A Secure Architecture enabling End-User Privacy in the context of Commercial Wide-Area Location-enhanced Web Services

Thibault Candebat, 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

July 2005

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 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 Id: 50194879

Date : 09/09/2005

Acknowledgements

This thesis is the result of my four years of PhD study at the School of Computing in Dublin City University. Many people have helped me achieve this goal and I am grateful to all of them.

Firstly, I would like to thank Dr David T. Gray for his excellent supervision and advice throughout the duration of my thesis. In particular, I am grateful for his support when arguing on crucial issues with the rest of the research group (*i.e.* Cameron), taking my side 90% of the time.

Of course, I want to thank Cameron, my colleague and friend, for his everyday optimism and enthusiasm. I will always remember those long afternoons spent debating geopolitics and EU regulations. I will not forget our long arguments about software engineering design either. I believe I have learnt a lot from them as well as from his expertise. Cheers for your support and insightful comments on my work during the past four years !

My close friends also really helped me relax and enjoy my time throughout my PhD. A big thank you to Nico, with whom I enjoyed my first year in Ireland as an Erasmus student, to Toufman for his willingness to come and join me in my western celtic territory explorations, and to Matthieu who almost bumped into my car on the way down to Doolin and that has remained since then, one of my best friends.

I thank all the people from the postgrad lab of the School of Computing for making this place a pleasant working environment. In particular, I would like to thank Kieran, Wu Hai and Q for welcoming me at the start of my PhD, back in the old days when the lab only counted twenty people.

I certainly would not have survived this long in Ireland without the support of the Dead Foie Gras Eater Society. Thank you JM, Bea and Maja for discussing the grounds of fundamental research while experimenting with latin food and coffees. Simona, I also haven't forgotten your great meals, especially the tiramisu, so thank you !

I believe I have to thank the people who helped me forget about my research as well. Therefore, thank you JR for those endless runs around Albert College Park twice a week: you just could not help speeding up... Jayzz !! The Caving Club and Sub Aqua Club enabled me to escape the lab many times during those four years, so thank you very much lads for the week ends away: it helped a lot !

Finally, a big thank you to Saibh, my petite, would really helped me get this thesis written in gramatically correct english. The examiners really appreciated :)

I would have liked to thank a lot more people. However, my thesis is not really about my time here in Ireland or about the people I have met, even though I believe they are all linked. I will therefore finish by gratefully acknowledging the Dublin City University School of Computing for funding my first year of research and thanking Enterprise Ireland for their support through the grants IF/2002/0336/N and PC/2004/446.

Contents

Abstract vi

List of Figures viii

1 Introduction 1

1.1 Introduction . . . 1

1.2 Drivers . . . 1

1.2.1 Mobile Telephony . . . 2

1.2.2 Location Information Provision . . . 3

1.2.3 Wireless Internet Connectivity . . . 4

1.2.4 Conclusion . . . 4

1.3 Location-Based Services . . . 5

1.3.1 Definition . . . 5

1.3.2 Classification . . . 6

1.3.3 Conclusion . . . 7

1.4 Privacy and Context Data Security . . . 7

1.4.1 Definitions . . . 8

1.4.2 Regulations . . . 9

1.4.3 Conclusion . . . 11

1.5 Problem statement . . . 11

1.6 Thesis Objectives . . . 11

1.6.1 Thesis Contributions . . . 12

1.6.2 Thesis Limitations . . . 13

1.7 Thesis Outline . . . 13

2 Background 14

2.1 Introduction . . . 14

2.2 Building Blocks . . . 14

2.2.1 Symmetric Cryptography . . . 15
2.2.2 Asymmetric Cryptography . . . 15
2.3 Public Key Infrastructure . . . 18
2.3.1 PGP . . . 19
2.3.2 SPKI/SDSI . . . 19
2.3.3 X.509 . . . 21
2.3.4 Identity-Based Encryption and Applications . . . 25
2.4 Mobile Phone Systems and their Security . . . 26
2.4.1 Global System for Mobile Communications (GSM) . . . 27
2.4.2 General Packet Radio Service (GPRS) . . . 31
2.4.3 Universal Mobile Telecommunications System (UMTS) . . . 32
2.4.4 Internet Access over Mobile Phone Networks . . . 33
2.5 Location Management in Mobile Networks . . . 38
2.5.1 Location Positioning Technologies . . . 38
2.5.2 Network Upgrades to Support Location Information . . . 41
2.5.3 Operations . . . 41
2.6 Refinement of the Thesis Objectives . . . 43

3 Analysis and Requirements **44**
3.1 Introduction . . . 44
3.2 Environment . . . 44
3.2.1 Functional Description and Requirements . . . 44
3.2.2 Threat Model . . . 45
3.2.3 Conclusion . . . 47
3.3 Middleware and Protocol . . . 47
3.3.1 Middleware . . . 48
3.3.2 Protocol . . . 50
3.3.3 Conclusion . . . 51
3.4 Public Key Infrastructure . . . 52
3.5 Conclusion . . . 52

4 Related Research **53**
4.1 Introduction . . . 53
4.2 Identity Blurring . . . 53
4.2.1 Mix Zones . . . 54
4.2.2 Location k-anonymity . . . 55

4.3 Location Blurring . . . 56
4.4 Machine Readable Location Privacy Policies . . . 57
4.5 Access Control to Location Information . . . 59
4.5.1 Multi-target access control for location information . . . 60
4.5.2 Certificate-based solutions . . . 61
4.5.3 Role-based Access Control Mechanisms . . . 62
4.5.4 Rule engine-based Access Control Mechanisms . . . 63
4.6 Secure Architectures for context aware computing . . . 63
4.6.1 Location-aware computing with no location disclosure. . . . 64
4.6.2 Proxy-based location information disclosure . . . 64
4.6.3 Token-based location information disclosure . . . 65
4.6.4 Secure platforms for location-based services provision . . . 66
4.7 Standards . . . 67
4.7.1 Open Mobile Alliance . . . 67
4.7.2 Parlay/OSA . . . 68
4.7.3 Geopriv . . . 69
4.7.4 The GSM Association . . . 69
4.7.5 3GPP . . . 70

5 Architecture, Infrastructure and Protocol **71**
5.1 Introduction . . . 71
5.2 Architecture . . . 71
5.3 Middleware . . . 72
5.3.1 Proxy-based approach . . . 72
5.3.2 Components . . . 73
5.3.3 Implementation . . . 80
5.3.4 Evaluation . . . 81
5.3.5 Conclusion . . . 82
5.4 Protocol . . . 83
5.4.1 Design . . . 83
5.4.2 Implementation . . . 84
5.4.3 Evaluation . . . 85
5.4.4 Conclusion . . . 85
5.5 Integration . . . 86
5.6 Conclusion . . . 87

6 The Location Blurring Algorithm 88
6.1 Introduction . . . 88
6.2 Presentation . . . 89
6.3 Definitions . . . 89
6.4 Threat Model . . . 91
6.5 Requirements . . . 93
6.6 Geometrical considerations . . . 94
6.7 Our *Location Blurring* Algorithm . . . 96
6.7.1 Principles . . . 96
6.7.2 Description . . . 97
6.8 Implementation . . . 106
6.8.1 Environment . . . 106
6.8.2 Algorithm . . . 108
6.9 Evaluation . . . 109
6.9.1 Security . . . 109
6.9.2 Accuracy . . . 112
6.9.3 Limitations . . . 115
6.10 Conclusion . . . 119

7 Public Key Infrastructure 120
7.1 Introduction . . . 120
7.2 Motivations . . . 120
7.2.1 Physical Layer Security . . . 121
7.2.2 Network Layer Security . . . 121
7.3 Architecture . . . 123
7.3.1 Principles . . . 123
7.3.2 Description . . . 124
7.4 Encryption and Signature Algorithms . . . 126
7.4.1 Formal Specification . . . 126
7.4.2 Group and Parameter Selection . . . 131
7.5 Prototype Implementation . . . 135
7.5.1 Cryptographic Operations . . . 135
7.5.2 Private Key Generator . . . 137
7.5.3 Security Mediator . . . 138
7.6 Scenario . . . 139
7.7 Evaluation . . . 140

7.7.1 Performance . . . 140
7.7.2 Security Analysis . . . 142
7.8 Conclusion . . . 146

8 Conclusions and Future Work 148
8.1 Introduction . . . 148
8.2 Thesis Summary . . . 149
8.3 Major Contributions . . . 150
8.3.1 A Location Blurring algorithm . . . 150
8.3.2 A Public Key Infrastructure . . . 151
8.4 Other contributions . . . 151
8.4.1 A critical survey of context information security . . . 151
8.4.2 A prototype for a secure architecture delivering *LBS* over the Internet 151
8.5 Future work . . . 152
8.5.1 The Orient Platform . . . 152
8.5.2 Location Blurring algorithm . . . 153
8.5.3 PKI . . . 154
8.6 Concluding remarks . . . 155

List of Acronyms 157

Bibliography 163

Appendices 184

A The Orient Protocol Stack 184

B The Privacy Engine User Interfaces 185

C Cryptographic API 188
C.1 User API . . . 188
C.2 Private Key Generator API . . . 189
C.3 Security Mediator (SEM) API . . . 190
C.4 Common API . . . 191

Abstract

Mobile location-based services have raised privacy concerns amongst mobile phone users who may need to supply their identity and location information to untrustworthy third parties in order to access these applications. Widespread acceptance of such services may therefore depend on how privacy sensitive information will be handled in order to restore users' confidence in what could become the "killer app" of 3G networks.

The work reported in this thesis is part of a larger project to provide a secure architecture to enable the delivery of location-based services over the Internet. The security of transactions and in particular the privacy of the information transmitted has been the focus of our research. In order to protect mobile users' identities, we have designed and implemented a proxy-based middleware called the *Orient Platform* together with its *Orient Protocol*, capable of translating their real identity into pseudonyms.

In order to protect users' privacy in terms of location information, we have designed and implemented a *Location Blurring* algorithm that intentionally downgrades the quality of location information to be used by location-based services. The algorithm takes into account a blurring factor set by the mobile user at her convenience and blurs her location by preventing real-time tracking by unauthorized entities. While it penalizes continuous location tracking, it returns accurate and reliable information in response to sporadic location queries.

Finally, in order to protect the transactions and provide end-to-end security between all the entities involved, we have designed and implemented a Public Key Infrastructure based on a Security Mediator (SEM) architecture. The cryptographic algorithms used are identity-based, which makes digital certificate retrieval, path validation and revocation redundant in our environment. In particular we have designed and implemented a cryptographic scheme based on Hess' work [108], which represents, to our knowledge, the first identity-based signature scheme in the SEM setting. A special private key generation process has also been developed in order to enable entities to use a single private key in conjunction with multiple pseudonyms, which significantly simplifies key management.

We believe our approach satisfies the security requirements of mobile users and can help restore their confidence in location-based services.

List of Figures

2.1 Architecture of GSM networks. 29
2.2 Architecture of 3G networks. 32
2.3 The WAP Architecture. 35
2.4 Accuracy of positioning techniques. 40
2.5 Network Areas in Wireless Networks 42

3.1 General Architecture Topology. 46

5.1 Revised General Architecture Topology 72
5.2 Charging Mechanism 78
5.3 Architecture of the Orient Platform. 81
5.4 Protocol stacks of the different entities involved in location-based content provision. 86

6.1 Intersection and border problems. 95
6.2 Overlapping grids. 96
6.3 Maximal distance between any two points belonging to two *Blurred Localities*. 99
6.4 Location Tracking through the use of the *Location Blurring* algorithm. 101
6.5 *Location Blurring* Algorithms Testing Framework. 108
6.6 Non Trusted LBS Collusion with same *Blurred Sighting* accuracy. 111
6.7 Non Trusted LBS Collusion with different *Blurred Sighting* accuracy. 112
6.8 Accuracy of the Algorithm in a Location Tracking process carried out every minute on a driving *Target*. 116
6.9 Accuracy of the Algorithm in a Location Tracking process carried out every minute on a cycling *Target*. 117
6.10 Accuracy of the Algorithm in a Location Tracking process carried out every minute on a walking *Target*. 118

7.1 The WAP Gap. 122

7.2 End-to-end secure communications in the WAP Architecture. 122
7.3 End-to-end secure communications in the Orient Platform's environment. . . 122
7.4 Public Key Infrastructure deployed in the Orient Platform's environment. . . 125
7.5 Private key share selection in the encryption process. 139

A.1 The *Orient Protocol* Stack. 184

B.1 Interface for registering privacy preferences for a particular *LBS*. 185
B.2 Interface for choosing the *Subject* and assigning her to a cluster. 186
B.3 Interface for visualizing and modifying privacy preferences for a particular *Subject* and *LBS*. 186
B.4 Interface for modifying a specific timeslot. 187

Chapter 1

Introduction

1.1 Introduction

This research project aims to provide software tools that enable mobile network operators to deploy mobile services based on the location of their users over the Internet. In particular, we focus on providing techniques that safeguard end-user privacy. The framework proposed in this thesis also aims at facilitating software development regarding location information manipulation by factoring out location related functionalities. It relieves developers of the burden of dealing with location information acquisition and security.

In this introductory chapter we provide a brief description of the technological and regulatory environments in which such location-based mobile services will operate. We state the fundamental challenge that we tackle throughout the research work carried out and give an overview of the structure of the dissertation.

1.2 Drivers

This section intends to point out the main factors that fostered the emergence of location-based services. In particular, we comment on some advances made in the field of mobile telephony during the past decade. We also emphasize the recent need for mobile phone location information management and describe the new regulatory framework designed to help emergency services deal with mobile users' emergency phone calls. Finally, we give a short introduction on wireless Internet access technologies and show how all this relates to the advent of location-based services.

1.2.1 Mobile Telephony

The concept of mobile telephony has been adopted in a short period of time. According to a recent study [158], the number of mobile phones in the world reached 1.2 billion in 2003, with production in excess of 450 million mobile phones the same year. The estimated number of units for 2006 exceeds 2 billion. With an ever increasing performance and miniaturization, mobile phones attract more and more users, reaching almost 70% of people in western countries. In some countries like India, the number of mobile phone users has already outstripped traditional landline connections [23]. Although it took near 50 years for landline personal communications to catch on, mobile telephony has been adopted in less than a decade. This can be explained by the fact that in general, developing countries find mobile phone infrastructures to be a very cost-effective solution for deploying telecommunication networks. In [26], the ever increasing number of mobile phones sold is also explained by the high rate of mobile phone replacements, which today account for more than 80% of all mobile phone purchases.

In this thesis, we will consider mobile devices that operate within mobile phone networks. In particular, we will consider handsets that are GSM, GPRS as well as UMTS enabled. More precise information about the nature of these wireless networks is provided in Section 2.4. Mobile phones can be used to send and receive voice calls like normal fixed line phones. In addition, basic handsets already support simple utility software such as calendars, calculators or alarm clocks. More advanced terminals, also known as "smart phones", present enhanced features such as the ability to download applications like games or even the possibility to browse the Internet. In fact, today's mobile terminals share a lot of similarities with computers. They are typically controlled by microprocessors that can run at a speed close to 200 Mhz using up to 32MB of ROM and RAM. They also run an operating system and can include a Java Virtual Machine tailored to mobile environments used to run either pre-installed applications or downloaded code. Today's mobile phones can also support a wide range of protocols used in the wired world such as POP3, IMAP4, and SMTP and of course TCP/IP. However, even though their limited size and weight constitute an advantage as far as mobility is concerned, they considerably reduce mobile phone capabilities in terms of display and user interface. To overcome the problem, accessories such as enhanced keyboards can be connected to the mobile phones. Alternatively, mobile terminals can be used as accessories, providing a laptop computer with mobile network connectivity wherever the wireless network coverage is available. In this case, the laptop only uses the mobile device as a means to connect to Internet using a wireless connection.

1.2.2 Location Information Provision

As mentioned in Section 1.2.1, the number of mobile phones is already significant compared to the number of landline phones. As a result, an increasing percentage of emergency calls are made using mobile phones. Location information is very helpful to emergency services as it enables them to save time and lives by knowing precisely where their help is needed. While locating landline phone users consists in looking up subscriber addresses in an electronic directory, retrieving mobile phone users' location has proved to be problematic.

In 1996, the United States' Federal Communications Commission (FCC) issued a mandate that required network operators to provide emergency services with callers' location information in order to assist emergency victims promptly. In order to fulfill the necessary requirements, the program called E911 for Enhanced 911 was divided into two different phases. The first one, already implemented, required that the callers' phone number as well as the location of the base station serving their mobile phone to be forwarded to emergency services using network-based or handset-based methods; see Section 2.5.1 for further developments. The second phase required network operators to be able to locate mobile phone users with even more accuracy, as described in Table 1.1. Mobile operators have however been quite reluctant to implement E911, seeing no potential benefit in deploying costly high-accuracy location technologies. The full implementation of E911 is due for completion by the end of the year 2005 and wireless carriers are already struggling to meet the deadline.

Solutions	*67% of calls*	*95% of calls*
Handset-based	50 meters	150 meters
Network-based	100 meters	300 meters

Table 1.1: Accuracy Required by the FCC for Locating Mobile Phones.

A similar initiative was launched in the E.U. by the European Commission in 2003. The location enhanced 112 emergency service (E112) [133] is an E.U. recommendation that aims to provide information about the callers' location in both fixed and cellular-based networks. While E911 took almost 10 years to implement, E112 may greatly benefit from the U.S. experience and be adopted more quickly [142]. As opposed to the situation in the U.S. in 1996, E112 benefits from a more favorable environment. GSM networks constitute the most widely used standard for 2G networks throughout Europe, which gives all the countries a common base to start implementing the recommendation. High-accuracy location technologies have also received comprehensive testing and are now seen as revenue generating network upgrades.

1.2.3 Wireless Internet Connectivity

Within a few years, the Internet has evolved to become the most popular and comprehensive source of information. This networked environment radically transformed the way people interact with each other and do business. People quickly adapted to free access whatever their location, provided they had access to a computer. With the advent of mobile communications, a need for mobile internet access consequently arose. However, while GSM mobile networks could offer a reasonable bandwidth to support voice communications, they were not initially designed to transfer data such as web pages. A first solution was proposed in 1999 by the WAP forum [5] under the name of Wireless Application Protocol. It consisted in providing an architecture that would tailor the Internet stack protocols to mobile phone weak capabilities; see Section 2.4.4 for further descriptions of WAP and related technologies. Once implemented, WAP was presented as the technology that would let one access the Internet through her mobile phone. However, Internet web pages had to be either converted or generated from scratch by content providers in order to adapt the new lightweight standard. Furthermore, handsets were technically limited and people became disappointed when they experienced slow and expensive connections combined with a poor WAP content delivery. As a result, WAP never really took up, not because the technology was not good enough but because of the way it had been marketed.

At the same time, a similar initiative was launched in Japan in order to enable mobile phone users to access the Internet from their mobile device. Running on more powerful wireless networks, I-mode was primarily introduced as a value added service and not only allowed users to access real Internet content but also provided them with an always-on connection with a more flexible billing scheme. Similar wireless networks such as GPRS or UMTS will soon be available worldwide and already allow for data transfers at a higher bandwidth than before. Furthermore, mobile handsets have evolved into "smart phones"; see Section 1.2.1, and are now comparable to portable computers. While the first approach to mobile internet access consisted in adapting the existing Internet technologies to mobile phone usage, the trend is now for mobile devices to access the wired Internet with very little or no modification to the standard protocols.

1.2.4 Conclusion

Mobile telephony has recently witnessed some profound technological evolutions that influenced the regulatory framework in which it operates. In particular, mobile phone networks and terminals are now powerful enough to support wireless connections and avail of Internet services like normal desktop computers. New regulations now require mobile network oper-

ators to be able to locate each of their mobile subscribers. While this location information could be restricted to internal use only, it is very likely that it will be made available to external third parties that may, in turn, provide location related services to both desktop and mobile users. Mobile phone network operators are indeed looking for new sources of revenues and consortiums such as the Parlay Group [202] push the development of mobile data services by proposing technologies to help them open their networks [136, 203].

1.3 Location-Based Services

The advent of mobile computing devices combined with the deployment of wireless networks and positioning technologies has given birth to a new field of research known as *context-aware computing*. Context-aware computing encompasses the study of applications that consider the environment or *context* in which they operate as a runtime parameter. Several definitions of *context* have been proposed in [19, 50, 32]. Within the scope of this thesis, *context* refers to mobile users' characteristics and properties, namely, users' identity and users' physical location combined with a time reference. We also consider a subset of this field known as *location-aware computing*, whose applications are only triggered by users' location. In this section, we define the concept of location-based service (LBS), which is the most common form of context-aware computing applications, as well as a taxonomy that classifies the different sorts of LBS.

1.3.1 Definition

Within the scope of this thesis, we refer to a Location-based service, or *LBS*, as a service that offers to its users a value-added service that exploits the location of a set of mobile entities at a particular time. From a more technical point of view, we see a *LBS* as a context dependent web service. The W3C Web Services Architecture Working Group [213] defines a web service as a "a software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A Web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols". Apart from the *LBS* itself, we identify two different actors involved in the provision of *LBS*. *LBS* users or *Subjects* are the service requestors and can use either fixed or mobile devices in order to avail of the location dependent service. Located entities or *Targets* typically hold a mobile device that connects to a wireless mobile phone network such as the ones described in Section 2.4. *Targets*' mobile devices have to be locatable as defined in Section 2.5.1, so that their location can be determined and transmitted by some entity to the *LBS*. In some cases, a *LBS* user may play the two different roles at the same

time. For example, she may be considered both as a *Subject* and as a *Target* when requesting a *LBS* that necessitates her own location.

Finally, we are interested in the deployment of commercial wide-area *LBS*. Indeed, while some of the *LBS* considered in this thesis may only be used within a building, we will not focus on indoor specific *LBS*. Wide-area *LBS* can potentially be accessed from all over the world, provided the wireless network coverage is efficient enough to allow for *Targets* positioning. We also assume a commercial relationship between *Subjects* and *LBS* providers such as the ones existing between e-commerce web sites and their users.

1.3.2 Classification

From a technical point of view, a *LBS* can be assimilated to a web site that generates context-related content for its users. From a more functional point of view however, the definition given in 1.3.1 suffers from a lack of precision regarding the different types of location related services that exist. In this section, we present a classification that categorizes the different kinds of *LBS*.

Giaglis *et al.* [93] have established a taxonomy of *LBS* and classified the different services in 6 categories. The results of the study are shown in Table 1.2.

Services	*Examples*	*Accuracy Needs*
Emergency	Emergency calls	****
	Automotive assistance	***
Navigation	Directions	*****
	Traffic management	***
	Indoor routing	*****
	Group management	**
Information	Travel services	****
	Mobile yellow pages	***
	Infotainment services	****
Advertising	Banners, Alerts, Advertisements	****
Tracking	People tracking	*****
	Vehicle tracking	*
	Personnel tracking	***
	Product tracking	*****
Billing	Location sensitive billing	**

Table 1.2: Accuracy Required by the FCC for Locating Mobile Phones.

Since several kinds of *LBS* exist, it is very likely that every one of them has different needs in terms of accuracy of location information. For example, car traffic management *LBS* will have low location information accuracy expectations compared to people tracking *LBS*. This is mainly due to their relative speed: the slower the speed the better the accuracy

should be in order to offer *LBS* users a meaningful service. Therefore, the authors believe that a useful *LBS* classification should also take location accuracy needs into account.

From the classification proposed, we also identify two different types of *LBS*.

- User-triggered *LBS* are location-based services that deliver a context related service upon user query or user presence in a predefined location. For example, *LBS* such as information services, advertising or billing services fall into this category. We will refer to this kind of services as *pull LBS*.

- Self-triggered *LBS* tend to perform intermittent or continuous user tracking. For example, fleet management *LBS* continuously track someone's assets over time. Also, some advertising *LBS* can perform intermittent tracking, provided they can link someone's location with her identity. From now on, we will refer to this kind of service as *tracking LBS*.

1.3.3 Conclusion

Mobile network operators have recently invested a lot of money in third generation network licenses and network infrastructures; see Section 2.4.3 for further details. They are still looking for the killer application that will help them generate new revenue streams in order to reduce their debt and increase their Average Revenue Per User (ARPU). Location-based services may represent a good opportunity for them to do so. However, *LBS* technologies already raise security and privacy concerns amongst potential users [128, 31]. The fear of privacy invasion where people's location information would be disclosed to some central authority is already here and recalls unsurprisingly George Orwell's novel *1984*. *LBS* acceptance will therefore depend on the deployment of sound security practices and privacy techniques, able to convince their users that their privacy will remain safe [82].

1.4 Privacy and Context Data Security

Location-based services technologies are now available and test-bed applications are starting to flourish. However, *LBS* development is hampered by people's concerns related to the disclosure of their location information [153]. Potential *LBS* users do not feel comfortable in giving away their location, a characteristic now considered as a private piece of information. This loss of control over personal information is however a privacy risk inherent to most context aware applications. If not considered carefully, this threat can potentially lead to disturbing or even dangerous situations. In particular, some location-based advertising practices could be perceived as spam if someone's location information was given away to

some non-trustworthy entity. Such organizations could also build up some personal data collections and sell them to other entities without user consent, as it happens nowadays with email addresses collections. More serious threats could involve criminals stalking and tracking mobile users, a problem that would become even more significant if the latter were young teenagers for example. Analyzing tracking information can also help malicious entities learn about someone's interests or habits and by cross analyzing data from various sources, this can disclose even more private details such as where someone lives.

In this section, we explain some security related terms and tailor their definition to the field of context-aware computing. We also give an overview of the privacy regulations in place in different countries.

1.4.1 Definitions

LBS security services are very similar to the services the Internet provides to secure online transactions. In this section, we describe them in our context and illustrate their use by considering the following scenario. "Friend Finder" is a *LBS* that lets its users find out where their friends are located in real-time. A *Subject* S wants to query the service in order to locate a *Target* T.

First, S needs to prove to the *LBS* that she is really who she claims to be in order to avail of the "Friend Finder" service. After identifying herself to the *LBS*, she needs to provide the *LBS* with an evidence that will enable her to verify S's identity. This is equivalent to showing a passport when entering a foreign country. This process is known as *Authentication*.

Then, S needs to be authorized by T to access her location information. This authorized access may depend on multiple parameters, such as S's identity or attributes for instance. In this case, the security service offered can be referred to as *Authorization*. If the parameters involve anything other then the S's attributes, such as the date and time of the day or even T's location itself, the concept of *Access Control* over a particular resource is generally used to describe the security service.

Once access to T's location information has been granted to S, the *LBS* needs to retrieve T's location details, process them and deliver the value added content to S. Eavesdropping or monitoring unprotected internet connections is known to be a relatively easy task to perform. In order to protect e-commerce transactions and in particular location data transmissions and retention, cryptography (see Section 2.2.1) is generally used to ensure that sensitive information is not disclosed to unauthorized parties. This security service is known as *Confidentiality*.

Finally, S receives the location related content from the *LBS*. While some *LBS* may pro-

vide some services free of charge, it is very likely that most of them will require their users to pay for the content they receive. In order to ensure that neither S nor the LBS can deny having requested or offered a service without the other party consent, mechanisms known as digital signatures (see Section 2.2.2) are generally used to enable the *Non-Repudiation* security service. In this context, digital signatures also allow for the detection of unauthorized location information alteration during the data transmission and this way ensures the *Integrity* of messages sent from one entity to another.

1.4.2 Regulations

Mobile network operators already know to some extent where every mobile phone user is located. This is necessary to implement a cellular network and usually, most people trust them not to misbehave with this data. However, the new positioning technologies described in Section 2.5.1 enable much finer grained location requests than before. Furthermore, this location information will be soon made available not only to emergency services but also to independent third parties in commercial relationships with mobile network operators.

In order to help preventing malicious usage of location information, some regulations regarding to the collection and processing of personal data must be enforced in non-emergency situations.

European Union

A detailed description of the European legal framework regarding to this issue is available in [153]. The two principal directives dealing with personal location information are presented here.

The General Privacy Directive (95/46) [200] deals with the protection of individuals with regard to the processing of personal data and the free movement of such data. According to this directive, the entity that performs data collection on a subject must inform the subject as well as a special authority that decides what processing is allowed, taking into account the purpose data collection. More sensitive data such as racial or ethnic origins for example are treated in a stricter manner. The Directive on Privacy and Electronic Communications (2002/58/EC) [201], which replaced the Privacy in Telecommunications Sector (97/66/EC) Directive, specifically addresses the use of location data. Article 9 of the directive states that:

> *"Where electronic communications networks are capable of processing location data other than traffic data, relating to users or subscribers of their services, these data may only be processed when they are made anonymous, or with the*

consent of the users of subscribers to the extent and for the duration necessary for the provision of a value addressed service."

This article clearly shows that the European Commission encourages 'opt-in' privacy policies [20] by giving users the right to express consent prior to the use of the location information. This automatically gives privacy protection by default to every user as well as a total control on which service will have access to their location information.

United States of America

While the European Union chose to address privacy concerns at an early stage of location positioning systems deployments, legislation regarding to location privacy came very late in the U.S.A. Defined in 2001, the Location Privacy Protection Act [204] proposes to regulate location information collection, use, retention or distribution and advocates "opt-in" privacy policies. The Wireless Privacy Protection Act [205] of 2003 goes even further by proposing to amend the Communications Act of 1934 with the following statement:

"To require customer consent to the provision of wireless call location information."

However, as stated in [21], the legislation is very unclear: for example, according to the law, location privacy is only considered when making voice calls. Browsing the web on a mobile phone or sending SMS messages could potentially expose users to location disclosure. This lack of regulatory guidance lead to a lot of confusion and the industry as well as individual states started to implement their own version of the Wireless Privacy Protection Act.

Japan

In Japan, a very clear legislation was put in place from the beginning and this legal framework is believed to have facilitated the development of location-based services. The "Guidelines on the Protection of Personal Data in Telecommunications Business" issued by the Japanese government in 1998 clearly states that location information cannot be disclosed by mobile operators to any other party without users' consent. In 2003, the "Personal Data Protection Law" clarifies the requirements for implementing 'opt-in' privacy policies mentioned in the 1998 guidelines.

1.4.3 Conclusion

The potential loss of privacy together with the fear of location data disclosure to unauthorized entities will undoubtedly slow down end-user adoption of *LBS*. Regulations have recently been put in place in order to restrict potential abuses related to location information gathering without user consent. However, implementing those regulations may turn out to be problematic since they guarantee total end-user privacy although enabling them to use *LBS* seems contradictory.

1.5 Problem statement

Mobile communications technologies, together with positioning and wireless Internet access technologies create tremendous opportunities for mobile operators to offer new value-added services to their users while generating revenues. Prototypes of such location-based applications have already been developed and some of them have already been deployed in Europe [152, 24, 61]. However, concerns have arisen regarding potential privacy threats that personal location information disclosure could entail [128, 31]. These concerns have been partially addressed by the development of sound regulatory frameworks in Europe, the U.S.A and Japan, see Section 1.4.2. In fact, the real issue lies in how these frameworks will be implemented. While mobile network operators may implement these location-based applications within their infrastructure, it is very likely that they will only act as location information providers to third party *LBS*, as explained earlier on. In the latter case, *LBS* end-users may be faced with a dilemma. On one hand, they must be willing to disclose a minimum of their personal location information in order to avail of *LBS*' services. On the other hand, revealing too much information may expose them to privacy and security threats.

The challenge here is therefore to be able to provide third party *LBS* with location information as accurate as possible, while maintaining a high level of privacy for their users.

1.6 Thesis Objectives

The goal of the research carried out and described within this thesis is to enable a secure and privacy preserving usage of third party location-based services by mobile phone users. More precisely, we wish to help *Targets* preserving their privacy by guaranteeing a fine-grained access control to their location information as well as the *Confidentiality* and the *Integrity* of their location data. We wish to supply *LBS* providers with reliable location information and enable *Subjects* to access the services they provide securely. Section 4.3

outlines some approaches that deal with *Subjects*' location information privacy. From this review, we identify three main problems related to location privacy:

- The location inference problem, by which a *LBS* can infer the exact location of a particular *Target* given some less precise location data retrieved from the *Mobile Operator*.
- The location path problem, by which a *LBS* can infer the location of a particular *Target* given some historical data.
- The location transfer problem, by which a *LBS* can redistribute to unauthorized entities the *Target*'s location data it retrieved from the *Mobile Operator*.

The research work presented in this thesis aims at addressing the location inference problem and, to a certain extend, the location transfer problem. In particular, while we do not address fully the location transfer problem, we provide a solution that may discourage such a practice. By offering a tailored quality of service in terms of location accuracy, our solution provides location information that may not be suitable for any other unauthorized *LBS* than for the one it was generated for.

We also intend to facilitate the development of *LBS* by providing an environment in which dealing with location information and security is made transparent to software developers. In this section, we briefly describe the outcomes of the research carried out.

1.6.1 Thesis Contributions

The expected contributions of this thesis are summarized in this Section as follows:

- We propose a design for a secure infrastructure that enables mobile phone users to avail of third party location-based services while preserving their privacy.
- We introduce the need and describe the functionalities of a software platform and a protocol that handle both user privacy, communication security and location information provision to *LBS*.
- We design a privacy engine that enables mobile phone users to provision their privacy preferences.
- We present a secure algorithm capable of degrading the quality of mobile users' location information so that it guarantees their privacy while remaining meaningful to *LBS*.
- We use elliptic curve-based server-aided cryptography in order to provide a Public Key Infrastructure that handles the security services needed within the architecture considered.

1.6.2 Thesis Limitations

The infrastructure proposed is designed to operate within the wireless mobile phone networks described in Section 2.4. Wireless networks such as WLAN and its related positioning techniques are not considered in this thesis. Also, the security of mobile phones or mobile terminals in general will not be covered and remain beyond the scope of the research carried out. Concerning location privacy, analytical attacks that involve the use of environment characteristics such as the presence of buildings, roads, direction of traffic flow, etc. to infer somebody's location, will not be considered in the design of our location privacy algorithm. However, research avenues describing potential approaches to prevent such attacks will be given in Section 8.5. Finally, we will not consider security services provided by the knowledge of location, namely location-based security services [186, 218] but instead focus on the security of location information.

1.7 Thesis Outline

The structure of this thesis is organized as follows:

Chapter 2 reviews the background necessary to understand the environment in which location-based services operate. In particular, we focus on computer security and mobile phone telecommunications.

Chapter 3 introduces the different players in the provision of *LBS* as well as their requirements in terms of security and privacy.

Chapter 4 presents the related research carried out in the field of location privacy, location privacy policies as well as location information access control. We also describe infrastructures that aim to achieve similar goals to ours.

Chapter 5 details the design of the architecture proposed. It describes in depth the software platform functionalities and the protocol used by *LBS* to access them.

Chapter 6 describes an algorithm used to modify the granularity of mobile users' location in order to guarantee their location privacy.

Chapter 7 outlines the design and implementation of the Public Key Infrastructure used in order to provide confidentiality in mobile users' communication with Location-Based Services.

Chapter 8 draws the conclusions of the work carried out and outlines the possible future research directions.

Chapter 2

Background

2.1 Introduction

In this section, we intend to provide the background necessary to understand the research work undertaken. The design of an architecture allowing for the provision of location-based web services necessitates a deep understanding of three research domains: internet computing, mobile telecommunications and computer security. Instead of covering these three research fields in depth, we will focus only on the last two and describe their overlap in the context of the Internet. We first give an overview of the building blocks of computer security and highlight some of their issues. We then present the mobile phone systems currently used and emphasize the security services they provide. Finally, we describe the standardized location positioning technologies that will enable the widespread of location-based services.

2.2 Building Blocks

Research in the field of cryptography became very active during the second world war where cryptography was used separately by Allies and Germans as a way to protect their vital information. Even though its usage proved successful in some cases, scientists also studied techniques to attack the cryptographic designs used, laying the grounds of cryptanalysis. Advances in the late 70s as well as the deployment of the Internet a decade ago enabled and fostered the need for personal secure communications. Nowadays, cryptography is considered as a non exhaustive set of techniques used to protect electronic data transfers, providing security services such as confidentiality. In this section, a brief overview of cryptography is given as well as some interesting cryptographic schemes relevant in the context of this thesis.

2.2.1 Symmetric Cryptography

Symmetric cryptography is also referred to as secret key cryptography. It enables the transformation of a meaningful message (plaintext) into an unintelligible one (ciphertext). This transformation is achieved using a secret piece of information called the cryptographic key and is called encryption while its inverse is named decryption. Both of these operations are indexed on the cryptographic key and define a bijection between the two sets of plaintexts and ciphertexts. The encryption and decryption processes combined together are usually referred to as a cryptographic algorithm.

Two entities sharing a cryptographic key can therefore securely communicate by enciphering their messages, provided that the algorithm and the key are considered secure. In 1883, Auguste Kerckhoffs stated six cipher design principles also known as the Kerckhoffs' laws [121]. Two of them are worth mentioning since security designs that do not take them into account are usually easily broken (see Section 2.4.2 for an example). The first principle deals with openness as opposed to security by obscurity. The design of cryptosystems should be made publicly available to researchers so that it can be studied and declared secure after enough scientific review. The only element that should be kept secret is the secret key. The second principle states that the key space should be large enough to prevent a brute force attack (exhaustive search of the key) but that it should not be considered as a sufficient condition to guarantee the security of a cryptosystem. Currently, the lower bound estimate for symmetric cryptographic key size is 128 bits of key material [130]. In theory, any sequence of bits that is long enough can be considered as a symmetric key. Therefore, the key generation process can remain as simple as a random number generation.

Nowadays, the most popular block cipher is the Advanced Encryption Standard (AES) [156]. It was chosen by the National Institute of Standard and Technology (NIST) in 2000 to be the successor of DES, which was later withdrawn from the FIPS standards for not being adequate enough to protect federal government information [157].

2.2.2 Asymmetric Cryptography

Asymmetric cryptography was discovered in the 1960s by an engineer and mathematician from GCHQ, James Ellis, who provided the proof of the possibility of non secret encryption. In 1976, this concept was officially discovered and published for the first time by Whitfield Diffie and Martin Hellman, who described a protocol enabling key exchange over an insecure communication channel [67]. More widely known as public key cryptography, it proposes an alternative to secret key cryptography. It solves the key exchange problem by allowing the use of a publicly available encryption key as well as a private decryption key, both of them

bound to any individual willing to conduct secure communications.

The key pair cannot be chosen at random as for a symmetric cryptographic key: the asymmetry property of a public key encryption scheme requires them to be mathematically related but not deducible from each other. The difficulty of recovering the private key from the public key relies on the widely believed difficulty of solving a hard problem. Examples of well known hard problems are as follows :

- Integer factorization for large enough integers is considered to be computationally difficult: if the prime factors of a large number n are known, it is easy to compute n. However, factoring n can be computationally infeasible for a large enough n.

- The Discrete Logarithm Problem (DLP) is defined as follows:

  > *Given an element g in a finite group G of order n and another element $h \in G$, find an integer x where x is in the range of $[0, n-1]$ such that $g^x = h$, provided that such an integer exists.*

  While computing discrete exponentiation is easy, solving the discrete logarithm problem is considered as difficult in some well chosen groups.

In other words, the security of public key cryptographic algorithms relies on a trap door function : a pseudo one-way function only invertible with a secret. This function enables the key generation process but makes it computationally infeasible to recover the private key from the public key. The size of the cryptographic keys used to guarantee a good level of security does not depend anymore on the ability of an attacker to perform an exhaustive search in the private key space. Indeed, since the key pair is mathematically generated, it is more relevant to attack its mathematical structure. As a result, depending on the algorithm considered and the hard mathematical problem involved, the acceptable key size will be different from one public key cryptosystem to another. Two main public key cryptosystems standards are currently used and have been studied for more than 20 years. We give a brief description of each of them as follows.

RSA. The RSA cryptosystem [177], named after its inventors Ron Rivest, Adi Shamir and Len Adleman, relies on the difficulty of computing the integer factorization for large integers. An interesting property of the RSA cryptosystem appears when swapping the public and the private key parameters in its algorithm. The RSA encryption algorithm turns into a new cryptographic primitive known as a digital signature scheme that can provide services like message integrity and non-repudiation. Encrypting a token such as the hash of a message using one's private key enables anybody having access to the corresponding

public key not only to check for the authenticity of the sender but also for the integrity of the associated message. The recommended minimum RSA key size is 1024 bits. This provides a comfortable level of security provided factorization is the only way to break RSA. For almost twenty years, the RSA cryptosystem has been subject to a certain number of attacks [145]. In his survey [41], Dan Boneh categorizes them in four different categories: attacks on low private exponents, low public exponents, on the implementation, and on misuse of the system. He concludes that most of them exploit flaws in the padding schemes used prior to the encryption process and can be avoided by following good practices.

Elliptic Curve Cryptography-based cryptosystems. Elliptic Curve Cryptography (ECC) -based cryptosystems constitute the second standard of public key cryptosystems that have undergone a thorough academic scrutiny. Independently discovered by Victor Miller and Neil Koblitz [122, 149], they rely on the difficulty of a special class of the discrete logarithm problem: the Elliptic Curve Discrete Logarithm Problem (ECDLP). Traditionally, the discrete logarithm problem can be stated as in Section 2.2.2. This definition is applicable to any group. However, the DLP problem in some groups may be harder to solve than in others. Furthermore, computations on the group elements can be made easier depending once again on the group considered. Therefore, finding a group of mathematical objects making the DLP problem the hardest as possible while making sure operations on the elements of this group can be made quick enough is a key to design a cryptosystem based on such a problem.

In the case of ECC, the group in question is defined as an abelian group G of n elements, known as points on an elliptic curve. An elliptic curve is a two-dimensional mathematical structure defined over any finite field $\mathbb{F}$. Thus, a point of an elliptic curve is defined by the two coordinates x and y taken from a finite field, for example a Galois field with a size of a power of 2. $GF(2^k)$ fields make it easy for computers to perform arithmetic since field elements can be represented as polynomials of a degree less than k, with coefficients in $\mathbb{F}_2$ [103]. In other words, x and y are bit strings, which is an efficient data representation for computer processing. The analogue problem of the DLP in ECC can now be defined. The Elliptic Curve Discrete Logarithm Problem (ECDLP) states that:

> *Given an elliptic curve E defined over a finite field $\mathbb{F}_p$ and two points P and Q, where P has order n, find an integer k where k is in the range of $[0, n-1]$ satisfying $Q = k.P$, provided that such an integer exists.*

In ECC, finding a suitable group where the DLP is hard while arithmetic operations on elements remains efficient is therefore a matter of finding the right elliptic curve. To

guarantee the hardness of the ECDLP, one requirement is that the number of elements of the curve is divisible by a large prime number in order to avoid the Pohlig-Hellman attack that reduces the ECDLP to instances of the same problem in subgroups of prime order [170]. For the same group size however, ECC cryptosystems are generally slower than the RSA cryptosystem. This is mainly due to the execution of the time consuming scalar multiplication used in the underlying field computations. Yet, ECDLP is believed to be significantly harder than DLP or the factorization problem. Therefore, cryptosystems based on ECDLP can achieve the same level of security as others by using a group with a smaller size. This means that the cryptographic keys used by ECC cryptosystems can be significantly shortened while providing equivalent security in terms of the time to break the cryptosystem. Therefore, the recommended minimum ECC key size is 163 bits. Attacks on ECC-based cryptosystems focus on their theory rather than on their actual implementation, mostly because ECC cryptosystems have not been used as much as RSA in the industry. More details about these attacks can be found in [104, 47, 194, 143, 90].

2.3 Public Key Infrastructure

In the real world, individuals are usually identified by their names even though the date and place of birth are sometimes needed to resolve any hypothetic collision. An identity card or a driving license certifying one's identity can be used to authenticate one to toll stations when crossing borders. These documents are usually issued by trusted government administrations and usually bind someone's name with a picture of herself. In the digital world, checking the identity of an individual is not an easy task since physical verification is not possible. Instead, individuals are given a cryptographic key pair that makes them identifiable when conducting online communications, at least in theory. While it is easy to remember someone's identity in the real world, it is much more difficult to associate somebody you know with a string of bits. Rather than trying to use directly a public key to identify somebody, a mapping binding a user's identity with her public key, signed by a trusted third party's private key is used: this is called a digital certificate. Issuing, managing or revoking certificates necessitate a lot of work and organization. As in the real world, a trusted entity following a set of policies is needed. In this section, three different Public Key Infrastructures (PKI) are presented. These are frameworks that include protocols and services used to manage identities and their associated key pairs. The PGP, SPKI/SDSI and X.509 approaches are highlighted as well as their pitfalls and an overview of recent advances and possible solutions is also given.

2.3.1 PGP

The Pretty Good Privacy (PGP) system [89] was created in 1991 by Phil Zimmermann and developed in various versions through out the years by many developers both from the corporate and open source world. PGP is mainly used to secure email exchanges, guaranteeing confidentiality, integrity and non-repudiation. It implements a hybrid cryptosystem where RSA or ElGamal is used for key agreement and IDEA, TripleDES and of course AES can be used as the symmetric block cipher. The principles of the PGP PKI lie in a concept called the "web of trust". There is no central authority in charge of managing trust: this is up to the users of the system to validate each others public keys. Every user generates her key pair and publishes the public key on a key server, on a web page or even includes it in an email. However such a key has no value since anybody could have impersonated an individual by creating and publishing the key under her name. To establish trust, a user's public key needs to be signed by a person that can state that the key actually belongs to this user. The public key, together with a name, an email address, a creation date and a list of digital signatures forms a PGP certificate. Every user is free to sign any certificate and to share it with anybody. When it comes to trust a particular public key, users have therefore the choice to follow one path on the "web of trust" in order to validate the ownership of the public key. Public key revocation is handled by end-users as well: a revocation certificate is sent by a user to his/her correspondents and to key servers. PGP is therefore well suited for informal and personal networks but not really for large organizations or the world wide web.

2.3.2 SPKI/SDSI

SDSI

The Simple Distributed Security Infrastructure [179] (SDSI) was designed in 1996 and, as with PGP, no central authority is required to issue certificates: everybody is free to do so. However, it improves on PGP by introducing the notion of a local name space. Each user of the system has its own name space and uses local names to refer to other entities. A local name is a public key together with an identifier, which is typically a common name or a nickname. The local name is only meaningful to its issuer and is bound to other public keys in the subject of a name certificate. Name certificates can be represented by a four-tuple structure and linked between each other through local names.

SPKI

At around the same time, the SPKI [215] project was developed following a predefined set of requirements. Its aim was to design an easy way to perform authorization rather than authentication through certificates. Like for PGP, users would be free to generate certificates and delegate access rights to other usersfor a predefined time interval clearly specified in the certificate. In order to avoid using complex naming mechanisms, the SPKI approach would give public keys a central role: access rights would be granted to a key rather than to an individual. The system would therefore be used completely without names. SPKI certificates would bind the issuer's public key with a subject and mention any relevant detail about authorizations and their delegation.

SPKI/SDSI

In 1997, the two research efforts merged to form the SPKI/SDSI public key infrastructure [75, 54]. The SPKI/SDSI approach took the best ideas from both worlds to provide a powerful PKI. The system implements the two types of certificates:

- Name certificates similar to SDSIs that bind local names to public keys. The SDSI naming convention is used to provide SPKI with a mechanism for getting public keys according to local names. Any certificate chain containing a certificate can be searched and if one path is not trusted, another one can be chosen.

- Authorization certificates like in SPKI. The system implements the SPKI delegation of rights mechanism. By issuing a certificate, a user grants another entity some or all her rights on a resource and these rights can be transferred depending on the issuer's requirements. Delegation of rights can therefore be seen as a way to share private key usage without disclosing the actual private key to anybody.

Revocation

The SPKI/SDSI philosophy states that there is no certificate revocation problem since SPKI/SDSI certificates are not meant to be revoked. As stated in [178], "This certificate is good until the expiration date. Period". As a consequence, SPKI/SDSI advocates that signers present separate evidence that the key pair used has not been compromised. In order to enable such a process, a separate infrastructure is proposed: a network of servers called "suicide bureaus" (SB) is set up on a high speed network. When users generate their key pair, they digitally sign a document known as a "suicide note" and keep it secret. They also register their public key with the SB network. If users are required to produce a proof

of validity for their key pair, they can request a "certificate of health" from a SB. This is a dated document stating there is no evidence that a private key has been lost or compromised. If their key ever gets compromised, they issue the suicide note to a SB that will broadcast it to the SB network which will stop issuing "certificates of health" for this public key.

2.3.3 X.509

The X.509 PKI is the most popular PKI used to carry out secure transactions over the Web. The X.509 original approach defines identity-based certificates binding a public key to a name. Unlike the previous approaches, these certificates are generated by a central authority and are distributed through online directories.

