinement of the contributions described in Section 1.5.

7.3.1 Architecture and Protocol

We have developed an architecture for users, LBSs, and an infrastructure, which is a trusted middleware entity that facilitates the operation of these LBSs over the Internet in a secure manner. In particular, the infrastructure provides common functionality regarding the users' identity information and sighting information.

We have also developed a protocol that is based on both the X.509 PKI and mediated identity based cryptography, and which uses this architecture, so that users, LBSs, and an infrastructure can achieve three-party mutual identification and authentication. This protocol allows users to simultaneously identify and authenticate themselves to the infrastructure using an account name, and to the LBS using a public name. Indeed, each user can be identified and authenticated using a different public name with each LBS. This is achieved without requiring the mobile device to have significantly greater resources, and without the need for additional messages to be exchanged. This usage of public names with LBSs provides users with increased privacy without necessarily reducing the usefulness of the LBSs. The confidentiality, integrity, and non-repudiation of all subsequent messages can be guaranteed, and we described how this can be used so that a user receives sighting information from an LBS, which in turn receives this sighting information from the infrastructure.

7.3.2 Access Control Model

We have developed an access control model that is implemented within the infrastructure within our architecture. Our access control model is based on users who are targets creating permissions for users who are indirect requesters, and for LBSs that are proxy requesters. There are two types of permission within our access control model - IAPs that are used by indirect requesters, and PAPs that are used by proxy requesters. These permissions specify which requesters are entitled to obtain sightings of which targets, under what circumstances these sightings are released, and the maximum accuracy of these sightings. Our access control model is then responsible for releasing users' sightings in accordance with their permissions. When an indirect requester and a proxy requester pair request a sighting of a target, they

must supply the infrastructure with an IAP and a PAP.

The main novelty of our access control model is that it enables users to specify two different types of permission - IAPs and PAPs. This has the effect of creating a whitelist for users, and a separate whitelist for LBSs. The access control model will only allow sightings to be released if it is presented with both a valid permission specified in terms of users, and a valid permission specified in terms of LBSs.

7.3.3 Sighting Blurring Algorithm

We have developed a sighting blurring algorithm that offers targets increased privacy by performing spatial blurring on the location components of their sightings. Instead of performing temporal blurring, our sighting blurring algorithm will not produce any new blurred sightings of the target until a specific duration has elapsed. This has the effect of limiting the frequency with which new sightings can be obtained for a target, and this frequency is a function of the area of the location component of the sighting that was produced by the spatial blurring.

There are three advantages of our sighting blurring algorithm compared to the sighting blurring algorithms presented in the related research. Firstly, our sighting blurring algorithm is designed to be resistant to mathematical attacks based on frequent sighting requests. Secondly, our sighting blurring algorithm generates blurred sightings with a consistent sighting accuracy. Thirdly, our sighting blurring algorithm is only dependent on the sightings of the targets that are directly involved in a current LBS invocation, and it is independent of the sightings of all other users.

7.4 Minor Contributions

There are several minor contributions of our work that are described in this thesis:

- We have developed a real-time charging service that uses our protocol, and this charging service is based on a revenue sharing model. Furthermore, our charging service enables users to offer payments to LBSs using vouchers. This is in contrast to LBSs taking payments from users. Therefore, our charging

service gives users complete control over how much they pay LBSs, which in turn reduces the amount of trust that these users must have in LBSs.

- We have collected more than 280,000 unique sightings from real users doing their typical activities. We classified these sightings based on the mode of transport used, and for each mode of transport we calculated the average journey distance, average journey duration, and average journey speed. Although previous studies have collected sightings from real users, this is the first study that we are aware of which collected sightings from users in Ireland.

- We used the sightings that we collected to calculate duration values corresponding to the magnitudes that we used in our sighting blurring algorithm testbed.

7.5 Future Work

Although this work in its current form is complete, we have identified several areas related to our architecture and protocol, our access control model, and our sighting blurring algorithm that require further investigation.

