PROVIDING TRUSTWORTHINESS TO THE OPERATION OF LOCATION BASED SERVICES IN MOBILE NETWORKS

BY SHU CHEN

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Wade Trappe

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2010

ABSTRACT OF THE DISSERTATION

Providing Trustworthiness to the Operation of Location Based Services in Mobile Networks

by Shu Chen

Dissertation Director: Wade Trappe

The development of low-cost, ubiquitous, wireless systems is leading to a future where location will define the next generation of computing applications. Location Based services (LBS) use location information as the basis for providing enhanced services to a mobile user. The development of LBS is still in a relatively young stage and many questions and challenges must be addressed before a robust, secure and trustworthy LBS system can be successfully developed. This thesis takes the viewpoint that the eventual successful development of LBS can be accomplished, if the system is carefully designed and appropriate techniques are integrated. In particular, this thesis maps out three steps needed to achieve the security and trustworthiness requirement: first, identify the security policies that regulate the LBS application; second, use appropriate mechanisms to enforce the security policies; and third, put the whole system into a secure framework to prevent manipulation by unscrupulous entities participating in the service.

In this thesis, we first capture the essential features of policies that regulate an LBS and formulate a security policy model. Then in order to enforce the security policies it is essential to get the accurate location information of the users. We propose to use the environmental data from embedded sensor networks to support localization and

position verification, and prove its viability through experiments. Further, we present a key distribution based location verification scheme. Particularly in this part, we investigate the relationships between the transmitter deployment density, the number of keys received and the location verification accuracy. Next, we take the viewpoint that the risk of location spoofing can be bypassed when an LBS infrastructure does not rely on a localization procedure to enforce the security policies. We show how this is made possible in spatio-temporal access control applications when we use the key distribution method.

Finally, we study a practical Mobile Location Based Service (MLS) and present an ultimate MLS framework and communication protocols that represent a culmination of the mechanisms we have examined in this thesis and support the primary theme of this thesis, which is that it is possible to design a secure location based service using an appropriate combination of tools.

Acknowledgements

This dissertation is an accumulation of much work, guidance, support, and the friendship of many people.

First and foremost, I would like to express my utmost gratitude to my advisor Prof. Wade Trappe, for all his guidance, support and patience throughout the course of my studies. He has been a constant source of inspirations, wisdom and encouragement. Besides research, I would also like to thank him for helping me improve my skills in presentations, writing and time management, and sharing life experience. I am also greatly indebted to my co-advisor Prof. Yingying Chen. Working with her was a pleasant experience and I was always inspired by her passion on research. In particular, I would like to express my gratitude to her for the many in depth discussions, sharing hands on experiences without reservations, and always willing to help. She has not only been a good advisor, but also a good friend. I have been very fortunate to have the opportunity to work with both of them.

I would also like to express my gratitude to Prof. Richard Martin and Prof. Michael Littman for serving on my dissertation committee and have given me valuable comments that have helped improve the quality of this thesis.

Further, I would like to thank my colleagues in Computer Science Department and in WINLAB at Rutgers University. In particular I must thank Brian Thompson for the valuable discussions and ideas on my research, for helping me with my experiments and for correcting my writings. I would also like to acknowledge the members of Prof. Trappes research group. They are Wenyuan Xu, Qing Li, Yu Zhang, Zang Li, Pandurang Kamat, Liang Xiao and Tingting Sun. I must thank for the various help and friendship you have given me which makes it a much joyful journey to pursue my Ph.D. degree.

Outside of Rutgers, I would like to extend my thanks to Sanjay Macwan and Cristina Serban at AT&T Security Research and Development center. The summer that I spent with them helped expand my horizons as a researcher.

Finally, to my parents, my brother, my sister in law and my nephew, thanks for giving me the love, support, understanding, and bringing me the happiness that keep me going through the whole graduate studies.