Description of the PKI

The early design of the X.509 PKI used the X.500 directory structure [219] to provide naming services to its end-users. However, the X.500 standard itself was never adopted by the Internet community because it was seen as far too complex to use. The X.509 PKI consists of the following entities:

- Certification Authorities (CA). The main role of CAs is to issue signed X.509 certificates. They are generally organized in a hierarchical structure whose root is called a root CA. This top level certification authority is a trusted entity that is in charge of propagating trust through other CAs using digital signatures. As opposed to normal CAs, the root CA can be off-line so that a high level of security is maintained. Every CA holds a public key with its associated certificate. The root CA certifies the public key of the CA that is directly below it in the CA hierarchy and so on until the end user's certificate is reached. Its certificate is self-signed, which means that it can only be used to check for the integrity of its content.

- Registration Authorities (RA). This is a component used as an interface between the certificate authority and the end-users. It is in charge of identifying and authenticating certificate requestors before the corresponding CA issues them a digital certificate. Depending on the nature of the certificate requested (self-signed, user, server or business certificates), RAs use different practices and procedures to establish the identity of certificate requestors [52]. Identity verification can be performed online without providing any credential or using a more secure "out-of-band" channel, comparing registration details with real identity papers such as passports of the requestors [22].

- One or more directories that store X.509 certificates.

Even though the X.509 Public Key Infrastructure primarily adopted the X.500 hierarchical structure, it also evolved in a networked structure enabling more flexible bilateral relationships. Cross-certificates are therefore used in order to bypass a hierarchy structure that would be too awkward to follow.

Certificate Validation

Every time a digital signature needs to be checked or an encrypted communication needs to be set up, the certificate validation process takes or at least should take place in order to be sure that a valid certificate is being used. Certificate validation can be described as a 3 step process.

- The certificate path discovery: this involves building a path of certificates from the certificate to be checked to a CA certificate that is considered trustworthy. This is usually carried out on the end-user side. Most of the time, the end-user already trusts the CA that issued the certificate and in this case, the certificate path is already built. However, when the PKI topology becomes complex, the burden of discovering the certificate path is usually left to the end-user in charge of validating the digital signature. [135] explores more in depth the issues that arise when certificate path discovery is performed in environments where cross certification between heterogeneous PKIs is involved.

- The certificate path verification: this involves checking the integrity of each certificates by validating the digital signatures of each of them in the path.

- The certificate status checking: this involves making sure that each certificate in the path is valid. To be considered as valid, a certificate should be used within its validity period (i.e should not have expired) and should not be revoked. Certificate revocation is a process by which an entity can decide to prematurely end the lifetime of a certificate and therefore invalidate the binding between the public key and the identity it contains. As stated in [80, 84], several reasons can lead to the revocation of a certificate, in particular:

    - Private key compromise or its suspicion: if a user's or CA's private key has been compromised, the binding between its identity and public key should no longer be trusted and the corresponding certificate should be revoked as soon as possible.
    - Change of affiliation: It occurs when any information mentioned in the certificate is no longer valid.

- Cessation of operation: A CA may cease to operate or a user may no longer need the certificate issued.

The most common and standardized method to perform certificate revocation is to use Certificate Revocation Lists (CRL) [114]. A CRL is a document digitally signed by every CA that implements the standard: it contains the list of all the serial numbers of revoked but not yet expired certificates issued by the CA. CRLs are periodically published by CAs and stored in the directory with the certificates issued by the same CA. They can be located using an extension field called CRL Distribution Point (CDP) [114] in the X.509 certificate. It contains a X.500 or LDAP directory entry or even a URL indicating the location of the CRL.

Problems

In [76], Ellison and Schneier conduct a critical analysis on PKIs and highlight their major flaws. In particular, they emphasize the fact that PKIs involve many components controlled by different entities for which different levels of trust are applied. Thus, they question the relative authority of CAs and RAs that may issue certificates to people that are not trustworthy but whose identity will not be verified. They also point out that the end user has not been taken into account when PKIs were designed: most PKIs assume that a private key holder is the end-user, forgetting that a private key is primarily manipulated by a computer program on a platform that may not be considered secure [188]. Ellison and Schneier's analysis also critics the naming conventions used and highlights the fact that name collisions are likely to happen, which of course defeats the purpose of such an infrastructure. In [134], the authors point out some more practical problems related to the X.509 PKI such as the certificate processing complexity, certificate costs as well as cross domain trust management involving translation of security policies.

However, the PKI issue that has received the most attention from the academic community is with no doubt the certificate revocation problem. The widely used CRL revocation system suffers from several major flaws [178], such as its size, the variable periodicity of its update and its lack of actual use by applications. Indeed, locating a CRL for a particular certificate involves checking the value of the X.509 extension field CRL Distribution Point present in the certificate. Most applications do not perform this check by default and need to be configured to do so. Moreover, even though [114] recommends support for this extension by CAs and applications, most certificate issuers do not bother filling the field (see [211] for an example). In order to try to overcome the problems listed above, several approaches have been studied [216, 217]. In [222], these methods are sorted into four different categories. We

briefly describe the most important here.

List-based Schemes. Delta CRLs were a first attempt to solve the size problem of CRLs by supplying users with a base CRL and some more lightweight delta CRLs that include only the serial numbers of the certificates that have been revoked since the base CRL was issued. Segmented CRLs and over-issued CRLs are also discussed in [216] but do not quite provide a realistic solution to CRLs problems.

Certificate Revocation status-based schemes. The first real innovation came from Micali in 1994 with his Certificate Revocation Status system (CRS), later refined and renamed Novomodo [147, 148]. It aims at getting rid of the burden of CRL checking as well as getting a more complete answer regarding the status of the certificate. It is based on hash chains and allows for a single certificate status checking at a time. The verifier is only left with the verification of this hash function-based lightweight signature and obtains a complete and satisfactory answer.

Tree-based Schemes. This approach was suggested by Kocher [124] and is based on Merkle hash trees [146]. It aims at saving time bandwidth and processing power by avoiding the downloading of a full CRL. The serial numbers of revoked certificates are stored in the leaves of the Certificate Revocation Tree (CRT) and its root is signed by the CA. When a certificate status is requested, a short proof of validity is sent to the verifier by the CRT issuer, which is the entity in charge of running the system on several CAs. The main drawback of this approach lies in the computational effort to carry out an update on the tree: a complete re-computation of the whole tree may be necessary.

Verifier transparent schemes. The strategy here is to off load time and resource consuming operations to a third party that carries out some operations on behalf of the end-user and/or CAs.

The Online Certificate Status Protocol (OCSP) [154] is a protocol used to request the status of a certificate online and therefore provides potentially more up-to-date information than CRLs. However, end-user clients encounter the same problem as in the CRL system: they must be able to locate the right OCSP server and even if its location can be found in some certificates, it is likely that it is not mentioned in most of them. Verifying a certificate chain can also become problematic, especially when each certificate has been generated by different CAs depending on different OCSP servers. In order to overcome this problem, an architecture has been designed in [74] to enable OCSP servers to work as part of a network

and provide OCSP clients with relevant responses even though the OCSP server contacted does not hold the information locally.

Other alternative certificate validation schemes are analyzed in [44]. Their main idea is to delegate as many operations as possible to a third party online server. The Delegated Path Discovery (DPD) [169] approach delegates the task of discovering the certificate chain to a server given a target certificate specified by the client. The DPD server does not need to be trusted since it only collects CA generated information and forwards it to the client. The Delegated Path Validation (DPV) [169] approach extends the idea of the delegation of tasks even further by delegating all the burden of certificate path validation as well as certificate path discovery to an online server. As opposed to DPD, DPV does not provide the client with CA-signed proofs that the information requested is trustworthy. Therefore the online DPV server used must be fully trusted and the connection between the server and the client must be secure. The authors of [44] provide also a detailed description of SCVP [140] and XKMS [83], two protocols implementing the concept of DPV.

2.3.4 Identity-Based Encryption and Applications

In 1984, Shamir first came up with the concept of Identity-Based Encryption (IBE), a public key cryptosystem that would not require users to exchange any public key or public key certificate [191]. Instead, users would use any string and in particular their identity as a public key in order to solve key authenticity problems without using PKIs. The approach required however a trusted key generation center to derive the private key corresponding to their identity. The private key then had to be retrieved through a secure channel by the end-user. Identity-Based Encryption remained a concept until Boneh and Franklin showed in 2001 that bilinear pairings on elliptic curves could be efficiently used to implement the first fully functional and secure IBE scheme [43]. Their design suffers however from two major problems. Identity revocation is awkward to perform since revoking one's email address, for instance, prevents anybody else from using it to communicate with that person. A solution to this problem was proposed by the same authors, and consisted in including a preset expiration date in the identity in order to produce ephemeral public keys. For example, "receiver-public-key || expiration-date" where "expiration-date" represents the current day, month or year depending on how long the key pair should be valid for, could be used as the recipient's public key. This solution is however not very flexible as it forces the recipient of the message to obtain a new private key within the timeframe of the "expiration-date". Another problem inherent to their design is key escrow, where one's private key is generated by the trusted key generation center. Nevertheless, their IBE system remains a breakthrough

in the field of pairing-based cryptography and fostered with no doubt research efforts in the area.

At the same time, Boneh *et al.* developed the concept of an on-line semi-trusted mediator (SEM) [42]. A SEM is a server that carries out cryptographic operations in conjunction with the end-user in order to provide a system that overcomes the certificate revocation problem. Based on threshold cryptosystems [91], this solution exploits a modified version of RSA called mediated RSA (mRSA) whose private key is split into two parts: one used by the end-user, the other one by the SEM. One of the main advantage of this approach is that it enables immediate revocation of security capabilities. In other words, the SEM checks whether the public key used has been revoked and if so, prevents immediately any entity from using the corresponding private key for decryption and/or digital signature generation by not completing the threshold cryptographic operations required. Therefore certificate validation no longer needs to be carried out by the end-user .

Soon after, Ding and Tsudik proposed a variant of the SEM-based security architecture that uses a modified version of mRSA based on IBE concepts: IBE-mRSA [68]. The approach eliminates the need for certificates but relies on the strong assumption that no user will ever be able to compromise the SEM, which would result in the total break of the system. Furthermore, Libert and Quisquater uncovered a flaw in the security proof provided, showing that the security of the system against inside attacks is no longer guaranteed [132]. They also proposed a mediated ID-based encryption scheme based on bilinear pairings. However, even though it improves on Ding and Tsudik's results, their system is still not secure against insiders that possess a user's private key share and conduct chosen ciphertext attacks. Recently, Baek and Zheng solved this problem by designing a scheme that is secure against such attacks [29]. Their work finally provides IBE with fine grained revocation but still does not solve the key escrow problem.

2.4 Mobile Phone Systems and their Security

The first wireless telephone services were deployed in the late nineteen forties in Saint Louis, USA. A team of engineers from Bell Labs had designed and made available the first wireless network to thousands of users [28]. The system could only handle three subscribers at a time in the same city and was only half duplex. Since then, advances in the field of mobile telecommunications have led to the deployment of highly sophisticated third generation mobile phone architectures throughout Europe. In this section, we will explain the main principles of mobile telephony, detailing the current architectures in place. Then, we will study the mechanisms used to enable Internet access through mobile phone systems with a

particular focus on their security.

2.4.1 Global System for Mobile Communications (GSM)

The GSM project [30, 99] started in 1982 at the European Post and Telecommunications conference (CEPT) where a proposal for a European cellular system was first discussed. The aim was to design a system that would replace the incompatible wireless mobile systems already in place in Europe. A study group called "Groupe Special Mobile" was formed with the objective to specify the GSM system. It would be cellular-based[1], digital and would primarily be designed for voice transmission. The first tests were carried out in the early nineties and in 1992, the first GSM cellular network was introduced to the public by France Telecom [85], with its Itineris network. The system was then renamed the Global System for Mobile Communications and became the most popular mobile phone system in the world.

Description

The GSM system is a revolution compared to previous systems. From the mobile operator point of view, the GSM network can handle even more mobile users due to a new technique called Time Division Multiple Access (TDMA) [117] and that replaces the frequency-division multiple-access (FDMA) system used in 1G wireless networks. The frequency dedicated to each mobile user is now divided into time slots which increases the carrying capacity of the network by enabling multiple mobile users to access it at the same time [55]. From the mobile users' point of view, the digital design provides them with a lot of new features, such as a superior speech quality achieved by using digital audio encoding or extended battery lifetime due to reduced power consumption. New services are available, such as SMS [15] or call waiting/forwarding services and the security of communications is now guaranteed [185]. Finally, it provides its users with a new international roaming capability, regarded as one of the key strengths of the whole system.

Architecture

The architecture of the GSM system can be split into three different components; see Figure 2.1.

- The Mobile Station (MS). This is the mobile handset used by a mobile user. It contains the Subscriber Identity Module (SIM card) which identifies a mobile user uniquely

[1]The wireless network is divided into multiple geographical areas also called cells in order to enable frequency reuse (in non adjacent cells).

through the use of the International Mobile Subscriber Identity number (IMSI number). It is also composed of the Mobile Equipment (ME), in charge of the digital audio signal processing and radio transceiving. The ME is identified by the International Mobile Equipment Identity (IMEI).

- The Base Station Subsystem (BSS), composed of two entities. The Base Station Transceiver (BST) is the radio transceiver relaying mobile users calls. Its radio coverage area is called a cell, which measures typically from 250m wide for high density urban areas to 30km wide for rural areas. The Base Station Controller (BSC) is a small switch that handles a finite number of BST and links them to the switching system component. The area covered by the BSC is called Location Area and its identifier Location Area Identity (LAI). In some cases, a LA may be covered by multiple BSCs. The BSS is also called the radio access network.

- The Network Subsystem (NSS). This is the component used to set up and maintain calls made over the network. A Mobile Switching Center (MSC) usually parents a number of BSCs. The area controlled by the MSC is the sum of all the LAIs of its BSCs, as further described in Section 2.5.3. It uses a distributed database system to manage mobile users' mobility. A Visitor Location Register (VLR) associated with each MSC stores information about mobile stations currently served by the MSC. The Home Location Register (HLR) carries IMSIs, service subscription information, location information (the identity of the currently serving Visitor Location Register (VLR) to enable the routing of mobile-terminated calls), service restrictions and supplementary services. It is generally associated with an Authentication Center (AuC) handling the authentication phase at the beginning of a connection. An Equipment Identity Register maintains a list of IMEI as well as their rights on the network. The switching system also provides interfaces to PSTN networks as well as TCP/IP networks.

Security

First generation wireless networks suffered from eavesdropping attacks and mobile phone cloning since the identity of the phone as well as the communications were transmitted in plaintext over the radio link [65, 109]. Therefore, there was a need to secure the radio link of cellular mobile networks. The aim was basically to make GSM communications at least as secure as ordinary PSTN ones. Thus, several security mechanisms have been implemented in the GSM system to ensure the authenticated anonymity of mobile users and the confidentiality of their communications as well as the protection of mobile network operators from billing frauds.

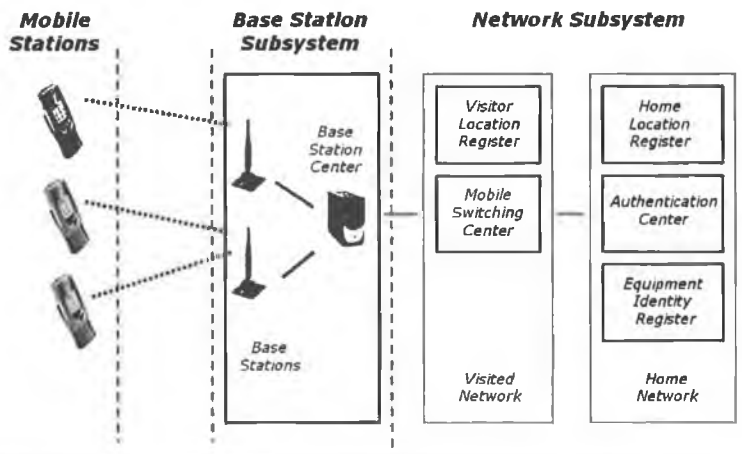


Figure 2.1: Architecture of GSM networks.

Anonymity. Anonymity of mobile users is provided through the use of a temporary pseudonym [172]. The MSC needs to identify the Mobile Station by its IMSI number in order to establish a connection between the Mobile Station and the Network Subsystem. However, sending this number as a plaintext over the network could reveal information about the Mobile Station. To ensure anonymity, the IMSI will be sent only once, when the Mobile Station is turned on for the first time. From then, a Temporary Mobile Subscriber Identity (TMSI) number is used instead and updated frequently depending on time and location of the subscriber [35]. The TMSI number is stored in non volatile memory so that it can be reused if the phone is switched off and switched back on.

Authentication. It occurs between the Mobile Station and the Network Subsystem. The Mobile Station contains a SIM card. It is a tamper resistant device also known as a smart card [174]. The SIM card stores temporary personal data such as mobile phone numbers or SMS text messages. It also stores permanent security related information such as the IMSI number, two proprietary security algorithms (A3,A8), a 128 bits user Key K_I and a PIN (Personal Identification number). The authentication mechanism follows a challenge/response protocol initiated by the Network Subsystem. It is extensively described in [98].

Confidentiality. Once the authentication step is successfully achieved, both the Mobile Station and the Network Subsystem generate an enciphering key K_C aimed at encrypting communications on the fly. K_C is a 64 bit key, is generated by the A8 algorithm and is also an operator dependent one way function. A8 and A3 are usually implemented in a single one way function. An example of such a function is the GSM Association's [99] COMP128, which

is implemented both on the AuC and on the SIM card of each subscriber's Mobile Station. The use of A3 and A8 as part of the authentication protocol constitutes an operator option. As a result, operators can specify and implement their own authentication algorithms. In practice, most operators do not wish to do so and usually use example implementations from either 3GPP [199] or the GSM Association [99]. In the latter case though, the COMP128 implementation is only made available to qualified industry parties. In order to allow for mobile phone users to roam without operators having to reveal authentication algorithms and K_I keys, triplets containing common parameters can be exchanged between different networks. Once K_C has been agreed between the two parties, communications between the Mobile Station and the Base Stations can be encrypted using a stream cipher A5. A5 is implemented on the Mobile Equipment part of the Mobile Station because it has to encrypt and decrypt data on the fly. It is also implemented on Base Stations Transceivers.

Discussion. The GSM security is however limited by a number of factors [181]. First and foremost, the security design only provides access security in the sense that information is only protected between the Mobile Station and its related Base Station. Signaling and communications can therefore be intercepted in clear within the fixed network as well as tokens like triplets potentially exchanged between network operators that can be reused to perform replay attacks. The authentication design lets Mobile Stations be authenticated by the network but does not allow them to authenticate it. An active attack can therefore be conducted by impersonating network elements. An example of such Man-In-the-Middle attack is presented in [35]. Algorithms used for both authentication and confidentiality represent another source of security weaknesses. Indeed, their design were not made available to public scrutiny and proved to be insecure after further investigations. Their implementation suffered from security flaws as well. Indeed, since 1998, a series of attacks has seriously hampered the credibility of the GSM security, see [214, 123]. The stream cipher A5 has also been subject to various attacks. Due to import/export restrictions on encryption technologies, several versions of A5 are available. A5/0 is a version of A5 used by countries under UN sanctions. A5/1 is used in Europe while A5/2 is used in Asia. Even though its design was never published, it leaked to two computer security researchers [214] and led in 1999 to two successful attacks on A5/2 and A5/1 [167, 40, 214].

In order to address these security issues, 3GPP [199] and the GSM Association Security Group [99] developed fully open security algorithms based on international standards. In 2003, they published the specifications of A5/3 and GSM Milenage [12] in order to replace the flawed algorithms initially used. Some attacks have already been performed on A5/3, but without noticeable success [198, 39, 193]. For the time being however, only a few networks

and Mobile Stations support these new algorithms.

2.4.2 General Packet Radio Service (GPRS)

The GSM system was primarily designed for voice communications. While data transmission is possible over such a type of network, it remains limited because of its circuit switched design. In order to provide a more flexible and faster data transmission over wireless networks, ETSI [1] defined a new standard in 1997, based on a packet switched design. This approach enables GPRS to provide a bandwidth theoretically 18 times higher than GSM, which is more suitable for data transmission.

A GPRS network is usually built on top of a GSM network architecture and is viewed as an upgrade of the 2G network [150]. Such a network is referred to as a 2.5G network. Both networks work in parallel; the GSM network still provides voice services while the GPRS network handles data transmissions. In order to sort the incoming data, the Base Station Controllers (BSC) are upgraded with a new piece of hardware called Packet Control Unit (PCU), a unit that routes the data to the appropriate GSM or GPRS network. Voice communications are handled by the GSM network while data communications are forwarded to two new functional units: the Serving GPRS Support Node (SGSN) and the Gateway GPRS Support Node (GGSN). The former is in charge of delivering data packets and handling handovers when mobile users move from one cell to another. The latter is a gateway between the mobile network and the Internet and may be in charge of assigning IP addresses. To avail of GPRS new services, Mobile Stations need to be upgraded as well.

Security

Security at the radio link remains the same as for the GSM system [45]. However, the Mobile Station (MS) performs the authentication step directly with the SGSN. The same algorithms are used but their implementation follows the standards published by 3GPP [12]. Encryption is performed at a higher layer in the protocol stack (Logical Link Layer (LLC)) and uses a new stream cipher with different input/output parameters: the GPRS Encryption Algorithm (GEA3) [79, 11], similar to A5/3. The secure link is extended further back in the network to the SGSN, as opposed to the Base Station Transceiver in the GSM system. The fact that GPRS was designed to enable mobile users to access Internet services means that some components of the GPRS system are potentially subjects to the same threats as online systems [175, 48]. Like GSM, the system still does not provide end-to-end security and must use some other technologies (see Section 2.4.4) to provide this security service.

2.4.3 Universal Mobile Telecommunications System (UMTS)

In Japan, third generation (3G) wireless networks were commercially made available to mobile users in late 2001 [2]. Like GPRS networks, 3G networks are primarily designed for data communications and adopted a packet switched design. By providing a higher bandwidth than GPRS networks, they enable mobile network operators to support enhanced mobile services such as faster internet access or video services.

The Euro-Japanese version of 3G is called Universal Mobile Telecommunications System (UMTS) [16]. Like GPRS, it was designed to be an upgrade of the GSM networks already in use. In order to provide a higher bandwidth, UMTS defines a new way to handle mobile communications by using a new multiplexing technique: W-CDMA [6], which allows multiple devices to transmit on the same frequencies at the same time using sequence numbers.

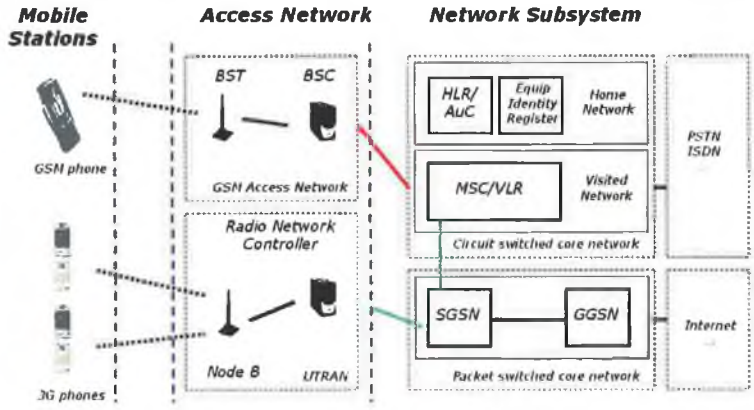


Figure 2.2: Architecture of 3G networks.

However, while UMTS still uses the same GSM/GPRS core network (mainly switches and registers), the radio access network cannot be recycled and needs to be upgraded in order to support W-CDMA [10] (see Figure 2.2 for a more detailed description of the architecture). This 3G radio access network, also known as UTRAN (UMTS Terrestrial Radio Access Network) [16], requires more Base Station Transceivers (BST) than before. Furthermore, due to the new multiplexing technique, BST sites that were considered optimum prior may not be optimum sites for 3G Base Stations. All this resulted in a relatively slow and costly 3G network implementation by mobile network operators. In order to provide similar services but at a lower quality of service, GPRS was implemented as an interim measure.

Security

Third generation network security is an evolution of GSM security [14]. The new security design aims at correcting the GSM security weaknesses as well as adding new security features required by new 3G services and new architectural network design. While anonymity is ensured by the same mechanisms, authentication and confidentiality have been slightly redesigned in order to meet new security requirements.

Authentication. One of the main problem with GSM authentication is that it is only one way. The Mobile Station authenticates to the Network Subsystem but has no proof it is dealing with a valid network . To address this issue, several upgrades have been made to the GSM authentication scheme to provide mutual authentication of the two parties. Two new sets of functions have been introduced in order to replace the A3/A8 and A5 algorithms. 3G authentication steps are similar to GSM authentication ones [172] but involve different parameters. In order to prevent replay attacks, the freshness of the authentication parameters is also verified by the USIM using a sequence number.

Confidentiality. The 3G security takes advantage of the lessons learnt from the GSM security flaws and uses open design and published algorithms. Confidentiality is now ensured from the Mobile Station to the switching center and uses longer keys to avoid potential brute force attacks. The new ciphering algorithm F8 [17] replaces the A5 algorithm from the GSM system and is used to encrypt both signaling and data messages. F9 [17] also guarantees the integrity of the signaling messages. Both F8 and F9 are protected against reuse of their output by using parameters such as time, identity direction of the communication and random numbers.

Network security. A significant effort has been carried out in order to secure the Network Subsystem [35]. Indeed, while confidentiality was only available between the Mobile Station and the Base Station Transceiver in the GSM system, it has been extended further to the switch (MSC). Mechanisms were also included to support security between different network operators: a Public Key Infrastructure enables networks to securely exchange shared session keys used to protect information shared as part of the roaming agreements.

2.4.4 Internet Access over Mobile Phone Networks

This section aims to present the evolution of mobile phone Internet access. In particular, it introduces the WAP technology along with a description of its security as a first attempt to provide Internet services to 2G mobile stations. It then discusses an Internet Access

protocol known as I-mode, a NTT DoCoMo's proprietary technology, that provided first the possibility to browse the Internet from a phone over a packet-switched network.

Wireless Application Protocol (WAP)

The need for mobile phone Internet access arose by the end of the nineteen nineties, pushed by the promising success of m-commerce as well as its wired equivalent e-commerce. In order to tackle the daunting challenge of enabling wireless internet access on 2G Mobile Stations, industry partners gathered in a joint effort and founded a democratic consortium known as the WAP forum [5]. The group merged later in 2002 with the Open Mobile Architecture Initiative to form the Open Mobile Alliance (OMA) [161]. In 2001, they released the specifications of the Wireless Application Protocol (WAP), which defines an open standard architecture as well as a set of protocols for the implementation of wireless Internet access.

Description. The WAP architecture is designed to enable communication between mobile terminals and network servers and can operate over any wireless network. It defines a new protocol stack [5] based on the TCP/IP architecture. However, in order to suit mobile devices needs, WAP comes with some optimizations and enhancements:

- Its transport layer combined with its transaction support provides enhanced packet handling compared to TCP and UDP.
- Content is provided to mobile clients through a new binary encoded markup language (WML) based on XML. Smaller than HTML with only 35 tags, it provides mobile devices with display limitations with simple and lightweight content.
- A gateway in charge of translating Web-based protocols to/from WAP-based ones. It is also in charge of DNS lookup and caching. Some gateways can act as transcoding proxies [38] by translating HTML into WML, on the fly.
- New services are provided to mobile users such as Push services and Wireless Telephony Application (WTA).

While a HTTP client receives the HTML code upon HTTP request, a WAP client receives a compiled byte stream instead of a WML page. Indeed, when a WAP client requests a WML page through the WAP protocol, the gateway fetches the WML page from the corresponding remote server on her behalf and converts it into WML bytecode before sending it back to her. The size of the compiled stream received is smaller than the normal WML content and it requires less than half of the packets needed to transport information in the equivalent TCP/IP communication.

Security Protocol. WAP security is based on the specifications of TLS 1.0 [66], but is adapted to the wireless environment. Known as the Wireless Transport Layer Security (WTLS) [9], the protocol supports both client and server authentication, key exchange, confidentiality and integrity. It differentiates itself from TLS by supporting datagrams, since WDP or UDP can be used as the transport layer. It provides a key refresh for long lived connections obtained through an optimized handshake that provides certificate information used to easily retrieve the corresponding certificate. Finally, WTLS uses a more compact certificate known as the WTLS Certificate that supports RSA as well as ECC cryptosystems [144]. However, all the changes made to TLS in order to simplify it resulted in introducing some new security flaws. A detailed description of the known attacks on WTLS that include exploits on both symmetric and asymmetric weak keys and Man-In-The-Middle attacks, is given in [183]. On top of all these security vulnerabilities, no end-to-end secure connection between the mobile client and the content provider can be established since the WAP Gateway sits in between the two entities [53]. Translating WTLS into TLS requires the WAP Gateway to decrypt and encrypt information to be forwarded to the content provider. This "WAP gap" (see Figure 2.3) makes the WAP Gateway the weakest link in the data path.

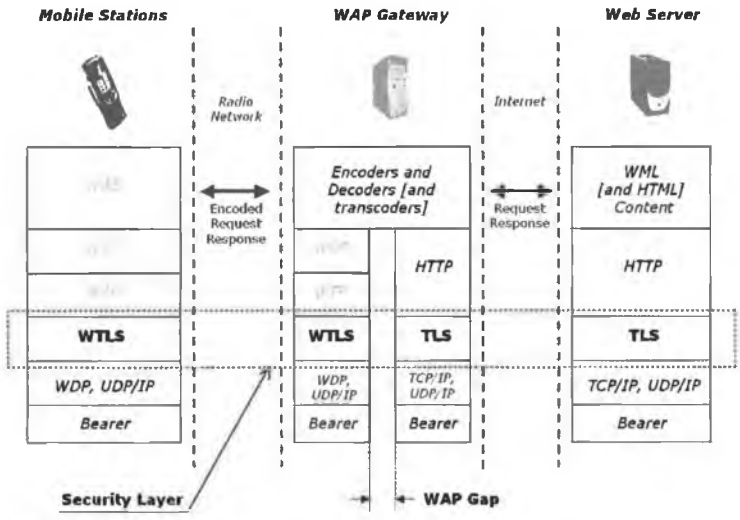


Figure 2.3: The WAP Architecture.

Wireless Public Key Infrastructure. In order to meet the security requirements of m-commerce and provide end-to-end security between a mobile client and a content provider, a wireless PKI (WPKI) [208] tailored to mobile devices was designed. This PKI is not entirely "wireless" and in fact the fundamental ideas behind this PKI are the same as for

a traditional PKI. However some enhancements and optimizations have been implemented to take into account the low processing power of mobile devices. The architecture of WPKI [220] is very similar to the IETF PKIX [114]. It requires the same components and the main difference between the two architectures is the Registration Authority that interoperates between mobile devices and traditional CAs. It is implemented as a network server and is referred to as the PKI portal. Mobile devices now carry a new tamper resistant computer chip [173], the WAP Identity Module (WIM) [206], along with the SIM card. Like a SIM card, it performs cryptographic operations at the transport layer (WTLS) but the WIM module supports public key technologies, and provides enhanced authentication and key agreement mechanisms. Furthermore, WIM modules provide application layer security by allowing mobile terminals to digitally sign WAP transactions using a set of cryptographic libraries known as WMLScript [207]. Users' private keys as well as a set of PKI root certificates are also stored on the WIM module.

Enhancements. WPKI introduces optimizations in PKI protocols, in cryptographic algorithms and keys as well as in certificate formats [182]. In particular, a public key cryptographic algorithm based on Elliptic curves [63] can be used in order to perform digital signature operations as well as the TLS handshake [66]. Also, a new server certificate format known as the WTLS certificate has emerged in order to minimize the amount of work required for transfer, processing and storage of public keys on mobile devices. It is used only for server authentication and may be sent over the air via the WAP protocol. Client certificates remain X.509v3 certificates [114] but these are not meant to be sent over the radio network. Instead mobile clients provide the server with the URL of their X.509v3 certificate so that the latter can retrieve it.

Limitations. The WPKI approach tends to reduce the amount of computation on the mobile device as well as the quantity of memory and bandwidth used. However, it does not introduce any new solution to solve the traditional PKI problem: certificate revocation. Indeed, CRLs or OCSP approaches have not been specified yet in the WPKI standard [63, 102]. To overcome the problem, short-lived server-side certificates are issued by CAs and can be used during typically 48 hours by mobile clients. To revoke a server's public key, a CA stops issuing a certificate for his key. Clients need only to perform a simple time check, provided they have access to a precise and reliable clock.

I-Mode

In 1999, NTTDoCoMo [3], a Japanese network operator, released I-Mode [159], a proprietary technology aimed at delivering mobile services to mobile phone users. Similar to WAP implementations, this protocol runs on Personal Digital Cellular Packet (PDC-P) networks, comparable to GPRS networks but proprietary, to establish wireless communications between mobile phones and transceivers. It then uses standard TCP/IP [196] to communicate with mobile service providers on the wired part of the network. The architecture remains quite simple [27]: a Mobile Message Packet Gateway (MMPG), part of the PDC-P network, is in charge of converting messages between the two protocols and a portal web server known as the I-mode server, hosts the different iMode sites. Software content is provided via Java applets [113], while static content is distributed using a subset of HTML 3.0, known as Compact HTML (cHTML [70]). No content translation is needed since mobile users can still view HTML but cHTML has been tailored to constrained mobile displays and therefore looks better.

I-Mode uses a packet switched network which makes it more suitable to transfer data and very attractive in terms of performance compared to WAP implementations [34]. The bandwidth available is higher and allows richer content to be downloaded. The "always on" connection does not require a long dial up (sometimes up to 40 seconds using WAP) and allows mobile subscribers to be charged for the content downloaded rather than for the time spent online.

However, I-mode was designed behind closed doors and none of its components' design specifications were ever made publicly available. Even though NTT DoCoMo adopted the Internet TCP/IP protocol from the beginning and therefore uses well known and studied technologies, no standard security was provided before 2001. All this means that security relied on the lower layers of the I-Mode protocol stack, such as the PDC-P radio link which was never published and was potentially subject to attacks. In 2001, NTT DoCoMo adopted the SSL/TLS [66] standard security layer which provides end-to-end security between mobile phones and the I-Mode server [4]. However, only server side certificates are handled by mobile clients which does not provide non-repudiation services, digital signature cannot be used by mobile users and the I-Mode server cannot authenticate mobile users, at least at that protocol layer.

Conclusion

Higher bandwidth availability due to network upgrade as well as technical advances in mobile device design will soon make the use of the IP stack in the wireless environment possible.

Version 2.0 [7] of the WAP specifications adds support for TCP/TLS/HTTP and provides IP support to mobile devices when it is available on the radio network [220]. WAP2.0 now encompasses both the TCP/IP stack and the old WAP protocol stack and provides enhanced security by guaranteeing end-to-end security between the mobile device and the content provider through the use of TLS [8]. As a result, the "WAP gap" problem disappears and the utility of the WAP Gateway is reduced to providing enhanced telephony functionalities as well as WAP push services.

2.5 Location Management in Mobile Networks

Location Management is the process by which a mobile network keeps track of its subscribers in order to allow the latter to initiate and/or maintain a wireless communication. With the advent of location-based applications, the positioning technologies previously used have become obsolete and some more accurate techniques are required to comply with the E112/E911 regulations recently adopted (see Section 1.2.2). In this section, an overview of the newly standardized positioning technologies is given along with some considerations about accuracy and ease of deployment. Finally, the network upgrades needed to handle location information requests and positioning are detailed together with the main location management operations.

2.5.1 Location Positioning Technologies

In order to comply with the new regulations aiming to render mobile phones locatable within wireless phone networks, the American National Standards Institute (ANSI) as well as the European Telecommunications Standards Institute (ETSI) have recently standardized the following location positioning systems [18].

Network-based Mobile Positioning Technologies

Enhanced Cell-ID. The Cell-ID technique is the most simple but also one of the least accurate positioning methods. As described in Section 2.4.1, since mobile phone networks are cell-based, they constantly need to know in which cell a mobile user is located in order to provide him/her with mobile network services. Therefore, the positioning technology is already built-in and a simple software upgrade can make it available for LBS use. However, this method suffers from a very low and variable accuracy due to differences in cell size. Another problem lies in the fact that the cell serving the mobile phone might not be the closest to the handset. Precision can therefore vary from a few hundred meters in cities up

to several kilometers in the countryside where cell concentration is much lower.

Enhanced Cell-ID takes advantage of the Cell-ID positioning technique associated with some features of the GSM system usually used for network management. Timing Advance (TA) is an internal parameter used for synchronization on the radio channel in TDMA systems (see Section 2.4.1). It basically holds the time it takes for the radio signal to travel from the mobile phone to the Base Station. However, its accuracy is very low (around 500 meters) and using it to locate a mobile phone would require the device to communicate with three Base Stations. Timing Advance is therefore only used in conjunction with Cell-ID to achieve higher accuracy by determining with precision the cell serving the mobile phone, when the cell radius is greater than 500 meters. In 3G networks (see Section 2.4.3), Timing Advance is replaced by the Round Trip Time (RTT) parameter.

Time of Arrival. Time of Arrival (TOA) is a technique based on Base Station (BS) triangulation and assumes that BS positions are accurately known. It also exploits the time difference between radio signals propagation for the same signal to reach two distinct BS. The location of a mobile phone is determined by the intersection of the three hyperbolas corresponding to its estimated location measured by three BS. This approach provides only a little better accuracy than Cell-based location techniques and is subject to quality of service problems due to potential multipath signal propagation and/or lack of Base Station coverage. Also, some Base Station upgrades known as Location Measurements Units (LMU) (see Section 2.5.2) are necessary in order to measure and triangulate users' position. These units need to be implemented in every Base Station and need to be synchronized. TOA enables the network to locate a mobile user with a precision of about 100 meters.

Handset-based Mobile Positioning Technology

Enhanced Observed Time Difference (E-OTD). Observed Time Reference refers to the time interval observed by a mobile phone between the reception of signals emitted by two different Base Stations. This positioning technique locates a mobile phone by triangulation, knowing the Base Stations' location, the arrival times of the signals emitted by each Base Stations as well as their the timing differences. It requires some software modifications in order to enhance existing measurements processes as well as some Location Measurements Units (LMU) (see Section 2.5.2). Its accuracy ranges from 50 meters to 150 meters.

Hybrid

Assisted GPS. The Global Positioning System (GPS) [111] is the oldest as well as the most widely used satellite-based positioning system. It relies on 24 satellites in orbit around

the earth. A receiver calculates the distances between itself and four satellites and is able to work out its location with a precision of up to ten meters. The system is however subject to coverage problems inside buildings and in very dense urban areas, and mobile phones need to be upgraded with these satellite receivers in order to avail of the system. A European version of this positioning technique called Galileo will be available in 2008. The system will be under civilian control and will be an alternative to the GPS system, controlled by the United States Department of Defense.

Assisted GPS, or A-GPS is a network-assisted GPS. It combines the GPS system with network-based positioning technologies to achieve better response time. Assisted GPS uses network resources to get relevant information such as the position of the appropriate satellites to contact and forwards it to mobile stations. The GPS receiver included in mobile phones can then:

- Process its location quicker and more precisely thanks to the additional correction data received. This approach is known as Mobile Station-based A-GPS
- Acquire the raw GPS signal and forward it to the mobile network. The mobile network can perform the necessary calculations on behalf of the mobile station and process its location. This approach is known as Mobile Station-assisted A-GPS

This hybrid approach achieves an accuracy of a few meters in open environments but remains limited in dense urban areas due to GPS lack of availability.

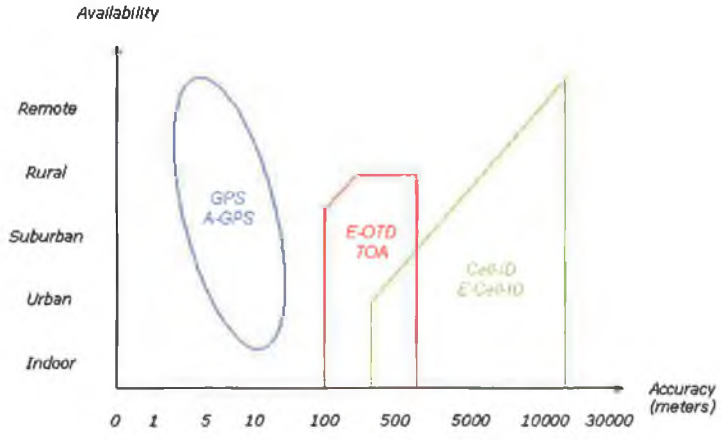


Figure 2.4: Accuracy of positioning techniques.

Some of the positioning techniques presented in this section are applicable whatever the geographical context but may not provide an optimal accuracy for some location-based services. Others are very accurate but are not available in every location. Almost all of them need some software and/or hardware upgrades, as described in Section 2.5.2. Therefore, because of all these parameters, network operators may implement only a subset of these standardized techniques, depending on the level of service they want to offer as well as the investments they are willing to engage in network upgrades. Figure 2.4 summarizes the accuracy of each method given the context environment in which they operate.

2.5.2 Network Upgrades to Support Location Information

The integration of these new positioning technologies implies both a hardware and software upgrade on the existing wireless networks. The authors of [71] describe three newly required hardware units as well as some existing hardware upgrades to be performed in order to correctly implement the new location positioning requirements.

The first hardware component needed is called the Gateway Mobile Location Center (GMLC). It is in charge of interfacing the network with location positioning requesters and its main tasks are requester authentication as well as access control. The Serving Mobile Location Center (SMLC) is a component located at the edge of the access network and is in charge of determining the position of mobile terminals. Finally, Location Measurement Units (LMU) assist SMLCs in determining mobile terminals' positions by carrying out Base Station synchronizations measurements when necessary.

As stated before, existing network components also need to be upgraded. Mobile Switching Centers (MSC) need to be able to work in conjunction with GMLCs in order to perform access control according to users' preferences. It also needs to be able to trigger mobile terminals for their location when handset-based mobile positioning technologies are used. Base Stations also need to be upgraded with the necessary LMUs.

2.5.3 Operations

Understanding the concept of network areas is essential to capture the location management mechanisms used to route voice and data phone calls. In this section, we briefly describe the different levels of location areas used within a wireless phone network and show how the latter manages to maintain an up-to-date location information for every registered mobile phone.

Network Areas

Wireless mobile phone networks cover a geographical area called the Public Land Mobile Network Area (PLMN). This is the area in which a network subscriber expects to be both reachable and able to initiate voice or data phone calls. Because of the very nature of cell-based wireless networks, the PLMN area is made up of multiple other subdivisions corresponding to the different network layers. Figure 2.5 presents the different levels of network areas.

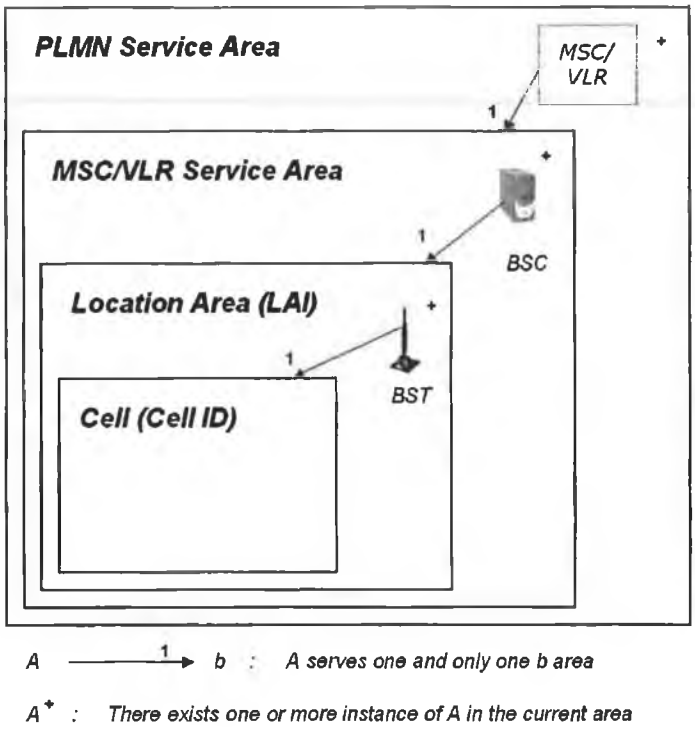


Figure 2.5: Network Areas in Wireless Networks

As explained earlier, routing voice/data phone calls involves the knowledge of the location of the receiver. In other words, the mobile network and more precisely the MSC must have up-to-date Mobile Stations' location information in order to establish the communication. Two processes enable the MSC/VLR to be aware of the Mobile Stations' position:

- *Location Update.* Every time a Mobile Station enters a new Location Area, it sends an update message to the corresponding MSC. This automated process does not occur as often as moving from one Base Station to another. It is not very precise since the Location Area returned may potentially be very large, but it reduces however the search space when the MSC needs to locate a Mobile Station. *Location Update* also occurs

when a Mobile Station is either switched on or off. In practice, Base Stations regularly broadcast the LAI of the Location Area they are part of. Mobile Stations compare this value with the previous one received and send an update to the corresponding VLR via the MSC if it has entered a new Location Area. The MSC provides Mobile Stations with a new TMSI and updates the different components of the Network Sub System.

- *Location Paging.* When the MSC needs to locate a particular Mobile Station in order to route a phone call for example, it *pages* the Mobile Station or polls the Base Stations located in the Location Area associated with this Mobile Station. The Mobile Station can then send a location update.

More information on the research efforts to optimize both *Location Update* and *Location Paging* can be found in [139].

2.6 Refinement of the Thesis Objectives

In the light of this section, we wish to refine here the research objectives previously stated. The research undertaken aims at designing a secure infrastructure to enable mobile phone users to avail of location-based web services over the Internet.

Mobile phones are here Internet enabled devices. They can connect to a 2G or 3G network and do not necessary embed any positioning technologies. However they need to be locatable in the sense that an entity, such as the mobile phone network for example, is able to position them and transfer their location safely to the mobile operator databases.

The location-based web services are web services run by third party entities. These third parties access mobile operator's network resources through the infrastructure to provide LBS over the Internet to both mobile and static clients.

The infrastructure design will make sure to take into account all the security issues arising from the nature of the radio network. An appropriate PKI suitable for mobile phones will be used as a way to authenticate mobile users and secure communications. Finally, the infrastructure will help mobile operators to comply with the EU directives stated in Section 1.4.2 by allowing 'opt-in' user consent and flexible access control to location information.

Chapter 3

Analysis and Requirements

3.1 Introduction

This section aims to provide the requirements for the design of an architecture that guarantees privacy for mobile phone users when considered as *Targets* for location-based services applications. Our analysis first describes the mobile environment in which the architecture will operate by introducing the different actors involved in the provision of location-based applications. By stressing each entities' security requirements, we introduce the need for a middleware together with its open protocol and PKI in order to interface and mediate the transactions between the different actors. We finally state the requirements for the new components considered.

3.2 Environment

We present here the basic requirements of the main actors in the context of the provision of location-based services in wireless phone networks. First and foremost, we give a general description of the entities involved and detail their respective requirements. We then study the threat model of the environment in which they operate and identify the main problems that the research work carried out aims at solving. Finally, we introduce the need for a middleware in order to implement the various requirements of each entity.

3.2.1 Functional Description and Requirements

The architecture considered in order to enable *LBS* provision is composed of 4 different entities as illustrated in Figure 3.1. Recalling the terminology defined in Section 1.3.1, we identify the requirements for the following entities:

- The *Subject*. The *Subject* is the entity querying a *LBS* through the Internet. She may or may not be registered with a *LBS* or with the *Mobile Operator*. The *Subject* can access a *LBS* using a fixed or mobile device through a fixed or wireless Internet access and may or may not be registered with the corresponding *LBS* depending on the type of service offered. We also refer to mobile *Subject* as a *Subject* connecting to a *LBS* through a mobile device.

- The *Target*. The *Target* is the entity to be located by the *Mobile Operator* and whose position will be used by *LBS* to offer a value-added service. She carries a mobile device connected to a wireless mobile phone network, as defined in Section 2.4 and that is locatable by either network or handset-based positioning techniques (see Section 2.5.1 for further information). The *Target* has to be registered with the *Mobile Operator* and may or may not be known to the *LBS*. Her identity and location should be protected in order to restrict their access by authorized *LBS*. Finally, the *Target* should have the possibility to set up and modify her privacy preferences regarding to the degree of intimacy she wishes to maintain between her and any *LBS* or *Subject* that may want to locate her. As mentioned in Section 1.3.1, the *Target* and the *Subject* entities may represent the same entity in some situations such as when a mobile *Subject* uses a *LBS* that requires her location to deliver its service.

- The *LBS*. This is the location-based service provided over the Internet. It can be accessed either anonymously or using pseudonyms by *Subjects*. The external third party that provides a *LBS* is refered to as a *LBS* provider.