7.5.1 Architecture and Protocol

The areas related to our architecture and protocol that require further investigation are:

- Our architecture and protocol are capable of supporting very flexible and novel commercial relationships between the users, the infrastructure, and the LBSs. We would like to identify these capabilities, and build upon them. Another aspect of the charging that we would like to focus on is the reduction of the trust needed by a user in a particular LBS.

- We would like to undertake more work on the implementation of our protocol. In particular, we would like to use real mobile devices, and we would like to do this in a way that requires the least amount of changes to their existing software.

7.5.2 Access Control Model

The areas related to our access control model that that require further investigation are:

- Our access control model enables users to create many permissions that support very complex and personalised security specifications. Therefore, it is possible that there will be many different combinations of IAPs and PAPs that an indirect requester and proxy requester pair can use in order to obtain a target's sightings. We would like to investigate methods that allow the indirect requesters and proxy requesters to select the most appropriate combinations of IAPs and PAPs according to some criteria.

- In the distributed implementation of our access control model, both the indirect requesters and the proxy requesters are responsible for managing the permissions that were created by the targets that they might need to sight. We would like to investigate both methods that enable users to publish their permissions, and methods that enable requesters to retrieve these permissions.

7.5.3 Sighting Blurring Algorithm

The areas related to our sighting blurring algorithm that require further investigation are:

- We would like to continue testing and evaluating our sighting blurring algorithm using new sample sightings from a wider group of participants. In particular, we would like to assess the correctness of the values that we are using as magnitudes and durations.

- We would like to develop an improved homogenisation process to pre-process raw sightings that contain circular location components. In particular, this improved homogenisation process should create input sightings with a consistent sighting accuracy, and this sighting accuracy should not vary depending on the target's location.

- We would like to investigate alternative approaches to deriving the values that we are using as durations. A possible starting point is to consider varying the durations based on the last released location of the target. This approach is based on an observation that we made while evaluating our sighting blurring algorithm that the mean distance and standard deviation between successive distinct blurred locations varied significantly for a given target and magnitude depending on the cartographical features of the location.

- We would like to study additional sociological techniques that can be used to attack a target's sightings that have been observed over an extended period of time. It is possible that the findings of this study could help targets understand the extent of sighting blurring that needs to be performed on their sightings, and hence increase their privacy.

7.6 Publications Arising

There are three significant international peer-reviewed publications that have arisen as a result of the work described in this thesis.

- The architecture and protocol described in Chapter 4 was presented at the *International Conference on Pervasive Services (ICPS) 2008* [28].

- The access control model described in Chapter 5 was presented at the *International Workshop on Security In Information Systems (WOSIS) 2008*, which was hosted as part of the *International Conference on Enterprise Information Systems (ICEIS) 2008* [29].

- The sighting blurring algorithm described in Chapter 6 was presented at the *International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI) 2008* [27].

The feedback regarding these three publications that was received from both the anonymous reviewers, and the conference attendees, has been incorporated into this thesis.

7.7 Conclusions

In this chapter we formed our conclusions to this thesis. In particular, we identified both the major contributions and the minor contributions of this thesis in terms of our architecture and protocol, our access control model, and our sighting blurring algorithm. We also identified several areas of the work described in this thesis that require further investigation. Finally, we identified the publications that have arisen as a result of the work described in this thesis.

The overall goal of the work described in this thesis was to reduce the amount of trust that users must have in both LBSs and other users by developing security techniques for use with these LBSs. In particular, we described an architecture and protocol, an access control model, and a sighting blurring algorithm which all increase users' security.

In conclusion, the security techniques described in this thesis can be used as part of an overall solution for reducing the amount of trust that users must have in LBSs, and that these security techniques form a significant contribution to any future development of LBSs that operate on the Internet.