Dedication

To my parents, Jijun Chen and Zhimin Chen

To my brother, Chen Chen

To my nephew, Yuze Chen

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

The recent advancements in wireless technologies has made it possible for the development of low-cost, ubiquitous wireless systems. Computing and communication is shifting from the static model of the wired Internet towards a new and exciting "anywhere-anytime" pervasive computing model. This is leading to a future where location will define the next generation of computing applications. In some applications, location information related to wireless devices or sensor nodes is indispensable such as in geographical routing [1–5] where location information is used to select the next forwarding node, or in environmental monitoring [6–9] where knowing where the monitored data originated from is essential for event detection or habitat monitoring. In these applications, the location information is part of the system. Location information can also be used to *add value to services that are provided to the end users.* These applications are referred to as **Location Based Services (LBS)**. When location information is combined with content, a broad range of location based services are made possible, for example providing quick and efficient assistance to mobile users who are in emergency, asset or personnel tracking, navigation, providing location-specific information services for tourists, controlling access to confidential information based on user's location, advertising merchandise to mobile users that are in vicinity, etc.

Since LBS represents a new paradigm for computing and communication services, their development is still in a relatively young stage. Consequently, many questions remain to be addressed before robust, secure and trustworthy LBS systems can be successfully deployed. In particular, several hurdles have to be conquered in order to provide trustworthiness to the operation of an LBS, which include: assurance that a

mobile entity's location information is a trustworthy and accurate representation of the entity's true location, the ability to provide the correct content or services to the requester, assurance that the content or services can only be accessed by valid users at the right time, and finally that LBS itself can be made secure so as to prevent manipulation by unscrupulous entities participating in the service. This thesis takes the viewpoint that it is possible to integrate appropriate techniques into location based services to ensure that they operate in a trustworthy and secure manner.

This thesis will first provide an overview of LBS and analyze the challenges facing the implementation of a trustworthy LBS system. Then we will focus on the policy representations that regulate LBS, proposing our solutions for trustworthy location verification, service distribution and communication between entities in LBS system.

1.2 Overview of Location Based Services

1.2.1 Concept of LBS

Location Based services (LBS) are services that use geographic or location information as the basis for providing enhanced service to a mobile user. The objective behind LBS is to add value to a service and should not be confused with LoCation Services (LCS) [10] which seek to find the location of an entity and to make the resulting location data available for external parties to use. LBS also has similarities with Geographic Information Systems (GIS) [11], but are different in the sense that GIS is mainly for professional applications that capture, manage, analyze and interpret the geographically referenced information [12], such as earth surface based scientific investigations, environmental impact study, or urban planning. GIS users are professional, experienced users, for example city and transportation systems planning offices. Usually GIS requires extensive computing resources, whereas LBS are emerging applications that will be used by the general public. LBS applications can be deployed on portable mobile devices with limited computing resources.

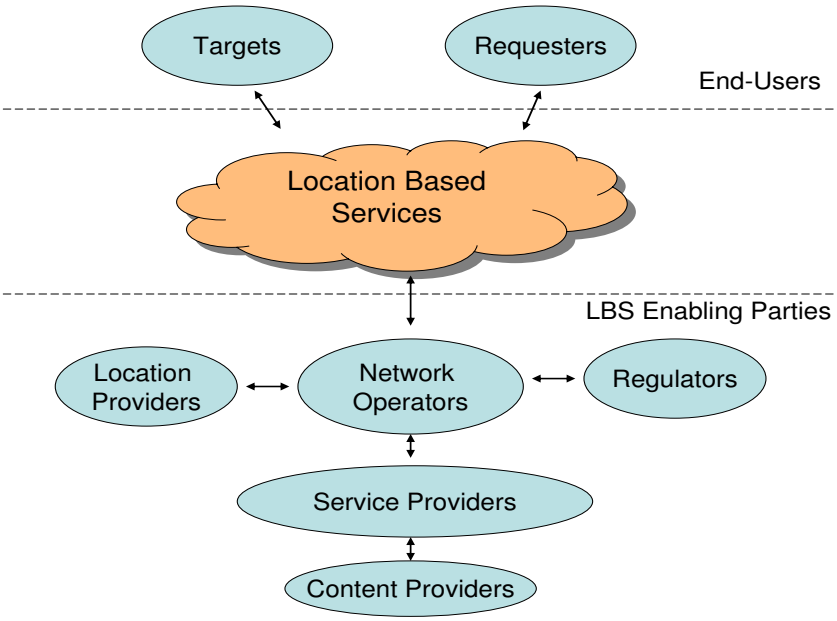


Figure 1.1. The components of an LBS system

1.2.2 LBS Participants

Several components are required in a typical LBS system: Targets, Requesters, Network Operators, Service Providers, Location Providers, Regulators. Figure 1.1 shows the relationships between these components.