- The *Mobile Operator*. This entity operates a wireless network and may be responsible for locating *Targets* and managing their location information, when network-based positioning techniques are used. The *Mobile Operator* is also in charge of delivering *Targets*' location information to *LBS*, according to their privacy preferences.

3.2.2 Threat Model

The *Mobile Operator* is the only completely trusted entity in this architecture. She knows the identity as well as the personal details of all of her mobile subscribers. She also already knows their location to a certain extend as this is a requirement for implementing a cellular network; see Section 2.4.1 for more information. Soon, when the new positioning techniques mentioned in Section 2.5.1 will be implemented, she will have access to rather more accurate location information. The *Mobile Operator* needs to be trusted not to misuse the data she already has or will have access to. *Targets* trust her not to disclose their location information

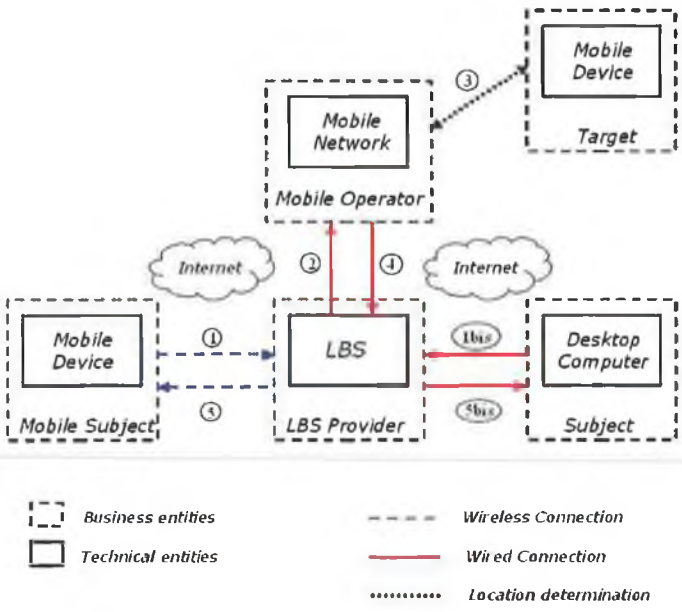


Figure 3.1: General Architecture Topology.

without considering their privacy preferences while *LBS* expect her to provide them with reliable location information upon authorized request.

The *LBS* may not be trusted by any other entity, with respect to location information handling. *LBS* providers do maintain a commercial relationship with the *Mobile Operator* and for this reason, we believe that they are not likely to give away the location information they have collected. However, we are aware that some information leaks may occur in some cases. E-commerce web sites that store credit card details in their database are usually much less trusted than web sites that outsource the billing transaction to a trusted banking institution, by implementing protocols such as SET [120] for example. Similarly, we believe that *LBS* that store location information represent a threat for *Targets* since this information could be stolen and misused by some other malicious entities, like credit card numbers could be. Therefore, we believe that *Targets* should be able to judge and decide whether the *LBS* requesting their location should be given full or limited access to it. This way, we avoid considering *LBS* as a "global hostile observer" and create several levels of confidence. Yet, *LBS* are expected not too cheat in order to help malicious *Subjects* gain more information than what they should receive. Indeed, *LBS* may query *Mobile Operators* for some *Target*'s location information on behalf of a particular *Subject*. By replacing the identity of the malicious *Subject* with the identity of a more trusted one, a *LBS* could retrieve more precise and sensitive information and forward it to the malicious *Subject*.

Subjects authorized to access a particular *Target*'s location information are generally believed by the latter not to misbehave with the location data they receive since there exist a trust relationship, external to the architecture described, that has been established prior to connecting to a particular *LBS*. An example of such a relationship lies in the identity management of Instant messaging services such as MSN Messenger, where someone needs to know her correspondent's email address before being able to establish a communication with this person. Malicious *Subjects* that collude with some malicious *LBS* can however become a threat to the architecture considered, as describe above.

The *Target* can potentially become an active entity with respect to location information provision when handset-based positioning techniques are used by the *Mobile Operator* to retrieve her location details. In this precise context, the entity is trusted to provide reliable location data. In other words, the *Target* is not considered as an attacker willing to modify her location information.

3.2.3 Conclusion

The requirements stated in this section outline the role of an intermediary that would make decisions regarding to who could access location information and when. The *Mobile Operator* is clearly in the best position, both from a technical and commercial point of view, to be given this responsibility. We therefore propose the design of an interoperable middleware, operated by a trusted third party, that will implement the technologies necessary to fulfill the requirements detailed in this section. The proposed middleware may be administrated and run by the *Mobile Operator* and will communicate with *LBS* using a specific protocol.

3.3 Middleware and Protocol

A middleware is generally defined as a software layer or "glue" that is used to interface two applications in order to facilitate their communication. In the context of this thesis, the middleware considered will interface location-based web services with low level positioning technologies. It will act both as a location information provider and as a security mediator between the different requestors and providers of location information. In this section, we outline the requirements for the design of such a middleware as well as the ones for the protocol used by *LBS* to access its functionalities.

3.3.1 Middleware

Security and privacy of *Targets* being the main reason why such a middleware is needed, we first describe the functional requirements it should fulfil. We then investigate the security features that it provides to *Targets*, *Subjects*, *Mobile Operators* and *LBS*.

Functional Requirements

The middleware proposed is designed to be implemented within a trusted third party's infrastructure. We consider the *Mobile Operator* as the candidate that provides the ideal environment to facilitate the implementation of such middleware. It will easily allow the latter to remain transparent for *Subjects*, *Targets* and *LBS* and will provide an easy access to *Targets*' location information. As a result, we will use it as the trusted third party entity in our description even though the architecture design will allow the middleware to be located in some other trusted third parties's infrastructures. The main functional requirements for the design of such a middleware are listed as follows:

- *Requirement 1.* It should be easily accessible to Web Service developers in terms of programming interfaces. Web Service designers and developers, as opposed to Telecom developers, are more familiar with high level programming languages. The middleware will therefore be implemented at the application level of the OSI model.

- *Requirement 2.* Its design should follow a network and mobile device independent approach. In other words, the middleware will abstract the location positioning technologies used to retrieve location information. It will also be independent of the wireless network used, provided the latter can provide a full Internet access to *Subjects*. Finally, the middleware is intended to be mobile device independent, which means that *Subjects* may use whatever type of devices they wish as long as the latter are connected to a suitable wireless network and, of course, locatable as defined in Section 2.5.1.

- *Requirement 3.* It should provide an interface capable of accessing some of the *Mobile Operator*'s resources. In particular, it should be able to retrieve *Targets*' location information from the Gateway Mobile Location Center (GMLC) when necessary.

- *Requirement 4.* It should provide an open interface in order to interoperate with similar infrastructures, run by other *Mobile Operators*, that facilitate the provision of location information. In particular, the middleware should handle roaming *Targets* in terms of positioning and charging.

- *Requirement 5.* It should provide an open interface so that any entity willing to implement a location-based service over the Internet using *Targets*' location information can access the necessary resources upon agreement with the *Mobile Operator.*

- *Requirement 6.* It should provide some charging mechanisms implementing the revenue sharing model by which a *LBS* provider and the *Mobile Operator* share the technical resources and the profits made from every *LBS* usage by *Subjects.*

- *Requirement 7.* It should provide different formats for location information. Indeed, while the world geodetic coordinate system [210], which describes location as a pair of coordinates *(longitude,latitude)* together with its altitude, would constitute a raw representation of location information, it is likely that other formats such as northing and easting coordinates or even civil location description such as "Dublin train station" will prove to be more meaningful to *LBS.* Therefore, the middleware should be able to perform translations between the different location formats.

Security Requirements

Identity management constitutes without any doubt a crucial characteristic of the security mechanisms designed as part of the middleware. *Targets*' identity must be known precisely to the middleware so that it can locate them upon request. However, *LBS* may or may not be given the exact identity of a particular individual either *Target* and/or *Subject*, depending on how trustworthy the former is or on what the latter is prepared to disclose in terms of privacy details. As a result, the middleware should implement pseudonymous access to *LBS*, where an alias known as a pseudonym is used to refer to a particular individual. Preferably, the pseudonyms used should represent long term identifiers such as the ones used in web services like instant messaging or webmail. The middleware should also be able to provide an anonymous usage of *LBS* by which the identity of either *Subjects* or *Targets* may not be a necessary parameter to avail of a particular location-based service. Authenticated access should also be considered, as a matter of completeness.

Location information management will also be a key requirement for the design of such a middleware. Delivering location information to a *LBS* could constitute a threat for a *Target*'s privacy if it discloses too many details about her location. Therefore, we believe that intentionally degrading the quality of location information in order to render it less meaningful but still relevant to *LBS* constitutes a key feature of the middleware as well as a powerful privacy enhancing technique. This process will be from now on refereed to as *Location Blurring* and is further investigated in Section 4.3 as well as Section 6 of this thesis.

Access control to location information will also be handled by the middleware. The access

control mechanism should allow *LBS* to access a particular *Target*'s location information on behalf of a particular *Subject*. Access control decisions should be made as quick as possible so that they do not affect the process of location information provision to *LBS*. The provision of privacy preferences by *Targets* regarding to access control should be handled by a component of the middleware. The main requirements for this privacy preferences provision tool are:

- It should allow *Targets* to precisely tell which *LBS* on behalf of which *Subject* is authorized to access her location information, depending on parameters such as the time.
- It should be accessible through the Internet, from both fixed and mobile devices.
- It should allow *Targets* to quickly and easily override their privacy preferences for a given period of time in order to face a particular and potentially unpredicted context. This process would avoid requiring them to enter once again all their privacy details.

3.3.2 Protocol

As already stated in Section 1.3.1, *LBS* providers are third parties interested in accessing *Mobile Operators*' network resources in order to provide location-based web services to their *Subjects*. In order to exchange the necessary credentials with the middleware and query for some *Targets*'s location information, *LBS* need the specification of an open protocol that will allow them to communicate securely with *Mobile Operators* and access their resources. In this section, we detail the functional and security requirements for such a protocol.

Functional Requirements

The protocol will mainly enable *LBS* to connect to the middleware over the Internet using standard technologies. It will also in theory enable *Targets* to access their personal details and preferences but it is likely that a web interface will be provided for convenience. The main services that the protocol should offer are summarized below.

- Access to location information upon presentation of the right credentials.
- Management of personal profiles for both *LBS* and *Targets*.
- Communications between different instances of the middleware.

The protocol should also allow for *LBS* providers to bill their *Subjects* through the middleware following a revenue sharing model. The *Mobile Operator* would charge a *LBS* provider for the provision of a particular *Target*'s location information. The *LBS* provider would add

value to the data and charge in turn the *Subject* requesting the *LBS* through the *Mobile Operator*. Both the *Mobile Operator* and the *LBS* provider would make profits out of the transaction, as further explained in Section 5.3.2.

Security Requirements

The two party protocol should be able to let the *LBS* and the middleware authenticate each other. The authentication protocol used should be negotiable prior to the invocation of one of the services listed in the previous section. If the authentication phase succeeds, a session is established and the entity initiating the protocol should be able to request one or more services from the other entity. The session should terminate upon request from the initiating party.

The protocol should allow for the confidentiality of the exchanged messages. The symmetric algorithm, as well as the symmetric key size should be agreed during a negotiation phase that may occur during the authentication protocol negotiation or as part of the protocol itself during a handshake phase as in SSL [66]. The middleware should only negotiate algorithms that have received a fair amount of study and preferably standardized. The negotiated key length should also follow the up-to-date recommendations established by experts in the field of cryptography. A key agreement phase may then take place as part of the authentication protocol. When the session is established between two entities, the shared symmetric key should be used to encrypt each message exchanged until the session terminates. The two entities should then discard the secret key.

Integrity and non repudiation of the messages should also be supported through the use of Digital Signatures. The algorithms choice should follow the same logic as for the symmetric algorithm used for confidentiality.

3.3.3 Conclusion

The security requirements of the protocol used by *LBS* to access the middleware can be fulfilled using a conventional X509 PKI. The same does not apply however to the communications that occur between *Subjects* and *LBS*, where the formers may use wireless mobile devices to connect to the latter. As for conventional web sites, these communications may or may not be protected depending on the sensitivity of the information transmitted. When secure connections are required between these two entities, a PKI that suits better the needs of such a mobile environment should therefore be employed.

3.4 Public Key Infrastructure

Mobile phones become more and more sophisticated but still remain less powerful than desktop computers. In Section 2.3.3, we pointed out the shortcomings of existing "wired" PKIs. The main motivation behind the proposal for an alternate PKI for mobile devices is, on one hand, to overcome these issues and on the other hand, to simplify client-side security-related operations. In particular, we wish to remove the need for key distribution and provide a fast and efficient key revocation mechanism. Furthermore, mobile phones are devices that are easily stolen, lost and that are more fragile than conventional computers. Therefore, the devised PKI must remain flexible enough to adapt to the mobile environment described and secure communications involving mobile *Subjects* and *Targets*. The PKI will also be used to secure communications between the "non mobile" entities. In particular, the communications between *LBS* providers and the middleware will be protected as well as the ones between the middleware and the *Mobile Operator*, when the former is not implemented as part of the latter's infrastructure for example.

3.5 Conclusion

The architecture described in this section comprises a secure location management middleware, a secure and open protocol used by *LBS* to access the *Mobile Operator*'s resources through the middleware, and a PKI to provide security services to *Subjects*. The requirements stated for each entity emphasize the need for a strong support of security and privacy techniques. In order to fulfil these requirements, our architecture design will benefit from the latest advances carried out in the field of cryptography and will implement innovative techniques to protect *Targets*' location information according to their privacy preferences.

Chapter 4

Related Research

4.1 Introduction

One of the big challenges in location aware computing lies in the end-to-end control of location information [166]. This involves securing the communications used to convey location information as well as guaranteeing located entities an appropiate level of privacy. Some research has been carried out in the field of location privacy at the network layer, with, in particular, the Mist routing project [25] and the MIXes approach used in mobile communication systems [81]. In this section, we concentrate our study on location privacy at the application layer. In particular, we analyze recent advances in privacy techniques that modify the location granularity of mobile users in order to provide privacy services. Two different philosophies co-exist:

- Techniques that offer anonymity by withholding a mobile user's identity from a third party *LBS*.
- Techniques that blur the location information supplied to *LBS*.

We describe both techniques which we will respectively refer to as *Identity Blurring* and *Location Blurring*. We also review some more general algorithms and tools and describe approaches in building secure infrastructures similar to ours. Finally, we give a brief description of related standards that are currently being defined.

4.2 Identity Blurring

Identity Blurring is a technique that aims at rendering mobile users unidentifiable so that linking location information with a particular individual is made impossible. It is commonly

admitted that identifying users with their real identity opens up too many privacy breaches, especially when location information is transmitted to non-trusted third parties. Therefore, most research work carried out in the field of context aware computing advocate either anonymous or pseudonymous usage for location-based services. In this section, we review the two main approaches that implement *Identity Blurring*. The first one uses *transaction pseudonyms* [168] as well as a technique to prevent linking location requests while the second one describes a new anonymization technique.

4.2.1 Mix Zones

Beresford and Stajano recently introduced the concept of *mix zones* as a way to provide location privacy in pervasive computing [36]. Their main idea is to provide a certain degree of anonymity between non trusted LBS and their end-users by proposing a secure framework capable of frequently changing pseudonyms in order to achieve anonymization of location information. End-users are located through a portable device by either some embedded hardware or by some external network resources. A shared event-based middleware is used as a trusted proxy in order to provide pseudonym management services to them. LBS register with the middleware specifying an *application zone*. As a result, users that have registered with this LBS through the middleware and that enter its *application zone* will have their location details transmitted pseudonymously to it.

Beresford and Stajano introduce the concept of *mix zones* based on David Chaum's *mix networks* [49] ideas in order to implement their approach. A *mix zone* is defined for a certain number of users and constitutes an area where none of the LBS currently serving them have registered an *application zone*. Users that enter a *mix zone* become invisible from LBS and can change their pseudonyms to later become untraceable by LBS when they leave it. The results they obtain when assessing their technique are not however quite satisfactory. In their laboratory environment, many factors such as the number of users as well as the geometry of *mix zones* force the middleware to wait a considerable amount of time before forwarding services requests. In [37], the authors refine the concept of *mix zones* as well as its evaluation and extend their work by providing feedback to end users on their level of privacy.

Beresford and Stajano propose a method to prevent user tracking using historical location data. The framework they provide targets LBS that accept pseudonyms but that neither need to keep any session information nor perform user authentication. Using this framework to enable commercial wide-area LBS is however questionable. A lot of LBS such as the ones based on the well known "friend finder" principles require identification and authentication

mechanisms. Also, the size of their *application zone* could potentially greatly exceed the size of a country leaving no room for *mix zones*. Point-Of-Interest LBS such as location-based advertising services can however benefit from such an approach.

4.2.2 Location k-anonymity

In [96], Grutezer and Grunwald propose a new technique called *location k-anonymity* to allow anonymous usage of LBS. Their idea extends the concept of *k-anonymity* developed by Samarati and Sweeney in the context of database privacy [197]. With the concept of *location k-anonymity*, Grutezer and Grunwald want to "de-identify" LBS queries by making a user indistinguishable from any *k-1* other. Their approach allows for anonymous connection to LBS and prevents any re-identification attempts or location prediction when they are given some location history data.

End-users communicate with non trusted LBS through a trusted location server acting as a proxy. They typically use handset-based location systems such as GPS receivers to locate themselves and periodically send their precise location over a cellular network to the location server. The middleware then applies the location *k-anonymity* algorithm on the data received. The basic idea behind it is to provide anonymity by decreasing the location accuracy of the LBS end-user position. Instead of precise privacy sensitive user location coordinates, LBS receive two points delimiting an area in which k other individuals are located. The algorithm starts dividing the area A, keeps the sub-area that includes the location L and repeats the process until the sub-area obtained only contains *k-1* individuals. The coordinates of the sub-area S obtained at the second last iteration (i.e. containing at least k individuals) is then sent to LBS. Here again, users are shielded behind the middleware which is the only entity conducting anonymous communications with LBS.

While this first approach consists in varying the size of the area sent to LBS in order to meet the privacy requirements, a variant of the algorithm described by the authors aims at providing anonymity by delaying LBS responses. The parameter k now denotes the number of entities that have visited the area S that will be revealed to LBS. The area S is now passed as a parameter to the new *location k-anonymity* algorithm. When it receives a LBS request, the middleware starts monitoring the area S and counts the number of entities visiting it. When this number becomes greater than k, the privacy requirements are met and the middleware sends an LBS response containing the area S as well as the time it took for k entities to go through S. As a result, this approach achieves better accuracy but includes a delay in location responses.

The main problem that arises with Grutezer and Grunwald's approach is that it only

specifies privacy as the size of a crowd in which the end-user can hide. This may not be relevant for some services other than location tracking LBS. Furthermore, when user density is low in a specified area, the area containing k other entities can become extremely large and useless when returned to a LBS. Also, knowing some external information such as where a user lives can enable attackers to identify her. In [95], Grutezer *et al.* investigate privacy issues that arise when attackers collect location and time information of a particular user, referred to as a *path*. Since their initial approach relies on a high number of users and therefore on simultaneous *paths* potentially tracked, they propose two ideas as a complement to their work to try overcome the issue. *Path Segmentation* aims at truncating paths to make user tracking more difficult to perform. *Minutiae Suppression* introduces black-out periods in their *path* during which users are not locatable, for example when their location gives away too much privacy.

4.3 Location Blurring

Location Blurring is a technique that aims at rendering location information less accurate by intentionally decreasing its granularity. It can be seen as a data perturbation technique, like adding noise to accurate data in order to make it less meaningful. The Global Positioning System (GPS) [111] constitutes a good example as it used to employ such a technique before to 2000 [209] as part of its Selective Availability (SA) system whose aim was to deny access to accurate location information to unauthorized users. *Location Blurring* can be used by mobile users in order to avail of LBS while preserving a certain level of privacy. The idea has been mentioned in multiple research projects but its implementation has only been carried out using "civil locations" ("Train station","Dublin", "Ireland") and not in the context of raw location data. In his thesis [131], Leonhardt pushes the idea further and exploits the hierarchical nature of location domains to provide a way to vary the granularity of the location information disclosed. For example instead of revealing that a user is at a Dublin train station, the system only discloses the name of the city. Also, Hengartner and Steenkiste consider the same technique and use it as part of their access control to location information environment [106]. In this section, we analyse an extension of the algorithm mentioned in Section 4.2 that was used to implement *Identity Blurring*. Here it perturbs location data in order to achieve a specified degree of privacy.

In [97], Grutezer and Liu investigate a technique based on their previous work on *location k-anonymity* and referred to as *Minutiae Suppression*; see Section 4.2.2. In their paper, the authors state that some areas are more sensitive than others and therefore should be protected from location tracking. However they point out that disabling location information

determination in those areas is not sufficient. Indeed, attackers may easily infer where the user has gone using simple linear interpolation. To address this issue, they propose to withhold any location information that could disclose the sensitive areas the user has visited in order to protect her *path* information.

The architecture considered is very similar to the one used to implement *location k-anonymity*. The authors propose three different algorithms in order to withhold location information when mobile users enter sensitive areas. The *Base Algorithm* discloses location information only when mobile users are located in non sensitive areas. The *Bounded-Rate Algorithm* does the same but adds more privacy by lowering the frequency of location updates when mobile users are located in non sensitive areas. Finally, the *k-area algorithm* follows the *Base Algorithm* philosophy but only releases location information when it does not disclose which of at least k sensitive areas a mobile user has visited.

Grutezer and Liu proposed a technique whose aim is to solve the location inference problem. In their security evaluation, they run the three algorithms they have designed on a mobility simulator and find that the *k-area algorithm* is very convenient as it discloses more than 85% of location updates in non sensitive areas while guaranteeing an acceptable level of privacy. However, the assumptions under which the evaluation was conducted may be considered as weak. Indeed, the sensitive areas described in their experiments are buildings situated in the city center of Manhattan while streets are considered as non sensitive regions. If the density of sensitive areas decreases, one can expect a change in their results.

4.4 Machine Readable Location Privacy Policies

Machine readable privacy policies are privacy policies represented in a machine-readable syntax so that the process of checking a particular action against such privacy policies can be automated. The P3P initiative developed by the W3C [60] proposes an automated mechanism that helps users gain more control over the use of personal information collected by the web sites they visit. In this section, we review research that builds on the P3P project and analyze how they manage to provide a more efficient control over location information usage. It is worth mentioning that P3P and P3P-derived approaches rely on the assumption that the different entities involved will follow some predefined rules. Indeed, no mechanism has been defined in order to check whether privacy policies are enforced or not. However, P3P provides indications on how to resolve disputes if any, and can be, for instance, implemented as part of a larger legal framework.

In [116], Indulska *et al.* present a location management system that aims at aggregating, processing and managing location information obtained from various sources. The authors

use P3P and APPEL [59] as part of their location access control mechanism. In their system, each location query made to the location server is sent together with a P3P policy file indicating the purpose of the location information collection. Using APPEL rule-sets representing target user's context-aware preferences, the location system is able to block, partially release or fully release the location information requested. The authors however recognize the necessity of a more fine grained control on location information release and state they will address this issue in future work. Another potential weakness of this approach is that location requestors need to be trusted, especially when generating their P3P policy file, since they could lie on the intended use of the information requested.

The pawS system [127] also uses P3P to encode data collection and location information usage practice. The system aims at letting users have control on the data collected as well as providing data collectors with tools to help them enforce the security policies correctly. In the architecture proposed, a user accesses services' resources through privacy proxies that handle the interactions between the two parties. A privacy beacon regularly announces potential data collections by services for a particular context aware environment. When a user wishes to avail of a particular service, she contacts her proxy which first retrieves the available privacy policies from the service proxy and selects the one that complies most with the user's preferences. The personal privacy proxy forwards the relevant and potentially context-aware data to the service proxy. The latter then issues an authentication token known as *agreement id* to the user for future reference, should she wish to update the information submitted. Collected data is finally stored in the pawSDB back-end database together with the corresponding privacy policies. From an implementation point of view, privacy policies are encoded in XML using the P3P namespace and users' preferences are described using the APPEL language. The authors recognize that such a system is only a privacy enabler and that a legal framework should be put in place in order to prevent fraudulent usage of personal data. They also plan to extend the P3P language to be able to formulate privacy policies that capture the location of the data collection as a parameter.

Myles *et al.* designed the *LocServ* middleware service [155] in order to support LBS usage and provide their users with a fine-grained control on the release of their location information. Users specify their privacy preferences as part of a *Validator*, which is a system component capable of making decisions regarding to location information release by checking users' privacy preferences against LBS's privacy polices. The authors used here again the P3P syntax to specify privacy policies. However, they also slightly extended its vocabulary in order to capture the concepts of user and third party-initiated location query. For example, they introduce the *solicited* and *unsolicited* classes of interactions between LBS and *LocServ* in order to differentiate between sporadic user-driven location queries and location tracking

in terms of privacy policies to apply. The authors also discuss the nature of *validators* and state that any kind of such component should be accepted by the system. As a result, they do not advocate the use of the APPEL language in particular and leave the choice of specifying and checking users' preferences to *validators* designers. Their implementation actually consists of a simple and relatively flexible language using basic attributes.

In [195], Snekkenes introduces relevant concepts for formulating location-related privacy policies. While allowing users to specify their own privacy preferences like in P3P, the author's approach differs however from the former on one significant point. Policy enforcement is not left to service providers but instead, is split between them and users. Not only users can refuse to use a service if it does not comply with their requirements, but they can also decide on the level of accuracy at which they will release their location information in order to preserve their privacy. Snekkenes also points out that P3P is tailored to web usage and that it does not suit particularly context-aware environments, where the collection of sensitive data is performed all the time as opposed to only once when connecting to a web site. As a result, the author specifies fragments of a language aiming at encoding personal privacy policies. The language captures concepts such as the accuracy of location information, the identity of the requestor, the time when the request was made, as well as the speed of the target user. As a conclusion, Snekkenes recognizes however that users may find the use of the language somewhat awkward and that further research needs to be done in this context. He is also concerned that entities could just ignore policies since no mechanism is used to check policy enforcement. The author mentions that cryptographic techniques could be used to address this issue but does not however study the possible solutions.

4.5 Access Control to Location Information

Two different categories of access control models co-exist. Mandatory access control (MAC) mechanisms such as the Bell-LaPadula model [94] implemented on a system let the latter define and enforce the access control policies for all users. On the contrary, Discretionary access control (DAC) mechanisms such as the Lampson's access matrix [126] let users define the access control policies. In general, MAC access control mechanisms are more secure than DAC ones. However, DAC mechanisms are considered more efficient in terms of performance and convenience to users.

This section shows that these models are not applicable as far as location information is concerned, due to the complexity of dealing with location and time. It then presents an alternative called the multi-target access control mechanism based on three sets of policies. Certificate-based access control mechanisms are then reviewed and the concept of role-based

access control is introduced as well as some applications. Finally, we discuss a rule-based access control framework used to provide mobile phone users with control on their privacy.

4.5.1 Multi-target access control for location information

In his thesis [131], Leonhardt proposes a generalised access control matrix in the form of multi-target access control policies in order to provide access control to location information. First, he recalls that standard models for access control do not work when applied to location information. Indeed, he states that neither location information nor the located object can be considered as a valid *target* since the two entities are inter-dependent. He then shows how to generalize the classic access control models in order to provide a solution to the access control to location information. In the context of the Lampson's access matrix, Leonhardt proposes to refine the security policy derived from the matrix. The standard policy can be stated as follows:

<subject> {<list of actions>} <target>

Leonhardt modifies it such that location information and located object can be addressed as *targets*:

<subject> {<action>} <target 1> ... <target n>

Using this definition, an authorization policy enabling a *subject* Joe to find out about the location of a *subject* named Fred when the latter is at school can be expressed as:

Joe { testForCollocation } Fred, Building=School

In the context of label-based access control, the author shows that the Bell-LaPaluda security model can also be applied to multiple subjects or targets such as location information and located objects. To conclude, he states that neither of the two approaches satisfies all environments and advocates a "mix-and-match" approach depending on the security requirements needed. Within the scope of his thesis, he designs a security mechanism that comprises three different layers, where only one or two are used at a time, depending on the requirements of the situation. He defines the following policies:

- *Access policies.* They are standard access policies that regulate authorized access to a particular resource or *target.*
- *Visibility policies.* They are policies regulating the level of location detail released. For example, instead of releasing the street name where a *target* is currently located, only the name of the town will be disclosed. This philosophy follows the author's tree-based hierarchical model for location information developed as part of his thesis.

- *Anonymity policies.* They regulate the level of detail released about the identity of a *target* at a particular location. This is the same idea as for visibility policies: a hierarchical model for identity is built and, depending on the level of security required, the identity of the *target* ranging from anonymous to his real first name and last name combination can be disclosed.

By combining those three levels of access control policies, the author is able to define more flexible policies, *higher-level policies*, applicable in a wider range of contexts.

4.5.2 Certificate-based solutions

In this section, we review two approaches that use authorization certificates as a way to implement Discretionary Access Control (DAC).

In [107], Hengartner and Steenkiste propose a design of an access control mechanism for a people location system. The architecture considered consists of a hierarchical structure of *location services*. *Location services* are entities that gather and/or process location information. Requests go through the *location services* chain and the response follows the inverse path. *Location services* can either be trusted or not, and the access control mechanism proposed relies on the use of location policies. In order to implement their location access control mechanism, the authors enclose trust information as well as location policies in SPKI/SDSI digital certificates, as described in Section 2.3.2. Users that want to use the system sign their location requests together with a timestamp and forward it to a *location service*. *Location services* can then use the certificates received as proofs when they require access to location information on behalf of somebody. The certificates, together with location requests are accessed through a virtual database called the Aura Contextual Service Interface designed in [119]. In their evaluation, the authors discuss the influence of delegation of access rights and admit that the cost of access control certificates generation and checking is quite high and that it only pays off when long certificate chains have to be checked. They finally conclude that their system provides good security but with a significant delay.

Hauser and Kabatnik also developed an access control system based on authorization certificates. In [105], they study the requirements for an access control mechanism suitable for location services. The authors describe an access control solution applicable within the NEXUS architecture for spatially aware applications [112]. They outline two ways in order to help users preserve their privacy while disclosing location information. The first solution lies in the frequency at which location updates are performed. The lower it is, the less accurate location information is. Users should be given control of the location sensors so that they can regulate or tune the accuracy of the information that is sent to the location

services. The other solution aims at reducing the information concerning users' identities by using pseudonyms. The solution for the access control is based on the use of authorization certificates binding the permissions of key holder directly to their public key. In their approach, *targets* share a unique pseudonym with the LBS with the particularity that the pseudonym in question is in fact the *target*'s public key. LBS also possess a cryptographic key pair. When a *target* wishes to let a *subject* locate her, the former issues an authorization certificate to the latter, as well as a LBS public key encrypted Token T containing her pseudonym together with a nonce to prevent replay attacks. Upon receipt of the query made by a *subject*, the LBS first checks its signature. It then decrypts the token T, revealing the *target*'s public key. Using it, the LBS attempts to validate the authorization certificate. Finally, if the authorization certificate's subject is the public key of the *subject* then this means that the *target* has granted the *subject* with the permissions stated in the authorization certificate. The LBS can then query its database to retrieve the *target*'s location information.

In a mobile environment, the main drawback of this approach would certainly be the *target*'s certificate generation and transmission over limited bandwidth. Furthermore, the authors point out that the revocation of permissions is quite problematic and can only be resolved by using short lived certificates. This however further increases the number of certificates to be generated by *targets*.

4.5.3 Role-based Access Control Mechanisms

The Role-based Access Control (RBAC) model [184] offers an alternative to MAC and DAC. As a non-discretionary access control, the model advocates the use of a *subject*'s function or *role*, instead of her identity. In this model, *subjects* are assigned *roles*, and each *role* is assigned a set of *permissions*. Most of the time, *permissions* granted to *roles* will not change while the association *subject-role* might vary more often. This results in a more flexible access control scheme regarding *permissions* assignments.

Zhang and Parashar propose the use of RBAC to build a flexible dynamic context-aware access control mechanism [221]. It aims at assigning dynamically *roles* to *subjects* as their context changes. However, the context and more precisely the location of the *object* considered is not taken into account when access control decisions are made.

In [58], Covington *et al.* extend RBAC by introducing *environment roles* as well as *object roles*. *Environment roles* capture the state of the environment at the time an access control request is performed. In particular, the time and location dimensions are used as environmental information in order to define *environment roles* of access control queries. *Object roles* capture properties of the *object* of the access control query. The Generalized

RBAC (GRBAC) approach is however very general and would need to be more tailored in order to provide access control to location information. In particular, the number of *roles* required to express an access control policy could potentially grow very large.

4.5.4 Rule engine-based Access Control Mechanisms

One of the main problems with traditional and RBAC mechanisms is that permissions and roles must be generated in advance. Access control decisions cannot usually be made on the fly depending on constantly changing parameters. Defining fine-grained access policies is therefore problematic, especially in context-aware environments where time and location are parameters that can potentially influence access control decisions.

In [115], Hull *et al.* describe the design of the *Houdini* framework which proposes a discretionary rule-based access control mechanism that enables mobile phone users to specify their privacy preferences regarding to their context information. In their paper, the authors focus on two key components of the *Houdini* framework:

- The framework used to support self-provisioning of preferences. Users can express their view on who should access their location information and when, by specifying a time and location dependent profile. Because the access control system is rule-based, users need not specify their privacy policies in details using a complicated language. Instead, they specify their preferences through web forms and let the underlying rule engine make access control decisions based on their settings as well other context data such as time and location.

- The *Privacy Conscious Personalization* engine. This is the component that makes access control decisions based on users' preferences and context information.

The authors discuss the choices made for the *Houdini* language used to encode rules and assess the performance of their design on the time it takes to render access control decisions. By achieving a 3 milliseconds response time per query, the system proposed is clearly efficient enough to handle multiple requests at the same time.

4.6 Secure Architectures for context aware computing

This section first presents a brief survey of the mechanisms that can be used to protect mobile users' privacy. Then, it reviews some architectures that implement part of the requirements formulated in Section 3 and analyses their strengths and weaknesses.

4.6.1 Location-aware computing with no location disclosure.

Priyantha *et al.* designed and implemented the *Cricket* system to enable indoor mobile location-dependent applications [171]. It is based on a network of fixed devices also known as *beacons* that advertise the geographical area they are located in. Each mobile device uses a combination of ultrasonic and RF sensors called *listeners* in order to receive signals from *beacons* and decide which one is the closest. *Cricket* is a location support system as opposed to a location tracking system, since location information is not centrally stored on a server. Instead, the system enables mobile clients to learn their location by listening to the *beacons* without disclosing any personal information. They can then decide whether they want to use it locally with an application that utilizes cached location-based information or transmit it to external entities.

In [187], Schilit *et al.* introduce the concept of *intermittent connectivity*. Like the Cricket system, their approach aims at using location information directly on the mobile device rather than connecting to a remote server. The mobile device computes its own location and uses cached location-based information. However, downloading and caching location information through multiple queries can potentially disclose someone's location. *Intermittent connectivity* advocates downloading geographically coded records in one go to avoid revealing precise location information when querying for location-based information. This approach is of course limited to LBS that can be run on a mobile device and for which location-based information is not subject to frequent updates.

4.6.2 Proxy-based location information disclosure

In [78], Escudero and Maguire propose the use of a proxy server to hide the network location of a mobile user as well as her identity. The proxy operates in an environment where the positioning technology used to locate a user is fully under her control and where LBS are accessed anonymously. When a mobile user wants to avail of a LBS, she retrieves her location, creates a SOAP request and forwards it to the proxy. The proxy hides her network address and acts as an intermediary between her and the LBS. To ensure confidentiality, they assume the presence of a shared secret key between mobile users and LBS in order to encrypt part of the SOAP request containing location information using XML Encryption. The encrypted location record submitted can then, either be forwarded to the corresponding LBS or published in a DNS LOC directory that can carry location information. Finally, Escudero and Maguire discuss the use of the proxy as a mix node [49] and show how the approach can anonymise communications.

A similar approach has been proposed by Konidala *et al.* in [69] in the context of wireless

networks. However, it strictly differs from Escudero *et al.*'s by its trust model. Indeed, the authors explicitly mention that the proxy used is trusted and should be implemented within the mobile operator's infrastructure. In order to secure the communication link between the mobile user and the proxy, the authors propose the use of a short lived shared secret key between the two entities. The communication link between the proxy and LBS is secured using a traditional X509 PKI.

4.6.3 Token-based location information disclosure

The security architecture described in this section gives mobile end-users a total control on the disclosure of their location information. Mobile end-users are given a token that they can release in order to provide third parties with their location.

In [180], Rodden *et al.* design a secure protocol based on the principle that mobile users should have control on the disclosure of their own location information. They propose a protocol that aims at retaining the mobile user's identity as long as no agreement is passed between her and a LBS. The location is then released to the LBS for a given time interval. The user generates a sufficiently large random number in order to avoid collisions and sends it to a location server in order to be used as a *transaction pseudonym* [168]. The location server then records the user's location as a triplet *(location,time,pseudonym)*. When a user wants to disclose her location to a LBS, she gives it her pseudonym. The LBS can then use it to retrieve her location information from the location server. By modifying the value of her pseudonym, she revokes the LBS' privilege to use her location information. This lightweight mechanism supposes of course that LBS do not collude with each other and that LBS do not need to identify users in order to keep session information.

Gajparia *et al.* [87] introduce a mechanism that enables end-users to control the use, storage and dissemination of their location information in the context of location-based services. The technique relies on the use of *constraints* [86], which are rules that dictate how location information should be handled. Constraints are statements bound to location information and are enforced by a trusted third party known as the *Location Information Preference Authority (LIPA)*, acting on behalf of the end-user. The authors then describe the protocol used to provide users with control on their privacy. Once the user has been located, her location information together with the related *constraints* form a token that is signed and encrypted using the *LIPA*'s public key. The token is then given to a LBS. The latter can identify whose location information the token contains but neither the location information itself nor the associated *constraints*. When necessary, the LBS sends the token to the *LIPA*, which can then verify the signature and decide to grant access to the location

information depending on the *constraints* imposed by the user. A message is then sent to the LBS, encrypted using her public key and signed by the *LIPA*. It contains either the location information together with some security parameters or a message stating the failure of the location request.

4.6.4 Secure platforms for location-based services provision

Little research has been carried out in order to offer access to location-based services over the Internet with a particular focus on end-users privacy. In this section, we describe three different initiatives whose objectives are relatively close to the Orient Platform overall project

Kurashima *et al.* designed a platform allowing LBS mobile users to control their privacy [125]. In particular, they developed a privacy control gateway to be run by mobile operators, whose aim is to match mobile users' privacy policies with location-based services'. The gateway uses the errors inherent to positioning systems in order to "blur" the location information, which does not provide a very high level of privacy. It also uses a policy-based privacy control mechanism. This mechanism is used to match LBS' policy with mobile users' preferences in terms of location granularity. It is implemented as a platform and run by the mobile operator. The authors have implemented their prototype and studied its integration within the existing 3G infrastructure. They do not, however, address the identification issues arising from the externalization of location information. It is also unclear whether they have designed and implemented an algorithm to adjust the granularity of location information. Finally, their platform does not provide location-based web services to its mobile users since it is implemented as a gateway where services are published. As a result, LBS deployment is not as flexible and transactions handling is left to the mobile operator.

Lee *et al.* propose a secure web services infrastructure to enable the use of location-based services through wireless networks [129]. However, it does not implement a comprehensive fine grained access control to location information since the profiles used as parameters to the policy creation do not even take time or location into consideration. Also, the approach implements the WAP1.1 architecture known to have encountered some security flaws (see Section 2.4.4 for more details).

Zuidweg *et al.* have designed a privacy control architecture in the context of the WASP project [224]. The WASP project aims to develop a context-aware service platform that enables 3G mobile phone users to access Location-Based web services [57]. The privacy control architecture is based on the P3P initiative and aims at implementing the *notice* and *consent* paradigms [1] in the context of location information. In his thesis [223], Zuidweg

[1] The "notice" concept refers to the process of informing users when information about them is collected while "consent" provides users with the choice between disclosing or retaining the publication of their

details how P3P can be extended to make it suitable for both web services usage and location information handling. In particular, he studies how to transport and reference P3P policies in the context of web services. He also advocates to update the P3P basic data types in order to include references to contextual information. Concerning user privacy preferences, the author proposes an extension of APPEL, known as the Context-Dependent Preferences Language, that takes location, time, date, day of the week and user activity as parameters. The WASP context aware privacy architecture provides a way to compare users' privacy preferences with service providers' privacy policies in the context of web services. However, the proposed architecture is entirely based on trust. Indeed, there is no mechanism defined in order to ensure that service providers will behave according to what is stated in their privacy policies. Furthermore, the platform's *Privacy Control Layer*'s decisions are not granular in terms of context information. For example, if a service's privacy policies do not comply with a user's privacy preferences or her context dependent preferences, the service request will be discarded: it's all or nothing. Finally, this approach does not support anonymous or even pseudonymous access to context aware services and does not therefore provide a high level of privacy for end-users.

4.7 Standards

In this section, we present the different standard bodies that influence the development of mobile telecommunications. We focus on the working groups that intend to specify security mechanisms to ensure a secure and private provision of location-based services over the Internet.

4.7.1 Open Mobile Alliance

The Open Mobile Alliance (OMA) [161] is an industry forum that was formed in 2002 in order to consolidate the multiple interoperability forums already existing in the area of mobile services. It brings together around 300 companies and aims at developing open technical specifications to guarantee interoperability between mobile services. In particular, the OMA Location Working Group continues the work carried out by both the Location Interoperability Forum (LIF) and the WAP forum in order to ensure interoperability of mobile location services on an end-to-end basis. The OMA currently develops two protocols that are relevant in the context of our research. These two protocols are being specified in order to comply with 3GPP Release 6 LCS Specification; see Section 4.7.5 for more details.

information.

The Mobile Location Protocol (MLP) [163] is an XML-based application-level protocol that can be used by third parties to communicate with mobile operators in order to access mobile users' location information. It is independent of the underlying location technologies and supports different transport mechanisms such as HTTP or SOAP. The Mobile Location Protocol offers five different services to its users. The first two services are initiated by the third party LBS and define a standard mobile user location request/response mechanism as well as an emergency one. The same services initiated by mobile users are also supported. Finally, a triggered location status update service can be requested by a third party LBS in order to continuously monitor the location of a particular mobile user. This protocol does not support any security or privacy techniques and provides location information "as is", upon request on MSISDN numbers.

The Location Privacy Checking Protocol (PCP) [162] is also an application-level protocol and it can be used by mobile operators to check mobile users' privacy policies. The privacy policies are stored in a server called Privacy Checking Entity (PCE) and the service offered by the protocol enables the mobile operator to query for the assertion of a mobile user's privacy settings prior to disclosing her location information to a third party. The mobile operator can then notify the mobile user and ask for her verification if necessary. The PCP assumes the existence of a PCE whose specifications are not available yet.

4.7.2 Parlay/OSA

The Parlay Group [202] is a non profit consortium of 65 companies from the IT and Telecom industries. They have designed the Parlay/OSA architecture [136] as well as a set of APIs to enable third parties to create telecommunication services using mobile network operators' resources. Parlay/OSA APIs give access to fixed, mobile and next generation IP-based network resources in a secure, controlled and accountable way. They are open APIs and can therefore be adopted by a large proportion of developers. However, IT developers are used to working with higher level programming interfaces and the Parlay/OSA APIs are primarily aimed at telecommunication developers. To address this problem, a sub group of the Parlay Group called Parlay X, designed the Parlay X web services API [203]. This API aims at simplifying the development of next generation network applications by IT developers who are not necessarily experts in telephony and telecommunications. The Parlay/OSA architecture's main component is a gateway whose role is to ensure that the resources made available by network operators through the Parlay/OSA APIs are not accessed by unauthorized users or applications. Applications request location information by providing the gateway with the MSISDN number of the end user, which means that applications need to be aware of

the real identity of the end users. While providing a very basic service, the Parlay/OSA initiative represents a significant first step towards the externalization of telecommunications services to third parties.

4.7.3 Geopriv

Geopriv is an IETF working group that focuses on privacy issues related to location information gathering and transfer [92]. In [151], they provide the requirements for the design of an architecture and protocol deployed over the Internet that aims at providing mobile users control on the delivery and accuracy of their location information. Their work focuses also on the definition of a *location object* that encapsulates location information as well as its associated privacy requirements. In [64], they carry out a security analysis where they explore the different threats the architecture and the protocol may encounter. Among other things, they advocate the use of short lived identities for target users and unlinked pseudonyms for location recipients to prevent location tracking. Finally, they discuss in [101] the design and the storage of policies rules produced by a component called the Rule-maker. In particular, they detail the two location data filtering strategies used to reduce location accuracy. The first one defines six different levels of precision, or "civil locations", ranging from building to country visibility levels, and taking into account the two extreme cases of full filtering and null filtering. The second strategy aims at decreasing geospatial location information by altering its raw representation. For example, this could consist of rounding up longitude or latitude coordinates to offer less accurate information.

4.7.4 The GSM Association

The GSM Association (GSMA) [99] was founded in 1987 in order to develop and foster the interests of GSM mobile operators throughout the world. In 2004, the association counts almost 660 members and still remains very active in terms of standardization activities. In particular, one of its subgroups known as the Service Group (SerG) has produced a reference document about location-based services [100]. In this document, the authors identify and outline the basic requirements for providing privacy services regarding to the collection and use of location information. They advocate the use of anonymous location information where applicable or, if necessary, the use of an *Opaque ID*, which they define as the encrypted identity of the mobile user used as a *transaction pseudonym*. They also emphasize that users should have a fine-grained control on who is entitled to use their location information. The authors also believe that the level of accuracy of location information can be used to enhance users' privacy. However they do not propose any specific algorithm and just evoke a

hierarchical model of location information where an upper layer is chosen in order to reduce the accuracy of location data. The low level architecture proposed is identical to the one presented in Section 4.7.5.

4.7.5 3GPP

The 3^{rd} Generation Partnership Project (3GPP) [199] is a collaboration agreement established in 1998 between some telecommunications standards bodies. Its aim was initially to propose technical specifications as well as technical reports in order to contribute to the design of a 3G mobile system. Its scope was later refined to include the maintenance of GSM and GPRS technical specifications. In the context of location information, [13] outlines the architecture used to retrieve and make mobile users' location information available for use (see also Sections 2.4.3 and 2.5.2 for more information). In particular, 3GPP introduces the low level architecture used within mobile operators' current architecture to provide location information to both internal and external clients. The document also details the interactions between the different entities involved.

The 3GPP research effort introduces 2 new low level components regarding to security and privacy of location information.

- The *Privacy Profile Register* (PPR). This component stores privacy profiles for target mobile users. The PPR also performs privacy checks upon GMLC requests. The privacy options available are organized in privacy classes depending on LBS requesters identities, LBS types, the serving network and allow for the positioning of a target mobile user automatically or upon user notification and potentially verification.

- The *Pseudonym Mediation Service* (PMS). This component is in charge of managing users' pseudonyms used when accessing certain LBS. 3GPP defines two types of identities for mobile end users: *verinyms* and pseudonyms. *Verinyms* correspond to identifiers used to refer unambiguously to mobile users. 3GPP advocates the use of MSISDN numbers or IMSI identifiers (see Section 2.4.1) as *verinyms*. An IP address can also be used in some cases as a *verinym*. The solution defined concerning the nature of pseudonyms is the one proposed by the GSM Association (see Section 4.7.4). A mobile user's *verinym* is encrypted with the mobile operator's public key and is used together with the corresponding PMS and GMLC addresses as a pseudonym when accessing certain LBS. The main role of the Pseudonym Mediation Service is therefore to perform identity translation between *verinyms* and pseudonyms upon GMLC requests.

Please refer to Sections 2.4.3 and 2.5.2 for further information regarding to the 3G architecture developed by 3GPP.

Chapter 5

Architecture, Infrastructure and Protocol

5.1 Introduction

This section introduces the architecture designed to enable mobile phone users to securely avail of location-based services through the Internet. It intends to satisfy some of the requirements presented in Section 3. In particular, we present a design for a middleware called the *Orient Platform* together with its protocol, namely the *Orient Protocol*, and describe their implementation. However, we do not cover the security aspects of this architecture as they are further detailed in Sections 6 and 7 of this thesis. Most results outlined in this section constitute joint work carried out with my colleague Cameron Ross Dunne [73]. While they do not represent the core of the research summarized in this thesis, they remain essential for the better understanding of the reader. The overall design presented reflects the security priorities outlined in Section 3, and lays a basis to implement comprehensive security mechanisms in order to enable mobile phone users to maintain a high level of privacy.