List of Acronyms

API Application Programming Interface

DAC Discretionary Access Control

DCU Dublin City University

EU European Union

GPS Global Positioning System

GSM Global System for Mobile communications

HTML HyperText Markup Language

HTTP Hypertext Transfer Protocol

IAP Indirect Access Permission

IBE Identity Based Encryption

IBS Identity Based Signature

LBS Location Based Service

MAC Mandatory Access Control

MLP Mobile Location Protocol

MMS Multimedia Messaging Service

OMA Open Mobile Alliance

PAP Proxy Access Permission

PDA Personal Digital Assistant

PKI Public Key Infrastructure

PoI Point of Interest

SMS Short Message Service

UCD University College Dublin

UMTS Universal Mobile Telecommunications System

URL Uniform Resource Locator

W3C World Wide Web Consortium

WGS 84 World Geodetic System 1984

XML eXtensible Markup Language

Bibliography

[1] 3rd Generation Partnership Project. *Technical Specification Group GSM/EDGE Radio Access Network; Functional stage 2 description of Location Services (LCS) in GERAN (Release 8) (3GPP TS 43.059 V8.0.0)*, November 2007.

[2] Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper, and Mike Pinkerton. Cyberguide: A Mobile Context-Aware Tour Guide. *Wireless Networks*, 3(5):421–433, 1997.

[3] Denise Anthony, Tristan Henderson, and David Kotz. Privacy in Location Aware Computing Environments. *IEEE Pervasive*, 6(4):64–72, Oct–Dec 2007.

[4] Claudio Agostino Ardagna, Marco Cremonini, Ernesto Damiani, Sabrina De Capitani di Vimercati, and Pierangela Samarati. Supporting Location-Based Conditions in Access Control Policies. In *ASIACCS '06: Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 212–222, New York, NY, USA, 2006. ACM.

[5] Claudio Agostino Ardagna, Marco Cremonini, Ernesto Damiani, Sabrina De Capitani di Vimercati, and Pierangela Samarati. Location Privacy Protection Through Obfuscation-Based Techniques. In Steve Barker and Gail-Joon Ahn, editors, *DBSec*, volume 4602 of *Lecture Notes in Computer Science*, pages 47–60. Springer, 2007.

[6] Vijayalakshmi Atluri and Heechang Shin. Efficient Enforcement of Security Policies Based on Tracking of Mobile Users. In Ernesto Damiani and Peng

Liu, editors, *DBSec*, volume 4127 of *Lecture Notes in Computer Science*, pages 237–251. Springer, 2006.

[7] Vijayalakshmi Atluri and Heechang Shin. Efficient Security Policy Enforcement in a Location Based Service Environment. In Steve Barker and Gail-Joon Ahn, editors, *DBSec*, volume 4602 of *Lecture Notes in Computer Science*, pages 61–76. Springer, 2007.

[8] Joonsang Baek and Yuliang Zheng. Identity-Based Threshold Decryption. In *Proceedings of PKC'04*, volume 2947 of *Lecture Notes in Computer Science*, pages 262–276. Springer-Verlag, March 2004.

[9] Louise Barkhuus and Anind K. Dey. Location-Based Services for Mobile Telephony: A study of users' privacy concerns. In *Proceedings of IFIP INTERACT03: Human-Computer Interaction*, page 709. IFIP Technical Committee No 13 on Human-Computer Interaction, 2003.

[10] Paolo Bellavista, Antonio Corradi, and Carlo Giannelli. Efficiently Managing Location Information with Privacy Requirements in Wi-Fi Networks: a Middleware Approach. In *ISWCS 2005 Main Symposium*, pages 91–95, Siena, Italy, September 2005.

[11] Alastair R. Beresford and Frank Stajano. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[12] Alastair R. Beresford and Frank Stajano. Mix Zones: User Privacy in Location-aware Services. In *PERCOMW '04: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, page 127, Washington, DC, USA, 2004. IEEE Computer Society.

[13] Tim Berners-Lee, Larry Masinter, and Mark McCahill. Uniform Resource Locators (URL). RFC 1738, IETF Network Working Group, December 1994.