5.2 Architecture

The overall architecture design is presented in Figure 5.1. It differs from the architecture presented in Section 3.2.1 by the presence of a trusted third party implementing the middleware called the *Orient Platform*. In this architecture, a mobile *Subject* requests a *LBS* connecting through the *Orient Plaform* while a fixed *Subject* connects directly to the *LBS* like a normal web user. Then, if the need arises, the *LBS* may in turn connect to the *Orient Platform* in order to access some of the resources of the *Mobile Operator*'s infrastructure.

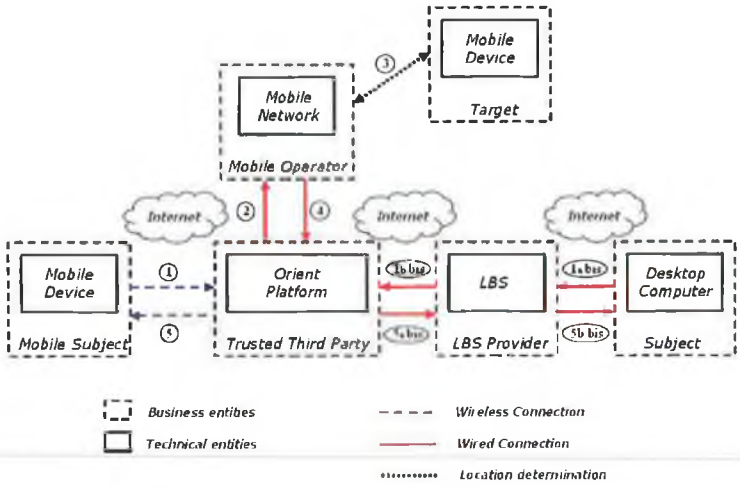


Figure 5.1: Revised General Architecture Topology

5.3 Middleware

The Orient Platform has been designed as a proxy server to fulfill the requirements stated in Section 3. We outline here the main reasons of such a choice, discuss its different components and present a prototype implementation.

5.3.1 Proxy-based approach

The Orient Platform is a middleware interfacing mobile phone users with *LBS*. One of its main roles is to guarantee mobile *Subjects*' privacy as well as the security of their communications. This involves authenticating them in order to:

- Protect their identity by shielding it using pseudonyms.
- Manage and maintain secure communications between them and *LBS*.
- Enable a reliable charging scheme involving them as well as both the *LBS* and the *Mobile Operator* they are registered with.

These operations should remain as transparent as possible to *Subjects*, especially to mobile *Subjects* for which the user interface of the mobile devices used are generally quite limited. As a result, *The Orient Platform*'s logical integration within the client/server architecture is performed by implementing it as a proxy server; see Figure 5.1. While *Subjects* connecting from a desktop computer do not directly connect to the *Orient Platform*, the latter is used later on to provide access control mechanisms to location information.

This approach has already been successfully applied to tailor web content to lightweight mobile devices capabilities [38]. In their context, the proxy server used is referred to as an "active proxy" or "transcoding proxy", as it intercepts and modifies the content of the HTTP responses to adapt to mobile devices' weak capabilities. This approach has also been used to provide location-based content to mobile users, as described in Section 4.6.2. In our context, the proxy helps provide much more transparency for *Subjects* when connecting to the Internet as it does not necessitate additional software to be installed on the client side. Furthermore, it allows the *Orient Platform* to monitor and control location-based content transmission in order to bill the appropriate entities for each transaction.

5.3.2 Components

The internal architecture of the *Orient Platform* can be described as a set of components implementing different functionalities. In this section, we describe the various units in which those components are organized.

Identity Management Unit

The Identity Management Unit (IMU) manages all forms of identity for any *Subject*, *Target* and *LBS*. There exist several kinds of *LBS* with different needs in terms of identity. For example, an *Point-of-Interest LBS* as defined in Section 1.3.2 does not need to be aware of the exact identity of a requestor. On the contrary, a *LBS* in charge of locating someone's child needs to be able to authenticate its *Subjects* in order to supply them with *Targets'* sensitive location-based information. As a result, the *Orient Platform* implements the following identification schemes:

- Anonymous access to *LBS*. This allows a *Subject* to access a *LBS* without providing her identity. In practice, this is achieved using *transaction pseudonyms* as defined in Section 4.2.

- Pseudonymous access to *LBS*. Most webmail services allow for the use of pseudonyms as usernames when setting up email accounts. Internet Messaging software such as Yahoo or MSN Messenger also provide this feature and enable people to create communities without using their real names. Since the *LBS* considered in our research are web-based, we believe that such long-term pseudonyms will also be used to identify both the *Subjects* and *Targets* of a particular *LBS*. For convenience however, we will limit *Subjects* and *Targets* to a maximum of one pseudonym per *LBS*.

- Fully-identified access to *LBS*. Banking web sites are the typical example of web sites

that necessitate to authenticate their users with their real identities. As for pseudonymous accesses, only one authenticated identity will be allowed per *Subject* and *Target*.

The Identity Management Unit' s main role is to translate the *Subjects*' real identity to the corresponding pseudonym to be used with a particular *LBS*. Also, the IMU must be capable of translating a *Target*'s pseudonym into her real identity since the latter may be used by the *Mobile Operator* to locate the *Target*.

Location Management Unit

The Location Management Unit (LMU) is the component in charge of all the processes involving location information. In particular, it looks after the task of retrieving location information for a particular *Target*. It is also in charge of translating location information from raw coordinates to more relevant and meaningful information, easily exploitable by *LBS*. Finally, the LMU is able to downgrade the accuracy of this location information to improve *Targets*' privacy using a process called *Location Blurring* defined in Section 4.3 and further analyzed in Section 6.

Location Retrieval. The *Orient Platform* provides *LBS* with secure location information. While *handset-based* or *hybrid* positioning technologies (see Section 2.5.1) may be used to locate a particular *Target*, it is likely that most location processes will involve *network-based positioning* techniques since not every device will be location aware. In any case, the *Orient Platform* has to have access to some of the *Mobile Operator*'s network resources for both location and identity information. As reviewed in Section 4.7.1, the Open Mobile Alliance (OMA) consortium has designed the Mobile Location Protocol (MLP) that enables third parties to query *Mobile Operators*' resources for *Targets*' location information. The Parlay initiative also aims at opening *Mobile Operators*' resources to authorized third parties. Althought the *Orient Platform* is more likely to be implemented within the *Mobile Operator*'s infrastructure, the middleware provides interfaces to both standards in order to enable remote access to *Mobile Operators*' resources if necessary.

Location Translation. *Mobile Operators* currently hold *Targets*' location information in terms of *Location Areas Identities*; see Section 2.5.3 for further details. However, other positioning technologies (see Section 2.5.1) may return a location information as a point, an area or even a volume. As a consequence, depending on the positioning technology used, the *LBS* requesting location details may end up with some information in a format she does not understand. Furthermore, different *LBS* may require different location formats, more relevant to the service they wish to offer. In order to cater for all the potential

LBS' needs, the *Orient Platform* provides an open interface so that location translation tables can be implemented by the *LBS* themselves and made available when retrieving location information. Of course, the *Orient Platform* provides already some default tables that enable, for example, to convert *longitude* and *latitude* coordinates into *Northing* and *Easting* coordinates.

Location Blurring. This functionality is part of the services provided by the *Orient Platform* that add value to a *Target*'s location information. Here, the value added lies in the transformation of the *Target*'s location information in order to become more private while still remaining useful to *LBS*. *Location Blurring* is the process defined in Section 4.3, that intentionally downgrades the quality of location information in order to provide *LBS* with the minimum acceptable accuracy in terms of *Targets'* position. *Targets* do not have to disclose their precise location if they believe the *LBS* is not trustworthy or if they wish to maintain a certain level of privacy. *Location Blurring* is designed as an algorithm that takes various parameters such as, for example, the accuracy of the desired location positioning. The *Orient Platform* is designed to implement different types of *blurring* algorithms, each *Target* being able to choose the one that corresponds most to the type of protection she requires. However, only one algorithm has been implemented and tested. Its design is further discussed in Section 6.

Profile Management Unit

The Profile Management Unit (PMU) provides *Targets* with the possibility to enter and modify their personal details as well as their privacy requirements into the system. They can either log on to the system from a desktop PC and set up their privacy preferences or directly do so from their mobile devices.

We present here a description of the privacy preferences provisioning component of the PMU. After having outlined the main motivations behind its conception and introduced its design, we will provide a description of its implementation and show how it relates to the other components within the *Orient Platform*.

Motivations. The field of context-aware security is relatively new. While the meaning of context as well as context aware applications have received extensive interest from the research community, very little work has focused on analyzing privacy issues and requirements in the context of mobile services. In Section 4.5.4, we describe the approach taken by Hull *et al.* [115] in order to design and implement a flexible rule engine-based access control mechanism that lets *Target* provision their privacy preferences. While a similar research

effort has also considered this idea in [88], we argue that given the conclusions of the studies mentioned earlier on, such systems can be efficiently replaced by simple server side software with database access.

Design. Our privacy engine was designed to remain as simple as possible, both from the programming and the user point of view. The design takes into account the results of three studies [32, 128, 56] that investigated the privacy requirements in context aware computing. The main ideas guiding our choices can be outlined as follows.

- In order to protect *Targets*' privacy, we consider that a *Target* wishes to remain non-locatable as long as she does not explicitly specify the contrary using the privacy engine. This avoids having *Targets* being potentially tracked by default when connecting to a *LBS* for the first time. We note that it also complies with the EU directives described in Section 1.4.2, that advocate opt-in user consent with regards to location information collection.

- The parameters enabling a *Target* to decide to what degree she wishes to disclose her location information include the identity of the *LBS* used, the identity of the *Subject*, the reason why the information is collected and her current situation. We however restrict the meaning of the last two parameters as follows. The reason for the location information collection does not reflect why a single location request is performed but more what the usage of the information collected by the corresponding *LBS* will be. As a result, upon registration with a *LBS* using the privacy engine, a *Target* is prompted with a detailed description of location information handling. As with P3P, there is no way of actually enforcing the policy. The other parameter, *i.e.* the *Target*'s situation when the request is performed, would usually encompass both *Target*'s location and time of the request. However, we chose to only retain the time of the request as a relevant parameter since we believed that the two are most of the time interlinked. This statement is supported by the fact that a *Target* usually knows where she is or will be at a particular time. Therefore, when setting up her privacy preferences, she will be able to accurately define when and where she wishes not to be tracked by only considering the time of a potential request.

- Instead of allowing a *Target* to set her privacy preferences by specifying an exact date, we restricted her choice to a weekly schedule. As reported in [56], some weekly *Target* mobility patterns exist, essentially due to working hours and daily office life. We exploit these patterns to provide *Targets* with a flexible way to set up their privacy preferences while minimizing their interaction.

- The authors of [160] point out the importance of "clusters" of *Subjects* to simplify privacy management. "Clusters" are defined as groups of entities that are treated similarly. In our context, this means that a *Target* will likely impose the same level of privacy to groups of people rather than a different one per person. As a result, the privacy engine provides pre-defined profiles (Public, Work, Family, *etc.*) that a *Target* can choose when allowing a *Subject* to indirectly look up her location.

- Finally, it is assumed that the profiles provided will be suitable in most situations. The design of the privacy engine allows however *Targets* to perform simple and quick changes on a particular profile, or, if the need arises, to create re-usable profiles from scratch. In particular, it is possible for them to create temporary profiles by specifying only the accuracy they wish to be located by. These temporary profiles can be used to override existing ones when, for example, an unexpected event occurs, which requires a *Target* to temporary obfuscate her location for example.

The use of pre-defined profiles and user interfaces used to create new profiles prevents *Targets* from declaring conflicting privacy preferences. Every time a location request is performed by a *LBS* on behalf of a *Subject*, the PMU queries the data stored by the *Target* for this *Subject* and *LBS* given the time of the query. The PMU then outputs the accuracy at which the *Target* should be located and forwards it to the *Location Blurring* algorithm of the Location Management Unit (LMU).

Implementation. The Java language was used in order to implement our privacy engine. More precisely, we used JSP to implement the backend of the web-based forms made available to *Targets*. The database of the *Orient Platform* was used to store the privacy preferences of each *Target* for each *Subject*. The user interface basically consists of multiple forms to fill in, four of them are shown in Appendix B. Upon request from a *LBS* through the LMU, the privacy engine can determine very efficiently what accuracy the *Location Blurring* algorithm implemented as part of the LMU needs to use in order to protect the corresponding *Target*'s privacy while complying with her privacy requirements. When connecting to the web interface of the PMU, *Targets* need to identify and authenticate themselves. This can be done either by using the facilities provided by the Public Key Infrastructure described in Section 7 to enable client side authentication or the traditional login/password mechanism.

Of course, we realize that our privacy engine is less flexible than the ones presented in [115] or [88]. However, we believe our approach reflects the needs of *Targets* since it is based on the results of recent studies that point out what privacy parameters really matter with

regards to location information disclosure. Furthermore, the response time of the privacy engine discussed in this section is reduced compared to the other approaches since privacy preferences are hard-coded in a database when provisioned by *Targets*. The authors of [115] recognize that hard coded privacy preferences may be preferable in some situations. We illustrated here how this approach could be efficient in our context.

Charging Unit

Following the recommendations established in the reference document [100], the charging mechanism used is based on the revenue sharing model, not only between the different players involved, but also with their peers. Indeed, a *Subject* may connect either to the network operator it is registered with or to a foreign one, while roaming for example. This means that the two network operators need to be able to share the cost of a location positioning request. Figure 5.2 illustrates a typical revenue sharing model involving a roaming *Subject*. The *Orient Platform* is considered here as not being part of the network operator's infrastructure but is very likely to be integrated by the latter.

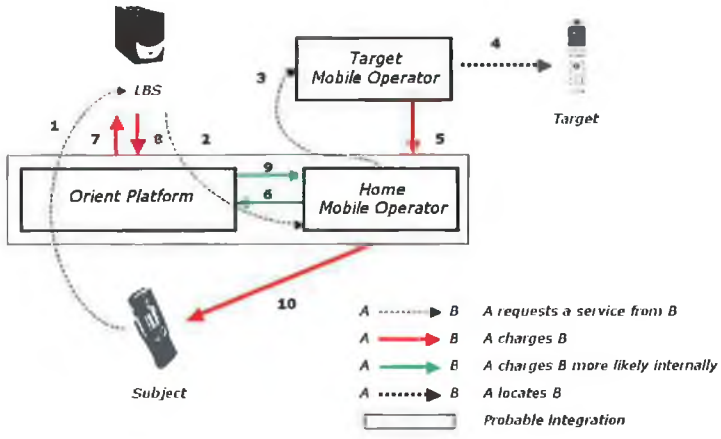


Figure 5.2: Charging Mechanism

The charge for providing location information is believed to be set up by every *Mobile Operator* individually. This charge will also depend on the positioning technology used to determine the *Target*'s location as well as on whether the *Target* is roaming or not. Thus, a possible charging scenario can be described as follows[1]:

1. A *Subject* requests a service from a *LBS* provider.

[1]This example does not consider the case where a *Subject* queries a *LBS* from her desktop computer.

2. The *LBS* contacts the *Orient Platform*, which forwards the request to the corresponding *Mobile Operator*[2].

3. The home *Mobile Operator* checks whether the *Target* is locatable within its own PLMN network and queries the relevant target *Mobile Operator* for the *Target*'s location if the need arises.

4. The target *Mobile Operator* retrieves the location of the *Target*.

5. The target *Mobile Operator* forwards the location of the *Target* to the home *Mobile Operator* and charges him the amount *pt* for the location request.

6. The home *Mobile Operator* then forwards the location information to the *Orient Platform* and charges her *ph+pt*.

7. The *Orient Platform* adds value to the location information and forwards it to the *LBS*, charging him *po+ph+pt*.

8. The *LBS* adds value to the location information by providing his service to the *Subject*, and charges the latter through the *Orient Platform pl+po+ph+pt*.

9. The *Orient Platform* forwards the charging details to the home *Mobile Operator*.

10. The home *Mobile Operator* then bills the *Subject*, pays the target *Mobile Operator pt* as well as the *Orient Platform pl+po*, which in turns pays the *LBS pl*.

We now present some comments and observations on this charging scheme. First, both *LBS* and *Subjects* should be aware of the maximum price of the total transaction. Thus, we advocate, like the GSM Association in [100], that *LBS* fix a maximum price and the accuracy for a particular location request prior to sending it to the *Orient Platform*. An estimation of the different prices *px* is given in Table 5.1[3][4].

	pl	*po*	*ph*	*pt*
Cost	20 cents	15 cents	10 cents	5 cents

Table 5.1: Estimation of the Shared Costs of a *LBS* Request.

Furthermore, if, for some reason, the location request fails, this should not be charged to either the *LBS* or the *Subject*. This charging scheme involves a lot of transactions between parties that may not fully trust each other. In order to ensure that any charging related

[2] In this section we voluntary simplify the *Orient Platform*'s actions for clarity purpose.
[3] On the basis of a *LBS* request costing 50p.
[4] As mentioned in [77] for 2003. Average price for a location request (*ph*) (O2, Vodafone, T-Mobile, Orange) in the UK.

conflict will be resolved, strong security mechanisms guaranteeing confidentiality, integrity and non-repudiation must be used in every transaction.

5.3.3 Implementation

A prototype implementation of the *Orient Platform* has been carried out using the Java language, mainly for robustness reasons. As a middleware platform, it interfaces directly with the two entities mobile *Subject* and *LBS* as a proxy. It also communicates with the *Mobile Operator* when a location request is received from the *LBS*. Finally, its PMS component is accessible to *Subjects* and *Targets* via a web interface. As a result, the *Orient Platform* prototype comprises the following components, as presented in Figure 5.3:

- A proxy server. A *Subject* may need to set up his browser preferences such that it connects transparently to the proxy when conducting any HTTP connection; see Section 5.5. The proxy server then forwards any HTTP request to the relevant web host when the request is not location-based related. On the contrary, when it involves location information retrieval, the request is filtered, modified and tunneled through a protocol called the *Orient Protocol*, further discussed in Section 5.4.

- A web server. It provides an interface for *LBS*, *Subjects* and *Targets*. Indeed, when a *LBS* is queried by a *Subject* using a desktop PC, as would any internet user do with a conventional web site, the former queries the *Orient Platform* for location information through the web interface using the *Orient Protocol*. When a *Subject* wishes to modify her personal details such as her pseudonyms used for *LBS* access for example, she contacts the PMU through the web interface. Finally, a *Target* can manage her privacy preferences by accessing the same unit through the same interface.

- A database. The database contains all the records of the identities of the different parties involved, mapping pseudonyms with real user identities. It also stores *Subjects*', *Targets*', and *LBS*' general preferences and privacy details where applicable.

- A set of protocols. The MLP or the Parlay protocol (see Section 4.7.1) is used in order to query the *Mobile Operator* for *Targets*' location information. The communications between *LBS* and the *Orient Platform* are established using the *Orient Protocol*; see Section 5.4 for further information. The protocols mentioned are implemented as Java APIs and distributed to the parties involved in their use.

In order to evaluate the different components of the *Orient Platform*, we ran a location simulator [176] whose task was to generate a set of sample *Targets* and simulate their movements. The simulator was acting as a server, just as a *Mobile Operator*'s GMLC, and was

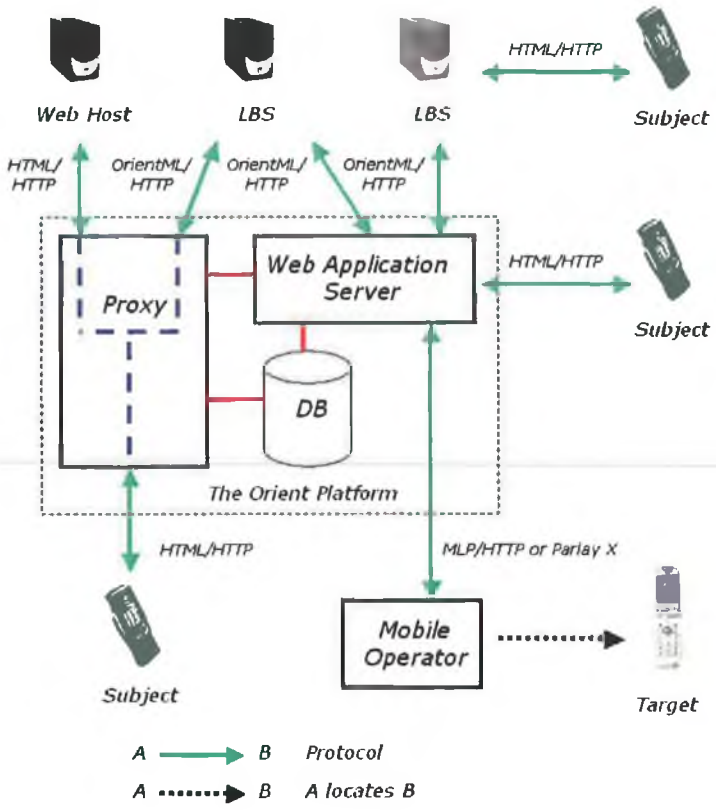


Figure 5.3: Architecture of the Orient Platform.

able to respond to queries upon *Targets*' pseudonym provision. This simulator, combined with prototypes of *LBS* developed as part of the *Orient Platform*'s project (see [73]), enabled us to implement a useful test bed for the *Orient Platform*.

5.3.4 Evaluation

The design of the *Orient Platform* presented in this section has taken into consideration the analysis presented in Section 3.3.1. In particular, the following design requirements have been fulfilled:

- *Requirement 1.* The middleware is implemented as a combination of a proxy server and a web application server. This high level design makes it easy for both *Subjects* and *LBS* to interact with the *Orient Platform*.

- *Requirement 2.* The middleware is mobile device independent, both in terms of *Subjects*' and *Targets*' equipment. The former needs only to be able to browse the Internet from her mobile device while the latter has no particular constraint imposed but to be

locatable by the *Mobile Operator*.

- *Requirement 3.* The middleware accesses directly the GMLC through the MLP or the Parlay X protocol, whether the *Orient Platform* is implemented inside or outside the *Mobile Operator*'s infrastructure.

- *Requirement 6.* A charging mechanism has been presented in this section and implemented as part of the *Orient Protocol*; see Section 5.3.2.

- *Requirement 7.* A location translation module enables the translation of the location information received from the GMLC into formats that are more suitable for processing by *LBS*. The translation table used are updatable and upgradable by *LBS*.

The *Requirements 4&5* will be further discussed in Section 5.4. Part of the security requirements also mentioned in the analysis have been met. In particular, the identity management implements an anonymous, pseudonymous and fully-identified access to *LBS*. Furthermore, a module implementing *Location Blurring* has been presented, but the algorithm itself will be further discussed in Section 6.

At this stage, the overall approach used to provide *LBS* access to *Subjects* does not represent significant architectural advances compared to what has been recently published in the related literature. The proxy approach has already been used as mentioned in Section 4.6.2. Our design differs however by the fact that the proxy is trusted, that it enables a web access to *LBS* and that it can handle multiple types of identities. It differs furthermore from the closely related research work described in Section 4.6.4 by the fact that it is not specifically meant to be used over a WAP connection and because it provides some other functionalities such as *Location Translation* and a charging mechanism. The security features of the *Orient Platform* present however some novelty that make this middleware a fairly unique tool to enable a secure *LBS* provision to mobile users over the Internet. This will be further discussed in Sections 6 and 7 of this thesis.

5.3.5 Conclusion

In this section, we introduced the design of the *Orient Platform* as a middleware that provides *Targets* with a high level of privacy and guarantees *LBS* a reliable provision of relevant location information when delivering a location-based service to *Subjects*. We then presented a prototype implementation and we refer to [73] for a critical evaluation of the overall design. We also briefly introduced the *Orient Protocol* used by both *LBS* and the *Orient Platform* to communicate with each other. The protocol will be discussed more in depth in the next section of this chapter.

5.4 Protocol

The *Orient Platform* and *LBS* need to be able to communicate using a protocol. Using an existing protocol such as HTTP is manageable but does not provide much flexibility in terms of session management and message handling. Indeed, the two parties need to exchange messages containing information about identities and location and thus keeping track of the message sequence may become problematic and demanding. Therefore, we have devised an application level protocol that can run on top of HTTP but also on UDP or TCP. We present here the resulting design and implementation of the *Orient Protocol*.

5.4.1 Design

As shown in Figure 5.3, the *Orient Protocol* is designed to allow communications between a *LBS* and the *Orient Platform*, but also between the latter and *Subjects* and different instances of the *Orient Platform*. Both *LBS* and the *Orient Platform* can initiate the communication and start a *session* during which one or more *services* can be invoked. A *service* is a sequence of at least two messages based on the request/response paradigm. The *Orient Protocol* defines four different *services*, two that can be invoked by the *Orient Platform* and two by the *LBS*:

- The *Client Initiated Service* (CIS). This service is initiated by the *Orient Platform* when a *Subject* requests a *LBS* through the proxy. The *Orient Platform* encapsulates the *Subject*'s request into a request message and forwards it to the *LBS*. The *LBS* can request further information from the *Orient Platform* before responding finally to the request.

- The *LBS Initiated Service* (LIS). This service is initiated by a *LBS* and consists basically in a simple request-response message exchange in order to retrieve a *Subject*'s location details.

- The *Profile Management Service* (PMS). It can be used by *Subjects*, *Targets* and *LBS* to update their personal details and preferences. In practice, this service invocation will be made transparently through a web interface.

- The *Inter-platform Communications Service* (ICS). This is the service that ensures that several instances of the *Orient Platform* can communicate with each other

Each *service* is made of two or more different messages. Each message shares the same basic structure composed of a header and a body. The header contains some session information, including the time the message was emitted as well as the identities of the sender

and the receiver. The body contains the actual content of a message including location requests/responses, charging details *etc.*. A complete description of the services and messages can be found in [46] and [73]. A complete protocol stack is also given in Appendix A of this thesis.

The *Orient Protocol* guarantees message confidentiality, authentication, integrity and non-repudiation. This is achieved through the use of encryption and digital signatures at the application level. Any entity involved in a message exchange authenticates herself to the other party at the beginning of a session, and agrees on a cryptographic session key. The choice of the authentication protocol used is left to the protocol initiator and is decided after a protocol negotiation. This mechanism is similar to SSL, but designed at the application level so that each party has control on the session key used and on the encrypted and signed message parts.

XML was preferred to ASN1 to specify the protocol messages, mainly because the technology is widely known and adopted in the Internet environment. A namespace called OrientML was created in order to describe each of the components of the messages. XML Schemas were used to validate the XML messages as opposed to DTDs because of their high level of reusability. Finally, the XML Encryption and Digital Signature namespaces were adopted to specify the security layer of the *Orient Protocol*.

5.4.2 Implementation

The *Orient Protocol* was implemented as a Java API. Built as a finite state machine, it generates the relevant XML message according to the state of the service and the session parameters. It handles the digital signatures and encryption processes over a specified part of the message using XML Signature and XML Encryption cryptographic libraries. The messages can then be sent over HTTP or any other transport protocol to the other party. An alternative to this approach would have been to use an existing standard, the Simple Object Application Protocol (SOAP), that provides a XML messaging framework that also supports XML Encryption and Signature. SOAP operates in a web service environment and is useful as far as RPC requests are concerned. However, it adds overheads by encapsulating messages in a SOAP envelope when the protocol only consists of sending and receiving a series of XML messages. We finally opted for an application specific XML protocol in order to implement the *Orient Protocol*. This approach still uses widely adopted standards and remains interoperable.

5.4.3 Evaluation

The requirements for the design of a protocol capable of accessing the middleware functionalities were stated in Section 3.3.2. We recall them here briefly and show how they are fulfilled by the design and implementation of the *Orient Protocol*.

Access to location information upon presentation of the right credentials. The CIS and LIS services of the Orient Protocol offer the possibility to a *LBS* to request location information of a particular *Target*. Whether the access to that information will be granted or not is left to the access control module of the Profile Management Unit of the *Orient Platform*. Thus, *Requirement 5* for the design of the *Orient Platform* is fulfilled.

Management of personal profiles for both *LBS* and *Targets*. The PMS service allows both entities as well as *Subjects* to enter, update and modify their personal profile stored in the Profile Management Unit. For convenience however, a web interface is used as a front-end and may request the PMS service transparently for its users.

Communications between different instances of the middleware. The ICS service ensures that multiple instances of the *Orient Platform* can communicate and forward requests and responses when the need arises, implementing the *Requirement 4* for the design of the *Orient Platform*.

The security requirements stated in Section 3.3.2 have also been fulfilled by using XML Encryption and XML Signature to provide confidentiality, integrity and non repudiation. Authentication is carried out at the beginning of every session and establishes a temporary session key as specified in Section 5.4.1

5.4.4 Conclusion

The *Orient Protocol* Java API has been evaluated by a group of undergraduate students, see conclusions in [73]. While the general services provided were used without any particular problem to implement *LBS*, it appears that there would be a need for a service that implements a location request based on an area as opposed to an identity. Indeed, querying an area to find out who is present at a certain place has been prevented by our design.

5.5 Integration

In this section, we show how the *Orient Platform* and the *Orient Protocol* fit in with the current *Mobile Operator*'s infrastructure. As stated previously, the *Orient Platform* can either be integrated within the *Mobile Operator*'s network or remain as a stand alone proxy server. The protocol stacks of each of the entities involved is described in Figure 5.4.

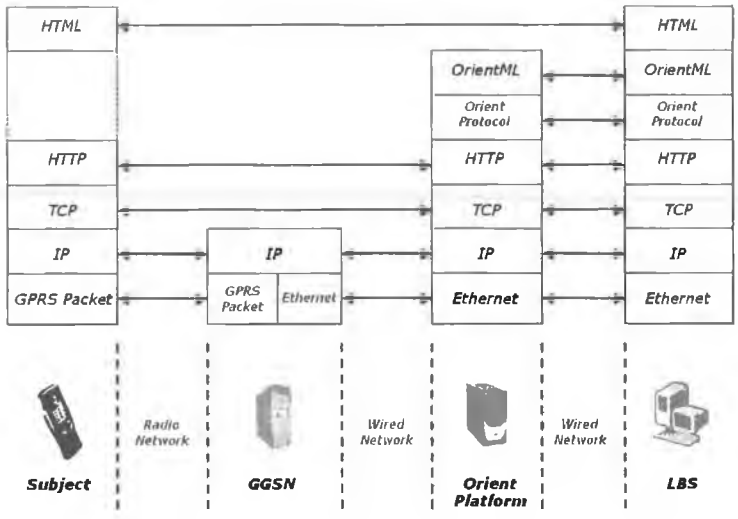


Figure 5.4: Protocol stacks of the different entities involved in location-based content provision.

The Gateway GPRS Support Node (GGSN) of the 2.5G network interfaces between the wireless and the wired networks converting the GPRS packet flow into Ethernet frames. The *Orient Platform* is implemented behind this gateway. It encapsulates *LBS*-related HTTP requests into Orient Protocol messages and forwards them to *LBS*. The latter receive the embedded HTTP requests, process them and respond by embedding the HTTP response into an *Orient Platform* message that is processed by the *Orient Platform* which sends the HTML content back to *Subjects*.

Configuration 1 : Within the *Mobile Operator*'s infrastructure

The *Orient Platform* is located within the *Mobile Operator*'s infrastructure, coupled with one of its GGSN. When a *Subject* wishes to use a location-based privacy-enhanced connection when browsing the Internet, she must configure the Access Point Name (APN) prior to establish her connection. This is an alias for the GGSN to be used. Upon receipt of the APN, the SGSN resolves the name into the IP address of the corresponding GGSN and connects to it. The latter now authenticates the *Subject* and allocates her an IP address through a

DHCP server. From now on, the *Orient Platform* acts as a proxy server, transparently to the *Subject*. It can communicate with the *Mobile Operator*'s GMLC using a protocol such as Parlay or MLP.

Configuration 2 : Outside the *Mobile Operator*'s infrastructure

When the *Orient Platform* acts as a stand alone proxy server, the *Subject* specifies the APN of the GGSN connecting to the network the proxy is implemented in. The *Subject* is then allocated an IP address either by the DHCP of the GGSN or the one from the network where resides the *Orient Platform*. Since the *Orient Platform* needs to identify and authenticate somehow its *Subjects* in order to be able to retrieve location information on behalf of both them and *LBS*, a mapping between a *Subject*'s IMSI and IP address needs to be maintained. This information resides already in the GGSN, as part of the PDP Context of the connection. In practice, infrastructures of the same nature as the *Orient Platform* (transcoding proxies like WAP Gateways for example) are supplied with this information through the use of the RADIUS protocol. This approach assumes of course that the entity running the *Orient Platform* has an agreement with the *Mobile Operator* in order to configure the APN and the GGSN such that they enable a connection as described previously.

5.6 Conclusion

The *Orient Platform* and the *Orient Protocol* presented in this section have been designed to fulfill the requirements stated in Section 3. In particular, they allow for accountable location information provision to *LBS* by implementing a revenue sharing model between the *Orient Platform*, *Mobile Operators*, *LBS* and *Subjects*. They also constitute the building blocks to provide privacy enhancing mechanisms to mobile users connecting to location-based applications. Further information on the *Orient Protocol* can be found in [73], where a detailed specification of the protocol messages and a description of the implementation choices are provided.

Chapter 6

The Location Blurring Algorithm

6.1 Introduction

The middleware presented in Section 5 provides the building blocks for the security design of an architecture that aims at delivering *LBS* to *Subjects* while preserving *Targets*' privacy. In particular, it implements identity hiding or *identity blurring* by combining the use of a web proxy and pseudonyms for both *Subjects* and *Targets*. However, several issues remain unsolved. Whilst *LBS* may not be fully aware of the real identity of a particular *Subject* or *Target*, they can easily gain more personal information by tracking them over time. *Targets* are then left with a dilemma:

- They must disclose their location details to an entity susceptible to track them if they wish to avail of a particular *LBS*.
- They must retain some of their personal location details if they wish to preserve their privacy.

Refusing to disclose enough location details may make it impossible for a *LBS* to provide a relevant service to her *Subjects*. Preserving *Targets*' privacy is therefore a matter of finding the subtle balance between location information retention and disclosure.

This section addresses this problem by proposing a *Location Blurring* algorithm that intentionally downgrades the accuracy of the location information supplied to *LBS*. Its role is to maintain a reasonable level of location privacy specified by *Targets* in terms of the size of the area they wish to be located in, while providing *LBS* with meaningful information.

6.2 Presentation

The *Location Blurring* algorithm presented in this section is implemented as part of the Location Management Unit (LMU) (see Section 5.3.2) of the *Orient Platform*. It is used to prevent a loss of privacy that may occur when *Targets*' location information is disclosed to *LBS*. For the better understanding of the reader, we first state some general definitions of terms that will be used through out this section. We then present the threat model of the environment in which the algorithm will operate and the requirements for its design. We point out how unique this location privacy feature is in the context mentioned and describe our approach after having outlined the main issues related to its design. We finally conclude on the relevance of our design.

6.3 Definitions

We understand that the terms quoted below may encompass notions and concepts that are somehow very similar. In our context however, we will clearly differentiate each of them by giving them a precise and subjective definition.

We first consider a three-dimensional Euclidian space S, where we define the following entities and concepts:

- A *Position* is a point $P \in S$, denoted by its three coordinates (x,y,z) where $x \in \mathbb{R} \cap (-180, 180]$, $y \in \mathbb{R} \cap [-90, 90]$ and $z \in \mathbb{R}^+$. A *Position* represents the real location of a particular entity.

- A *Locality* is defined as a volume V defined within S, and that contains P. A volume is defined as the amount of 3-dimensional space contained in a polyhedron or in a union of polyhedra. A *Locality* may for example define the volume whose base is delimited by a cell when the Cell-ID location positioning technology is used. Another example of *Locality*, denoted as *Precision*, can be defined as the quantification of location determination measurements errors. The *Precision* of a particular *Target* can therefore be assimilated to the kind of location information a GPS system would provide in reality. However, we will from now on consider that a GPS system returns a precise *Target*'s location and therefore, the *Precision* of this location system will be reduced to its *Position*.

- A *Locality* L_1 constitutes a *blurring* of a *Locality* L_2 if the former contains the latter, *i.e.* $\forall x \in L_2 \Rightarrow x \in L_1$. When context permits, we will use the term *Blurred Locality* to denote a *Locality* which is the result of applying the *Location Blurring* algorithm.

We note that a *Blurred Locality* contains the original *Locality* it is defined from in order to meet one of the requirements mentioned in Section 6.5, which states that a *Target*'s *Blurred Locality* should always contain her original *Position*.

- An *Instant* is a number $T \in \mathbb{N}$ defined as the absolute time measured with a certain accuracy from some epoch.

- A *Sighting* S is defined as a *Locality* together with an *Instant*. When the *Orient Platform* queries a location server for a *Target*'s location, it receives a *Sighting* as an answer. We note here that the *Instant* of a *Sighting* may not reflect the time of the query since some location servers may return the last location of a *Target* with a timestamp if they cannot currently locate her.

- A *Sighting* S_1 constitutes a *blurring* of a *Sighting* S_2 if the *Locality* associated with S_1 constitutes a *blurring* of the *Locality* associated with S_2 and if the *Instant* associated with S_1 is equal to the *Instant* associated with S_2. When context permits, we will use the term *Blurred Sighting* to denote a *Sighting* output by the *Blurring Algorithm*. Typically, a *LBS* querying the *Orient Platform* for a *Target*'s location is provided with a *Blurred Sighting*.

We note here that, for a particular *Sighting*, we only introduced the concept of *Spatial Blurring*. The bigger the *Blurred Locality* is, the more difficult it is to pin point exactly where a *Target* is located, *i.e.* a space range or volume is given rather than a clearly defined point in the space S. The same concept could have been applied to the temporal dimension, as in providing a time range r, for which a *Target* is located in the associated *Locality* at some *Instant* $T \in r$. However, we decided to focus on *Spatial Blurring* and leave *Temporal Blurring* for further research.

We now present some metrics used to quantify *Blurred Sightings*.

- *Spatial Granularity* represents the level of detail with which the *Locality* of a *Sighting* (and respectively *Blurred Locality* for a *Blurred Sighting*) is described, *i.e.* is a measure directly proportional to the volume of the *Locality*. The notion of *Temporal Granularity* cannot be defined as such since we decided not to investigate *Temporal Blurring* when designing our algorithm. However, the concept of *Temporal Freshness* may be defined as the time during which a *Locality* or *Blurred Locality* will be presented as the most up-to-date geographical area where the corresponding *Target* is located. We note that the *Temporal Freshness* is maximal at the *Instant* when the *Locality* is disclosed, and that it decreases over time.

- *Quality* refers to the degree of relevance of the location information supplied to a *LBS*. It strongly depends on her needs and is maximal when the *LBS* is able to provide an optimal service to her *Subjects* using the location information received. It can be quantified using a scale ranging from *Poor Quality* to *Perfect Quality*. *Poor Quality* would be defined as the minimum acceptable *Spatial Granularity* together with the minimum *Temporal Freshness* tolerated for any *Blurred Sighting* received. *Perfect Quality* would represent the most precise location information achievable, as in a *Blurred Locality* reduced to its *Position* together with an *Instant* corresponding to the time of the request.

- *Accuracy* illustrates a subjective notion used to describe the level of privacy of a particular *Target*. It can be described in terms of *Spatial Granularity*, *Temporal Freshness* or in terms of any parameter relevant to the *Target*, such as a civil location (i.e. "home", "workplace"). Generally, *accuracy* will be used as a synonym of *Spatial Granularity* through out this chapter.

Considering the fact that we will be studying *Targets*' mobility on the earth surface, it is legitimate to think of reducing S to a 2-dimensional space[1]. In this case, all the volumes defined become areas. In particular, a *Locality* in its simplest form may be a circle or a square. In this simple case, *Spatial Granularity* can be defined as the radius of the former or the square side of the latter.

6.4 Threat Model

A complete threat model of the architecture, with a particular emphasis on the trustworthiness of *LBS* has already been proposed in Section 3.2.2. We propose here to refine it with respect to the *Location Blurring* algorithm, and to describe it in more depth. The context of this section mainly involves two entities.

- The *Orient Platform*, which will apply the *Location Blurring* algorithm on a *Sighting*.

- The *LBS* that receives the resulting *Blurred Sighting*.

The *Orient Platform* is considered as responsible for the "security" of the *Sighting* released to the *LBS*. In other words, we assume that the location and timestamp provided to the *LBS* cannot help her gain a more *accurate* knowledge on where and when a particular *Target* is in real-time.

[1] While a *Target* may travel relatively freely on the ground, she may not, in most cases, be able to travel along the altitude axis.

It is admitted that a *LBS* can track a *Target* as accurately as the most privileged *Subject* registered with this *LBS* and authorized to do so. However, we do not assume that a *LBS* can request a *Target*'s *Blurred Sighting* on behalf of a particular *Subject* without her consent. It could indeed be detected as overcharging a *Subject* and could lead to legal complaints. Instead, we assume that *LBS* can cache *Blurred Sightings* requested by their *Subjects* and perform location tracking. Therefore, this is the responsibility of the *Target* to restrict *Subjects* from obtaining very accurate location information if she does not wish to grant full visibility to a particular *LBS* that she does not trust entirely.

Finally, we also assume that a *LBS* will not cheat and use other resources to gain a smaller *Locality* of a *Target*. Nevertheless, we believe that some of them may try to collaborate and collude with each other in order to gain more *accurate* or fresher location information about the current *Position* of a *Target*. We will therefore consider three categories of *LBS* providing different degrees of location privacy:

- *Fully Trusted LBS*. They are *LBS* that are fully authorized to track their users. Real user credentials can be used for identification and authentication and no *Location Blurring* needs to be applied to the *Sighting*. This category encompasses emergency *LBS* and services that require trust and confidentiality such as child tracking services for example. In the World Wide Web environment, services such as online banking facilities share similar properties.

- *Semi Trusted LBS*. Most *LBS* considered fall into this class of services. They are authorized to track a *Target* to the extent of her privacy preferences but are trusted not to share any identity or location information with any other entity. An analogy of *Semi Trusted LBS* in the WWW environment is the service provided by Yahoo! Inc[2]. Users are identified using pseudonyms and while they accept to be tracked using cookies or preference provisions, they trust the company not to disclose these personal details or behaviors to other entities.

- *Non Trusted LBS*. They are *LBS* that may not only track but also share *Blurred Sightings* with other entities. Typically, anonymity is used when applicable. Any web site used in the WWW environment that does not require identification or authentication may fall under this category as no formal or informal contract has been passed between the entities with regards to the identity and location information collected.

As mentioned earlier, we assume that this is the responsibility of the *Target* to ensure she is fully aware of the type of *LBS* that she authorizes to look up her location details.

[2]My Yahoo! http://my.yahoo.com.

6.5 Requirements

In this section, we first outline the requirements for the design of our *Location Blurring* algorithm and show that no approach has yet tackled such a challenge.

- First and foremost, the *Orient Platform* may indirectly use different positioning technologies to locate a particular *Target*. For example, Assisted GPS (see Section 2.5.1) may locate a *Target* with negligible positioning errors. The result of such a positioning can therefore be assimilated as a point, or, as defined earlier on, as a *Position*. The Cell-ID positioning technique associated with the knowledge of the cell planning locates a *Target* within the shape of a polygon. The *Location Blurring* algorithm defined must therefore be able to cope with the different *Localities* or geometrical location objects.

- The *Location Blurring* algorithm should support anonymous, pseudonymous and fully identified *Targets*.

- The *Blurred Sighting* representing the location details of a *Target* should always reflect the reality, *i.e.* the corresponding *Blurred Locality* should always contain her *Position*. The *Spatial Granularity* parameter of the *Blurred Sighting* may vary in order to maintain a reasonable level of privacy while providing accurate time and location information to *LBS*.

- The *Blurred Sighting* returned to the *LBS* should have the maximum accuracy requested by this *LBS* and allowed by the corresponding *Target*. If this is not possible, the *Blurring* algorithm will return an error message.

- The *Blurred Sighting* returned to the *LBS* should contain a *Locality* in which a *Target* could potentially be anywhere with equal probability.

The requirements stated above emphasize the need for a location privacy algorithm that has no equivalent in the related research work summarized in Section 4. Indeed, the approaches based on *Identity Blurring* (see Section 4.2) are not applicable in our context since *Targets* need to be identifiable in some ways by *LBS*. The *Location k-area* [97] approach presents a similar threat model as ours and represents the closest research effort to ours carried out to date, since it aims at hiding the location of a *Target* rather than her identity. Their approach is based on sensitive areas which are generated according to the fraction of the population considered that has visited the area: the less frequented areas are classified as *sensitive*. Their concept of location privacy is therefore based on the difficulty in distinguishing one person from another: the more difficult it is, the less sensitive the area is. While we respect this notion of location privacy, we believe that it should be based on

Targets' personal criteria rather than imposing the same model of privacy to all of them. A *Target* may not feel comfortable letting a *LBS* know she is in a particular place, no matter how many people surround her. Also, the *Location k-area* approach stops disclosing location information when a *Target* enters a *sensitive area*, which goes against one of our requirements. Furthermore, the authors do not provide any precise description of the algorithm such as the list of parameters influencing the frequency of the location updates, and finally it is not clear whether the system would be efficient in a semi deserted area.

6.6 Geometrical considerations

This section points out the main issues regarding the development of a *Location Blurring* algorithm. In particular, it illustrates some of the geometrical constraints encountered when designing the algorithm so that it prevents location information leaks during a continuous location tracking. It demonstrates how we define the *Blurred Locality* to be sent back to the *LBS*. Throughout this section, we will consider two *Blurred Localities* that belong to the same *Target*. As a matter of simplicity and as mentioned previously, we will use a 2-dimensional Euclidian space S and define the two *Blurred Localities*. In this context, we choose a simple *Blurred Locality* reduced to a circle of center P (its *Position*) with a radius r as an example. Another assumption requires the two *Sightings* passed as parameters to the algorithm to be consecutive in order to simulate a continuous location tracking.

Intersection issues. Intersection problems may arise when the *Target*'s speed is negligible and when two *Blurred Sightings* of the same *Target* are requested consecutively by a *LBS*. If a different *Blurred Locality* is generated at each request, we then face such a problem. As shown in Figure 6.1, the *LBS* gains more precise location details than what the *Location Blurring* algorithm is supposed to reveal, since the area covered by the intersection of the two *Blurred Localities* is smaller than either one of them.

Border issues. Bearing in mind the same configuration as mentioned above but now considering the *Target*'s speed as non negligible, we define border problems as the higher probability for a *Target* to be located either in the intersection or near the intersection of the two *Blurred Localities*. The border problem is illustrated in Figure 6.1

Intersection problems can be avoided easily by imposing a constraint on all the *Blurred Localities* returned to a particular *LBS*. As long as they remain the same or disjoints but do not overlap, the combination of these regions will not leak more information than a single

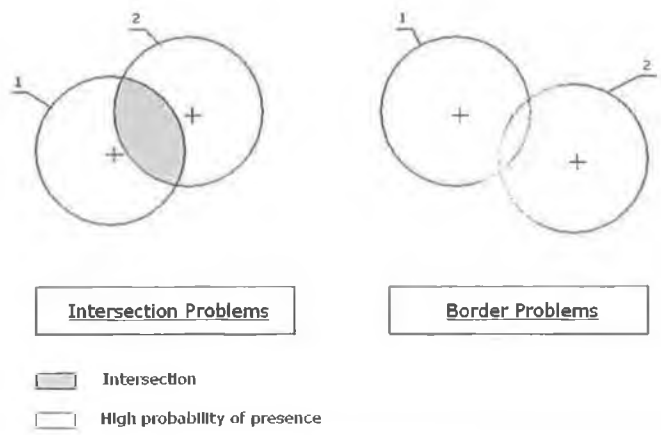


Figure 6.1: Intersection and border problems.

Blurred Locality.

Solving *real-time* border problems [3] involves working with the *Blurred Sighting* itself rather than with the *Blurred Locality* only. It basically amounts to guaranteeing that a particular *Target* can be located anywhere in the *Blurred Locality* rather than at its borders. This can be achieved by giving the *Target* enough time to potentially move to any *Position* within the *Blurred Locality* before the *Blurred Sightings* is released to the *LBS*. Solving *retrospective* border problems [4] is more delicate since it involves releasing *Blurred Sightings* that do not leak information about the past *Positions* of a *Target*, but also that will not disclose sensitive information either when associated with more up-to-date *Blurred Sightings*.

The shape of *Blurred Localities* is also of importance. As stated in Section 6.5, the *Location Blurring* must be able to reflect the reality, *i.e.* being able to release a *Blurred Sightings* for every location in a predefined subset of a 2-D space, corresponding to the PLMN of a *Mobile Operator* for example. This implies that the space considered is fully covered by predefined *Blurred Localities*. Therefore, such a collection of *Blurred Localities* must form a tiling of the plane. This can be easily achieved using shapes like squares, equilateral triangles, and regular hexagons (wireless network cells constitute an example of such a tiling).

[3] Border problems that may affect the *Position* of the *Target* at the *Instant* of the *Blurred Sighting* received.

[4] Border problems that may affect the *Position* of a *Target* at the *Instant* of a *Blurred Sighting* previously received.

6.7 Our *Location Blurring* Algorithm

We introduce here the algorithm used as part of our approach to enable *Targets* to maintain a specified level of privacy. The algorithm presented aims at solving intersection as well as *real-time* border problems. We first introduce its main principles and then provide a detailed description of its design.

6.7.1 Principles

The main principles of our *Location Blurring* algorithm follow the recommendations stated above. The area covered by the algorithm is limited to the PLMN of the *Mobile Operator* in charge of locating a *Target*. A square tiling of the PLMN has been chosen as a matter of simplicity. The side length of each tile represents the *accuracy* by which a particular *Target* wishes to be located by a particular *Subject*. For efficiency purposes, multiple overlapping grids of squares of different sizes are pre-computed. Each of these grids represents a level of privacy and each square in a grid is a potential *Blurred Locality*, as shown in Figure 6.2.

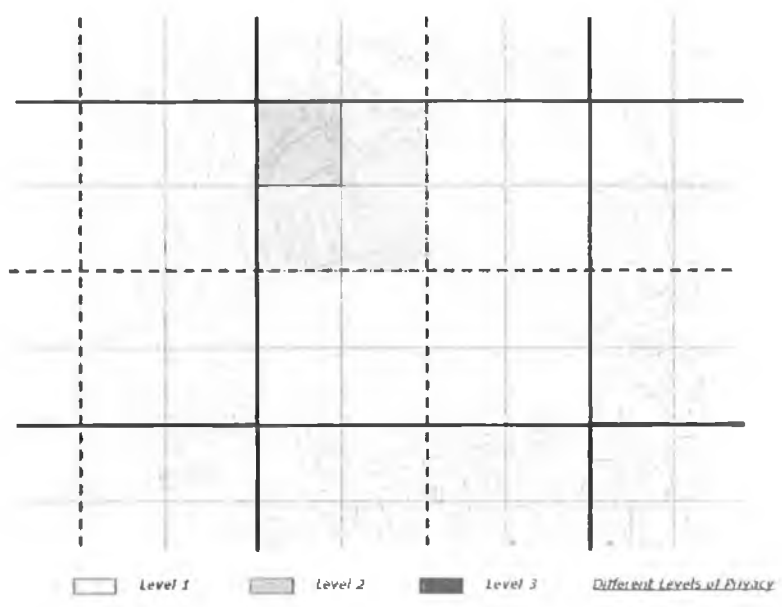


Figure 6.2: Overlapping grids.

The grid selection is performed by looking at the *Target*'s entry for a *Subject* in her privacy preferences in terms of *accuracy*. In fact, this is the entry of a *Subject*'s pseudonym for the *LBS* requesting a *Sighting* that is examined. Then, the *Blurred Locality* in which the *Target* is located is determined from the actual *Locality* of the *Target*.

While the control of *Spatial Granularity* of the *Blurred Sighting* released to the *LBS* is left to the preference of the *Target*, the *Temporal Freshness* of the *Blurred Sighting* is managed by the *Location Blurring* algorithm itself in order to prevent intersection and *real time* border problems. This parameter may depend on multiple factors such as:

- The *Target*'s *Locality*. A *Target* may wish to remain invisible to a particular *Subject* for a given period of time, which could correspond, for example, to the duration of her stay in the *Locality*.

- The *Target*'s speed. The algorithm must guarantee that the *Blurred Sighting* returned to the *LBS* contains a *Blurred Locality* in which the *Target* can be uniformly located. We would like to draw the attention of the reader on the fact that this claim will be supported by results of experiments described in Section 6.9.2.

- The *Target*'s accuracy preferences, which is of significant relevance for the same reasons as mentioned above.

The *Location Blurring* algorithm has of course access to the latest *Target*'s *Locality*. The *accuracy* preferences of the *Target* for a particular *Subject* and *LBS* are made available via the Profile Management Unit (PMU) of the *Orient Platform*. The *Target*'s average speed needs however to be calculated from real data anytime it needs to be used.

6.7.2 Description

We consider a *Target* T whose *Position* is denoted P. A *LBS* requests the *Orient Platform* at the *Instant* t for her location on behalf of a *Subject*. The *Orient Platform*'s *Location Management Unit (LMU)* then queries a location server such as a *Mobile Operator*'s for a *Sighting* representing T's location and passes it to the *Location Blurring* algorithm. The *Blurred Sighting* BS produced by the algorithm is then forwarded to the *LBS*.

Within the algorithm, we consider a data structure that reflects the past location requests made for any *Target*, specifying the *LBS* that carried it out, the *Subject* that initiated it, the *accuracy* requested by the *Target* as well as the latest *Blurred Sighting* BS_{latest} returned together with its *release time* t_{rel} ($R_{latest} = (BS_{latest}, t_{rel})$). We note here that the *release time* of a *Blurred Sighting* corresponds to the date when it was released and may be different from its *Instant* that denotes its original *Sighting*'s date of creation. A *Blurred Sighting* repository ($BSstore$) is also kept in this data structure in order to store pre-computed *Blurred Sightings* together with their future *release date* until their release or destruction when their *release date* has passed.

At a Glance

The algorithm described in the next section relies on the following principle. The *Blurring* engine will release a *Blurred Sighting* if and only if it is considered as secure, i.e if the *Target* can be located uniformly within its *Blurred Locality*. Our hypothesis for this to happen can be stated as follows : considering the instantaneous speed of a *Target*, the time it takes for her to travel the distance between the two extreme corners of the last *Blurred Locality* released and the one she is currently located in, constitutes the time delay necessary to ensure she can be located uniformly within its current *Blurred Locality*. This claim is justified by the algorithm performances in the simulation presented in Section 6.9.2 of this thesis.

In this section we note I_{now} the current time, *i.e.* the time of a particular *Blurred Sighting* query.

Iterations. The first iteration of the algorithm computes the *Blurred Locality* $BL_{current}$[5] from the *Target*'s *Position* $P_{current}$ received from the *Location Management Unit (LMU)*. It then constructs the *Blurred Sighting* $BS_{current} = (BL_{current}, I_{current})$ where $I_{current}$ is the *Instant* of the *Sighting* received from the LMU, and releases it. We note that $I_{current} \leq I_{now}$.

Subsequent iterations of the algorithm will repeat the step described above and generate a new $BS_{current} = (BL_{current}, I_{current})$. The algorithm keeps two data structures of relevance in memory: the latest *Blurred Sighting* BS_{latest} [6] that has been released to the *LBS*, and a repository $BSstore$ containing *Blurred Sightings* that were computed on previous iterations but not released. The *Blurred Sighting* released by the algorithm to a *LBS* will be one of the following candidates : $BS_{current}$, BS_{latest}, or one element of the $BSstore$.

- BS_{latest} when the *Target* is still located in the *Blurred Locality* BL_{latest} associated with BS_{latest}, whatever her speed.

- $BS_{current}$ when the *Target* is not located in the *Blurred Locality* BL_{latest} associated with BS_{latest}, but when her speed is equal to 0 [7]. Such a situation occurs when, for example, a *Target* has only moved by an insignificant distance during a large period of time since the last location poll carried out by the *LBS*. We note that this is equivalent to restarting the algorithm.

- An element of $BSstore$ for which the *release time* has passed (*i.e.* $t_{rel} \leq I_{now}$). This element is chosen such that its original *Sighting*'s *Instant* is the closest to I_{now} in order to provide the most up-to-date location information. Such an element is released

[5] A square defined within a predefined grid that contains the *Target*'s *Position*.
[6] At the second iteration, BS_{latest} is equal to $BS_{current}$ released at the first iteration.
[7] A speed lower or equal to $1km.h^{-1}$ is approximated to 0, as explained in Section 6.7.2.

when the *Target* is not located in the *Blurred Locality* BL_{latest}, but when her speed is different from 0. This corresponds to the general behavior of the algorithm, as in the general behavior of a moving *Target*.

Blurred Sighting Repository Management. The *BSstore* repository is fed with particular *Blurred Sightings* generated from *Targets' Sightings* received from the *Location Management Unit (LMU)*. These particular *Blurred Sightings* (denoted as $BS_{current}$ in the previous section) cannot be released directly to *LBS* because they would give away too much information. This is the case when the *Target* is effectively located in the *Blurred Locality* of one of these *Blurred Sighting*, but cannot be located with equal probability anywhere in it (see hypothesis mentioned in Section 6.7.2). Such a *Blurred Sighting* is valid, but releasing it would help a *LBS* exploit *real-time* border problems and compromise the *Target*'s location. Therefore, the algorithm stores it for future use together with a *release time* calculated as follows. We note that each request performed on the *BSstore* repository updates its content by removing all the *Blurred Sightings* for which the original *Sighting*'s *Instant* is lower or equal than the one of the *Blurred Sighting* returned in the corresponding response.

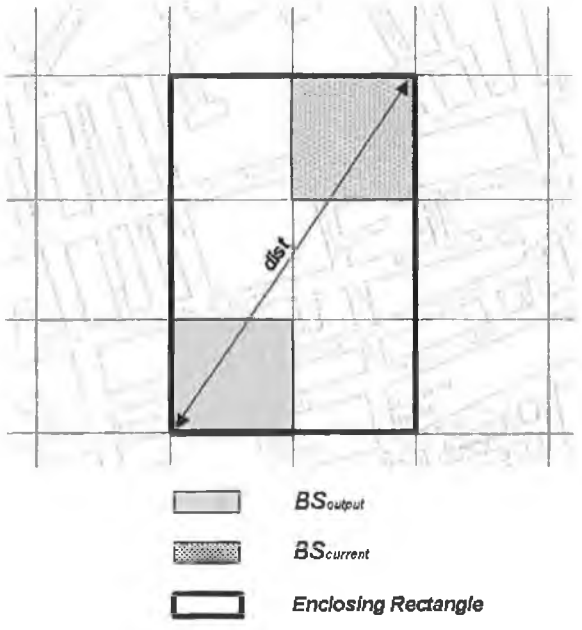


Figure 6.3: Maximal distance between any two points belonging to two *Blurred Localities*.

The algorithm is run and selects one of the three *Blurred Sightings*[8] mentioned in Section 6.7.2 to be output. We will refer to this *Blurred Sighting* as $BS_{output} = (BL_{output}, I_{output})$.

[8]The candidate selected will obviously be either BS_{latest} or one of the element of *BSstore* for which the *release time* has passed.

The original *Sighting* from which it was produced is denoted as $S_{output} = (P_{output}, I_{output})$ and is available to the algorithm. Then, the algorithm computes:

- The maximal distance between BL_{output} and $BL_{current}$ as shown in Figure 6.3, **dist**.
- The distance between P_{output} and $P_{current}$, **d**.
- The apparent speed of the *Target* between P_{output} and $P_{current}$, **speed**.
- The release time of $BS_{current}$, as being the time by which the *Target* would have traveled the remainder of the distance given it has already traveled **d** (See Equation 6.1), $dist_{remainder}$.

$$\begin{aligned} dist &= \max(\text{distance}(BL_{current}, BL_{output})) \\ d &= \text{distance}(P_{current}, P_{output}) \\ speed &= \frac{d}{I_{current} - I_{output}} \\ dist_{remainder} &= dist - d \\ releaseTime &= I_{current} + \frac{dist_{remainder}}{speed} \end{aligned} \tag{6.1}$$

At each iteration, the algorithm queries *BSstore* for the most up-to-date *Blurred Sighting* available in the repository [9]. This *Blurred Sighting* qualifies as one of the three different candidates to be released, see Section 6.7.2. This process is performed by retrieving the *Blurred Sighting* whose release time is the closest to the time of the query.

Figure 6.4 depicts a typical scenario where a *LBS* performs its first *Blurred Sighting* query while the *Target* is stationary. It illustrates also what happens when the *Target* starts moving by listing the content of the two variables R_{latest} and *BSstore* through out the process.

Algorithms

In this section, we will consider data structures X whose inner components such as private members or methods y are accessible using the operator ".". $X()$ will denote a constructor for such a data structure and finally $\leftarrow$ will be used as the store operator. We first describe auxiliary methods to finish by a detailed description of the *getBlurredSighting()* function.

[9] Except for the first iteration since *BSstore* is empty.

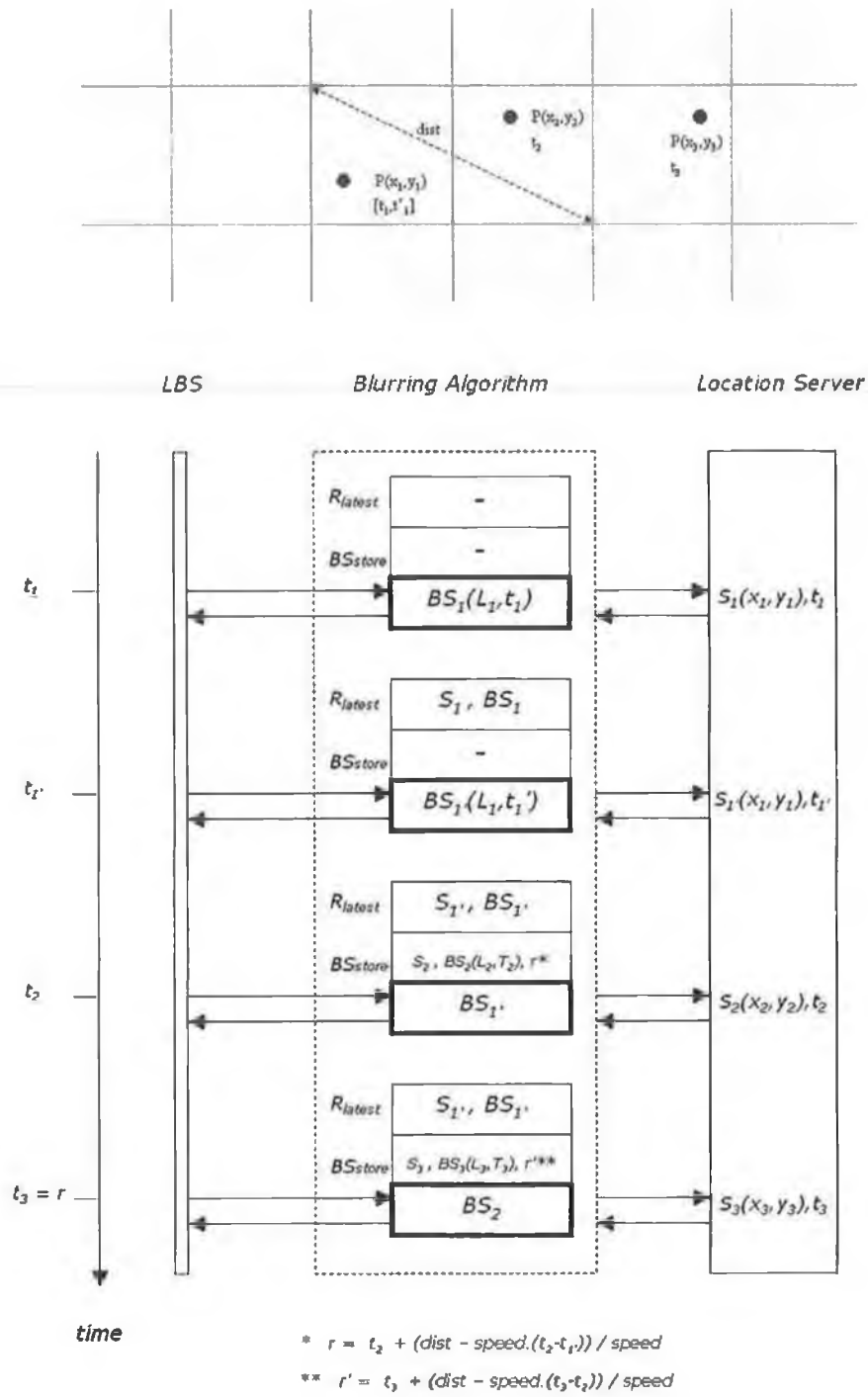


Figure 6.4: Location Tracking through the use of the *Location Blurring* algorithm.

The *getMUTDBlurredSighting()* method. This method supposes the availability of the $BSstore$, which is a repository where some *Blurred Sightings* to be released at a latter date than the current one are stored. The most up-to-date *Blurred Sighting* (MUTDBlurred-Sighting) $BS_{upToDate}$ is retrieved from $BSstore$ using a method described in Algorithm 1. It is chosen such that its *release time* is the closest lower or equal to the current time, *i.e.* $t_{rel} \leq I_{now}$.

Algorithm 1 getMUTDBlurredSighting(I_{now}, R_{latest})

Require: Access to $BSstore$, I_{now} and R_{latest}
Ensure: The most up to date (MUTD) *Blurred Sighting* is returned.
  $result \leftarrow R_{latest}.BS$
  while $BSstore.hasMoreBlurredSightings()$ **do**
    $(S_{temp}, BS_{temp}, releaseTime_{temp}) \leftarrow BSstore.next()$
    $mostRecentReleasable \leftarrow 0$
    if $(releaseTime_{temp} \leq I_{now})\&\&(mostRecentReleasable \leq releaseTime_{temp})$ **then**
      $mostRecentReleasable \leftarrow releaseTime_{temp}$
      $result \leftarrow BS_{temp}$
    end if
  end while
  $BSstore.update(result)$
  return $result$

The *update()* method. A call on the method *update* (see Algorithm 2) can be performed at any time in order to sort and discard unnecessary or outdated *Blurred Sightings* in the $BSstore$. This is done by removing any entry whose initial *Sighting* was generated prior to the *Blurred Sighting* to be released. Indeed, all these entries reflect the location of the *Target* before the released *Blurred Sighting* was generated. They have thus become out of date.

Algorithm 2 update($BS_{current}$)

Require: Access to $BSstore$
Ensure: $BSstore$ only contains records created after $BS_{current}$
  while $BSstore.hasMoreBlurredSightings()$ **do**
    $(S_{temp}, BS_{temp}, releaseTime_{temp}) \leftarrow BSstore.next()$
    if $((BS_{current}.I \geq BS_{temp}.I)$ **then**
      $BSstore.removeBlurredSighting(S_{temp}, BS_{temp}, releaseTime_{temp})$
    end if
  end while

The *getLongestDistance()* method. This method computes the distance between the farthest away points of two *Blurred Localities*, here $BS_{upToDate}.L$ and $BS_{current}.L$ ($dist$). We note that this method returns 0 when the two *Blurred Localities* passed as arguments

denote the same area. Indeed, if a *Target* is said to be in a certain *Blurred Locality*, she can be anywhere in it and therefore, in any of its corners.

The *getSpeed()* method. In the algorithms presented, the speed of the *Target* is used to predict whether she will be able to reach any location within a *Blurred Locality*. However, the value calculated remains an approximation since it is evaluated from the *Positions* and *Instants* of the two *Sightings* generated by the location server following two consecutive location queries from a *LBS*. The traditional expression of the speed ($distance = speed \times time$) can therefore no longer be used when two consecutive queries are separated by a significant period of time while the *Position of the Target* only varies by a few meters [10]. Indeed, this would result in the computation of a *Blurred Sighting* whose release date grows significantly due to a very low estimation of its apparent speed. In order to overcome this problem, we propose to impose a threshold on the estimated speed of a *Target*. If she is traveling at less than $1km.h^{-1}$ (bearing in mind that the slowest average speed considered is the one of a walking *Target*, which is approximately $5km.h^{-1}$), the speed is set to $0km.h^{-1}$ and this reinitializes the algorithm. One side effect of this strategy is that it cuts off the path of a *Target* that starts traveling along a loop at the time of a first query and that comes back to where she started at the same time as the second location query occurs. We note that this is also a general behavior of the algorithm and that it is beneficial since it hides part of a *Target*'s path while remaining truthful and efficient in terms of *Blurred Sighting* freshness.

The *getBlurredSighting()* method. The *Blurring Algorithm* is activated by a call on the *getBlurredSighting* method (see Algorithm 3). This method first retrieves the current location of the *Target* T and constructs the corresponding *Blurred Sighting* from the *Position* and the *Instant* of the *Sighting* received. This *Blurred Sighting* may or may not become the one released to the *LBS*, depending on its tracking history. If such history exists, and if the accuracy used is the same as the parameter a, then the algorithm retrieves it as the following tuple : $(BSstore, R_{latest})$. If there is a tracking history but with no record using the accuracy a, then only the last *Blurred Sighting* is retrieved and $BSstore$ is initialized to an empty array. Finally, when no tracking history exists, the algorithm initializes itself by constructing the relevant data structures according to the side length a of the *Blurred Localities* to be used and provided as part of the *Target*'s privacy preferences taken from the privacy engine, see Section 5.3.2. $BS_{upToDate}$ is then retrieved using the *getMUTDBlurredSighting()* method. The distance *dist* between $BS_{upToDate}.L$ and $BS_{current}.L$ is computed using the

[10] This could easily happen when a *LBS* queries for the location of a *Target*, once a day, at a certain time during the day, when the *Target* is likely to be located at the same place.

getLongestDistance() method and the approximate speed of the *Target* is obtained by a call on the *getSpeed()* method.

We now consider three different cases :

- If the distance *dist* is equal to 0, this means that the *Target* has not moved from her previous *Blurred Locality*. In this case the algorithm outputs $BS_{upToDate}$. We note that $BS_{upToDate}.L$ and $BS_{current}.L$ represent the same area, however, $BS_{upToDate}.I$ is lower than $BS_{current}.I$. By not releasing $BS_{current}$ we prevent the *Target* from being tracked as soon as she quits $BS_{current}.L$. Indeed, a *LBS* performing a continuous location tracking on a *Target* would notice that, when the *Target* stays in the same *Locality*, a different *Instant* is released for every query while as soon as she quits it, the same *Blurred Sighting* is returned. As soon as the *LBS* gets two identical consecutive *Blurred Sightings*, he would be in a position to locate the *Target* near the borders of the corresponding *Blurred Locality*.

- If the distance *dist* is different from 0 but the speed is equal to 0, this means that the *Target* has traveled since the last *Blurred Sighting* query, but that the algorithm needs to be reinitialized since the apparent speed of the *Target* is null. This corresponds to the case where the location tracking process had stopped and starts again. The previous stored *Blurred Sightings* are out of date and the very low speed would engender non relevant other. $BS_{current}$ is therefore released.

- if neither the distance nor the speed equals 0, it is still uncertain whether the $BS_{current}$ or $BS_{upToDate}$ will be released. In order to decide if $BS_{current}$ is the good candidate, the maximum distance achievable by the *Target* at her current speed, departing from the furthest corner of $BS_{upToDate}.L$ is compared with *dist*. If the former is lower than the latter, this means that $BS_{current}.L$ is not ready yet to be released as the *Target* needs more time to, in theory, be able to reach any of its *Positions*. As a consequence, $BS_{current}$ is stored together with the release time at which the *Target* may have covered the distance mentioned above. $BS_{upToDate}$ becomes therefore the released *Blurred Sighting*. On the contrary, if the *Target* has traveled far enough, $BS_{current}$ can be released to the *LBS*. This *Blurred Sighting* is the most accurate a *LBS* can be issued since it was created on the fly from its original *Sighting* without any delay regarding to its *Instant*.

Following every *Blurred Sighting* release, a call on the *update()* method ensures that *BSstore* contains only relevant *Blurred Sightings*. A description of the *getBlurredSighting()* method is given in Algorithm 3.

Algorithm 3 getBlurredSighting(lbs,T,a)

Require: T is the *Target* for which the location should be blurred. T must be locatable, R_{latest} contains the latest *Blurred Sighting* released together with the original *Sighting* it was made from. a represents the side length of the *Blurred Localities* chosen as a privacy preference. lbs is the name of the *LBS* carrying out the location request.*subject* is the name of the subject on behalf of which the location request is performed.

Ensure: The latest (most secure) *Blurred Sighting* is returned.

$I_{now} \leftarrow getCurrentTime()$
$S_{current} \leftarrow getSighting(T)$
$L_{current} \leftarrow getLocality(S_{current}.P)$
$I_{current} \leftarrow S_{current}.I$
$BS_{current} \leftarrow BS(L_{current}, I_{current})$
if $(hasEntry(subject, lbs, T))$ **then**
  if $(hasEntry(subject, lbs, T, a))$ **then**
    $(BSstore, R_{latest}) \leftarrow getEntry(subject, lbs, T, a)$
  else
    $(BSstore, R_{latest}) \leftarrow ([\sim], getR_{latest}(subject, lbs, T))$
  end if
else
  $(BSstore, R_{latest}) \leftarrow ([\sim], R(S_{current}, BS_{current})$
end if
$(S_{upToDate}, BS_{upToDate}) \leftarrow BSstore.getMUTDBlurredSighting(I_{now}, R_{latest})$
$dist \leftarrow getLongestDistance(BS_{upToDate}.L, BS_{current}.L)$
$speed \leftarrow getSpeed(S_{current}, S_{upToDate})$
if $(dist = 0)$ **then**
  $BSstore.update(BS_{upToDate})$
  $R_{latest} \leftarrow R(S_{upToDate}, BS_{upToDate})$
  return $BS_{upToDate}$
else
  if $((dist > speed \times (I_{current} - BS_{upToDate}.I)) \&\& (speed \neq 0))$ **then**
    $releaseTime \leftarrow I_{current} + \frac{(d - speed \times (I_{current} - BS_{upToDate}.I))}{speed}$
    $BSstore.addBlurredSighting(S_{current}, BS_{current}, releaseTime)$
    $BSstore.update(BS_{upToDate})$
    $R_{latest} \leftarrow R(S_{upToDate}, BS_{upToDate})$
    return $BS_{upToDate}$
  else
    $BSstore.update(BS_{current})$
    $R_{latest} \leftarrow R(S_{current}, BS_{current})$
    return $BS_{current}$
  end if
end if

6.8 Implementation

The *Location Blurring* algorithm is implemented as part of the Location Management Unit of the *Orient Platform*. Like for the other components, the Java language was used mainly because it achieves robustness and because it constitutes the programming environment used throughout the project. We also developed a testing framework in order evaluate the algorithm's behavior, assess its security and test different versions of its design. We note here that while the design of the algorithm remains relatively general, its implementation and evaluation was restrained to the geographical boundaries of Ireland as justified later on. This section introduces this framework and highlights specific issues regarding to the implementation of the algorithm.

6.8.1 Environment

The *Orient Platform* prototype is fully implemented and its functionalities have been tested against proof-of-concepts *LBS* in our test bed environment, see *cameron*. However, we cannot expect to have access to real time data such as *Targets*' *Sightings*, simply because this would require an agreement with a mobile operator and location-based services to be deployed. Instead, we decided to gather real location information independently from the *Orient Platform* project and use it to evaluate our *Location Blurring* algorithm.

Data

We primarily used a GPS receiver connected to the serial port of a laptop computer. This device was given to a set of participants for a certain period of time and these were asked to remember approximately their activity while the GPS system would track their outdoor movements. We also used a PDA and a mobile phone, both of them carrying a GPS sensor, as well as a wrist GPS device. We gathered the equivalent of 210 000 *Sightings* and grouped them by *sessions* subdivided into *activities*. A *session* is defined by the period during which the locatable device was allocated to a *Target* while an *activity* represents a *Target*'s specific mobility pattern such as "commuting", "hiking", "doing the shopping", *etc.* For each *activity*, we also logged the mode of transport used as well as a general description that reflected the context of the *Target*'s path. The data gathered can thus be described as forty *sessions* subdivided into a sixty *activities* corresponding to the mobility traces of ten different *Targets* tracked around Ireland and more precisely in the Dublin area. The *Targets* considered were four female and six male individuals respectively aged 20,23,24,58 and 23,27,27,33,45,58. Their *Sightings* were recorded during disjoints periods of a week (from monday to sunday) and their *activities* mainly involved commuting and walking around in

town. The *Targets* were given the possibility to turn off their tracking device or even delete some data recorded during periods they would consider as privacy-invasive. However, none of the *Targets* expressed any interest in the latter option while they turned on the tracking device everytime they thought it would acquire relevant data. The data was collected at different frequencies, depending on the capabilities of the tracking devices, ranging from once every five seconds to once every minute. We visualized most *activity* traces using a graphical interface and discarded obvious "off-course" *Sightings* resulting from the lack of precision of the tracking device at that particular time. These mainly represented data acquired at the beginning of a *session*, when GPS-based tracking devices need a certain amount of time to initialize, or when sattelites were not directly visible to GPS tracking devices [11]. All data collected was stored in a database following a schema reflecting the different groupings mentioned earlier on.

Framework

A testing framework was designed to evaluate various *Blurring* algorithms in the exact same conditions as if they were run within the *Orient Platform*. Since such algorithms would not be used in conjunction to real time data, there was a need to simulate a consistent release of *Sightings* with regards to the time flow. We therefore decided to implement our testing framework based on a discrete event system simulator. Such a simulator reflects the dynamic behavior of a system by controlling the time advance using a centralized clock. In our context, the system can be described as a set of *Targets* releasing *Sightings* at different *Instants*. Figure 6.5 shows a detailed description of the system whose processes can be outlined as follows.

1. The *Sightings* being stored in a database, an entity known as the *Location Server* makes the relevant records available upon query on a particular *session*.
2. The *Simulation Executive* then collects the *Sightings* from the *Location Server* and starts ticking the clock at a pre-defined frequency.
3. The location tracking process is initiated when the *Location Tracker* entity starts querying the *Blurring Component* entity for a particular *Target*'s *Blurred Sightings*.
4. The *Blurring Component* in turn queries the *Simulation Executive* which sends back the latest *Sighting* available for the *Target*, according to the current time.
5. The *Blurring Component* then applies the *Blurring Algorithm* on the data received and sends it back to the *Location Tracker* entity.

[11] This usually happened when a *Target* entered a building or was traveling close to high buildings

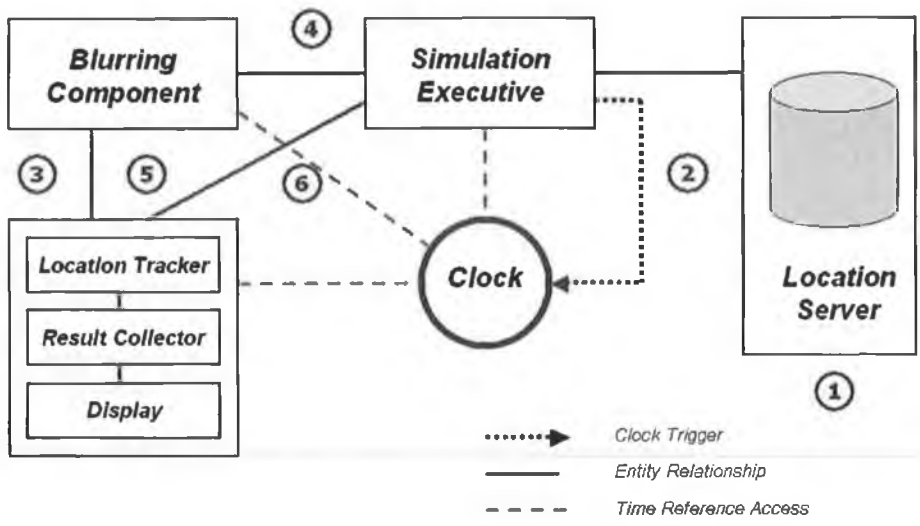


Figure 6.5: *Location Blurring* Algorithms Testing Framework.

6. While querying the The *Blurring Component*, the *Location Tracker* also queried the *Simulation Executive* for a copy of the original *Sighting* in order to help the *Result Collection* component perform experiments regarding to the efficiency and security of the algorithm. The results of the experiments can then be displayed by the *Display* sub component.

In order to reflect the reality of location tracking, the *Simulation Executive* implements a time slicing approach of time management. Time is incremented by a factor that can help speeding up or slowing down the time flow. This time management strategy was preferred to the "next event" approach were the *Simulation Executive* would jump in time in order to reach an *Instant* when a new *Sighting* would be released. The main reason is that, while being time efficient, the latter approach makes it more difficult to analyze the results of experiments, especially when they have to be displayed in real time on a graphical interface for example.

6.8.2 Algorithm

The *Location Blurring* algorithm was implemented within the framework described above. As it involves computations over distances, areas and *Target*'s speeds , there was a need to consider a bounded flat surface in order to perform accurate measurements. Most of the *Sightings* gathered having their WGS84 coordinates [210] contained within the boundaries of Ireland, we opted for using the Irish Grid plane co-ordinate system [164] as a projection

plan. In order to the get a sense of the output of the algorithms tested, we also implemented a graphical interface.

6.9 Evaluation

This section presents an evaluation of the *Location Blurring* algorithm. We first discuss the security provided by the algorithm and then evaluate its efficiency in terms of *Quality*.

6.9.1 Security

The *Blurring Algorithm* is made publicly available as we saw throughout this thesis that security by obscurity was not an option. Therefore, we must ensure that it constitutes a true one way function in the sense that the *Blurred Sighting* output should not leak any identity, time or location information that could enable an eavesdropper to gain a more precise *Blurred Sighting* than the one he was given, at real-time. A more precise *Sighting* is either:

- A *Blurred Sighting* whose *Blurred Locality* is smaller than the original one's but still reflects reality.
- A *Blurred Sighting* whose *Instant* is higher than any of the *Blurred Sighting*'s *Instants* already received but still reflects reality.

Attacker Model

The attacker is modeled as a *LBS* attempting to track a *Target*'s movements. The attacker can query the *Orient Platform* and therefore the *Blurring algorithm*, at arbitrary intervals. This way, the *LBS* can get newly generated *Blurred Sightings* as soon as they are available.

Attacks

Real-time Border Issues. The algorithm releases a *Blurred Sighting* to the *LBS* when the *Target* it is trying to protect may be located uniformly anywhere within the corresponding *Blurred Locality*, according to the hypothesis formulated in Section 6.7.2. Therefore, if the *Target* travels at a sufficient speed, she may have already left the *Blurred Locality* when the *LBS* receives the *Blurred Sighting*. Indeed, the *release time* of a *Blurred Sighting* is calculated over the maximum distance that a *Target* can achieve between the two *Blurred Localities* concerned. Also, if the *Target* accelerates in between two *Blurred Localities*, there is even a higher probability that she is outside the area returned to the *LBS* when the

Blurred Sighting is released. However, at this time, it is not possible to infer that the *Target* has sped up since the speed taken into account in order to compute the release date of the *Blurred Sighting* is lower than the one of the *Target* when she reaches the border of the *Blurred Locality*. Also, the projected *Blurred Sightings* [12] are made irrelevant since they are always overridden by *Blurred Sightings* created on the fly from the real *Sighting* received. If the *Target* decelerates, the projected *Blurred Sightings* will be released one after the other and if the *Target* stops, a *Blurred Sighting* with the same *Instant* will be released for every *LBS* query. This avoids disclosing the fact that the *Target* still resides in the *Blurred Locality* or has left it. As a result, acceleration and deceleration do not seem to disclose any extra identity, time or location information enabling the attacker to exploit *real-time* border issues (see Section 6.6).

Retrospective Border Issues. Concerning *retrospective* border issues however, it is easy to see that an attacker can infer a smaller *Blurred Locality* of a particular *Blurred Sighting* using, in some circumstances [13], the next *Blurred Sighting* received. This is done by computing the distance d_{sep} separating the two successive *Positions* of the *Target* from the Equation 6.1. The disc whose diameter is d_{sep} can then be used to work out the region where the *Target* is the most likely to be.

Intersection Issues. Intersection issues are prevented by the design of contiguous *Blurred Localities*, see Section 6.6. However, such issues may arise due to a change in the *Blurred Sighting*'s *Spatial Granularity* released for a particular *Target*. Such a change may occur at a time t when a *Target* sets her privacy preferences in order to be located at an accuracy lower than the one used before t. If a *LBS* is performing a continuous location tracking on that particular *Target* at this time, the intersection of the two consecutive *Blurred Localities* may reveal a smaller area than either of them, enabling the *LBS* to locate the *Target* more precisely than she is allowed to. Therefore, when it detects a decrease in the *accuracy* used from a_1 to a_2 [14], the algorithm computes the next *Blurred Sighting* using a_2 but keeps releasing the latest *Blurred Sighting* stored in R_{latest} until the former becomes valid. This way, even though there might have been an intersection between the two *Blurred Localities* consecutively released, the time delay introduced in between the two releases removes intersection issues.

[12] *Blurred Sightings* produced with a release date greater than the current date.
[13] When the *Blurred Sighting* has not been repeatedly transmitted.
[14] When there are records of location tracking for a particular *Target* but not with the same accuracy.

LBS Collusion. As seen in Section 6.4, we consider *LBS* that may differ from each other by the level of trust put into them. Thus, *LBS* that belong to the *Fully Trusted LBS* category do not require any *Location Blurring* since they are completely trusted not to misuse location information. *Semi Trusted LBS* are provided with *Blurred Sightings* but are trusted not to share them with other entities. *Non Trusted LBS* may share such a piece of information. They should therefore be the most penalized in terms of location information *Quality* since they may cheat and disclose private details to non authorized entities. By colluding, two *Non Trusted LBS* can increase their knowledge of a particular *Target*'s location:

- Firstly because if they query the *Orient Platform* successively and associate their results at real time, they can gain more relevant information (they indeed get a *Blurred Sighting* with the highest *Temporal Freshness*). They may thus be in a position to estimate more precisely the location of a *Target* since they know she is located close to the border of the two *Blurred Localities*. See Figure 6.6 for more details on this *Border* issue.

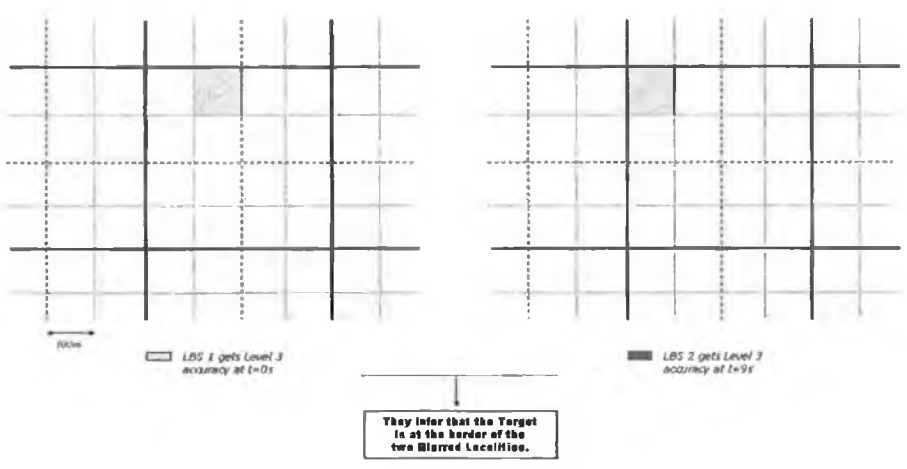


Figure 6.6: Non Trusted LBS Collusion with same *Blurred Sighting* accuracy.

- Secondly because they may cache *Blurred Sightings* requested by different *Subjects* requiring different levels of *accuracy* for the same *Target* at the same time. They may also share this information with other *LBS*. Only the *Non Trusted LBS* that had the less knowledge gains more knowledge in this case. See Figure 6.7 for more details. This is a natural consequence of *Intersection issues*.

Of course, *Non Trusted LBS* may also exploit *retrospective* border problems to infer more precise past location information. In order to prevent collusion between *Non Trusted LBS*,

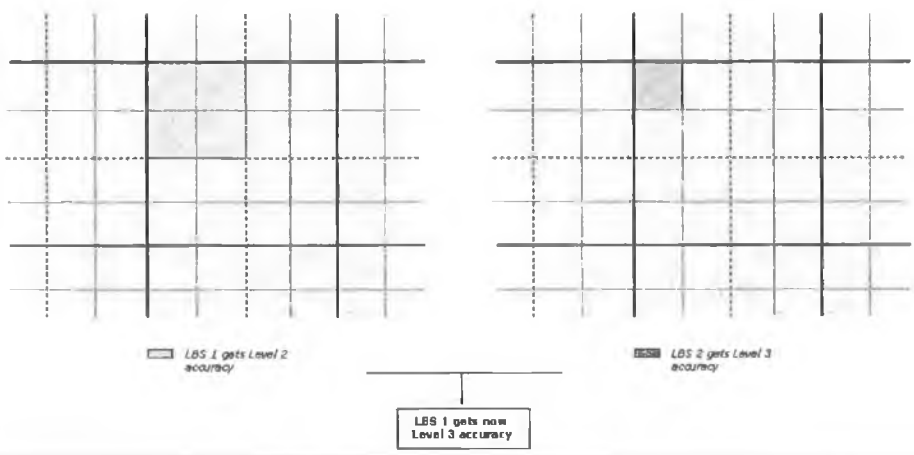


Figure 6.7: Non Trusted LBS Collusion with different *Blurred Sighting* accuracy.

one solution may consist in providing them with the same location information. This would imply providing them with the same *Blurred Sighting* for a predefined period of time and update this *Blurred Sighting* every time a new period starts. We note that while lowering considerably the location information *Quality*, it solves *retrospective* border problems since a *Blurred Sighting* is no longer released according to the speed of the *Target*. However in order to make sure this does not breach one of the algorithm requirements, *i.e.* using the maximum accuracy allowed by a *Target* for a particular *LBS*, the *Blurred Locality* must be chosen as the maximal one in terms of area amongst all the ones required by *Non Trusted LBS*. Yet, we notice that none of these algorithms provide any real solution to the *Location Transfer* problem.

6.9.2 Accuracy

In this section, we attempt to demonstrate that our algorithm performs with reasonable efficiency. To do so, we conducted two experiments on three different sets of *Sightings*, which differ from each other by the mode of transport used by the *Target*. Each set represents ten activities. On each of these sets, we applied our *Location Blurring* algorithm for an *accuracy* value [15] that ranged from 10 to 40000 meters, and set the tracking frequency to one query per minute.

The first experiment aimed at evaluating the distance between the *Blurred Sighting* returned and its original *Sighting* in order to assess the *Quality* of the location information provided to *LBS*. We recall here that a *LBS* always receives location information with a

[15] The side length of the *Blurred Locality*.

Spatial Granularity equal to the side length of the *Blurred Locality*. However, the algorithm introduces a delay (*Temporal Freshness*) in the distribution of *Blurred Localities* in order to protect the *Target*'s privacy. Therefore, at an *Instant t*, a *Target* may be located at a certain distance of the borders of the *Blurred Locality* returned to the *LBS*. If this distance is too significant, the *Quality* of the location information provided to *LBS* becomes weak since it does not reflect a relevant approximation of the current *Target*'s location. The results of this experiment can be found in a graph present in Figures 6.8(a), 6.9(a) and 6.10(a). Considering that the *Target* is responsible for choosing a relevant *accuracy* for her *Blurred Sightings* in order to preserve her own privacy, we claim that a decent level of location information *Quality* is reached when the distance separating a *Sighting* from its blurred version does not exceed the value of its *accuracy*.

By analyzing the results, we note that this level is reached for an *accuracy* greater than 800 meters for both a cycling and a driving *Target*. For a walking *Target* though, the algorithm seems to perform a little bit better since it achieves this level for *accuracies* greater than 100 meters. We also note that the distance between a *Sighting* and its blurred version slightly decreases until the accuracy reaches a certain value (50 meters for a walking *Target* and 100 meters for a cycling and driving *Target*), and this, on the three graphs presented. This can be explained by bearing in mind that we calculate the distance between a particular *Sighting*' *Position* and the closest side of the corresponding *Blurred Locality*: the bigger the *Blurred Locality* is, the closer it is from the *Position*. At some stage, when it is big enough, it can even contain this particular *Sighting*' *Position*.

This last fact partially explains why the average distance between a *Sighting* and its blurred version augments after a certain limit (here 200 meters on the three graphs): there is indeed now less *Sighting*' *Positions* outside their *Blurred Localities* than before. This can be observed by noticing how the significant difference between the minimum and maximum distance between *Sightings* and *Blurred Sightings* in the high accuracies influences the average value of this distance. The other factor influencing this increase is the fact that the distance *dist* used to calculate whether a *Target* is able to reach any point in the next *Blurred Locality*, see Section 6.7.2, becomes disproportionate compared to the real distance traveled by the *Target* [16]. As a consequence, the release date of the next *Blurred Locality* is slightly overestimated, which leaves more time to the *Target* to travel further. This phenomenon is observed at least twice in every graph.

Finally, we note a similar behavior of the algorithm for both driving and cycling *Targets*. This can be explained first by the fact that the average speed of a cyclist and a car in the

[16]The distance between the two furthest corners of two *Blurred Localities* is indeed closer to the real distance traveled by a *Target* when the accuracy chosen is small. In this case, the *Blurred Localities* can be approximated to the *Position* of the *Sightings* they have been generated from.

Dublin area is higher than for a walker (respectively 42 km.h^{-1}, 20 km.h^{-1} and 5km.h^{-1} for non stationary entities). Also, a cyclist is less subject to traffic jam than a driver, which may compensate in a way the fact that he travels slower. The behavior of the algorithm for a walking *Target* presents similarities with the two other kinds of *Targets* considered in the low accuracies. However, it clearly hides any *Targets*' movement when the *accuracy* used gets too high compared to the straight line distance actually traveled. As a result, one should not consider the results for accuracies greater than 10000 meters as relevant, for a walking *Target*.

The second experiment illustrates the average *Temporal Freshness* of *Blurred Sightings* depending once again on the *accuracy* decided by the *Target*. Three graphs represent here again the results and can be found in Figures 6.8(b), 6.9(b) and 6.10(b). A qualitative analysis shows a phase whereby the *Temporal Freshness* remains fairly constant on average for *accuracies* lower than 100 meters (driving *Target*) to 200 meters (cycling *Target*). This reflects the comments stated earlier on as in the distance *dist* used to calculate the next *Blurred Sighting* remains the same while the distance between the current *Sighting* and the last *Blurred Locality* released decreases. The graph obtained for the walking *Target* presents many more up and down variations but still illustrates the same process as described above.

This series of graph reflects in general the same conclusions as the first series but they also give a quantitative estimation of the typical *Temporal Freshness* of the *Blurred Sightings* released. Thus, we notice that the average *Temporal Freshness* recorded slightly exceeds 10 minutes for a driving *Target* and 6 minutes for a cycling *Target*. Concerning a walking *Target* though, it can reach 50 minutes but may on average be close to 17 minutes. These results are to be carefully considered since the more the accuracy grows the less likely it is that the *Target* is going outside its current *Blurred Locality*. This is especially true in the case of a walking *Target* that may not travel very far away along a straight line but rather around the same place. Furthermore, a certain *accuracy* may be used for a fast *Target* but would not satisfy a slow one because not adapted to her situation. While we do not have a clear idea on whether these delays are acceptable or can be considered as satisfying for most *LBS*, we claim it does not prevent most location-based services to provide a meaningful service.

Finally, we notice that the results of these simulations support the claim that a *Blurred Sighting* returned to the *LBS* for a particular *Target* contains a *Blurred Locality* in which the *Target* can be uniformly located. The algorithm proposed approximated *Targets*' location paths to straight lines between two consecutive *Blurred Sightings*. While this may be considered as somehow restrictive, it must be noted that the algorithm considered the longest straight line distance achievable between the corresponding two *Blurred Localities*. As a result, for the claim to hold, *Targets* must at least be able to travel the distance *dist*

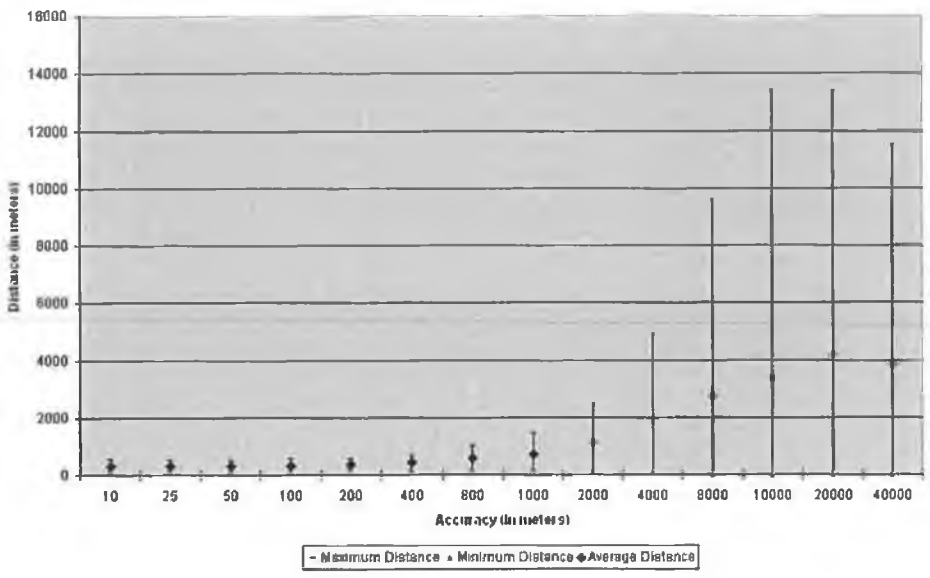
between the two extreme corners of the two *Blurred Localities* belonging to two consecutive *Blurred Sighting* releases. According to the results presented in this section, we see that on average, *Targets* not only passed through the *Blurred Localities* returned but also traveled further away from them when the corresponding *Blurred Sightings* were released. This shows that our approximation was enough to guarantee the claim of equally likelihood of *Targets*' locations inside the *Blurred Localities* of the *Blurred Sightings* returned.