[14] Harini Bharadvaj, Anupam Joshi, and Sansanee Auephanwiriyakul. An Active Transcoding Proxy to Support Mobile Web Access. In *Symposium on Reliable Distributed Systems*, pages 118–123, 1998.

[15] Charles Brooks, Murray S. Mazer, Scott Meeks, and Jim Miller. Application-Specific Proxy Servers as HTTP Stream Transducers. In *Proceedings of the 4th International World Wide Web Conference*, pages 539–548, 1995.

[16] Mauro Brunato and Roberto Battiti. PILGRIM: A Location Broker and Mobility-Aware Recommendation System. In *PERCOM '03: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, page 265, Washington, DC, USA, 2003. IEEE Computer Society.

[17] Thibault Candebat. *A Secure Architecture enabling End-User Privacy in the context of Commercial Wide-Area Location-enhanced Web Services.* PhD thesis, Faculty of Engineering and Computing, School of Computing, Dublin City University, Dublin, Ireland, July 2005.

[18] Thibault Candebat, Cameron Ross Dunne, and David T. Gray. Pseudonym management using mediated identity-based cryptography. In *DIM '05: Proceedings of the 2005 workshop on Digital identity management*, pages 1–10, New York, NY, USA, 2005. ACM Press.

[19] Thibault Candebat and David T. Gray. Secure Pseudonym Management Using Mediated Identity-Based Encryption. *Journal of Computer Security*, 14(3):249–267, 2006.

[20] Reynold Cheng, Yu Zhang, Elisa Bertino, and Sunil Prabhakar. Preserving User Location Privacy in Mobile Data Management Infrastructures. In George Danezis and Philippe Golle, editors, *Privacy Enhancing Technologies*, volume 4258 of *Lecture Notes in Computer Science*, pages 393–412. Springer, 2006.

[21] Xiangguo Cheng, Lifeng Guo, and Xinmei Wang. An Identity-based Mediated Signature Scheme from Bilinear Pairing. *International Journal of Network Security*, 2(1):29–33, January 2006.

[22] Keith Cheverst, Nigel Davies, Keith Mitchell, and Adrian Friday. Experiences of Developing and Deploying a Context-Aware Tourist Guide: The GUIDE

Project. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 20–31, New York, NY, USA, 2000. ACM.

[23] Commission for Communications Regulation. *Irish Communications Market - Quarterly Key Data*, June 2008.

[24] George Danezis, Stephen Lewis, and Ross Anderson. How Much is Location Privacy Worth? In *Fourth Workshop on the Economics of Information Security*, June 2005.

[25] Anind K. Dey, Daniel Salber, Gregory D. Abowd, and Masayasu Futakawa. The Conference Assistant: Combining Context-Awareness with Wearable Computing. In *ISWC '99: Proceedings of the 3rd IEEE International Symposium on Wearable Computers*, page 21, Washington, DC, USA, 1999. IEEE Computer Society.

[26] Matt Duckham and Lars Kulik. A Formal Model of Obfuscation and Negotiation for Location Privacy. In Hans-Werner Gellersen, Roy Want, and Albrecht Schmidt, editors, *Pervasive*, volume 3468 of *Lecture Notes in Computer Science*, pages 152–170. Springer, 2005.

[27] Cameron Ross Dunne, Thibault Candebat, and David Gray. A Frequency Based Sighting Blurring Algorithm for Use with Location Based Services on the Internet. In *International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI) 2008*, pages 3–12, New York, NY, USA, 2008. ACM.

[28] Cameron Ross Dunne, Thibault Candebat, and David Gray. A Three-Party Architecture and Protocol that Supports Users with Multiple Identities for Use with Location Based Services. In *ICPS '08: Proceedings of the 5th international conference on Pervasive services*, pages 1–10, New York, NY, USA, 2008. ACM.

[29] Cameron Ross Dunne, Thibault Candebat, and David Gray. An Access Control Model for Location Based Services. In Alfonso Rodríguez, Mariemma Inmacu-

lada Yagüe del Valle, and Eduardo Fernández-Medina, editors, *WOSIS*, pages 49–58. INSTICC Press, 2008.