6.9.3 Limitations

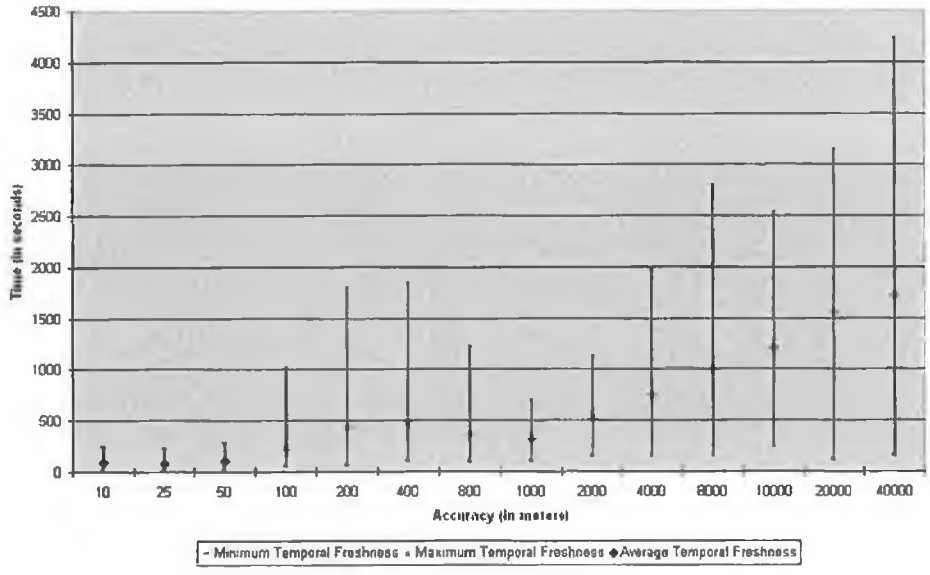
The algorithm proposed works well in the environment described but looses its efficiency if analytical techniques [17] are applied. In their approach to *Identity Blurring*, Beresford and Stajano (see Section 4.2 of this thesis or [37]) describe the attacks based on such techniques as being very complex to avoid since they involve a lot of parameters including the topology of the environment (buildings, roads, traffic flow directions) and user mobility patterns. Thus, if we assume an "a priori" knowledge of the *Target*'s home and workplace locations, it is easy to detect and predict where a particular *Target* is heading to. Documents such as maps or other publicly available information can also be used to restrain the area covered by a *Blurred Locality*, enabling *LBS* to gain more knowledge on a *Target*'s location than what they should be getting. Because of this, extending our algorithm to a three dimensional environment may prove to be difficult since a *Target* is very likely to be located either on the ground surface or in a restrained *Blurred Locality* such as one of the floors of a particular building. Similarly, a *Target* traveling in a hilly region may expose herself to more privacy threats since the more she goes up in altitude, the less area she can be located in. Interpolation and Extrapolation can also be used in conjunction with publicly available information. Considering a *Target* traveling on a straight line, it may be possible to infer her next *Blurred Locality* and by using a map, to reduce the admissible area where the *Target* can be located. Gruteser and Grunwald's approach to *Identity Blurring* (see Section 4.2 of this thesis or [96]) overcome this issue by proposing a technique that depends on the *Target* density rather than on a predefined area where a single *Target* can be uniformly located. However, their technique clearly does not meet the same requirements as ours in terms of *Target* privacy. For all these reasons, analytical attacks such as the ones pre-cited constitute an open problem considered outside the scope of this thesis. In Section 8 however, we give some research directions which may lead to some solutions to these problems.

Also, the claim that the *Target* can be located uniformly within the *Blurred Locality* as part of the *Blurred Sighting* released only holds as far as the results reported in Section

[17] Techniques that use environment characteristics such as the presence of buildings, roads, direction of traffic flow, etc.

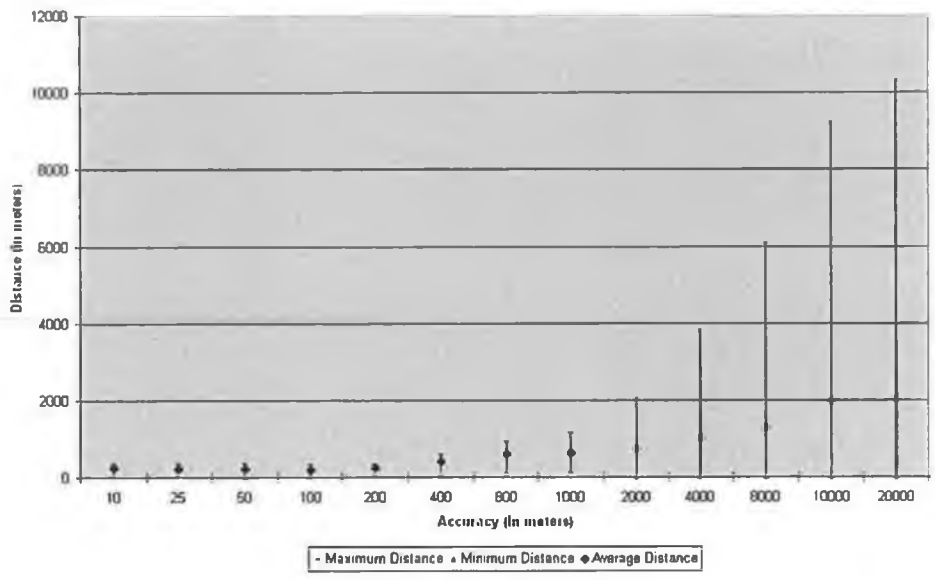


(a) Quality

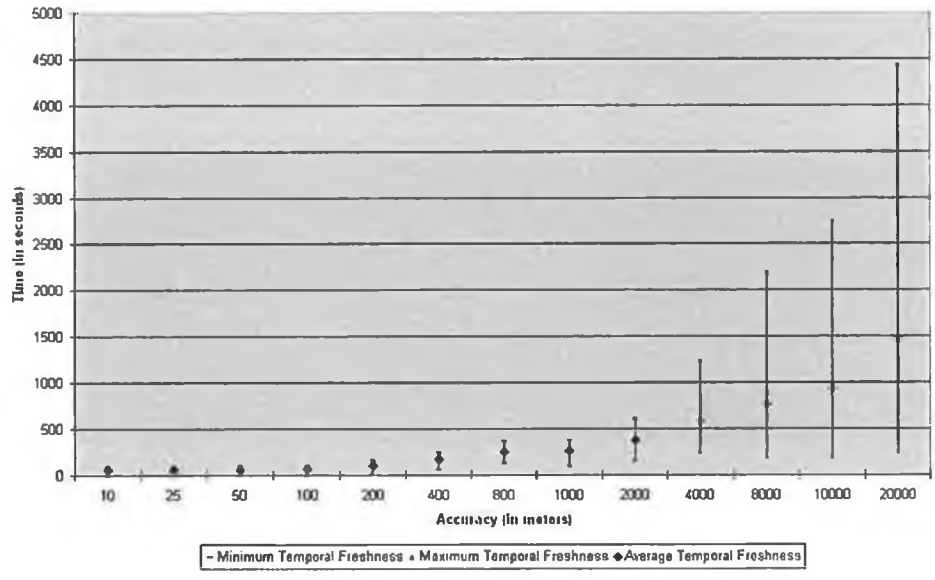


(b) Temporal Freshness

Figure 6.8: Accuracy of the Algorithm in a Location Tracking process carried out every minute on a driving *Target*.

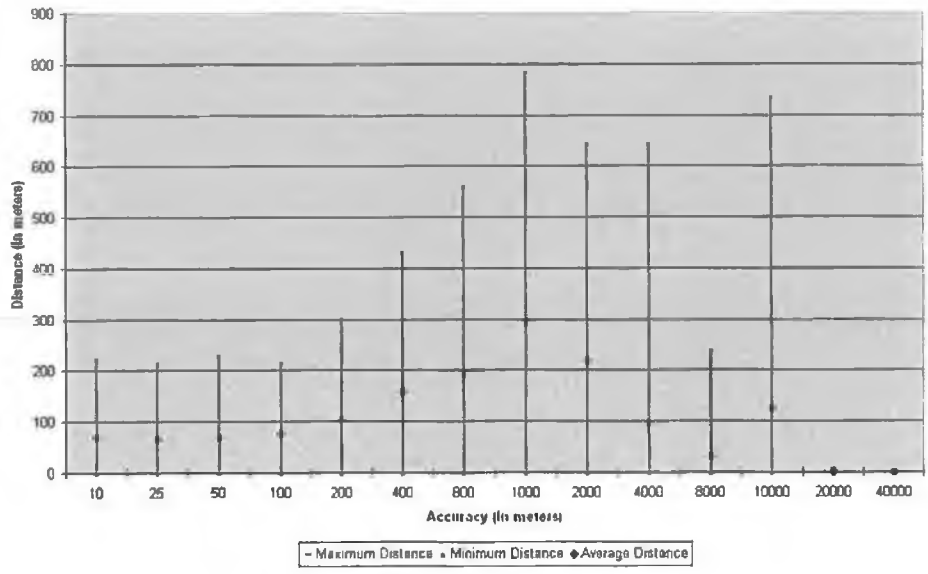


(a) Quality

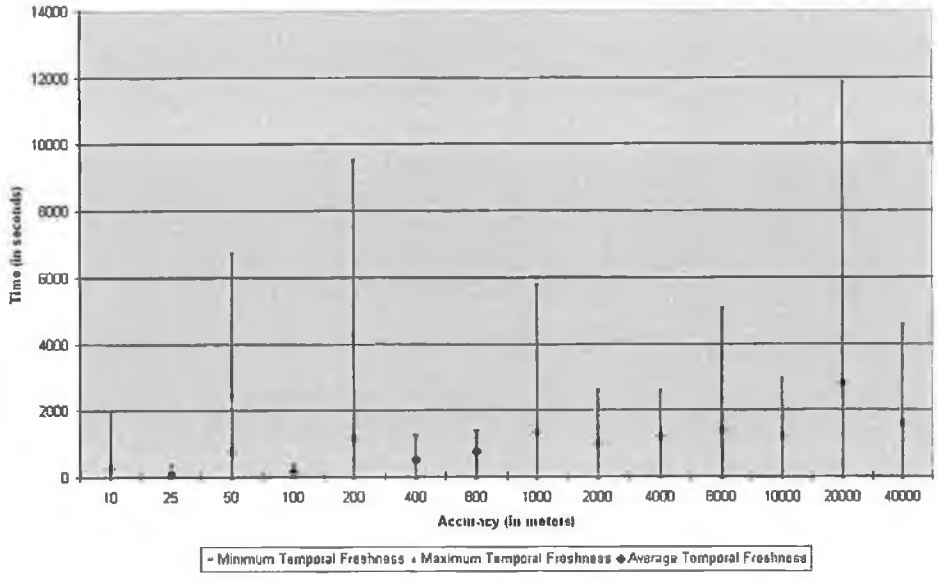


(b) Temporal Freshness

Figure 6.9: Accuracy of the Algorithm in a Location Tracking process carried out every minute on a cycling *Target*.



(a) Quality



(b) Temporal Freshness

Figure 6.10: Accuracy of the Algorithm in a Location Tracking process carried out every minute on a walking *Target*.

6.9.2 are concerned. While we believe these simulation results are representative of realistic behaviors, we recognize the fact that more comprehensive datasets would further support this claim.

6.10 Conclusion

The design of the *Location Blurring* algorithm presented here aimed primarily at solving the location inference problem at run-time, stated in Section 1.6. However, it is not clear to what extent the algorithm addresses the location path problem since *retrospective* border problems may become exploitable in some situations. Furthermore, while it intends to tackle the location transfer problem by discouraging entities to share the *Blurred Sightings* collected, it cannot effectively prevent them from doing so [18].

On the other hand, the approach taken to design our *Location Blurring* algorithm protects *Subjects'* privacy by maintaining a low but meaningful resolution of location information. It lets them specify the accuracy at which they wish to be located by *Subjects*. For *Semi-Trusted LBS*, the algorithm provides a good *accuracy* in terms of *Spatial* and *Temporal Granularity* of the first *Blurred Sighting* requested, with a maximal *Temporal Freshness*. If a second request is performed within the validity of the first *Blurred Sighting*, the same *Blurred Sighting* is returned. This way *location tracking* is penalized while sporadic location queries get accurate and secure information. In practice, *Semi Trusted LBS* may indeed only perform sporadic queries as opposed to *Fully trusted LBS* such as a child tracker which would intensively poll for the location of a particular *Target* in order to monitor exactly where the child is located. Another approach provides also a way to penalize *Non Trusted LBS* that may share their information by setting a common *Temporal Freshness* and *Blurred Locality* for their *Blurred Sightings*.

The concept of overlapping grids used as part of our *Location Blurring* algorithm has also recently proven successful in the field of location-based services discovery. In [165], Pashtan *et al.* show how to determine a geographical area of relevance for a particular mobile user according to her mobility characteristics. While their main motivations do not include providing location privacy, their example shows that such an approach is practical and scalable. It is also worth mentioning that the concept of overlapping grids to provide privacy may be extended to a 3-dimensional environment where *Localities* become cubes. The *Location Blurring* algorithm would have to be slightly modified however, as the new added dimension could leak location information.

[18] All *Non Trusted LBS* requesting a *Blurred Sighting* during the same predefined period of time receive the same information. Since they are paying for it, they are, somehow, discouraged from sharing it. However, they can still do it if they wish.

Chapter 7

Public Key Infrastructure

7.1 Introduction

Traditional Public Key Infrastructures generally suffer from various issues as commented in Section 2.3.3 of this thesis. In the mobile environment, they tend to present even more drawbacks, mainly because mobile devices owned by end-users have less capabilities than desktop computers and because the network bandwidth used to transmit certificate chains or revocation information is rather low.

In this chapter, we first outline our motivations regarding to the design of a PKI suitable for the mobile environment in the context of location-based services. We then introduce the architecture of the PKI together with the algorithms involved, and comment on their implementation and performance. Finally, a scenario showing how the PKI integrates within the location-based services architecture is presented.

7.2 Motivations

In the context of location-based services deployment, the critical piece of information to be protected remains, of course, the *Target* location. This encompasses the identity of the *Target* as well as her location together with a timestamp, referred to as a *Sighting*. When handset-based technologies (see Section 2.5.1) are used as part of the location determination process, a *Sighting* may have to be transmitted through the mobile network. Apart from confidentiality, authentication services may also be required in order to certify the originator of a specific *Sighting*. All this has serious physical and network layer security implications.

7.2.1 Physical Layer Security

The underlying wireless radio networks (GSM, GPRS or UMTS) already provide some encryption and authentication mechanisms. These are low level and are usually utilized between *Mobile Stations* and the *Network Subsystem*. However, they do not meet the requirements of end-to-end security since the link between the *Mobile Network* and the *LBS* provider is unprotected. It is also worth mentioning that even if reliable encryption algorithms may be in use in 2G networks between *Mobile Stations* and *Base Stations*, the weakest point of the architecture remains the link between those and *Base Station Controllers*. Communications are indeed transmitted in plaintext through microwave, which may allow for eavesdroppers to tap into 2G network users' connections.

Furthermore, if we assume the radio link is secure despite the several flaws detailed in Section 2.4.1, a man-in-the-middle attack can still be performed between a *Mobile Station* and a *Base Station* in order to attempt to turn off encryption. Such an attack is achievable using commercially available devices known as "IMSI Catchers". These devices, primarily used to perform network configuration tests, can act as a fake *Base Station* and force *Mobile Stations* to use the A5/0 version of the encryption algorithm, resulting in no encryption of the data stream sent to the *Mobile Network*. The device then establishes a normal connection with a real base station and relays the communications.

End-to-end communication security is clearly not achieved at the physical layer. Therefore, this needs to be taken care of at a upper layer.

7.2.2 Network Layer Security

Initially, the WAP forum introduced a wireless PKI (WPKI) in order to secure communications at the transport layer. However, as explained in Section 2.4.4, the approach taken could not guarantee end-to-end security between a mobile user and a WAP server since the WAP gateway had to relay communications between the WTLS layer and the TLS layer. This involved decrypting received data and re-encrypting it in order to forward it to the corresponding party using the right protocol. The presence of this so-called "WAP gap" presented a significant security issue that had to be addressed in some way (see figure 7.1).

The authors of the WAP2.0 specifications opted for the use of Internet protocols in general and TLS in particular as the security layer directly on the mobile device, removing the need for protocol translation at the WAP gateway level. It now guarantees end-to-end security since data sent between mobile users and content providers remain encrypted while passing through the WAP gateway (see Figure 7.2).

The architecture proposed in this thesis shares similarities with the WAP architecture.

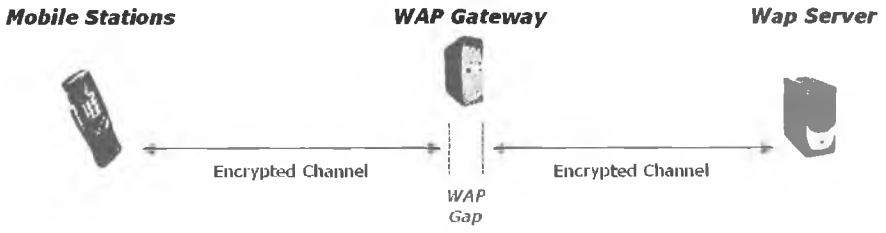


Figure 7.1: The WAP Gap.

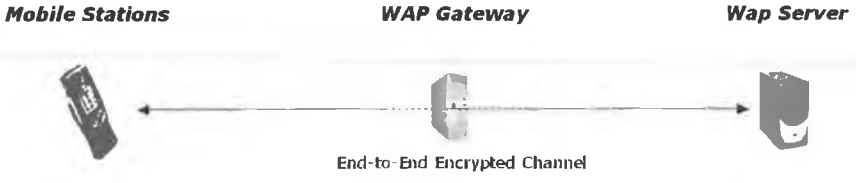


Figure 7.2: End-to-end secure communications in the WAP Architecture.

In particular, the WAP gateway and the *Orient Platform* both act as transcoding proxies whose aim is to assist wireless communications to some extent. However, they differ by at least one significant fact: the WAP2.0 Gateway no longer provides any security services to any of the parties involved, but simply acts as an intelligent router. On the other hand, one of the *Orient Platform*'s main role is to perform *Subjects*' and *Targets*' identity translation from their real identity to the pseudonym they wish to use when connecting to a particular *LBS*. Interaction with the *Orient Platform* is therefore always necessary to ensure pseudo-anonymity, and so every time a *LBS*-related request occurs. As a result, while secure end-to-end communications may still be required, in particular for sensitive data other than identities, a TLS tunnel is not applicable in this configuration, see Figure 7.3.

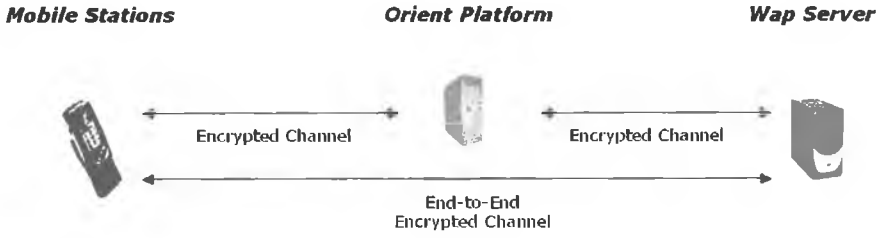


Figure 7.3: End-to-end secure communications in the Orient Platform's environment.

The physical and network layer issues outlined in this section demonstrate that there is definitely a need for an application level security layer guaranteeing end-to-end secure communications between a *Subject/Target* and the *LBS* she is connecting to. Using a conventional X509 PKI in this context may indeed be a solution but, as explained earlier on, it requires the client to perform resource intensive operations such as cryptographic computations and certificate validations. In the mobile environment, client devices are generally considered as lightweight, have low computational capabilities and have access to limited network bandwidth. In order to provide location-based services to such devices over the Internet, we introduced in Section 5 the *Orient Platform*, which factors out all the functionalities related to location information management and makes it transparent and lightweight for *Subjects* to connect to location-based services using a simple web browser. Following this philosophy, we believe that some PKI-related operations can be off-loaded to a trusted entity such as the *Orient Platform*. Of course, encryption and signature processes must be carried out on the mobile device since they are the building blocks of authentication mechanisms needed to connect securely to the *Orient Platform*. However, certificate validation is fundamental in the context of a PKI but is rarely carried out on desktop PCs and almost never carried out on mobile devices because it is considered too much time and resource consuming. Furthermore, certificate revocation is a process that may occur more often in the mobile environment than in the fixed one since mobile devices get more easily stolen or lost. As a result, this provides the right justification to off-load certificate validation to the *Orient Platform*.

7.3 Architecture

In light of the facts mentioned in the previous section, we opted for a PKI design based on a mediated server architecture.

7.3.1 Principles

In Section 2.3.4, we briefly described what a *SEM (Security Mediator)* architecture was. To put it in a nutshell, it consists of deploying an architecture where a user private key is split between two entities, one of them being a trusted *SEM*. An entity that wishes to encrypt a message uses the public key of her counterpart to cipher her message. The *SEM* will then assist the decryption process provided the security credentials of the recipient are considered as valid. Concerning digital signatures, the *SEM* will only assist the signing process if the signer's security capabilities have not been revoked. As a result, the encryption and decryption processes can be defined as follows:

Let pk_1 and pk_{SEM} be the two private key shares of a particular entity and Pk_1 her public key. Let P be a plaintext and C be the corresponding ciphertext, E being the encryption process and D the decryption process.

$E_{Pk_1}(P) = C.$
$D_{pk_1}(D_{pk_{SEM}}(C)) = P.$

Similarly, we define the signing and verification processes as follows:

Let pk_1 and pk_{SEM} be the two private key shares of a particular entity and Pk_1 her public key. Let P be the message to be signed and S be the corresponding signature, $Sign$ being the signing process and $Verif$ the verification process.

$Sign_{pk_{SEM}}(Sign_{pk_1}(P)) = S.$
$Verif_{Pk_1}(S) = True\ or\ False.$

We note that neither of the two parties involved can generate a valid signature or recover a plaintext from a ciphertext by themselves. Instead, they are required to cooperate with each other to achieve this.

7.3.2 Description

The architecture of the Public Key infrastructure considered in our context involves three different entities (see Figure 7.4):

- the *User*. This entity represents the parties that wish to conduct secure transactions. Therefore, a *User* can be a *Subject* wishing to securely access a *LBS*, a *Target* that requires to securely send her location or a *LBS* that receives connections from the two other types of *Users* cited.

- the *Security Mediator* (SEM). This is the entity that helps complete the decryption and signing processes depending on the *User*'s key pair revocation status. It will refuse to do so if her security capabilities have been revoked. This entity owns a share of every *User*'s private key.

- the *Private Key Generator* (*PKG*). Because the design of the PKI is based on a identity-based threshold cryptosystem, the private key shares of the two entities men-

tioned previously need to be mathematically related. The *Private Key Generator* is in charge of the generation of the decryption and signing private key shares and also of their distribution to the corresponding parties.

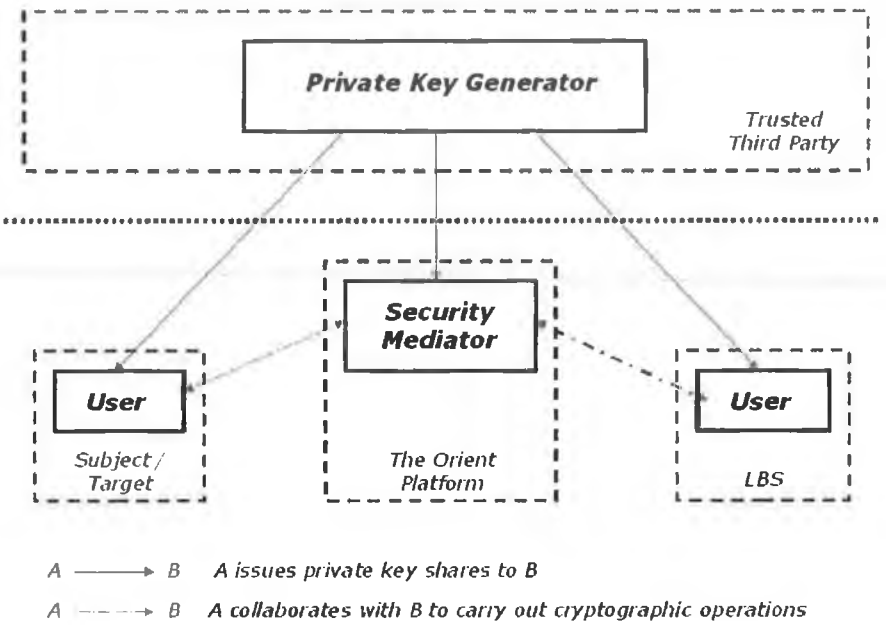


Figure 7.4: Public Key Infrastructure deployed in the Orient Platform's environment.

A *SEM* security architecture is very suitable in the environment considered. First and foremost, it solves the key revocation problem by enabling instant revocation of security capabilities. This is achieved by instructing the *SEM* not to assist cryptographic operations if the credentials of one of the parties have been revoked. The other reason that justifies such a choice is that the topology of the *SEM* architecture is very similar to the architecture to provide location-based services on the Internet considered in this thesis (see Figure 7.3). This results in a seamless integration of the Security Mediator as *the Orient Platform* already acts as an intermediary in the location-based services provision. One of the benefits of such an integration is that *Orient Platform* is able to transparently provide pseudo-anonymity services by redirecting a specific *LBS* request to the corresponding entity using the appropriate pseudonym, while allowing end-to-end communication security. This is detailed in Section 7.5.

Different configurations in which the entities described are owned by different parties may coexist. However, the most likely configuration is the one where the *Mobile Operator* owns the *PKG* and where the *SEM* is considered as a component of the *Orient Platform*.

7.4 Encryption and Signature Algorithms

In this section, we present the core algorithms used as part of the encryption and signing processes. The Baek and Zheng's threshold decryption algorithm as well as a newly designed mediated version of Hess' identity-based signature constitute the building blocks for providing encryption and signing services. We first formally define the algorithms used. Since they are based on mathematical groups defined over elliptic curves, we then recall some notions outlined in Section 2.2.2 and study more in depth some mathematical concepts in order to explain how to safely choose the system parameters and provide a concrete example specification.

7.4.1 Formal Specification

In this section, we will consider two cyclic groups in which the discrete logarithm problem (DLP) is considered as hard. $(G, +)$ represents an additive group while $(V, .)$ is a multiplicative group, both of order l. We will also consider a non-degenerate bilinear map e. This function is defined such that $e : G \times G \rightarrow V$, and has the following properties:

- *Bilinearity* : $\forall g_1, g_2, g_3, g_4 \in G$, $e(g_1 + g_2, g_3) = e(g_1, g_3).e(g_2, g_3)$
  Also, $e(g_1, g_3 + g_4) = e(g_1, g_3).e(g_1, g_4)$.
- *Non degeneracy* : $\exists g_1, g_2 \in G : e(g_1, g_2) \neq 1$
- *Computability* : $\forall g_1, g_2 \in G$, there exist an efficient algorithm to compute $e(g_1, g_2)$.

In addition to this, we also assume that given $g_1 \in G$, $e(g_1, g_2) \in V$, finding $g_2 \in G$ remains hard. This is known as the *Bilinear Pairing Inversion Problem (BPIP)*. Finally, the *Computational Bilinear Diffie Hellman Problem (CBDH)*, which proposes to compute $a.b.g_1$ given a generator g_1 of G, $a, b \in \mathbb{Z}_l{}^*$ and the values $a.g_1 \in G$ and $b.g_1 \in G$, must also be computationally intractable. Details on how to obtain such a bilinear map are provided later on in Section 7.4.2 of this chapter.

We now define the following functions used as part of the algorithms proposed. X^* denotes $X - \{O\}$ where O is the *identity* element of the law associated to the group X.

- $H_1 : \{0,1\}^* \rightarrow G^*$.
- $H_2 : V \rightarrow \{0,1\}^{len_1}$ where len_1 is the length of the plaintext to be encrypted.
- $H_3 : G^* \times \{0,1\}^{len_1} \rightarrow G^*$
- $H_4 : \{0,1\}^{len_2} \times V \rightarrow \mathbb{Z}_l{}^*$ where len_2 is the length of the message to be signed.

H_1 is used to map a string (representing an identity for example) to a element of the group G. H_2 hashes the result of the pairing to a string of length *len* and H_3 combines it with an element of the group G in order to output an element in the same group. H_4 is used to hash the message to be signed together with the result of a pairing.

Encryption Algorithm

We give here a formal description of the threshold encryption algorithm taken from [29] and used as part of the PKI in our architecture.

Public Parameters Generation. The *PKG* is in charge of generating and making the public parameters available to the other entities. The *PKG* first picks uniformly at random a number x in $\mathbb{Z}_l{}^*$ and keeps it secret. x will be referred to as the *PKG master key.* The *PKG* also picks uniformly at random an element $P \in G^*$, generator of G and computes $Y_{pkg} = xP$, where Y_{pkg} constitutes its public key. Then, he makes available to the other entities the following parameters : H_1, H_2, H_3, g_1 and Y_{pkg}, together with the any information related to the groups used.

Key Generation. The key generation process is also left to the *PKG*. For each user identity, the *PKG* computes:

- $Q_{ID} = H_1(ID)$, where ID represents the identity (public key) of a user.
- $D_{ID} = xQ_{ID}$, where D_{ID} represents the private key corresponding to a single user identity, to be split between her and the *SEM.*

While *Lagrange Interpolation* is generally used in threshold cryptosystems in order to split private credentials, the *SEM* and user private key shares are derived here from D_{ID} by a simple arithmetic operation since the threshold and the number of parties involved are equal to two. $D_{ID,sem}$ is chosen as a uniformly random elment in G and $D_{ID,user}$ is given by:

$D_{ID,user} = D_{ID} - D_{ID,sem}$

The private shares $D_{ID,user}$ and $D_{ID,sem}$ are then safely delivered to the corresponding parties.

Encryption. A Sender that wishes to issue an encrypted message then computes the following :

- A uniformly distributed random number $r \in \mathbb{Z}_l{}^*$.

- $U = rP$
- $k = e(H_1(ID), Y_{pkg})^r$, where ID is the identity of the corresponding receiver.
- $V = H_2(k) \oplus M$, where M is the message to be encrypted.
- $W = rH_3(U, V)$

The ciphertext $C = (U, V, W)$ is now ready to be sent to the *SEM* for assisted decryption[1].

Decryption. The decryption process occurs in two phases.

- The *SEM* first checks whether the user identity ID has been revoked. If this is the case, it forwards a message to the recipient of the original message, stating that he could not decrypt the ciphertext since ID has been revoked. On the contrary, if the identity has not been revoked, the *SEM* performs the following operations:
  - $h3 = H_3(U, V)$
  - Check of whether the ciphertext received is valid by computing $e(P, W) = e(U, h3)$. If the ciphertext is invalid, a message is forwarded to the recipient of the original message in order to inform him of the status of the decryption. If it is valid, the *SEM* computes $k_{sem} = e(D_{ID,sem}, U)$ and forwards it to the recipient of the message, together with C.
- The Receiver performs the same check as the *SEM* in order to verify whether the ciphertext received is valid. She then computes $k_{user} = e(D_{ID,user}, U)$, reconstitutes $k = k_{user} * k_{sem}$ and recovers the message M by computing $M = H_2(k) \oplus V$.

Signature Algorithm

In [132], Libert and Quisquater detail an approach to provide a *SEM* architecture with pairing-based signature capabilities. The authors point out that only a few signature schemes can be practically adapted to the *SEM* architecture. The first reason is that such signature schemes must allow for a secure threshold version to be derived from them, i.e, a scheme whereby neither of the private key shares of the parties involved are disclosed. The second reason reflects the practicability of the deployment of such a scheme in the *SEM* architecture. Indeed, most threshold signature schemes are probabilistic in the sense that they require signers to collaborate in order to perform a distributed random number generation. Such a

[1] In the scenario depicted in [29], the receiver is first given the ciphertext and interacts with the *SEM* in order to decrypt it. In our architecture, the *SEM* is implemented on the *Orient Platform* proxy server and performs transparently the key pair revocation status verification and ciphertext decryption.

secret sharing protocol, see [190], introduces communication overheads and would indeed, in our context, be clearly unacceptable as the *SEM* must remain as transparent as possible to its *Users*. As a solution, the authors propose a mediated version of the GDH signature scheme. However, this scheme is not identity-based which means that using it would complicate key management and defeat the purpose of using the identity-based encryption algorithm detailed in this section.

In this section, we propose to extend Hess' identity-based signature scheme [108] in order to provide the *SEM* architecture with identity-based signature capabilities. The identity-based signature scheme further described in Section 7.4.1 does not qualify as a threshold identity-based signature scheme, but rather as a server aided signature generation scheme (the *SEM* does not process the message to be signed but rather its signature). The signer produces a signature using her private key share, and this token is then processed by the *SEM* upon verification of the signer credentials in order to generate a signature verifiable by the corresponding public key identifier. This section is divided as follows: we start by a description of Hess' original scheme and then describe our contributions. Security considerations are discussed in Section 7.7.2 of this chapter.

Hess' Identity-based Signature Scheme

System Parameters Generation. The *PKG* is in charge of generating and making the public parameters available to the other entities. The *PKG* first picks uniformly at random a number x in $\mathbb{Z}_l^*$ and keeps it secret. As previously mentioned, x is referred to as the *PKG master key*. The *PKG* also chooses a random element P defined on the group G and computes $Y_{pkg} = xP$, where Y_{pkg} constitutes its public key. Then, he makes available to the other entities the following parameters : H_1, H_4, P and Y_{pkg}, together with the any information related to the groups used. These parameters are formally defined at the beginning of Section 7.4.1 of this thesis.

Key Generation. The *PKG* then performs the following computation for each user identity: $S_{ID} = x.H_1(ID)$, where ID represents the identity or public key of a user, and S_{ID} the corresponding private key.

Signature. The Hess' signature process can be described as the computation of the following elements:

- An element $P_1 \in G^*$ picked uniformly at random.
- An element $k \in \mathbb{Z}_l^*$ picked uniformly at random.

- $r = e(P_1, P)^k$
- $v = H_4(m, r)$, where m is the message to be signed.
- $u = vS_{ID} + kP_1$

(u, v) represents the signature associated with the message m.

Verification. The verifier computes $r = e(u, P).e(H_1(ID), -Y_{pkg})^v$. She then verifies the signature by checking whether $v = H_4(m, r)$.

Our mediated version of Hess' Identity-based Signature Scheme

System Parameters Generation. This phase remains identical as for the original version of Hess' Identity-based Signature Scheme.

Key Generation. After having calculated the values $S_{ID} = x.H_1(ID)$ for each identity ID considered, the PKG derives the following private key shares, as for the identity-based encryption algorithm presented earlier on:

$S_{ID,user} = S_{ID} - S_{ID,sem}$

$S_{ID,user}$ represents the private key share of the *User* identified by ID while $S_{ID,sem}$ is the corresponding private key share given to the *SEM*. $S_{ID,sem}$ is picked as a uniformly random element in G.

Signature. The signing process now involves two entities : a *User* and a *SEM*. The *User* signs a message with her private key share and the signature is then completed by the *SEM* using its share.

User. This entity computes:

- An element $P_1 \in G^*$ picked uniformly at random.
- An number $k_1 \in \mathbb{Z}_l{}^*$ picked uniformly at random.
- $r_1 = e(P_1, P)^{k_1}$
- $v = H_4(m, r_1)$, where m is the message to be signed.
- $u_1 = vS_{ID,user} + k_1 P_1$

(u_1, v) represents the signature share associated with the message m. (u_1, v) is forwarded to the *SEM*.

SEM. This entity computes:

- A element $P_2 \in G^*$ picked uniformly at random.
- A number $k_2 \in \mathbb{Z}_l{}^*$ picked uniformly at random.
- $r_2 = e(P_2, P)^{-k_2}$
- $u_2 = u_1 + vS_{ID,sem} + k_2 P_2$

(u_2, v, r_2) represents the full signature associated with the message m and is forwarded to a verifier.

Verification. Finally, a *Subject* wishing to verify the signature performs the following actions: she computes $r = r_2.e(u_2, P).e(H_1(ID), -Y_{pkg})^v$. She then verifies the signature by checking whether $v = H_4(m, r)$.

We note here that the signing process is identical for the signer, both in the original and in the *SEM* version of the scheme. However, it is slightly different for the *SEM* since it just needs to complete the signature using its own private key share. The correctness together with a discussion on the security of the scheme presented are discussed in Section 7.7.2.

7.4.2 Group and Parameter Selection

The two cryptographic schemes presented in the above section use bilinear maps as part of their algorithms. Such bilinear maps can be efficiently implemented using mathematical functions defined over groups of points on elliptic curves. In this section, we provide some definitions based on the basic concepts outlined in Section 2.2.2. We then show how to choose the cryptographic parameters necessary to efficiently implement bilinear maps. We do not intend to be exhaustive in our explanations regarding to elliptic curve terminology or pairing-based cryptography. Instead, we will provide the reader with the minimum information for his better understanding. We wish to acknowledge Martijn Maas for his very interesting thesis on pairing-based cryptography [138] and base our explanations on his work.

Elliptic Curves

An elliptic curve E is a mathematical object defined over a finite field F_q. The letter q represents the number of elements in the field (the *order of the field*) and is usually a large prime or a power of a large prime $q = p^m$, where p is called the *characteristic* of the field.

Likewise, the number of elements of the group formed by the points on the elliptic curve (denoted $E(F_q)$) defined over the underlying field F_q is denoted as the *order of the curve* and is referred to as $\#E(F_q)$. For each element P of this group, the *order of a point* P can be defined as the least positive integer r such that $rP = O$ where O is the *identity* of the group also known as the *point of infinity*. Also, a point P whose *order* divides a particular n is known as a *n-torsion point*. Thus, for a particular n, a subgroup formed by all the *n-torsion points* of $E(F_q)$ can be defined as a *n-torsion point* group. We will refer to it as $E(F)[n]$ such that :

$$E(F)[n] = \{P \in E(F) : [n]P = O\}$$

Finally, an elliptic curve is also defined by a constant t called the *Trace of Frobenius* and which is given by the following relation:

$$t = q + 1 - \#E(F_q)$$

When the characteristic p of a curve E/F_q divides its *Trace of Frobenius*, i.e. $t \equiv 0 \bmod p$, the curve is said to be *supersingular*. This property has some implications in the reminder of this section.

Bilinear Maps

The underlying mathematical structure of the encryption and signature algorithms also relies on bilinear maps (see Section 7.4.1 for a formal definition of a bilinear map). From such a bilinear map, one can construct a bilinear pairing on the group of points on an elliptic curve such as the Tate pairing. The Tate pairing is defined as follows :

Let E be an elliptic curve defined over a field F_q. Let l be relatively prime to q and such that there exists a point $P \in E(F_q)$ that has order l. We now define k as the smallest integer assumed to be greater than 1 such that l divides $(q^k - 1)$. k is often referred to as the *embedding degree* of the curve with respect to l. The smaller k is, the more efficient the algorithm will be but k should also be large enough to guarantee a good level of security, *i.e.* so that the discrete logarithm problem in F_{q^k} remains intractable (see Section 2.2.2 for a description on attacks on the discrete logarithm problem). A simplified version[2] of the Reduced Tate paring of order l, as defined in [33], is the bilinear map e_l given as:

$$e_l : E(F_q)[l] \times E(F_{q^k}) \rightarrow F_{q^k}^*$$

In addition to *bilinearity*, the Tate Pairing has also the following properties :

[2]This definition slightly differs from the standard one given in [138] but is considered as more suitable for cryptography purposes [33].

- $\forall P \in E(F_q)[l]$, $e_l(P,P) = 1$ *(identity)*

- $\forall P \in E(F_q)[l]$ and $Q \in E(F_{q^k})$, $e_l(P,Q) = e_l(Q,P)^{-1}$.*(alternation)*

- $\forall P \in E(F_q)[l]$, $\exists Q \in E(F_{q^k}) : e_l(P,Q) \neq 1$. *(non degeneracy)*

The first argument of the bilinear pairing is a point P on the elliptic curve E defined over the finite field F_q and whose order is l. The second argument is taken from a separate subgroup, here the group of points on the same elliptic curve defined on the extension field F_{q^k}. The choices of l and k ensure that the two points P and Q remain linearly independent from each other in order to satisfy the *non-degeneracy* property. Finally, the Tate Pairing outputs a result in the extension field $F^*_{q^k}$.

Modified Tate Pairing

The Tate pairing takes its arguments in two groups of points on the same elliptic curve defined over two finite fields, so that they remain linearly independent from each other. Choosing these two points involve working on two separate subgroups and over different fields and while this does not present any insurmountable challenge, being able to pick those two elements from the same group of points is certainly more convenient. This can be achieved by slightly modifying one of the two point's coordinates using a map known as a *distortion map.* A *distortion map* can be defined as follows.

In our context, a *distortion map* with respect to $Q \in E(F_q)$ is a group endomorphism $\phi \in End(E)$ that maps the point Q to a point $\phi(Q)$ defined on the curve E over the extension field F_{q^k} that is linearly independent from Q.

$$\phi : E(F_q) \rightarrow E(F_{q^k})$$

Additionally, this *distortion map* facilitates the choice for the Tate pairing arguments as they do not need to be points defined on a curve over separate fields anymore. This leads to the definition of the *Modified* Tate pairing:

$$\hat{e}_l : E(F_q)[l] \times E(F_q) \rightarrow F^*_{q^k}$$
$$\hat{e}_l(P,Q) = e_l(P,\phi(Q)) = \mu \text{ where } \mu \in F^*_{q^k}$$

We note here that the Modified Tate pairing does not satisfy the *identity* property mentioned earlier on, since P and $\phi(P)$ are now linearly independent. Therefore, computing $\hat{e}_l(P,P)$ will always lead to a non-trivial result. Also, the *alternation* property is no longer true since the pairing is now symmetric ($\hat{e}_l(P,Q) = \hat{e}_l(Q,P)$). In fact, this *symmetry* property makes the Modified Tate pairing even more convenient since most points in $E(F_q)$ will qualify as its second argument.

The computation of the Modified Tate pairing is considered as more efficient, not only compared to the traditional Tate pairing but also considering other admissible pairings such as the Weil pairing and its variants.

Parameters

Given all the notions introduced through out this section, we define here the system parameters that will be used to compute the Modified Tate pairing as part of the cryptographic algorithms used. Extreme caution is to be exercised when choosing these parameters since a mistake could lead to a total break of the system.

We wish to design a system based on the Modified Tate pairing, mainly for implementation efficiency. As seen earlier on, the *embedding degree* k of the chosen curve needs to be small enough for efficiency purpose but large enough to guarantee a good level of security. In [138], we note that curves that satisfy these constraints are very sparse. Only *supersingular* curves and so called *MNT* curves seem to fulfill these requirements (please refer to [138] for more information on *MNT* curves). Also, the Modified Tate pairing necessitates *distortion maps*, which do exist for all *supersingular* curves but only for some *MNT* curves.

We therefore select the following *supersingular* curve, defined on a field F_q, where q is a large prime:

$$E(F_q) : y^2 = x^3 + x$$

q is chosen as $q \equiv 3 \bmod 4$ as stated in [118] to ensure the *supersingular* character of the curve. In order to avoid attacks on the elliptic curve discrete logarithm problem, q needs to be chosen large enough. The authors of [189] advocate that q^k should be at least 1024 bits, where k is the *embedding degree* of the curve. They also advise to choose l, the group order of the first element, so that it is at least 160 bits long in order to avoid the Pohlig-Hellman attack mentioned in Section 2.2.2 of this thesis. l also needs to be relatively prime to q and must divide $\#E(F_q)$. Since $\#E(F_q) = q + 1$[3], we are now left with the computation of the values of q and l. While picking values for q and l that satisfy the properties previously cited is relatively easy, finding the ones that will make the most efficient computation of the pairing remains tricky. In [33], the authors advocate the use of a Solinas[4] prime for l. l has therefore a very low Hamming weigh relatively to its size, which speeds up considerably the Miller algorithm used to compute the Tate pairing. q is computed such that l is a prime factor of $q + 1$, such that $q \equiv 3 \bmod 4$ and such that it is a prime. Therefore, we have:

$$l = 2^{159} + 2^{17} + 1 = 730750818665451459101842416358141509827966402561 \text{ (160 bits)}$$

[3] See [33] and Section 5.3 on Curves with Small Embedding Degree in [138] for a proof of this result.
[4] A Solinas prime is a prime number that is the sum or difference of a small number of powers of 2.

q = 129018690069575864324485985108131770743485602383987917238774467

941022363774792713545975430967683811538557630166372039133559375078O

4962628312606218934271059 (512 bits)

We have l divides $(q+1)$, which divides $(q+1)(q-1)$, which also divides (q^2-1). The *embedding degree* k of the curve with respect to l is therefore 2 and we will denote F_{q^2} as the extension field. An element of such a field can be seen as the operation $a+ib$ where $a, b \in F_q$ and i defined such that $i^2 = -1$. The *distortion map*[5] to be used is defined in [138] and reads as follows:

$$E(F_q) \rightarrow E(F_{q^2})$$
$$(x, y) \rightarrow (-x, iy)$$

As a result, if we recall the formal parameters used to described groups in Section 7.4.1, we will, from now on, use $E(F_q)[l]$ as G and $F^*_{q^2}$ as V and implement these algorithms in consequence.

7.5 Prototype Implementation

In this section, we present a prototype implementation of the public key infrastructure used as part of our architecture. While the implementation may require some optimizations, we believe it achieves reasonable performance as shown in Section 7.7. The Java language was chosen to implement the PKI for two main reasons: first because Java achieves robustness and portability and also because this is the language used to implement the *Orient Platform*. We show here how it was used to implement the low level cryptographic operations as well as the high level components of the infrastructure.

7.5.1 Cryptographic Operations

There are very few Java cryptographic libraries available that implement pairing-based cryptography. One reason may be that the technology is fairly new. Another reason might be that the computations involved are resource consuming and that therefore, implementations in C or C++ are preferred. The authors of [72] propose a Java-based API that provides building blocks in order to help developers design identity-based cryptosystems. While being

[5] We note here that such a *distortion map* does not map every points to a separate subgroup since for y=0 the resulting point is linearly dependent to the argument of the map. Also, the identity of $E(F_q)$ (the so-called *point of infinity*) is mapped to itself so the *distortion map* works for all the points but those two.

very well thought in terms of design and easily extendable, this API remains painfully slow when it comes to implementing a single pairing using secure enough parameters.

Our first approach has been to try to extend their library in order to support native calls. Indeed, since some operations such as the execution of the Miller algorithm are considered as the bottleneck of pairing-based cryptographic implementations, implementing them in C would certainly improve their performance. This was achieved very easily from the Java library point of view as it only required to extend the class called *ModifiedTatePairing* and to provide the native method signatures, methods that would be called through the Java Native Interface (JNI). On the other hand, the native code implementation proved to be a little bit more tricky. It was carried out using the MIRACL C big number library [192] and necessitated to implement part of the F_{q^2} field arithmetic as the library only provides an incomplete set of functions. Also, all the optimizations mentioned in [33] were implemented in order to gain in speed of execution. While we achieved reasonable results on a desktop PC, it proved too slow on the mobile device. It was later decided to abandon this Java API and to implement all the cryptographic operations in a C++ library that could be called using JNI from Java. The work presented was however very useful for understanding the basics of pairing-based cryptography and testing the correctness of some example cryptographic schemes.

Our second approach was therefore to build a complete native implementation of an API that would be used to encrypt/decrypt and sign/verify messages using the algorithms presented in 7.4.1. Here again, we used the MIRACL library to implement it as it provides optimized code for field operations. In particular, it provides an example of a C++ implementation of an optimized *Modified Tate Pairing* along with the corresponding Miller algorithm. Building on these examples, we implemented the encryption and signature algorithms using this library. The API interface is described in Appendix C. We then created two software dll libraries:

- One compiled for x86 processors that can be used on Windows 98/2000/XP operating systems. It is used by the Private Key Generator, the Security Mediator and *LBS*.

- One compiled for StrongARM processors that can be used on the Windows CE operating system. It is used by the mobile devices held by *Targets* or *Subjects*.

We realize that we lose the portability of the code by using dll libraries. However, the performance of the implementation of the algorithms improved by a factor of 10 compared to the first approach.

7.5.2 Private Key Generator

This entity is implemented as a Java component and belongs to the *Mobile Operator*'s infrastructure.

Private key share management issue. We recall here that one of the main functions of the *Orient Platform* is to allow *Subjects* to hide their real identity using a pseudonym. When encrypting a piece of information or verifying a signature, *LBS* need utilize a public identifier for the *Subject/Target* they are dealing with. The most appropriate solution for them is therefore to use the pseudonym of the *Subject/Target* to perform these operations. While this constitutes a neat solution for *LBS*, it forces *Subjects* and *Targets* to possess one encrypting and one signing private key share for each *LBS* they may connect to. This poses significant key management problems, in particular concerning private key generation, transmission and revocation.

Solution. In order to resolve this issue, we propose a process whereby a *Subject* or *Target* owns one and only one private key share while any of her pseudonyms may be used as her corresponding public key by *LBS*. This can be achieved by having the *PKG* generate as many *SEM* private key shares per *Subject/Target* as needed in order to accommodate the number of *LBS*. In other words, when a *Subject* or a *Target* wishes to communicate with a particular *LBS*, the *SEM* will assist the corresponding cryptographic operation by choosing the private key shares that correspond to the pseudonym. This is made possible by the fact that the *PKG* can derive as many private key shares as required from the *User*'s one, provided the underlying mathematical structure of the scheme is large enough. This approach is valuable for two main reasons:

- Firstly, because as stated earlier on, it preserves *Subjects* and *Targets*' privacy by using their pseudonyms as their public key.
- Secondly because it does not increase the complexity of key management. *Users* do not have to request a new private key share every time they wish to connect to a new *LBS*. They do not have to revoke a large number of key pairs but only one if, for example, their mobile device is stolen.

Implementation. In order to achieve this, the *PKG* initially generates a unique decryption private key for every *User*, namely $D_{R_{ID}} = xH_1(R_{ID})$ where R_{ID} constitutes the *User*'s real identity. Then by picking uniformly at random a number $D_{R_{ID,user}}$ which will represent the *User*'s private key share corresponding to R_{ID}, the *PKG* derives the corresponding *SEM*

private key share as follows $D_{ID,sem} = D_{R_{ID}} - D_{R_{ID,user}}$. The $D_{R_{ID,user}}$ private key share is delivered to the *User* and from now on, every time a *Subject* registers to a *LBS* and obtains a pseudonym ID, the *PKG* performs the following operations:

- It computes $Q_{ID} = H_1(ID)$.
- It computes $D_{ID} = xQ_{ID}$ where x is the *PKG*'s private key.
- It then retrieves the *Subject*'s private key share $D_{R_{ID,user}}$ and computes $D_{ID,sem} = D_{ID} - D_{R_{ID,user}}$.
- The *PKG* then delivers the private key share corresponding to the pseudonym ID, namely $D_{ID,sem}$ to the *SEM*.

In other words, the *User* receives a unique private key share $D_{R_{ID,user}}$, and this key is used by the *PKG* to generate or derive *SEM* private key shares every time a new pseudonym is required. We note that the exact same key generation process is applied regarding the signing private key shares.

The encryption and signature processes that follow key generation are similar to the ones already detailed (see Section 7.4.1) for *Users*. The *Security Mediator*, however, must now actively search for the appropriate signing or decryption private key share to use, *i.e* the one corresponding to the *User*'s pseudonym to be used.

The next section details how the private key shares are managed by the *Security Mediator* in order to provide an end-to-end secure channel between *Users* while enabling their pseudo-anonymity.

7.5.3 Security Mediator

The *Security Mediator (SEM)* is implemented as a Java component of the *Orient Platform* and acts as a filter on its proxy interface. When *Users* connect to *LBS* through the *Orient Platform*, they transmit ciphertexts or partial signatures as HTTP POST variables. The *SEM* retrieves those variables and checks that the *User*'s key pair has not been revoked. The *SEM* then looks up a table that maps *Users*'identity, their pseudonym used for a particular *LBS*, and the private key share to be used in this precise case and completes the cryptographic process. We note that the *SEM* only assists cryptographic operations when the revocation status of the corresponding key pair is valid. Figure 7.5 illustrates the key distribution and usage between the different entities in an encryption process.

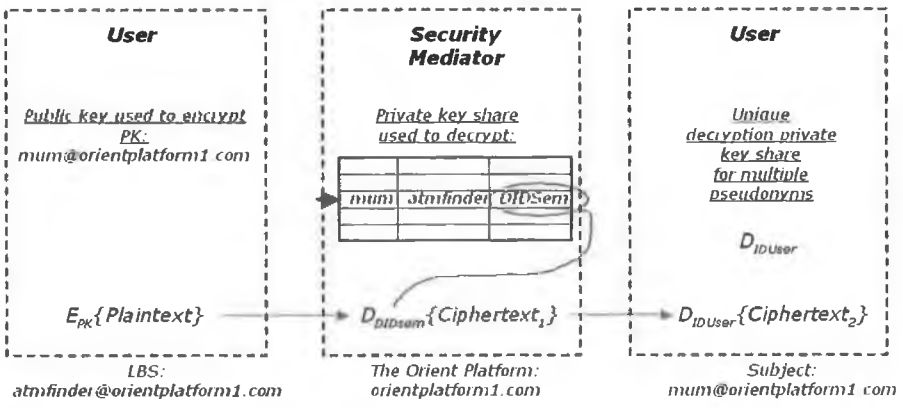


Figure 7.5: Private key share selection in the encryption process.

7.6 Scenario

In this section, we present a scenario in which a mobile user, acting both as a *Subject* and *Target*, connects to a Point-of-Interest Locator *LBS* and transmits her location securely.

We consider a *Subject* that uses an application on her mobile device in order to securely send and receive location information about herself and her relatives. We describe here a confidential transaction between a *Subject* and a *LBS*.

- The *Subject* is registered with the *Orient Platform*. The middleware is identified by a domain name"orientplatform1.com".
- The *Subject* connects to a *LBS* called "ATMfinder" under the pseudonym "mum" and wants to find out how far away she is located from the nearest ATM.
- Her mobile device uses a GPS sensor in order to determine its location.
- She wishes to know exactly how long she needs to walk to find the nearest ATM. She is also concerned about the security implications for her own safety regarding her request being intercepted by somebody else. Hence, she wishes to use end-to-end encryption with the *LBS* in order to transmit her location [6], even if this means bypassing the privacy modules implemented as part of the *Orient Platform*. Of course, she fully trusts the *LBS* not to share her details.

As soon as she connects to the *LBS*, a *Sighting*, which is a piece of information composed of her location details and a timestamp is produced. In order to send it securely to the

[6]Note : the PKI can be used to transmit any confidential piece of information between the two entities such as a session key for example, but we will use location information here as an example.

LBS, she encrypts it using the *LBS*' public key : "atmfinder@orientplatform1.com". The ciphertext is then transmitted using the HTTP protocol as a POST variable part of the HTTP request. The *Orient Platform*'s proxy server receives the request, the *SEM* retrieves the ciphertext from the POST variable and partially decrypts it depending on the *LBS*' security capabilities' status. Then the middleware initiates the Client Initiated Service (CIS) of the *Orient Protocol* (see Section 5.4 for more information about the protocol) with the *LBS* and forwards the HTTP request together with all the POST variables including the updated ciphertext as part of the first message. As shown in Appendix A, the CIS service is composed of a series of four messages. Generally, the four messages are used since the *LBS* needs to enquire about some *Targets*' location. In the present case, since the location of the *Subject* (also acting as a *Target*) is included as part of the request, only the first and the last message of the service will be used. Upon reception of the first message, the *LBS* completes the decryption of the ciphertext, generates the appropriate information including the *Subject*'s location as well as the one of the nearest ATM and encrypts it using the *Subject*'s public key : "mum@atmfinder@orientplatform1.com". The content is then encoded in base64, inserted in the fourth XML message of the CIS service and is sent back to the corresponding *Orient Platform*. The middleware extracts the value added content of the message, partially decrypts the ciphertext and forwards it to the *Subject*, who is left with the decryption using her private key share. In this scenario, we notice that the *Orient Platform* only acts as an pseudo-anonymising proxy as opposed to a secure pseudo-anonymising location server.

7.7 Evaluation

This section provides an evaluation of the overall Public key Infrastructure. Some experiments have been carried out to assess the performance of the algorithms described, both on desktop and mobile devices. Security issues are then discussed at the end of the section.

7.7.1 Performance

The experiments were conducted on :

- A desktop PC to simulate the *Orient Platform*'s behavior as well as *LBS*'s. It consists of a 1.7Ghz Intel Pentium 4 with 1G of memory, running Windows 2000 as well as the Java Virtual Machine 1.4.2.

- A PDA to simulate a *Subject*'s or *Target*'s mobile device. It consists of a 206 MHz Intel StrongARM 32 bit RISC Processor with 32M of RAM, running Windows CE 3.0.

The Java Virtual Machine used is the IBM j9 1.4.

We report here the results for the following operations carried out on the relevant devices. The experiment consists of ten runs of the same operation in order to calculate the mean and the standard deviation for each of them. The results are summarized in Table 7.1.

Operation	*Mobile Device*	*Desktop PC*
encrypt	4583 (44.1)	148 (8.86)
decryptSEM	-	141 (0.48)
decrypt	3720 (4.22)	142 (5.26)
sign	1475 (5.46)	47 (0)
signSEM	-	47 (0)
verify	3305 (34.9)	114 (7.53)
encrypt (optimized)	1680 (7.83)	51.5 (7.49)

Table 7.1: Performance results for Mediated Identity-Based Cryptography operations in milliseconds. The standard deviation is given within brackets.

While it may be difficult to assess the security equivalence in terms of key length between the Mediated Identity-based PKI and an RSA-based PKI, we report here the timings of RSA operations on the same platforms in Table 7.2. We intend here to compare the performances of the widely used RSA algorithm against the implementation of our mediated identity-based cryptographic algorithms on a mobile device in order to see if the execution times are significantly different.

Operation	*Mobile Device*	*Desktop PC*
encrypt (1024 bit key)	25 (0.3)	0 (0)
decrypt (1024 bit key)	893 (4.81)	25 (8)
sign (1024 bit key)	895 (3.4)	22 (7.9)
verify (1024 bit key)	22.6 (0.7)	0 (0)
encrypt (2048 bit key)	92.3 (1.1)	7.9 (8.33)
decrypt (2048 bit key)	6875 (8.4)	189 (15.4)
sign (2048 bit key)	6885 (10.1)	188 (14.6)
verify (2048 bit key)	80.7 (0.4)	6.3 (8)

Table 7.2: Performance results for RSA operations in milliseconds. The standard deviation is given within brackets.

The experiments carried out clearly demonstrate that:

- The Mediated Identity-based PKI (MIB-PKI) cryptographic operations carried out on the mobile device are relatively slow compared to RSA. This can be explained by the number of ECC point multiplication, modular exponentiation and Tate pairing computations to be carried out as opposed to only one modular exponentiation for RSA.

Also, these timing measurements are considered in addition to the JNI overhead. We note however that the MIB-PKI encryption process can be sped up by pre-computing some operations, see Table 7.1.

- While being slow, the MIB-PKI remains however practicable, offering good enough performance on the desktop platform. Furthermore, it removes the need of finding out the certificate revocation status by enabling the Security Mediator to revoke instantly the security capabilities of one of the parties involved. Therefore, in order to compare rigorously the two PKI, the time to retrieve and check the validity of a RSA public key certificate against the corresponding CRL should be taken into account. This experiment has not been carried out but one could expect a significant delay resulting from the CRL discovery, download, verification and parsing for potentially every single certificate concerned.

7.7.2 Security Analysis

The security analysis of the PKI first recalls the mathematical problems on which the threshold decryption and signature algorithms are built on. We also review the security features offered to *Users* by the overall system, focusing in particular on the public key revocation.

Algorithms

Encryption. The threshold decryption algorithm specified in [29] is based on the assumption that the *Computational Bilinear Diffie Hellman Problem (CBDH)* is computationally intractable. [29] provides a detailed formal proof of security of this result. The scheme ensures in particular security against *chosen ciphertext attacks* whereby an attacker is allowed to ask decryption shares for messages and identities of its choice other than those corresponding to the challenge, being in possession of a *User*'s private share. This result makes this scheme unique in the setting of mediated cryptography in the sense that it is the only one so far that provides this level of security.

Signature. The mediated Hess' identity-based signature scheme must satisfy some security properties in order to qualify as a secure and efficient scheme. In particular, we are interested in demonstrating that the following notions hold in our context:

- **Key Secrecy.** Neither the *User*'s private key share, nor the *SEM*'s should be derivable from any information available to any entity, excluding the *PKG*. This includes public parameters, signature shares, or any information given to a particular entity. We do

not consider the case where the *SEM* colludes with any other entity as it acts as an honest player in our setting.

- **Unforgeability.** It should not be possible to generate a valid signature for a particular message without the cooperation of a *User* and a *SEM*.
- **Non repudiation.** Neither the *User* nor the *SEM* should be able to deny having signed a message, provided a valid signature verifiable using one of the *User*'s identity exists.

We first prove here the correctness of the scheme by showing that the first operation of the signature verification leads to the relevant parameter r, necessary to produce the digest in order to verify the signature.

$$\begin{aligned}
r &= r_2.e(u_2, P).e(H_1(ID), -Y_{pkg})^v \\
&= e(P_2, P)^{-k_2}.e(u_1 + vS_{ID,sem} + k_2P_2, P).e(H_1(ID), -xP)^v \\
&= e(P_2, P)^{-k_2}.e(u_1, P).e(vS_{ID,sem}, P).e(k_2P_2, P).e(xH_1(ID), P)^{-v} \\
&= e(P_2, P)^{-k_2}.e(P_2, P)^{k_2}.e(vS_{ID,user} + k_1P_1, P).e(vS_{ID,sem}, P).e(S_{ID}, P)^{-v} \\
&= e(v(S_{ID,user} + S_{ID,sem}), P).e(k_1P_1, P).e(S_{ID}, P)^{-v} \\
&= e(S_{ID}, P)^v.e(S_{ID}, P)^{-v}.e(k_1P_1, P) \\
&= e(P_1, P)^{k_1}
\end{aligned}$$

The Hess' identity-based signature scheme relies on the hardness of the Diffie-Hellman problem in the random oracle model. We argue that its mediated version also relies on the hardness of the CBDH problem. Relying on these assumptions, we discuss the following properties in its mediated adaptation.

Key Secrecy. Upon receipt of the token (u_1, v) generated by a *User*, the *SEM* is not able to infer any information on the *User*'s private key share. The signing process being the same as in the original signature scheme, inferring a *User*'s private key share from her ID would amount to the DLP problem being tractable as it requires to determine x in $Y_{pkg} = xP$ to be able to generate private key shares. The token (u_2, v, r_2) issued by the *SEM* does not disclose any more information. r_2, just as r_1, contains information necessary to verify the signature. In the original protocol, r_1 is not transmitted as it is recovered during the first step of the verification process. If it was sent however, the scheme would rely on the hardness of the CBDH problem since recovering P_1 and k_1 would result in the total break of the system. The release of r_2 does not disclose any relevant information to

a *User*, no matter if this *User* is also the original signer. Indeed, from u_2, the latter can recover $vS_{ID,sem} + k_2P_2$. Even though, she knows the value of v, she still needs k_2 and P_2 to infer the *SEM*'s private key share. Finding these two values from r_2 is considered as intractable provided the hardness of the CBDH problem holds. Therefore, an attack on the mediated Hess' identity-based signature scheme can be viewed as a reduction to the attack on the original scheme.

Unforgeability. We consider here three different cases. First, we assume a *User* wishing to forge a valid signature on behalf of another *User*, but still using the *SEM* to complete the signature. In order to be valid, the *User* must be able to produce a value $S_{ID,user}$ which, when combined with $S_{ID,sem}$ would result as S_{ID}. The *User* must therefore either have the knowledge of S_{ID} and $S_{ID,sem}$ or have the knowledge of $S_{ID,sem}$ and being able to compute S_{ID}. Since S_{ID} and $S_{ID,sem}$ are kept secret from all *Users*, this means that the attacker must be able to compute S_{ID}, which is computationally infeasible since ECDLP is intractable, unless the *PKG*'s private key is disclosed. Secondly, the *SEM* alone trying to forge a valid signature leads to the same situation, as in he must either find the appropriate $S_{ID,user}$ corresponding to a particular ID or generate a new S_{ID}, the latter problem being intractable considering the hardness of the DLP problem. Finally, a *User* trying to bypass the *SEM* in order to produce a valid signature will encounter the same problems as if the *SEM* alone was trying to forge a signature.

Non-repudiation. In our context, Non-repudiation is achieved bearing in mind that the *PKG* is fully trusted not to use the private key shares he generates. Indeed, the value S_{ID} is needed to generate a valid signature and as seen earlier on, both the *User* and the *SEM* are required in order to jointly compute it. Since the private key shares used are unique, secretly kept by their *Users* and *SEM* and mathematically related, a valid signature can only be produced by the *User* owning the private key share corresponding to the ID used to verify it. This also means that the *SEM* completed the signature, which puts its responsibility at stakes.

The cryptographic algorithms used as part of the mediated identity-based signature scheme are based on Hess' formally proved identity-based signature scheme. In fact, the *User*'s signature process in the mediated identity-based PKI remains rigorously the same as in Hess' original scheme, while the *SEM* interaction only consists in completing the signature using the same scheme, both entities using a mathematically related private key shares. While we commented on the security of the overall protocol by attempting, with no success, to find scenarios whereby an attacker could break the scheme, we recognize the need for further investigation in order to find out whether any information leakage is possible. In particular, techniques such as model checking could be used in order to formally prove that

the protocol defined is secure.

System

Key generation is performed by the *PKG*. Since it is responsible for the generation of the two private key shares of a particular *User*'s private key, the *PKG* could in theory impersonate any *User* of the system. Hence, this entity needs to be fully trusted and should preferably be put offline. As seen earlier in Section 7.3, each user only owns one key pair for encryption and one for signature, no matter how many *LBS* they are registered with.

The security analysis of the algorithms shows that the decryption and signing processes must involve both a *User* and a *SEM* in order to produce a relevant result. A *SEM* will only participate in a decryption or signature process if the security capabilities of the corresponding *User* have not been revoked. Therefore, if a *SEM* is instructed to stop assisting cryptographic processes, *Instant (or immediate) Revocation* is provided in the sense that a *User* wishing to decrypt a message with a compromised private key share will not be issued with the decrypted token generated by the *SEM*. Also, a *User* trying to impersonate another entity will not be able to generate a full signature on a message since the *SEM* will refuse to cooperate with its corresponding private key share.

Following such a public key revocation, a *User* is required to obtain a new private key share for the corresponding revoked pseudonym, following the same key pair generation process as described earlier on. She can however still use her identity as her public key since the new private key shares will be derived from the original D_{ID}, see Section 7.5.2. We note also that, in the case of a *SEM*'s key pair revocation, only the *SEM* will necessitate a new private key share. However, if both the *SEM*'s and the *User*'s private key shares have been compromised, a digital signature produced by these two entities will always be verifiable, no matter if a new set of private key shares have been generated and issued to the relevant parties. This is due to the fact that only the *PKG*'s public key and the *User*'s identity are used to verify a signature. In this case, the *PKG* will be required to generate a new key pair for itself, making its public key available to all parties, and to re-generate and distribute all the *Users*' private key shares. In traditional identity-based encryption schemes, the revocation of one's private key implies that the corresponding public key can no longer be used (or at least for a certain period of time if it contained an expiry date [7]) unless the *PKG*'s system parameters are re-initialized. In our setting, such a situation may occur if and only if both *User*'s and *SEM*'s private key shares have been disclosed. This is obviously less likely to happen and, therefore, constitutes an advantage of using identity-based server

[7] A short lived public key may be defined as an expiry date concatenated to an identity, such as *bob@orientplatform1.com-01012005*.. This approach requires however frequent updates of private keys.

aided cryptography.

Once the revocation process is achieved, we note however that messages signed by the revoked key pair prior to their revocation and for which the signature was assisted by the *SEM* can still be validated. Plaintexts that were encrypted by both the *User*'s and the *SEM*'s private key shares cannot however be recovered with the new private key share. This is a natural consequence inherent to our design resulting from the fact that the same public key is always used while the corresponding private key may change.

7.8 Conclusion

In this chapter, we attempted to provide the *Orient Platform* with a security infrastructure that would remain as transparent as possible to *Users*. In particular, our main objectives were to:

- Identify a suitable mechanism providing end-to-end security at the application level.
- Attempt to simplify key management by minimizing the impact of key revocation.
- Provide a solution that could be integrated in a distributed Orient Platform architecture.

To achieve this, we employed a threshold asymmetric key encryption algorithm, whereby two parties are required to work in conjunction in order to produce a valid ciphertext decryption. As for the signature capabilities, we extended an identity-based signature algorithm to achieve the same goals without requiring the second party to actually sign the message but rather assist the signing process. By using mediated cryptography, we allow for the *Orient Platform* to check the validity of the *Users*' key pair every time a message passes through its proxy architecture and to provide therefore, *Instant Revocation* of security capabilities. The identity-based character of the encryption and signature algorithms simplifies the public key retrieval process since the public identifiers used are pseudonyms, known to all the parties involved. A slightly modified key generation process has also been designed in order to allow the pseudo-anonymity character of the transactions between *Users*. This process allows *Subjects* and *Targets* to own a single private key share corresponding to as many public key identifiers or pseudonyms as needed by the system.

We notice however that such a PKI fails to provide its *Users* with a way to sign and encrypt a message at the same time. There is therefore a necessity for signcryption schemes applicable in the context of mediated cryptography to be designed in the future. This may prove to be a complex problem since the *SEM*, upon verification of a *User*'s key pairs status,

would have to assist both the decryption and the signature of a message at the same time. Also, key escrow is inherent to the PKI design which, in some circumstances, could be problematic. However, since the *PKG* is trusted by all entities, it is understood that it will not try to impersonate any *Users* or tap into their communications. A possible solution would be to distribute the process of key generations between several *PKG* so that none of them is fully aware of a *User*'s or *SEM*'s private key share.

Whereas reasonably efficient on desktop computers, the PKI implementation on mobile devices demonstrates modest results, even though comparable to existing systems' performances. However, it remains more secure than currently used PKI since it provides instant revocation capabilities when other systems fail to enforce public key revocation checking. Also, improvements in the algorithms used and deployment of identity-based encryption technologies on embedded devices [8] will certainly contribute to enhance the performances of systems similar to ours [9].

[8] See `http://www.gemplus.com/press/archives/2004/id_security/02-11-2004-Identity-Based_Encryption.html`, where Gemplus announces the world's first Identity-Based Encryption implementation for smart cards.

[9] In this context, the PKI presented in this section would allow mobile users to store their private key shares in the SIM card. Mobile users would be issued a new SIM card every time their key pair is to be revoked.

Chapter 8

Conclusions and Future Work

8.1 Introduction

Software services that take into account the location of a particular entity already exist and are currently used in some applications. The first and best example remains the proximity services accessible through a 2G mobile network from a WAP-enabled mobile phone. Cinemas, restaurants and other points of interest can be located in the vicinity of the requestor using location technologies already implemented within mobile network architectures. Another example is the deployment of fleet management services where GPS devices are embedded in vehicles in order to track their position over time. The locations of vehicles are then transmitted to a central server using technologies such as SMS or MMS, depending on the radio network used. Stand-alone software or a Web interface may then be used to view, in real-time, the location of the tracked entities. Car insurance companies are also starting to use similar systems in order to provide their customers with tailored rates that depend on their driving habits.

For all these examples, the *LBS* are trusted not to misuse the location data of the entities tracked. Because they operate in a relatively closed environment, people tend to minimize their privacy concerns regarding the disclosure of their location information. However, the development of location determination technologies within mobile operators' architectures may change this attitude. Indeed, new regulations now require mobile operators to precisely locate their subscribers for emergency purposes. In order to minimize the costs of deploying such technologies and generate new revenue streams, mobile operators will open their networks to third parties, enabling them to use their resources and share their data. On one hand, this fosters the development of a new kind of *LBS*, where people can now, for example, enquire for the location of a mobile subscriber directly over the Internet. On

the other hand, this raises significant privacy issues regarding who can access this location information, how it will be used and how long it will be stored. The European Union, the U.S.A and Japan have already defined some privacy regulations in an attempt to protect mobile subscribers. However, implementation of these guidelines remain sporadic and it is proving to be a daunting task.

The research work carried out and summarized in this thesis identifies the privacy risks inherent to location information disclosure and discusses some interesting approaches that tackle related problems. It then proposed a possible solution implemented as a middleware platform and focuses on two of its main components. In this chapter, the main contributions of the work are summarized, along with some possible future research directions.

8.2 Thesis Summary

In the introductory Chapter 1, we explored how connected mobility, *i.e.* mobile devices with Internet access, together with the development and deployment of location-based technologies gave birth to a new range of mobile services based on location. After having stressed the growing privacy related concerns with regards to personal location information disclosure, we presented a first legal attempt to regulate location information handlers' practices in the European Union, the U.S.A. and Japan. We then defined the scope of this thesis by identifying the main problem it intends to tackle and outlined the expected contributions of the research work together with its limitations.

This thesis addresses user privacy in the context of location-based services. This field encompasses a broad range of concepts that find their roots in the three main areas of computer security, mobile telecommunications and Internet computing. In Chapter 2, we introduced the field of computer security, stressing the main principles of security algorithm design, describing the main infrastructures used and pointing out the two interesting areas of elliptic curve and threshold cryptography. We then provided a detailed description of currently used mobile phone networks. In particular, we identified their shortcomings in terms of security and discussed how they handled cellular-based location information. This chapter provided the background knowledge necessary to comprehend the notions developed in the following chapters.

Chapter 2 outlined the software and hardware environments in which location-based services operate. In Chapter 3, we refined the definition of this environment by analyzing the role of each entity and proposed a threat model. We then stated three main problems related to location privacy and pointed out the need for a middleware platform in order to interface mobile users with location-based services. The requirements for an application

level protocol in order to help *LBS* communicate with the middleware were also described, together with the requirements of a Public Key Infrastructure required for the environment considered.

After having defined what we expected in terms of research outcome in Chapter 3, we analyzed where the current research efforts were directed and how they could benefit the project. Chapter 4 presented an exhaustive review of the research related to location privacy in general, spanning from high level protocols and algorithms used to provide privacy, to mechanisms ensuring confidentiality and access control to location information. We refer the reader to Section 8.4 for a more precise description of this contribution.

In Chapter 5, we presented a prototype of the middleware and protocol devised in the previous sections, namely the *Orient Platform* and the *Orient Protocol*, and showed how they fulfilled the corresponding requirements. We then illustrated their use by providing the reader with an explanation of their integration amongst the various entities. A scenario was finally provided in order to show how a typical *LBS* request was handled by the newly defined architecture.

Chapter 6 introduced an algorithm called the *Location Blurring* algorithm that intentionally downgrades the quality of location information in order to provide *LBS* with the minimum acceptable accuracy in terms of mobile users' location while still remaining meaningful. This chapter provided the motivations, description of the main design issues, implementation and evaluation of this algorithm.

Finally, in Chapter 7, we described a Public Key Infrastructure that may be used in the environment considered. The algorithms outlined provide the architecture with the basic security services such as confidentiality, authentication, integrity and non-repudiation, by enabling an entity to perform encryption and digital signatures in an efficient way. We refer the reader to Section 8.3 for a more precise description of this contribution.

8.3 Major Contributions

8.3.1 A Location Blurring algorithm

The algorithm proposed in Chapter 6 is used to blur the location of a mobile user by returning an area containing her location in such a way that it is not possible for a *LBS* or any other entity to compute a more accurate location. This presents the advantage of maintaining a low but meaningful resolution of location information while preserving mobile users' privacy. The algorithm penalizes the location tracking process by providing cached location information when detected. On the contrary, it enables sporadic location queries

to get maximum location information accuracy. The *Location Blurring* algorithm solves the location inference problem in real-time. Along with the algorithm, we designed and implemented a testing framework as a discrete event system simulator that can be used to assess the efficiency of *Location Blurring* algorithms.

8.3.2 A Public Key Infrastructure

The PKI proposed in Chapter 7 is based on a Security Mediator (SEM) architecture that integrates seamlessly within the architecture we have designed in order to enable *LBS* provision over the Internet. The cryptographic algorithms used as part of the PKI are identity-based, which considerably simplifies public key management. In particular, the PKI implements a novel identity-based signature scheme based on Hess', which is, to our knowledge, the first identity-based signature scheme proposed in the SEM context. As a result, the PKI enables *Instant revocation* of security capabilities and removes the need to perform certificate retrieval, path validation and revocation in our setting. Also, a special private key generation process enables PKI users to use the same private key in conjunction with multiple pseudonyms as their public key. This way, they can access *LBS* under different identities without having to use different private keys.

8.4 Other contributions

8.4.1 A critical survey of context information security

In Chapter 4, we provided a detailed survey of the research efforts carried out to date in order to protect context information. We analyzed in particular the notions of *Identity Blurring* and *Location Blurring*, some approaches to context privacy policy definitions and different access control models for location information. Finally, we studied secure architectures designed for context aware applications and provided the reader with a description of the relevant standards.

8.4.2 A prototype for a secure architecture delivering *LBS* over the Internet

Chapter 5 presents a joint work carried out with Cameron Ross Dunne [73]. This work constitutes the basis for the research outlined in this thesis. It presents a proxy-based platform, *Orient Platform*, that handles *LBS*-related requests, performs identity translation and location presentation to *LBS*. A privacy interface lets users provision their privacy details

and a protocol called the *Orient Protocol* is proposed as a way to interface *LBS* with the *Orient Platform*.

8.5 Future work

8.5.1 The Orient Platform

There are three main areas that may need further research.

First, the middleware does not currently handle area-based queries; such queries would return all the locatable entities currently located in a specific area. Some *LBS* may indeed require this kind of information to provide a service. Some further versions of *Location Blurring* algorithms may also require the knowledge of other entities' location and conceal or disclose blurred location information in consequence. Therefore, there would be a need to extend both the *Orient Platform* and the *Orient Protocol* in order to support this change. Technically, this would involve extending the XML schemas specifying the *Orient Protocol*, and more particularly, the schema of the *LBS Initiated Service* (LIS) of the *Orient Protocol*. Instead of incorporating a location query for one individual, the *Target*'s identity XML object would be replaced by a shape XML object already used to define the location area returned by the *Location Blurring* algorithm. From the *Orient Platform*'s point of view, the technical upgrade would involve invoking the appropriate MLP [163] area query service and applying the *Location Blurring* algorithm to every *Target*'s *Sighting* returned. The *Charging Unit* internal calculations would need to be amended in order to take into account the new charging schemes supporting this kind of queries.

Secondly, fourth Generation networks (4G) are currently being defined. One of their main characteristics is that they aim to extend the roaming concept to different mobile network technologies, see [137]. For example, they may handle hand-overs between WiFi and UMTS networks. In our context, this means that the *Orient Platform* may need to cope with new kinds of users traveling in new environments. Technically, any wireless network involved in such hand-overs may individually locate its *Targets*. In the case of WiFi networks for example, *Targets* could be located relatively to their base stations using triangulation methods similar to those defined in Section 2.5.1. The same concept as the Location Translation tables defined in Section 5.3.2 could be used by the *Orient Platform* when acquiring location information from such a WiFi network : they would translate the relative positioning of a *Target* within the network into an absolute location by considering the fixed location of the network's access points.

Finally, there may be a need to inform the mobile user that she is being tracked and

give him the possibility to acknowledge the positioning request. At present, users can only access location request logs and cannot interactively influence the result of a request, by, for example, denying it. While this may prove to be awkward to implement and unnecessary since users may not wish to receive a notification every time their location is requested, some research in Human Computer Interaction may lead to an interface that could help provide a feedback on the information collected without disturbing the user experience. An example of such interface could be described as a simple web form pushed by the *Orient Platform* to the online *Target* everytime a request is performed in order to get her location, letting her grant or deny the access permission.

8.5.2 Location Blurring algorithm

The *Location Blurring* algorithm presented in this thesis constitutes the first step towards solving the daunting problem of location tracking. In particular, while it intends to solve real-time border issues, there is still research to be done in order to determine how to tackle retrospective border issues in order to make sure that a mobile users' past locations cannot be accurately inferred. One possible research direction may involve the study of the behavior of a variant of the algorithm presented in Section 6. This algorithm pre-computes *Blurred Sightings* from the last *Blurred Locality* released using the *Target*'s actual *Position* in order to get a value for the parameter *dist*. It then stores them in a *BSStore* repository. The variant of the algorithm would perform the same calculation but using the latest *Blurred Sighting* stored in the repository together with the *Target*'s actual *Position*. This would present the advantage of producing *Blurred Sightings* more readily available and may defeat some retrospective border attacks when the *BSStore* repository is not empty. Indeed the distance *dist* may not be accessible to the attacker since the *Blurred Sighting* it is calculated from might not have been released from the *BSStore* repository.

The most significant challenge to tackle remains, however, to propose a *Location Blurring* algorithm that is not subject to analytical attacks [1]. Some approaches already proposed in [96] rely on the assumption that the privacy of an individual depends on the number of people he can be mistaken with. Therefore, *Blurred Localities* do not have a constant shape and do not depend on the environment characteristics but more on people's behaviors. Our *Location Blurring* algorithm releases a new *Blurred Sighting* depending on the time it takes a *Target* to travel the longest straight line distance achievable between two consecutive *Blurred Localities*. The size of *Blurred Localities* is fixed and corresponds to a privacy preference set by the *Target*. One possible approach to tackle analytical attacks may therefore consist in

[1] Attacks that use environment characteristics such as the presence of buildings, roads, direction of traffic flow, etc. to infer somebody's location.

dynamically calculating *Blurred Localities* depending on the effective area a *Target* can be located in. For example, when a *Target* specifies she wishes to be located with an accuracy of 100 meters (or located in a disc of area approximatively equal to $3141m^2$), the next *Blurred Sighting* released for this *Target* will have its *Blurred Locality* computed such that the *Target* can be located uniformly in an acceptable area of dimension $3141m^2$. By acceptable area, we mean an area where the *Target* is the most likely to be located (a road when travelling at a certain speed with a direction parallel to that road for example). This will involve carrying out calculations with vector maps in real-time and may significantly increase the size of *Blurred Localities*. However, this approach, combined with other techniques such as *Temporal Blurring*, may lead to a practical solution to analytical attacks.

Finally, there is also a need for a mathematical model to be developed in order to formally assess the security and efficiency of *Location Blurring* algorithms. In order to do so, research fields such as Geographic Information Systems (GIS) and Spatial Analysis [62] may provide a basis to design such model. In particular, one approach could indeed consist in applying probability theory to the current *Location Blurring* algorithm in order to assess to what extent retrospective border issues can leak privacy sensitive location information. Furthermore, by using location estimation techniques [110], it may be possible to predict the probability of presence of a *Target* in a particular *Blurred Locality*, which would provide a valid reason to release or to retain the corresponding *Blurred Sighting* to the requestor.

8.5.3 PKI

We believe that the concept of a SEM architecture suits perfectly the security needs of mobile users in the context of location-based services delivery. However, we also believe that several aspects of the work presented in Chapter 7 need to be further investigated in order to improve on the PKI current performance. In particular, there is a need to investigate cryptographic algorithms that use less pairings and less scalar multiplications in order to enable mobile devices to quickly perform cryptographic operations. These two operations are indeed considered as the most resource consuming ones in the algorithm proposed.

Also, signcryption schemes may represent a possible future research direction since they do not exist yet in the context of a SEM architecture and constitute an open problem in this field. The issues lies in the fact that while a PKI *User* may perform the signing and encryption processes at the same time, the *SEM* would have to partially sign the token received and partially decrypt it at the same time. One possible approach may involve researching a cryptographic primitive that allows the encryption and signing operations to commute in order to allow for this to happen.

Finally, we recognize that the Public Key Infrastructure developed may suffer from a certain number of issues. One of them is the key escrow problem, whereby a private key is generated by the *Private Key Generator* and kept such that it can generate new private key shares every time a *Subject* needs a new pseudonym when registering to a new *LBS*. Certificateless public key encryption [51] advocates the use of a user generated private key together with a public identifier published by the same user, but which does not need to be validated, in order to remove the key escrow problem in identity-based cryptosystems. Adapting such a concept to our environment could remove the need for a *PKG* and therefore the need for key escrow. The other main issue that faces our mediated PKI is scalability. Indeed, only one central instance of the Security Mediator in not conceivable since it may go down or even become compromised. Some solutions to this problem have been proposed in [212], where the authors present some designs that allow for backup and migration processes between *SEM* instances. Since IBE-mRSA (see Section 2.3.4) is used as the main cryptographic primitive, it would be interesting to investigate whether their approach is adaptable to our context and how it can be extended in order to provide services such as protection against Denial-Of-Service attacks.

8.6 Concluding remarks

In the introduction of this chapter, we provided the reader with some examples of *LBS* and pointed out the possible risks of location tracking. In this thesis, we have attempted to introduce new approaches in order to tackle these location privacy issues. However, one could wonder whether all these privacy enhancing techniques are really necessary. Tracking an individual using her mobile phone may indeed raise privacy concerns amongst the general public. However, as noted in [141] by Martin Dodge, people are already being tracked in their day-to-day life and do not seem to be concerned about it. According to this study, there exist three main forms of people tracking:

- Sporadic tracking by transactions. Transactions made with debit or credit cards, digitally controlled physical access can disclose one's location with a precise space and time resolution.

- Visual tracking by cameras. The best example remains London's "Ring Of Steel" network of surveillance where CCTV cameras are installed in most buses, trains but also at crossroads and roundabouts in order to monitor traffic and ensure people safety.

- Mobile tracking, as defined in this thesis.

The author points out however that mobile tracking may prove to be more harmful than the other two since it potentially allows for more continuous and complete geosurveillance[2]. This provides the adequate justification to pursue research in this field and attempt to prevent Big Brother from watching us.

[2]A mobile phone subscriber is known to the mobile network operator it is connected to, in terms of identity and location.

List of Acronyms

3GPP 3rd Generation Partnership Project

ACL Access Control List

AES Advanced Encryption Standard

A-GPS Assisted GPS

AMPS Advanced Mobile Phone System

ANSI American National Standards Institute

API Application Programming Interface

APN Access Point Name

APPEL A P3P Preference Exchange Language

ARPU Average Revenue Per User

A-S Anti-spoofing

AuC Authentication Center

BSC Base Station Controller

BSS Base Station Subsystem

BST Base Station Transceiver

CA Certificate Authority

CDMA Code Division Multiple Access

CDP CRL Distribution Point

CL Capabilities List

CRL Certificate Revocation List

CRS Certificate Revocation Status

CRT Certificate Revocation Tree

DAC Discretionary Access Control

DAP Directory Access Protocol

DES Data Encryption Standard

DHCP Dynamic Host Configuration Protocol

DIT Directory Information Tree

DLP Discrete Logarithm Problem

DNS Domain Name System

DPD Delegated Path Discovery

DPV Delegated Path Validation

DSA Directory System Agent

DUA Directory User Agent

ECC Elliptic Curve Cryptography

ECDLP Elliptic Curve Discrete Logarithm Problem

E-OTD Enhanced Observed Time Difference

ETSI European Telecommunications Standards Institute

FCC Federal Communications Commission

FDMA Frequency Division Multiple Access

GEA GPRS Encryption Algorithm

GGSN Gateway GPRS Support Node

GMLC Gateway Mobile Location Center

GPRS General Packet Radio Service

GPS Global Positioning System

GSM Global System for Mobile communication

GSMA GSM Association

HLR Home Location Register

IBE Identity-Based Encryption

IETF Internet Engineering Task Force

IMEI International Mobile Equipment Identity

IMSI International Mobile Subscriber Identity

ISDN Integrated Services Digital Network

ISP Internet Service Provider

JNI Java Native Interface

LAI Location Area Identity

LAN Local Area Network

LBS Location-Based Service

LDAP Lightweight Directory Access Protocol

LLC Logical Link Layer

LMU Location Measurement Unit

LMU Location Management Unit

MAC Mandatory Access Control

MAC Message Authentication Code

ME Mobile Equipment

MIME Multipurpose Internet Mail Extensions

MLP Mobile Location Protocol

MMPG Mobile Message Packet Gateway

MS Mobile Station

MSC Mobile Switching Center

MSISDN Mobile Station International ISDN Number

NIST National Institute of Standards and Technology

NSA National Security Agency

NSS Network Subsystem

OCSP Online Certificate Status Protocol

OMA Open Mobile Alliance

P3P Platform for Privacy Preferences Project

PCU Packet Control Unit

PDC-P Personal Digital Cellular Packet

PGP Pretty Good Privacy

PIN Personal Identification Number

PKG Public Key Generator

PKI Public Key Infrastructure

PLMN Public Land Mobile Network Area

PMU Profile Management Unit

PSTN Public Switched Telephone Network

RA Registration Authority

RADIUS Remote Authentication Dial In User Service

RBAC Role-based Access Control

RDN Reduced Distinguish Name

RF Radio Frequency

RTT Round Trip Time

SA Selective Availability

SB Suicide Bureau

SDSI Simple Distributed Security Infrastructure

SEM Security Mediator

SET Secure Electronic Transaction

SGSN Serving GPRS Support Node

SIM Subscriber Identity Module

SMLC Serving Mobile Location Center

SMS Short Message Service

SOAP Simple Object Access Protocol

SPKI Simple Public Key Infrastructure

SSH Secure Shell

SSL Secure Socket Layer

TA Timing Advance

TACS Total Access Communication System

TDMA Time Division Multiple Access

TLS Transport Layer Security

TMSI Temporary Mobile Subscriber Identity

TA Time of Advance

UMTS Universal Mobile Telecommunication System

URI Uniform Resource Identifier

URL Uniform Resource Locator

USIM UMTS Subscriber Identity Module

UTRAN UMTS Terrestrial Radio Access Network

VLR Visitor Location Register

W3C World Wide Web Consortium

WAP Wireless Application Protocol

WIM Wireless Identity Module

WCDMA Wideband CDMA

Wi-Fi Wireless Fidelity

WLAN Wireless LAN

WPKI Wireless PKI

WTLS Wireless Transport Layer Security

WWW World Wide Web

Bibliography

[1] European Telecommunication Standard Institute (ETSI). http://www.etsi.org/.

[2] List of officially launched WCDMA networks. Available on: http://www.umtsworld.com/industry/livelist.htm.

[3] NTT DoCoMo. Available on: http://www.nttdocomo.co.jp.

[4] The unofficial independent iMode FAQ. Available at Eurotechnology.com: http://www.eurotechnology.com/imode/faq.html.

[5] The WAP Forum. Available on: http://www.wapforum.org.

[6] W-CDMA Specifications. Available on: http://www.umtsworld.com/technology/wcdma.htm.

[7] WAP 2.0 Technical Specifications. Available on: http://www.openmobilealliance.org/tech/affiliates/LicenseAgreement.asp?DocName=/wap/technical_wap2_0-20020813.zip.

[8] WAP TM Transport Layer E2E Security Specification. Available on: http://www1.wapforum.org/tech/documents/WAP-187-TransportE2ESec-20000711-a.pdf.

[9] Wireless Application Protocol Wireless Transport Layer Security Specification (WTLS). Available on: http://www1.wapforum.org/tech/documents/WAP-199-WTLS-20000218-a.pdf.

[10] 3G an upgrade of 2G. Land Mobile magazine Available on: http://www.landmobile.co.uk/download/upgrade-2g.pdf, May 2001.

[11] 3GPP TS 35.202 KASUMI specification, Specification of the 3GPP Confidentiality and Integrity Algorithms, Document 2, ETSI/SAGE. Available on: http://www.3gpp.org/ftp/Specs/archive/35_series/35.202/35202-600.zip, 2005.

[12] 3GPP Technical Specification Group Services and System Aspects. Specification of the MILENAGE Algorithm Set: An example algorithm set for the GSM Authentication and Key Generation functions A3 and A8 (Release 6). Available on: `http://www.gsmworld.com/using/algorithms/docs/55205-600.pdf`, December 2002.

[13] 3GPP Technical Specification Group Services and System Aspects. 3GPP TS 23.271 V6.9.0, Functional stage 2 description of Location Services (LCS) (Release 6). Available on: `http://www.3gpp.org/ftp/Specs/archive/23_series/23.271/23271-690.zip`, September 2004.

[14] 3GPP Technical Specification Group Services and System Aspects. 3GPP TS 33.102 - Technical Specification Group Services and Security Aspects 3G Security architecture (Release 6). Available on: `http://www.3gpp.org/ftp/Specs/archive/33_series/33.102/33102-630.zip`, December 2004.

[15] 3GPP Technical Specification Group Services and System Aspects. Technical realization of the Short Message Service (SMS). Available on: `http://www.3gpp.org/ftp/Specs/archive/23_series/23.040/23040-650.zip`, September 2004.

[16] 3GPP Technical Specification Group Services and System Aspects. 3GPP TS 21.101 Technical Specifications and Technical Reports for a UTRAN-based 3GPP system. Available on: `http://www.3gpp.org/ftp/Specs/archive/21_series/21.101/21101-620.zip`, June 2005.

[17] 3GPP Technical Specification Group Services and System Aspects. 3GPP TS 35.201 - Technical Specification Group Services and System Aspects 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms Document 1. f8 and f9 Specification (Release 5). Available on: `http://www.3gpp.org/ftp/Specs/archive/35_series/35.201/35201-600.zip`, January 2005.

[18] 3GPP TS 03.71 version 8.9.0 Release 1999. Digital cellular telecommunications system (Phase 2+). Location Services (LCS). Functional description. Stage 2. Available on: `http://webapp.etsi.org/action%5CPU/20040720/ts_101724v080900p.pdf`, June 2004.

[19] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a Better Understanding of Context and Context-Awareness. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, pages 304–307, Karlsruhe, Germany, 1999. Springer-Verlag.

[20] L. Ackerman, J. Kempf, and T. Miki. Wireless Location Privacy: Law and Policy in the US, EU and Japan. Online article available on: `http://www.isoc.org/briefings/015/index.shtml`, 2003.

[21] Linda Ackerman, James Kempf, and Toshio Miki. Wireless Location Privacy: Law and Policy in the U.S.A., EU and Japan. Available as an Isoc Member Briefing #15 on: `http://www.isoc.org/briefings/015/briefing15.pdf`, 2003.

[22] C. Adams and S Farrell. Internet X.509 Public Key Infrastructure Certificate Management Protocols, RFC 2510. Available on: `http://www.faqs.org/rfcs/rfc2510.html`, March 1999.

[23] Agence France Presse. Mobile Phones Outstrip Landline Connections in India. Available on: `http://www.mobimarketing.com/tips/index.php?p=21`, October 2004.

[24] Aircoach. Available on: `http://www.aircoach.ie/`.

[25] Jalal Al-Muhtadi, Roy Campbell, Apu Kapadia, M. Dennis Mickunas, and Seung Yi. Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments. In *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02)*, pages 74–83. IEEE Computer Society, 2002.

[26] Pierre Antoine. Understanding The Mobile Phone Market Drivers. Alcatel Telecommunications Review. Available on: `http://www.mobimarketing.com/tips/index.php?p=21`, 2004.

[27] Paul Ashley, Heather Hinton, and Mark Vandenwauver. Wired versus Wireless Security: The Internet, WAP and iMode for E-Commerce. In *Proceedings of the 17th Annual Computer Security Applications Conference (ASAC'01)*, pages 296–208, Sheraton New Orleans, Louisiana, U.S.A., December 2001. IEEE Computer Society.

[28] ATT. Technology Timeline. Available on: `http://www.att.com/attlabs/reputation/timeline/46mobile.html`.

[29] Joonsang Baek and Yuliang Zheng. Identity-Based Threshold Decryption. In *Proceedings of Practice and Theory in Public Key Cryptography (PKC'04)*, volume 2947 of *Lecture Notes in Computer Science*, pages 262–276, Singapore(SG), March 2004. Springer-Verlag.

[30] David M. Balston. The pan-European system: GSM. *Cellular Radio Systems*, 1993.

[31] L. Barkhuus and Anid .K. Dey. Location-Based Services for Mobile Telephony: a Study of Users' Privacy Concerns. In *Proceedings of Ninth IFIP TC13 International Conference on Human-Computer Interaction (INTERACT'03)*, pages 709–712, Zurich, Switzerland, 2003. ACM Press.

[32] Louise Barkhuus. Context Information in Mobile Telephony. In *Proceedings of the Mobile Human Computer Interaction 2003 (MobileHCI'03)*, pages 451–455, Udine, Italy, 2003. ACM Press.

[33] Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. Efficient Implementation of Pairing-Based Cryptosystems. *Journal of Cryptology*, pages 321–334, September 2004.