[30] A. Escudero-Pascual and Jr. Maguire, G. Q. Role(s) of a proxy in location based services. In *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*, volume 3, pages 1252–1256, September 2002.

[31] European Union. *Directive 2002/58/EC of The European Parliament and of The Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications)*, July 2002.

[32] European Union. *Special Eurobarometer: Issues Relating To Business And Consumer E-Commerce*, March 2004.

[33] Roy T. Fielding, James Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen, Larry Masinter, Paul J. Leach, and Tim Berners-Lee. Hypertext Transfer Protocol - HTTP/1.1. RFC 2616, IETF Network Working Group, June 1999.

[34] Bugra Gedik and Ling Liu. Location Privacy in Mobile Systems: A Personalized Anonymization Model. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pages 620–629, Washington, DC, USA, 2005. IEEE Computer Society.

[35] Google Maps. Available at `http://maps.google.com/` (Accessed: 28/2/2008).

[36] Marco Gruteser, Jonathan Bredin, and Dirk Grunwald. Path Privacy in Location-aware Computing. In *MobiSys 2004 Workshop on Context Awareness*, June 2004.

[37] Marco Gruteser and Dirk Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42, New York, NY, USA, 2003. ACM.

[38] Marco Gruteser and Xuan Liu. Protecting Privacy in Continuous Location-Tracking Applications. *IEEE Security and Privacy*, 2(2):28–34, 2004.

[39] Christian Hauser and Matthias Kabatnik. Towards Privacy Support in a Global Location Service. In *IFIP Workshop on IP and ATM Traffic Management, Paris*, pages 81–89, 2001.

[40] Urs Hengartner. Hiding Location Information from Location-Based Services. In *8th International Conference on Mobile Data Management (MDM 2007), Mannheim, Germany, May 7-11, 2007*, pages 268–272, 2007.

[41] Urs Hengartner and Peter Steenkiste. Implementing Access Control to People Location Information. In *SACMAT '04: Proceedings of the ninth ACM symposium on Access control models and technologies*, pages 11–20, New York, NY, USA, 2004. ACM.

[42] Urs Hengartner and Peter Steenkiste. Protecting Access to People Location Information. In *Security in Pervasive Computing: First International Conference, Boppard, Germany, March 12-14, 2003. Revised Papers*, volume 2802 / 2004, pages 25–38, 2004.

[43] Florian Hess. Efficient Identity Based Signature Schemes Based on Pairings. In *SAC '02: Revised Papers from the 9th Annual International Workshop on Selected Areas in Cryptography*, pages 310–324, London, UK, 2003. Springer-Verlag.

[44] Baik Hoh and Marco Gruteser. Protecting Location Privacy Through Path Confusion. In *SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 194–205, Washington, DC, USA, 2005. IEEE Computer Society.

[45] Richard Hull, Bharat Kumar, Daniel F. Lieuwen, Peter F. Patel-Schneider, Arnaud Sahuguet, Sriram Varadarajan, and Avinash Vyas. Enabling Context-

Aware and Privacy-Conscious User Data Sharing. In *Mobile Data Management*, pages 187–198. IEEE Computer Society, 2004.

[46] ITU. *Information Technology - Open Systems Interconnection - The Directory: Authentication Framework (ITU-T Recommendation X.509)*, 1997.

[47] D Kesdogan, P Reichl, and K Junghärtchen. Distributed Temporary Pseudonyms: A New Approach for Protecting Location Information in Mobile Communication Networks. In *Fifth European Symposium on Research in Computer Security*, volume 1485, pages 295–312, Louvain-la-Neuve, Belgium, 16–18 1998. Springer-Verlag.

[48] John Krumm. Inference Attacks on Location Tracks. In *Pervasive*, volume 4480 of *Lecture Notes in Computer Science*, pages 127–143. Springer, 2007.

[49] Scott Lederer, Jennifer Mankoff, and Anind K. Dey. Who wants to know what when? Privacy preference determinants in ubiquitous computing. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 724–725, New York, NY, USA, 2003. ACM.

[50] Ulf Leonhardt and Jeff Magee. Security Considerations for a Distributed Location Service. *Journal of Network and Systems Management*, 6(1):51–70, 1998.

[51] Iqbal Mohomed, Alvin Chin, Jim Chengming Cai, and Eyal de Lara. Community-Driven Adaptation: Automatic Content Adaptation in Pervasive Environments. In *WMCSA '04: Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications*, pages 124–133, Washington, DC, USA, 2004. IEEE Computer Society.

[52] Ginger Myles, Adrian Friday, and Nigel Davies. Preserving Privacy in Environments with Location-Based Applications. *IEEE Pervasive Computing*, 2(1):56–64, 2003.

[53] Ajith K. Narayanan. Realms and States: A Framework for Location Aware Mobile Computing. In *WMC '01: Proceedings of the 1st international workshop on Mobile commerce*, pages 48–54, New York, NY, USA, 2001. ACM.

[54] National Imagery and Mapping Agency. *Department of Defense World Geodetic System 1984*, NIMA TR8350.2, third edition, January 2000.

[55] Open Mobile Alliance. Available at `http://www.openmobilealliance.com/` (Accessed: 28/2/2008).

[56] Open Mobile Alliance. *Mobile Location Protocol (MLP) - Candidate Version 3.3*, April 2008.

[57] Ordnance Survey Ireland and Ordnance Survey of Northern Ireland. *The Irish Grid*, 1996.

[58] Ordnance Survey Ireland and Ordnance Survey of Northern Ireland. *Making Maps Compatible with GPS*, 1999.

[59] B. W. Parkinson and J. J. Spilker, editors. *Global Positioning System: Theory and Applications, vol. I*, volume 163 of *Progress in Astronautics and Aeronautics*. American Institute of Aeronautics, Inc., Washington DC, 1996.

[60] The Parlay Group. Available at `http://www.parlay.org/` (Accessed: 28/2/2008).

[61] The Parlay Group. *Open Service Access (OSA); Application Programming Interface (API); Part 12: Charging SCF (Parlay 6))*, March 2007.

[62] The Parlay Group. *Open Service Access (OSA); Application Programming Interface (API); Part 6: Mobility SCF (Parlay 6))*, March 2007.

[63] The Parlay Group. *Open Service Access (OSA); Parlay X Web Services; Part 6: Payment (Parlay X 3)*, June 2007.

[64] The Parlay Group. *Open Service Access (OSA); Parlay X Web Services; Part 9: Terminal Location (Parlay X 3)*, June 2007.

[65] Andreas Pfitzmann and Marit Hansen. Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology. Technische Universität Dresden, Version v0.31, 15/2/2008.

[66] Indrakshi Ray and Mahendra Kumar. Towards a location-based mandatory access control model. *Computers & Security*, 25(1):36–44, 2006.

[67] Tom Rodden, Adrian Friday, Henk Muller, and Alan Dix. A Lightweight Approach to Managing Privacy in Location-Based Services. Technical Report Equator-02-058, University of Nottingham and Lancaster University and University of Bristol, 2002.

[68] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 47–53, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[69] Todd Simcock, Stephen Peter Hillenbrand, and Bruce H. Thomas. Developing a Location Based Tourist Guide Application. In *ACSW Frontiers '03: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003*, pages 177–183, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[70] W3C. *HTML 4.01 Specification*, December 1999.

[71] W3C. *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*, April 2002.

[72] W3C. *Extensible Markup Language (XML) 1.0 (Fourth Edition)*, August 2006.

[73] Ge Zhong, Ian Goldberg, and Urs Hengartner. Louis, Lester and Pierre: Three Protocols for Location Privacy. In Nikita Borisov and Philippe Golle, editors, *Privacy Enhancing Technologies*, volume 4776 of *Lecture Notes in Computer Science*, pages 62–76. Springer, 2007.