[34] Elisa Batista. WAP or I-Mode: Which Is Better? Available at Wired.com: `http://www.wired.com/news/wireless/0,1382,38333,00.html`, August 2000.

[35] Olivier Benoit, Nora Dabbous, Laurent Gauteron, Pierre Girard, Helena Handschuh, David Naccache, Stephane Socie, and Claire Whelan. Mobile Terminal Security. Cryptology ePrint Archive, Report 2004/158, 2004. Available on: `http://eprint.iacr.org/`.

[36] Alastair R. Beresford and Frank Stajano. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[37] Alastair R. Beresford and Frank Stajano. Mix Zones: User Privacy in Location-aware Services. In *Proceedings of the First IEEE International Workshop on Pervasive Computing and Communication Security (PerSec'04)*, pages 127–132, Orlando, Florida, U.S.A., March 2004.

[38] Harini Bharadvaj, Anupam Joshi, and Sansanee Auephanwiriyakul. An Active Transcoding Proxy to Support Mobile Web Access. In *Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems*, pages 118–126, Washington, DC, U.S.A., 1998. IEEE Computer Society.

[39] Eli Biham, Elad Barkan, and Nathan Keller. Instant Ciphertext-only cryptanalysis of GSM encrypted communications. In *Proceedings of Advances in Cryptology (CRYPTO'03)*, volume 2729 of *Lecture Notes in Computer Science*, pages 600–616, Santa Barbara, California, U.S.A., 2003. Springer.

[40] Alex Biryukov, Adi Shamir, and David Wagner. Real Time Cryptanalysis of A5/1 on a PC. In *Proceedings of the 7th International Workshop Fast Software Encryption*

(FSE 2000), volume 1978 of *Lecture Notes in Computer Science*, pages 1–18, New York, NY, U.S.A., 2001. Springer.

[41] Dan Boneh. Twenty Years of Attacks on the RSA Cryptosystem. *Notices of the American Mathematical Society (AMS)*, 46(2):203–213, 1999.

[42] Dan Boneh, Xuhua Ding, Gene Tsudik, and C. Wong. A Method for Fast Revocation of Public Key Certificates and Security Capabilities. In *Proceedings of the 10th USENIX Security Symposium*, pages 297–310, Washington, D.C., U.S.A., 2001. USENIX.

[43] Dan Boneh and Matthew Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

[44] Marc Branchaud and John Linn. Extended Validation Models in PKI: Alternatives and Implications. In *Proceedings of the 1st Annual PKI Research Workshop*, pages 37–44, NIST, Gaithersburg MD, U.S.A., April 2002.

[45] Charles Brookson. GPRS Security. Available on: `http://www.brookson.com/gsm/gprs.pdf`, December 2001.

[46] Thibault Candebat, Cameron Ross Dunne, and David Gray. Architecture and Protocol for delivering Location Based Services over the Internet. Technical Report DCU-CS-08-01, Faculty of Engineering and Computing, Dublin City University, Dublin, Ireland, August 2004.

[47] Certicom. Certicom Announces Elliptic Curve Cryptosystem (ECC) Challenge Winner (ECCp-109). Available on: `http://www.certicom.com/index.php?action=company,press_archive&view=121`, November 2002.

[48] Dung Chang. Security Along the Path Through GPRS Towards 3G Mobile Telephone Network Data Services. Available on: `http://www.sans.org/rr/papers/68/165.pdf`, January 2002.

[49] David L. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[50] Guanling Chen and David Kotz. A Survey of Context-Aware Mobile Computing Research. Technical Report TR2000-381, Department of Computer Science, Dartmouth College, U.S.A., November 2000.

[51] Zhaohui Cheng and Richard Comley. Efficient Certificateless Public Key Encryption. Cryptology ePrint Archive, Report 2005/012, 2005. Available on: `http://eprint.iacr.org/`.

[52] S. Chokhani, W. Ford, R. Sabett, C. Merrill, and S. Wu. Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework, RFC 3647. Available on: `http://www.faqs.org/rfcs/rfc3647.html`, November 2003.

[53] Niels Christian and Juul Niels Jrgensen. Security Issues in Mobile Commerce using WAP. In *Proceedings of the 15th Bled Electronic Commerce Conference e-Reality: Constructing the e-Economy*, pages 444–462, Bled, Slovenia, June 2002.

[54] Dwaine E. Clarke. SPKI/SDSI HTTP Server / Certificate Chain Discovery in SPKI/SDSI. Master's thesis, Department of Electrical Engineering and Computer Science, Massachussetts Institute of Technology, September 2001.

[55] Tom Clements. Making Sense of Cellular. Available on: `http://developers.sun.com/techtopics/mobility/getstart/articles/radio/`, July 2002.

[56] Sunny Consolvo, Ian E. Smith, Tara Matthews, Anthony LaMarca, Jason Tabert, and Pauline Powledge. Location Disclosure to Social Relations: Why, When, & What People Want to Share. In *Proceedings of the SIGCHI conference on Human factors in Computing Systems (CHI'05)*, pages 81–90, Portland, Oregon, U.S.A., 2005. ACM Press.

[57] Patricia Dockhorn Costa, Luis Ferreira Pires, Marten van Sinderen, and Jose Goncalves Pereira Filho. Towards a Services Platform for Mobile Context-Aware Applications. In *Proceedings of the First International Workshop on Ubiquitous Computing (IWUC 2004)*, pages 48–62, Porto, Portugal, April 2004.

[58] M. Covington, M. Moyer, and M. Ahamad. Generalized Role-based Access Control for Securing Future Applications. In *Proceedings of the 23rd National Information Systems Security Conference*, pages 115–126, Baltimore, MD, U.S.A., May 2000. IEEE Computer Society.

[59] Lorrie Cranor, Marc Langheinrich, and Massimo Marchiori. A P3P Preference Exchange Language 1.0 (APPEL1.0), W3C. Available on: `http://www.w3.org/TR/P3P-preferences/`, April 2002.

[60] Lorrie Cranor, Marc Langheinrich, Massimo Marchiori, Martin Presler-Marshall, and Joseph Reagle. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, W3C. Available on: `http://www.w3.org/TR/P3P/`, April 2002.

[61] Cybit Ltd. mapAmobile. Available on: `http://www.mapamobile.com/`.

[62] Daniel L. Falbo and Lloyd P. Queen and Charles R. Blinn. Introduction to Data Analysis Using Geographic Information Systems. Regents of the University of Minnesota, 2002. Available on: `http://www.extension.umn.edu/distribution/naturalresources/DD5740.html`.

[63] J. Dankers, T. Garefalakis, R. Schaffelhofer, and T. Wright. Public Key Infrastructure in Mobile Systems. *Electronics and Communication engineering Journal*, 12:180–190, October 2002.

[64] M. Danley, D. Mulligan, J. Morris, and J. Peterson. Threat Analysis of the Geopriv Protocol, RFC3694. Available on: `http://www.ietf.org/rfc/rfc3694.txt`, February 2004.

[65] Dorothy E. Denning. *Information Warfare and Security.* Addison-Wesley Publishers, 1999.

[66] T. Dierks and C. Allen. The TLS Protocol - Version 1.0 RFC 2246. Available on: `http://www.faqs.org/rfcs/rfc2246.html`, 1999.

[67] Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.

[68] Xuhua Ding and Gene Tsudik. Simple Identity-Based Encryption with mediated RSA. In *Proceedings of Progress in Cryptology (CT-RSA'03)*, volume 2612, pages 193–210, Berlin, Germany, 2003. Springer-Verlag.

[69] K. M. Divyan, R. Deng, J. Zhou, and K. Kim. A Secure and Privacy Enhanced Location-based Service Transaction Protocol in Ubiquitous Computing Environment. In *Proceedings of the Symposium on Cryptography and Information Security (SCIS'04)*, pages 931–936, Sendai, Japan, January 2004.

[70] NTT DoCoMo. DoCoMo's iMode: Towards Mobile Multimedia in 3G,presentation given at IETF Meeting, IETF 47, Plenary Session. Available on: `http://www.ietf.org/proceedings/00mar/slides/plenary-imode-00mar/sld002.htm`, March 2000.

[71] Marie-Anne Dru and Stephane Saada. Alcatel Telecom Review: Location-Based Mobile Services: The Essential. Available on: `http://atr.alcatel.de/hefte/01i_1/gb/pdf_gb/14drugb.pdf`, 2001.

[72] Adam Duffy and Tom Dowling. An Object Oriented Approach to an Identity Based Encryption Cryptosystem. In *Proceedings of the 8th IASTED International Conference*

on Software Engineering and Applications, volume 436, pages 45–52, Cambridge, MA, U.S.A., November 2004. ACTA Press.

[73] Cameron Ross Dunne. *An Infrastructure for Location Based Services on the Internet.* PhD thesis, Faculty of Engineering and Computing, School of Computing, Dublin City University, Dublin, Ireland, July 2005.

[74] Serge Egelman, Josh Zaritsky, and Anita Jones. Improved Certificate Revocation with OCSP. Available on `http://www.guanotronic.com/~serge/paper.pdf`.

[75] Carl Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian Thomas, and Tatu Ylonen. SPKI certificate theory, RFC 2693. Available on: `http://www.faqs.org/rfcs/rfc2693.html`, September 1999.

[76] Carl Ellison and Bruce Schneier. Ten Risks of PKI: What You're Not Being Told About Public Key Infrastructure. *Computer Security Journal*, 16(1):1–7, 2000. Available on: `http://www.schneier.com/paper-pki.pdf`.

[77] EPL Communications Limited. Location Based Services - Product Information. Available on: `www.zimepl.com/products/pdfs/datasheets/lbs-datasheet.pdf`, 2003.

[78] Alberto Escudero-Pascual and Gerald Q. Maguire. Role(s) of a Proxy in Location Based Services. In *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'02)*, volume 3, pages 1252–1256, Lisbon, Portugal, September 2002.

[79] ETSI/SAGE. 3GPP TS 55.217 Specification of the A5/3 Encryption Algorithms for GSM and EDGE, and GEA3 Encryption Algorithm for GPRS, Document 2: Implementators Test Data. Available on: `http://www.3gpp.org/ftp/Specs/archive/55_series/55.217/55217-610.zip`, 2003.

[80] Scott Fairbrother. Certificate Revocation in Public Key Infrastructures. Available on: `http://www.giac.org/practical/GSEC/Scott_Fairbrother_GSEC.pdf`, July 2003.

[81] H. Federrath, A. Jerichow, and A. Pfitzmann. MIXes in Mobile Communication Systems: Location Management with Privacy. *Information Hiding*, pages 121–135, 1996.

[82] Shereen Fink. The Fine Line Between Location-Based Services & Privacy. Sun Microsystems. Available on: `http://www.jlocationservices.com`.

[83] Warwick Ford, Phillip Hallam-Baker, Barbara Fox, Blair Dillaway, Brian LaMacchia, Jeremy Epstein, and Joe Lapp. XML Key Management Specification (XKMS). Available on: `http://www.w3.org/TR/xkms/`, March 2001.

[84] Barbara Fox and LaMacchia. Certificate Revocation: Mechanics and Meaning. In *Proceedings of the International Conference on Financial Cryptography*, volume 1465 of *Lecture Notes in Computer Science*, pages 158–164. Springer-Verlag, 1998.

[85] France Telecom Group. Available on: `http://www.francetelecom.com`.

[86] Anand S. Gajparia, Chris J. Mitchell, and Chan Yeob Yeun. Using Constraints to Protect Personal Location Information. In *Proceedings of the IEEE Semiannual Vehicular Technology Conference (VTC'03)*, volume 3, pages 2112–2116, Orlando, Florida, U.S.A., October 2003. IEEE Press.

[87] Anand S. Gajparia, Chris J. Mitchell, and Chan Yeob Yeun. The Location Information Preference Authority: Supporting User Privacy in Location Based Services. In *Proceedings of the 9th Nordic Workshop on Secure IT-systems (Nordsec'04)*, pages 91–96, Helsinki, Finland, November 2004.

[88] Fabien L. Gandon and Norman M. Sadeh. Semantic Web Technologies to Reconcile Privacy and Context Awareness. *Journal of Web Semantics*, 1(3):241–260, 2004.

[89] Simson Garfinkel. *PGP: Pretty Good Privacy.* O'Reilly and associates, 1995.

[90] P. Gaudry, F. Hess, and N. Smart. Constructive and destructive facets of weil descent on elliptic curves. *Journal of Cryptology*, 15:19–46, 2002.

[91] Peter S. Gemmell. An Introduction to Threshold Cryptography. *RSA's Laboratories CryptoBytes (Technical Newsletter)*, 2(3):7–12, Winter 1997. Available on: `ftp://ftp.rsasecurity.com/pub/cryptobytes/crypto2n3.pdf`.

[92] Geographic Location/Privacy (geopriv). Available on: `http://www.ietf.org/html.charters/geopriv-charter.html`.

[93] George M. Giaglis, Panos Kourouthanassis, and Argirios Tsamakos. Towards a Classification Framework for Mobile Location Services. *Mobile Commerce: Technology, Theory, and Applications*, pages 67–85, 2003.

[94] Dieter Gollmann. *Computer security.* John Wiley & Sons, Inc., 1999.

[95] Marco Gruteser, Jonathan Bredin, and Dirk Grunwald. Path Privacy in Location-aware Computing. Available on: `http://www.sigmobile.org/mobisys/2004/context_awareness/papers/paper.pdf`, June 2004.

[96] Marco Gruteser and Dirk Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of the First International*

Conference on Mobile Systems, Applications and Services (MobiSys'03), pages 31–42, San Francisco, CA, May 2003. USENIX Press.

[97] Marco Gruteser and Xuan Liu. Protecting Privacy in Continuous Location-Tracking Applications. *IEEE Security and Privacy*, 2(2):28–34, March–April 2004.

[98] GSM Association. GSM Security Algorithms. Available on: `http://www.gsmworld.com/using/algorithms/index.shtml`.

[99] GSM Association. GSM World. Available on: `http://www.gsmworld.com/index.shtml`.

[100] GSM Association. Location-Based Services, version 3.1.0. Available on: `http://www.gsmworld.com/documents/lbs/se23.pdf`, January 2003.

[101] H. Schulzrinne, et al. A Document Format for Expressing Privacy Preferences for Location Information, Internet Draft. Available on: `http://www.ietf.org/internet-drafts/draft-ietf-geopriv-policy-03.txt`, October 2004.

[102] Vesa Hametvaara. Certificate management in mobile devices. Master's thesis, Department of Computer and Information Sciences, University of Tampere, Finland, 2002.

[103] Yongfei Han, Peng-Chor Leong, Peng-Chong Tan, and Jiang Zhang. Fast Algorithms for Elliptic Curve Cryptosystems over Binary Finite Fields. In *Proceedings of Advances in Cryptology (Asiacrypt'99)*, volume 1716 of *Lecture Notes in Computer Science*, pages 75–85, Singapore, Singapore, November 1999. Springer-Verlag.

[104] Darrel Hankerson and Alfred Menezes. Elliptic Curve Discrete Logarithm Problem. Available on: `http://www.win.tue.nl/~henkvt/ecdlp.pdf`, May 2004.

[105] Christian Hauser and Matthias Kabatnik. Towards Privacy Support in a Global Location Service. In *Proceedings of the IFIP Workshop on IP and ATM Traffic Management (WATM/EUNICE'01)*, pages 81–89, Paris, France, September 2001.

[106] Urs Hengartner and Peter Steenkiste. Protecting Access to People Location Information. In *Proceedings of First International Conference on Security in Pervasive Computing (SPC'03)*, Lecture Notes in Computer Science, pages 25–38, Boppard, Germany, March 2003. Springer-Verlag.

[107] Urs Hengartner and Peter Steenkiste. Implementing access control to people location information. In *Proceedings of the ninth ACM Symposium on Access Control Models and Technologies*, pages 11–20, Yorktown Heights, New York, U.S.A., 2004. ACM Press.

[108] Florian Hess. Efficient Identity Based Signature Schemes Based on Pairings. In *Proceedings of the 9th Annual International Workshop on Selected Areas in Cryptography (SAC'02)*, pages 310–324, London, UK, 2003. Springer-Verlag.

[109] Sean Heyen. Cellular phone conversation easy prey for eavesdroppers. Access Control and Security Systems. Available on: `http://securitysolutions.com/mag/security_cellular_phone_conversation/`, August 1997.

[110] Jeffrey Hightower and Gaetano Borriello. Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study. In *Proceedings of the Sixth International Conference on Ubiquitous Computing (Ubicomp 2004)*, volume 3205 of *Lecture Notes in Computer Science*, pages 88–106. Springer-Verlag, September 2004.

[111] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System, Theory and Practice*. Springer-Verlag Wien, New York, 2001.

[112] Fritz Hohl, Uwe Kubach, Alexander Leonhardi, Kart Rothermel, and Markus Schwehm. Next Century Challenges: Nexus. An Open Global Infrastructure for Spatial-Aware Applications. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 249–255, Seattle, Washington, United States, 1999. ACM Press.

[113] Cay Horstmann and Gary Cornell. *Core Java 2, Volume I: Fundamentals (6th Edition)*. Pearson Education, August 2002.

[114] R. Housley, W. Ford, W. Polk, and D. Solo. Internet X.509 Public Key Infrastructure. Certificate and Certificate Revocation List CRL Profile. Internet RFC 3280. Available on: `http://www.faqs.org/rfcs/rfc3280.html`, April 2002.

[115] Richard Hull, Bharat Kumar, Daniel Lieuwen, Peter F. Patel-Schneider, Arnaud Sahuguet, Sriram Varadarajan, and Avinash Vyas. Enabling Context-Aware and Privacy-Conscious User Data Sharing. In *Proceedings of the IEEE International Conference on Mobile Data Management (MDM'04)*, pages 187–198, Berkeley, California, U.S.A., January 2004. IEEE Computer Society.

[116] J. Indulska, T. McFadden, M. Kind, and K. Henricksen. Scalable Location Management for Context-Aware Systems. In *Proceedings of the the Fourth IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems*, volume 2893, pages 224–235, Paris, France, November 2003. Springer-Verlag.

[117] International Engineering Consortium. Time Division Multiple Access (TDMA). Available on: `http://www.iec.org/online/tutorials/tdma/`, 2004.

[118] K. Ireland and M. Rosen. *A Classical Introduction to Modern Number Theory.* Springer Verlag, 1990.

[119] Glenn Judd and Peter Steenkiste. Providing Contextual Information to Pervasive Computing Applications. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*, pages 133–142, Fort Worth, Texas, USA, March 2003. IEEE Computer Society.

[120] Y. Kawatsura. RFC 3538 - Secure Electronic Transaction (SET) Supplement for the v1.0 Internet Open Trading Protocol (IOTP). Available on: `http://www.faqs.org/rfcs/rfc3538.html`, June 2003 1997.

[121] Auguste Kerckhoffs. La Cryptographie Militaire. *Journal des Sciences Militaires*, IX:161–191, February 1883.

[122] Neil Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48:203–209, November 1987.

[123] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *Proceedings of Advances in Cryptology (CRYPTO'99)*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397, Santa Barbara, California, U.S.A., 1999. Springer.

[124] Paul C. Kocher. On Certificate Revocation and Validation. In *Proceedings of the Second International Conference on Financial Cryptography*, pages 172–177, Anguilla, British West Indies, 1998. Springer-Verlag.

[125] Akihisa Kurashima, Akira Uematsu, Kenichi Ishii, Masato Yoshikawa, and Jun ichi Matsuda. Mobile Location Services Platform with Policy-Based Privacy Control. *NEC Research and Development Special Issue on Devices and Systems for Mobile Communications*, 44(4):368–373, October 2003.

[126] Butler W. Lampson. Protection. *ACM SIGOPS Operating Systems Review*, 8(1):18–24, 1974.

[127] Marc Langheinrich. A Privacy Awareness System for Ubiquitous Computing Environments. In *Proceedings of the the Fourth International Conference on Ubiquitous Computing (UbiComp'02)*, volume 2498 of *Lecture Notes in Computer Science*, pages 237–245, Goteborg, Sweden, 2002. Springer-Verlag.

[128] Scott Lederer, Jennifer Mankoff, and Anind K. Dey. Who Wants to Know What When? Privacy Preference Determinants in Ubiquitous Computing. In *Proceedings of the SIGCHI conference on Human factors in Computing Systems (CHI'03)*, pages 724–725, Ft. Lauderdale, Florida, U.S.A., 2003. ACM Press.

[129] Minsoo Lee, Jintaek Kim, Sehyun Park, Jaeil Lee, and Seoklae Lee. A Secure Web Services Architecture for Location Based Services in Wireless Networks. In *Proceedings of the Third International IFIP-TC6 Networking Conference (NETWORKING'04)*, volume 3042 of *Lecture Notes in Computer Science*, pages 332–344, Athens, Greece, May 2004. Springer-Verlag.

[130] Arjen Lenstra and Eric Verheul. Selecting Cryptographic Key Sizes. *Journal of Cryptology*, 14:255–293, 2001.

[131] Ulf Leonhardt. *Supporting Location-Awareness in Open Distributed Systems.* PhD thesis, Department of Computing, Imperial College, London, May 1998.

[132] Benoit Libert and Jean-Jacques Quisquater. Efficient Revocation And Threshold Pairing Based Cryptosystems. In *Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing*, pages 163–171, Boston, Massachusetts, 2003. ACM Press.

[133] Erkki Liikanen. Commission Recommendation on the Processing of Caller Location Information in Electronic Communication networks for the Purpose of Location-enhanced Emergency Call Services. Available on the Official Journal of the European Union: `http://europa.eu.int/eur-lex/pri/en/oj/dat/2003/l_189/l_18920030729en00490051.pdf`, July 2003.

[134] John Linn. An Examination of Asserted PKI Issues and Proposed Alternatives. Available on `http://middleware.internet2.edu/pki04/proceedings/issues_alternatives.pdf`, April 2004.

[135] Steve LLoyd. Understanding Certificate Validation Path. The PKI Forum Technical Group (White Paper), September 2002. Available on: `http://www.pkiforum.org/pdfs/Understanding_Path_construction-DS2.pdf`.

[136] Zygmunt Lozinski. Parlay/OSA - a New Way to Create Wireless Services. Available on: `http://www.parlay.org/docs/2003_06_01_Parlay_for_IEC_Wireless.pdf`, June 2003.

[137] Maarten Wegdam and Jeroen van Bemmel and Ko Lagerberg. User Location in Heterogeneous Mobile Networks. Available on the AlbatroSS Project Website: `http://www.ist-albatross.org/`, January 2004.

[138] Martijn Maas. Pairing Based Cryptography. Master's thesis, Department of Mathematics and Computing Science, Technische Universitit Eindhoven, January 2004.

[139] Ratul Kr. Majumdar, Krithi Ramamritham, and Ming Xiong. Exploiting User Mobility Patterns for Adaptive Location Management. *IEEE Distributed Systems*, December 2003. Available on: `http://dsonline.computer.org/0312/f/oz002.htm`.

[140] A. Malpani, R. Housley, Vigil Security, and T. Freeman. Simple Certificate Validation Protocol (SCVP), Internet Draft. Available on: `http://www.ietf.org/internet-drafts/draft-ietf-pkix-scvp-15.txt`, July 2004.

[141] Martin Dodge and Michael Batty and Rob Kitchin. No Longer Lost in the Crowd. Prospects of Continuous Geosurveillance. Association of American Geographers Conference, Philadelphia, 2004. Available on: `http://www.casa.ucl.ac.uk/martin/aag_geosurveillance.pdf`.

[142] John Melcher. E-112 Is "Best Effort" the Best Approach ? Business Briefing: Wireless Technologies. Available on: `http://www.trueposition.com/lrc/E-112-Approach.pdf`, 2004.

[143] A. Menezes, T. Okamoto, and S. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39:1639–1646, 1993.

[144] Alfred J. Menezes. *Elliptic Curve Public Key Cryptosystems*. Kluwer Academic Publishers, 1993.

[145] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[146] R. C. Merkle. A Certified Digital Signature. In *Proceedings of Advances in Cryptology (CRYPTO'89)*, volume 435 of *Lecture Notes in Computer Science*, pages 218–238. Springer-Verlag, 1990.

[147] Silvio Micali. Efficient Certificate Revocation. Technical Report MIT/LCS/TM-542b, Massachusetts Institute of Technology, U.S.A., Cambridge, MA, USA, 1996.

[148] Silvio Micali. NOVOMODO: Scalable Certificate Validation and Simplified PKI Management. Available on `http://www.cs.dartmouth.edu/~pki02/Micali/paper.pdf`, April 2002.

[149] Victor Miller. Use of elliptic curves in cryptography. In *Proceedings of Advances in Cryptology (CRYPTO'85)*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426, Santa Barbara, California, U.S.A., 1986. Springer-Verlag.

[150] MorganDoyle Limited. GPRS tutorial. Available on: `http://www.item.ntnu.no/fag/tm8100/Pensumstoff2004/GPRS_Tutorial.pdf`.

[151] J. Morris, D. Mulligan, J. Peterson, and J. Polk. Geopriv Requirements, RFC3693. Available on: `http://www.ietf.org/rfc/rfc3693.txt`, February 2004.

[152] mTrack Services Ltd. KidsOk. Available on: `http://www.kidsok.net/`.

[153] Birgit Muehlenhaus. The Diffusion of Retail Location-Based Services: Geography and the Privacy Issue. Master's thesis, Department of Geography, College of Earth and Mineral Sciences, Pennsylvania State University, U.S.A., 2004.

[154] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure: Online Certificate Status Protocol (OCSP), IETF RFC 2560. Available on `http://www.faqs.org/rfcs/rfc2560.html`, June 1999.

[155] Ginger Myles, Adrian Friday, and Nigel Davies. Preserving Privacy in Environments with Location-Based Applications. *IEEE Pervasive Computing*, 2(1):56–64, 2003.

[156] National Institute of Standards and Technology (NIST). Advanced Encryption Standard. Available on: `http://csrc.nist.gov/CryptoToolkit/aes/`.

[157] National Institute of Standards and Technology (NIST). FIPS 46-3 (DES Specifications) Withdrawal. *Federal Register*, 69:44509–44510, July 2004.

[158] Rafe Needleman, Michael V. Copeland, and Om Malik. The Next Big Thing...the Cell Phone. Online article available on: `http://gigaom.com/articles/Cell%20Phone.pdf`, January 2004.

[159] NTT DoCoMo. All About iMode Index. Available on: `http://www.nttdocomo.com/i/index.html`.

[160] Judith Olson, Jonathan Grudin, and Eric Horvitz. A Study of Preferences for Sharing and Privacy. In *Proceedings of the conference on Human Factors in Computing Systems (CHI'05)*, pages 1985–1988, Portland, Oregon, U.S.A., 2005. ACM Press.

[161] Open Mobile Alliance. Available on: `http://www.openmobilealliance.com`.

[162] Open Mobile Alliance. Location Privacy Checking Protocol 1.0, Draft Version 2004-06-23. Available on: `http://member.openmobilealliance.org/ftp/public_documents/loc/Permanent_documents/OMA-LOC_PCP_Spec-V1_0-20040623-D.zip`.

[163] Open Mobile Alliance. Mobile Location Protocol 3.2, Draft Version 2004-10-05. Available on: `http://member.openmobilealliance.org/ftp/public_documents/loc/Permanent_documents/OMA-LOC_MLP_Spec-V3_2-20041005-D.zip`.

[164] Ordnance Survey ireland. The Irish Grid Reference System. Available on: `http://www.osi.ie/pdf/irish_grid.pdf`.

[165] Ariel Pashtan, Andrea Heusser, and Peter Sheuermann. Personal Service Areas for Mobile Web Applications. *IEEE Internet Computing*, 8(6):34–39, December 2004.

[166] Cynthia A. Patterson, Richard R. Muntz, and Cherri M. Pancake. Challenges in Location-Aware Computing. *IEEE Pervasive Computing*, 2(2):80–89, April–June 2003.

[167] Slobodan Petrovic and Amparo Fster-Sabater. Cryptanalysis of the A5/2 algorithm. Cryptology ePrint Archive, Report 2000/052, 2000. Available on: `http://eprint.iacr.org/`.

[168] Andreas Pfitzmann and Marit Koehntopp. Anonymity, Unobservability, and Pseudonymity – A Proposal for Terminology. In *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *Lecture Notes in Computer Science*, pages 1–9, Berkeley, CA, USA, 2000. Springer.

[169] D. Pinkas and R. Housley. Delegated Path Validation and Delegated Path Discovery Protocol Requirements, RFC 3379. Available on: `http://www.faqs.org/rfcs/rfc3379.html`, September 2002.

[170] S. Pohlig and M. Hellman. An Improved Algorithm for Computing Logarithms over GF(p) and Its Cryptographic Significance. *IEEE Transactions on Information Theory*, 24:106–110, 1978.

[171] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket Location-Support System. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 32–43, Boston, Massachusetts, United States, 2000. ACM Press.

[172] Jeremy Quirke. Security in the GSM System. Available on: `http//www.ausmobile.com`, May 2004.

[173] Wolfgang Rankl and Wolfgang Effing. *Smart Card Handbook*. John Wiley and Sons, January 2004.

[174] Josyula R. Rao, Pankaj Rohatgi, Helmut Scherzer, and Stephane Tinguely. Partitioning Attacks: Or How to Rapidly Clone Some GSM Cards. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 31–41, Berkeley, California, U.S.A., 2002.

[175] Jussi Rautpalo. GPRS Security - Secure Remote Connections over GPRS. Available on: `http://www.hut.fi/~jrautpal/gprs/gprs_sec.html`.

[176] Redknee Inc. Location API. Available on: `http://www.redknee.com/site/index.php`.

[177] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications ACM*, 21(2):120–126, 1978.

[178] Ronald L. Rivest. Can We Eliminate Certificate Revocation Lists? In *Proceedings of the International Conference on Financial Cryptography*, volume 1464 of *Lecture Notes in Computer Science*, pages 178–183, Anguilla, British West Indies, February 1998. Springer-Verlag.

[179] Ronald L. Rivest and Butler Lampson. SDSI - A Simple Distributed Security Infrastructure. Available on: `http://theory.lcs.mit.edu/~rivest/sdsi10.ps`.

[180] 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, U.K., 2002. Available on: `http://www.equator.ac.uk/PublicationStore/2002-rodden-1.pdf`.

[181] Gregory Rose. A precis of the new attacks on GSM encryption. QUALCOMM Australia Available on: `http://www.qualcomm.com.au/PublicationsDocs/GSM_Attacks.pdf`, 2003.

[182] Jae-Cheol Ryou. The Current Status and Future Trends of Wireless PKI Worldwide. Available on: `http://www.japanpkiforum.jp/symposium/presentation/session_4/Ses4_Ryou.pdf`, September 2002.

[183] Markku-Juhani Olavi Saarinen. Attacks against the wap wtls protocol. In *Proceedings of the IFIP TC6/TC11 Joint Working Conference on Secure Information Networks (CMS'99)*, pages 209–215, Deventer, The Netherlands, The Netherlands, 1999. Kluwer, B.V.

[184] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-Based Access Control Models. *IEEE Computer*, 29(2):38–47, 1996.

[185] SANS Institute 2001 Information Security Reading Room. The GSM Standard (An overview of its security). Available on: `www.sans.org/rr/papers/58/317.pdf`, 2001.

[186] Naveen Sastry, Umesh Shankar, and David Wagner. Secure Verification of Location Claims. *RSA's Laboratories CryptoBytes (Technical Newsletter)*, 7(1):16–28, Spring 2004. Available on: `http://www.rsasecurity.com/rsalabs/cryptobytes/Spring_2004_Cryptobytes.pdf`.

[187] Bill Schilit, Jason Hong, and Marco Gruteser. Wireless Location Privacy Protection. *The IEEE Computer Magazine*, 36(12):135–137, December 2003.

[188] Bruce Schneier. Why Digital Signatures Are Not Signatures. Available on Crypto-Gram Newsletter's website: `http://www.schneier.com/crypto-gram-0011.html#1`, November 2000.

[189] Michael Scott and Paulo S. L. M. Barreto. Generating more MNT elliptic curves. Cryptology ePrint Archive, Report 2004/058, 2004. Available on: `http://eprint.iacr.org/`.

[190] Adi Shamir. How to Share a Secret. *Communications of the ACM*, 11:612–613, 1979.

[191] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of Advances in Cryptology (CRYPTO'84)*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53, Santa Barbara, California, United States, 1985. Springer-Verlag New York, Inc.

[192] Shamus Software Ltd. Multiprecision Integer and Rational Arithmetic C/C++ Library. Available on: `http://indigo.ie/~mscott/`.

[193] Judy Siegel-Itzkovich and Damian Carrington. GSM phone encryption can be cracked. *New Scientist*, September 2003.

[194] Nigel Smart. The discrete logarithm problem on elliptic curves of trace one. *Journal of Cryptology*, 12:193–196, 1999.

[195] Einar Snekkenes. Concepts for Personal Location Privacy Policies. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 48–57, Tampa, Florida, U.S.A., 2001. ACM Press.

[196] W. Richard Stevens. *TCP/IP Illustrated Volume 1: The Protocols.* Addison-Wesley Professional Computing Series, 1994.

[197] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):571–588, 2002.

[198] Hidema Tanaka, Chikashi Ishii, and Toshinobu Kaneko. On the Strength of KASUMI without FL Functions against Higher Order Differential Attack. In *Proceedings of the Third International Conference Information Security and Cryptology (ICISC2000)*, volume 2015 of *Lecture Notes in Computer Science*, pages 14–21, Seoul, Korea, 2001. Springer.

[199] The 3rd Generation Partnership Project. Available on: `http://www.3gpp.org`.

[200] The European Parliament. EU Directive 95/46/EC The Data Protection Directive. Available on: `http://europa.eu.int/comm/internal_market/privacy/docs/95-46-ce/dir1995-46_part1_en.pdf`, October 1995.

[201] The European Parliament. EU Directive 2002/58/EC The Data Protection Directive. Available on: `http://europa.eu.int/eur-lex/pri/en/oj/dat/2002/l_201/l_20120020731en00370047.pdf`, July 2002.

[202] The Parlay Group. Available on: `http://www.parlay.org/`.

[203] The Parlay X Working Group. Parlay X Web APIs - White Paper, December 2002.

[204] The U.S. Congress. Location privacy protection act. Available on: `http://www.techlawjournal.com/cong107/privacy/location/s1164is.asp`, 2001.

[205] The U.S. Congress. Wireless location privacy protection act. Available on: `http://www.theorator.com/bills108/hr71.html`, 2003.

[206] The WAP Forum. Wireless Application Protocol Identity Module Specification. Available on: `http://www1.wapforum.org/tech/documents/WAP-198-WIM-20000218-a.pdf`.

[207] The WAP Forum. Wireless Application Protocol WMLScript Crypto Library Specification. Available on: `http://www1.wapforum.org/tech/documents/WAP-161-WMLScriptCrypto-19991105-a.pdf`.

[208] The WAP Forum. Wireless Public Key Infrastructure Specifications. Available on: http://www1.wapforum.org/tech/terms.asp?doc=OMA-WAP-217_105-WPKI-SIN-20020816-a.pdf.

[209] The White House, Office of the Press Secretary. Statement by the President regarding the United States' decision to stop degrading GPS sytem accuracy. Available on: http://www.ngs.noaa.gov/FGCS/info/sans_SA/docs/statement.html, 2000.

[210] The World Geodetic System. Available on: http://www.wgs84.com/.

[211] Eddie Turkaly. Securing Certificate Revocation List Infrastructures. Available on the SANS Institute 2001 Information Security Reading Room website: http://www.sans.org/rr/whitepapers/vpns/748.php, 2001.

[212] Gabriel Vanrenen and Sean Smith. Distributing Security-Mediated PKI. In *Proceedings of the EuroPKI Conference*, volume 3093 of *Lecture Notes in Computer Science*, pages 218–231, June 2004.

[213] W3C Web Services Architecture Working Group. Web Services Architecture Requirements, W3C Working Draft 2002. Available on: http://www.w3.org/TR/2002/WD-wsa-reqs-20020819.

[214] David Wagner. GSM Cloning. Available on: http://www.isaac.cs.berkeley.edu/isaac/gsm.html, 1998.

[215] Yulian Wang. SPKI - Introduction. Helsinki University of Technology, December 1998. Available on: http://www.hut.fi/~yuwang/publications/SPKI/SPKI.html.

[216] Jan Willemson. Certificate Revocation Paradigms. Available on the Cybernetica, Estonia website: http://www.cyber.ee/dokumendid/additional/certificate.pdf, 1999.

[217] Petra Wohlmacher. Digital Certificates: A Survey Of Revocation Methods. In *Proceedings of the ACM Workshop on Multimedia*, pages 111–114, Los Angeles, California, United States, 2000. ACM Press.

[218] Chris Wullems, Mark Looi, and Andrew Clark. Enhancing the Security of Internet Applications using location: A New Model for Tamper-resistant GSM Location. In *Proceedings of the 8th IEEE International Symposium on Computers and Communications*, pages 1251–1258, Kemer-Antalya, Turkey, June 2003. IEEE Computer Society.

[219] W. Yeong, T. Howes, and S. Kille. The Directory — Overview Of Concepts, Models And Services. CCITT X.500 Series Recommendations. Available on: `http://www.itu.int/itudoc/itu-t/aap/sg17aap/recaap/x500/x500.html`, December 1988.

[220] Chan Yeob Yeun and Tim Farnham. Secure M-Commerce with WPKI. Available on: `http://www.iris.re.kr/iwap01/program/download/g07_paper.pdf`, October 2001.

[221] G. Zhang and M. Parashar. Context-aware Dynamic Access Control for Pervasive Applications. In *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS'04)*, pages 219–225, San Diego, California, U.S.A., January 2004.

[222] Peifang Zheng. Tradeoffs in Certificate Revocation Schemes. *SIGCOMM Computer Communication Review*, 33(2):103–112, 2003.

[223] Martijn Zuidweg. A P3P-based Privacy Architecture for a Context-Aware Services Platform. Master's thesis, Faculty of Electrical Engineering, Mathematics and Computer Science, Department of Computer Science, University of Twente, August 2003.

[224] Martijn Zuidweg, Jos Gonalves Pereira Filho, and Marten van Sinderen. Using P3P in a Web Services-based Context-Aware Application Platform. In *Proceedings of the 9th EUNICE Open European Summer School and IFIP Workshop on Next Generation Networks, EUNICE 2003*, pages 238–243, Budapest,Hungary, September 2003.

Appendix A

The Orient Protocol Stack

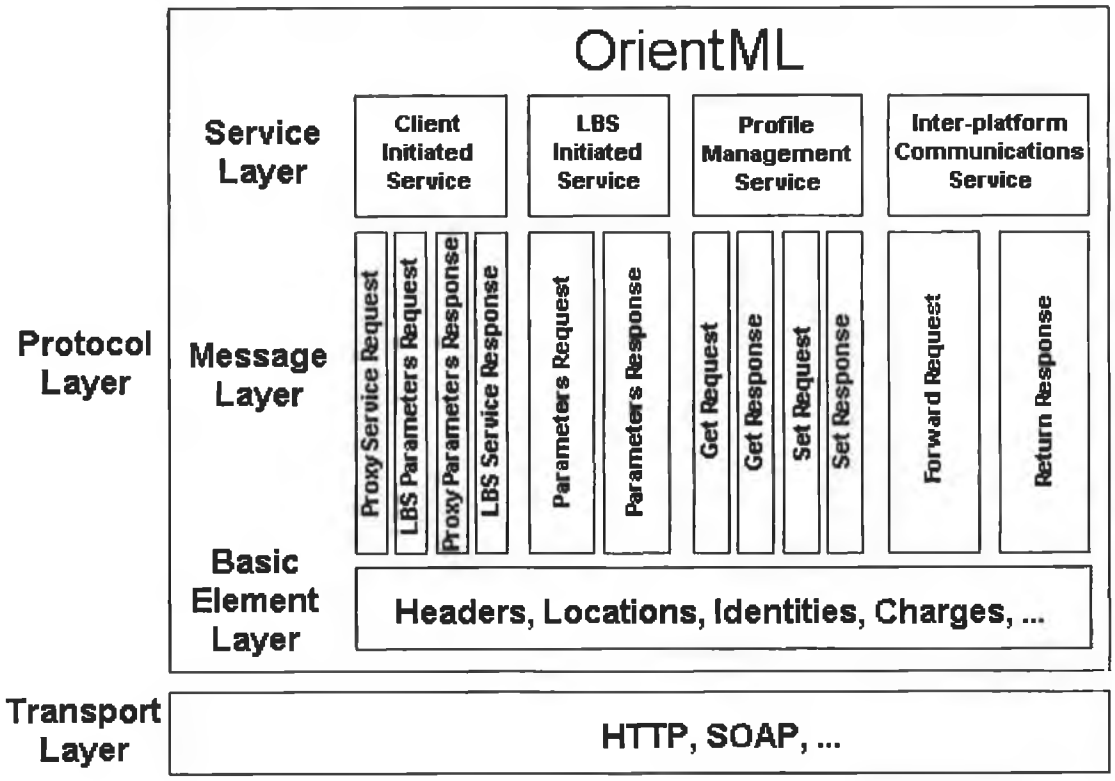


Figure A.1: The *Orient Protocol* Stack.

Appendix B

The Privacy Engine User Interfaces

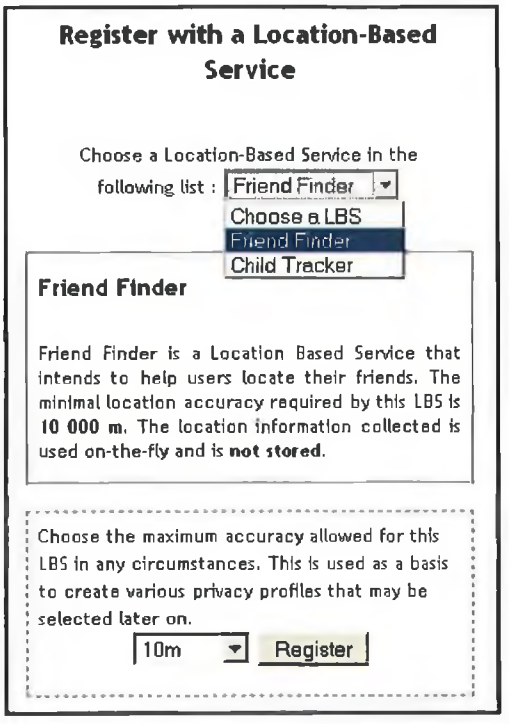


Figure B.1: Interface for registering privacy preferences for a particular *LBS*.

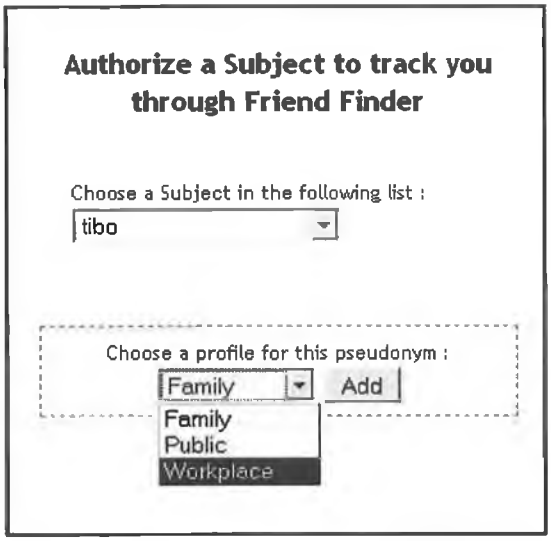


Figure B.2: Interface for choosing the *Subject* and assigning her to a cluster.

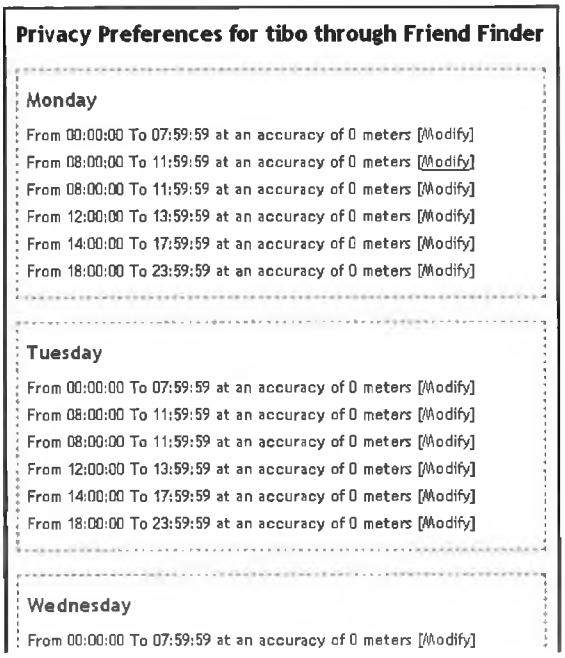


Figure B.3: Interface for visualizing and modifying privacy preferences for a particular *Subject* and *LBS*.

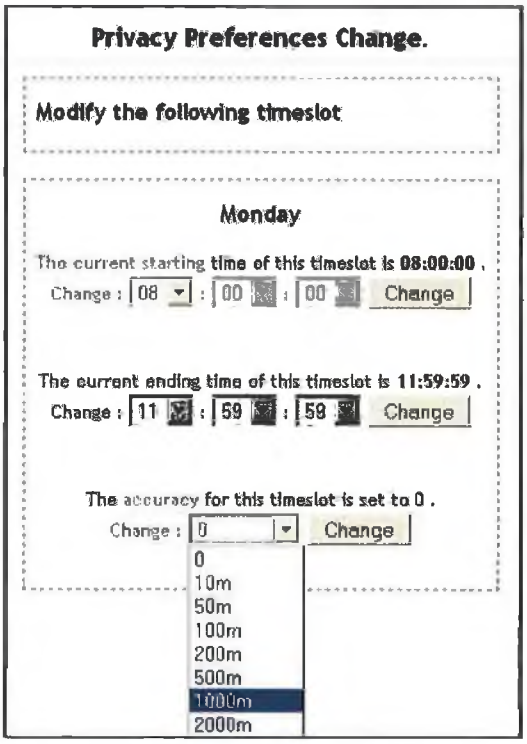


Figure B.4: Interface for modifying a specific timeslot.

Appendix C

Cryptographic API

C.1 User API

void setUserPartPrivateDecryptionKey(String x, String y)

This function is used to set up the User's private decryption key share.

void setUserPartPrivateSignatureKey(String x, String y)

This function is used to set up the User's private signature key share.

void setUserPublicParametersKey(String x, String y)

This function is used to set up the User's public signature key.

String[] encrypt(byte[] plain, String Id)

This function is used to encrypt a message with the identity "Id" of a particular User.

byte[] decrypt(String[] cipher, String[] ksem)

This function is used to decrypt a message using a parameter "ksem" received from the SEM.

String[] sign(byte[] message)

This function is used to sign a message.

boolean verify(byte[] message, String[] signature, String Id)

This function is used to verify the signature of a message with the identity "Id" of a particular user received from the SEM.

C.2 Private Key Generator API

void setUpParameters(String p, String q, String x, String Px, String Py)

This function is used to set up the elliptic curve related parameters p and q, the PKG private key x as well as the point P.

void keyGen(String Id)

This function is used to generate the set of keys for a particular identity "Id".

void setUp()

This function is used to generate default sample parameters including the ones generated by setUpParameters() and keygen().

String[] getPointP()

This function is a getter for the point P.

String[] getPointYkpg()

This function is a getter for the public key of the PKG.

String[] getPointDId()

This function is a getter for the decrypting private key of a User.

String[] getPointDIdSem()

This function is a getter for the decrypting private key share of the SEM for the User considered.

String[] getPointDIdUser()

This function is a getter for the decrypting private key share of the User considered.

String getPointSId()()

This function is a getter for the signing private key of a User.

String getPointSIdUser()()

This function is a getter for the signing private key share of the User considered.

String[] getPointSIdSem()

This function is a getter for the signing private key share of the SEM for the User considered.

String getX()

This function is a getter for the private key of the PKG.

String getP()

This function is a getter for the elliptic curve related parameter p.

String getQ()

This function is a getter for the elliptic curve related parameter q.

String getL()

This function is a getter for the elliptic curve related parameter l.

C.3 Security Mediator (SEM) API

void setSemPartPrivateDecryptionKey(String x, String y)

This function is used to set the SEM's private decryption key share.

void setSemPartPrivateSignatureKey(String x, String y)

This function is used to set the SEM's private signature key share.

String[] decryptSem(String[] cipher, String Id)

This function is used by the SEM to partially decrypt a ciphertext using its corresponding private key share.

String[] signSem(String[] userSig)

This function is used by the SEM to help a User sign a ciphertext using its corresponding private key share.

C.4 Common API

void loadUpParameters(String p, String q, String Px, String Py, String Ypkgx, String Ypkgy)

This function is used by most parties in order to load up their default parameters.