The **end-users** in an LBS system can be categorized into two types: the targets and the requesters. The **Targets** are the users whose locations are requested by someone or some device. More precisely, it can be thought of as the wireless devices on the targets that are to be located. The **Requesters** are the ones who ask for services or information provided by the LBS. An end-user can be both the target and the requester in an LBS system.

On the other side, the operation of Location Based Services relies on some functional entities. These **Location Based Services Enabling Parties** include network operators, location providers, regulators and service providers. **Network Operators** are the wireless carriers that are responsible for data transfer, such a transfer maybe the

location data or information that is to be sent to the requester. **Location Providers** are the parties that make it possible to locate the mobile devices. They provide the hardware and software required for localization. **Service Providers** are the companies that develop the LBS applications. They create the software which implement the service logic and user interfaces and interfaces between the parties in the LBS system. The services are offered through network operators. Service providers do not necessarily own the content or information needed for the services [13]. In some cases there are other parties, **Content Providers**, who provide the data. For example, NAVTEQ provides digital maps, while Yelp is used to provide reviews for the point of interests, weather channel provides weather reports, etc. The **Regulators** set up rules, policies, and laws that an LBS system needs to follow. The regulations may address issues such as the privacy of the users, the access authorization, etc. All the Location Providers, Service Providers and Regulators connect with the Network Operators and their functionality are supported through the Network Operators. We note that it is possible that one entity plays multiple roles in the infrastructure.

1.2.3 Types of LBS

Generally, LBS can be categorized as two types of services: Pull services and Push services.

In **"Pull" services**, the user initializes a service request, and the customized services or information is delivered to the user based on its location and requesting time. In pull services, it is very important to get the correct location of the user to ensure service quality. For example, a user in an emergency requests an ambulance by pressing a button on its device. The service provider has to know the user's location in order to dispatch an ambulance that can reach the user correctly and at the earliest time. The user also has to give the permission for the service to get its location.

In **"push" services**, the service or information is delivered without the instant request from the user. It can be triggered by a timer such as in the case that an user subscribed to a weather forecast service that delivers the local weather to the user every morning, or it can be triggered by some event such as when the dog being tracked runs

out of scope, the owner will receive a notifying message. It can also be some service triggered when a user enters an area, for example advertisement messages can be sent to customers when they enter a store. In push services, it is critical to decide when and what to send to the users, in other words, to ensure the services or information is delivered to the users at the the right location/area at the right time. The user must give permission to the service to send information to its device. Usually the user's location is required so that the service provider knows where the information or service is pushed to. In some implementations, however, the localization procedure can be bypassed. We will present an example of such an architecture in Chapter 5.

It should be noted that an LBS application can be implemented both as a pull service and as a push service. For example, an asset tracking service let the user track where an item in transit is. This service can be provided as either a pull service or a push service or both. In the pull version, when the user sends a request, the current location of the target is sent to the users devices. In the push version, whenever the target asset is handed over at intermediate transition points or destination, a message is sent to the user automatically.

1.2.4 The Current Status of LBS

In 1990, both Mitsubishi Electric and Pioneer announced the first GPS-based auto navigation system, which was the first obvious application of LBS. Since then navigation has thrived as an application and reached the mass market. Currently, a broad range of LBS applications have been deployed and more are being proposed. Figure 1.2 gives a summary of the existing LBS applications, grouped by application areas. It is easy to envision a future with a much more expended list.

We talk about some of the popular applications in more detail here:

- **Emergency Services** involve the ability to locate a mobile user who is in emergency situation. In emergency situations, immediate response to the assistant request is usually acquired but the requester may not know his location or it is unable to be communicated. Hence, being able to automatically locate the

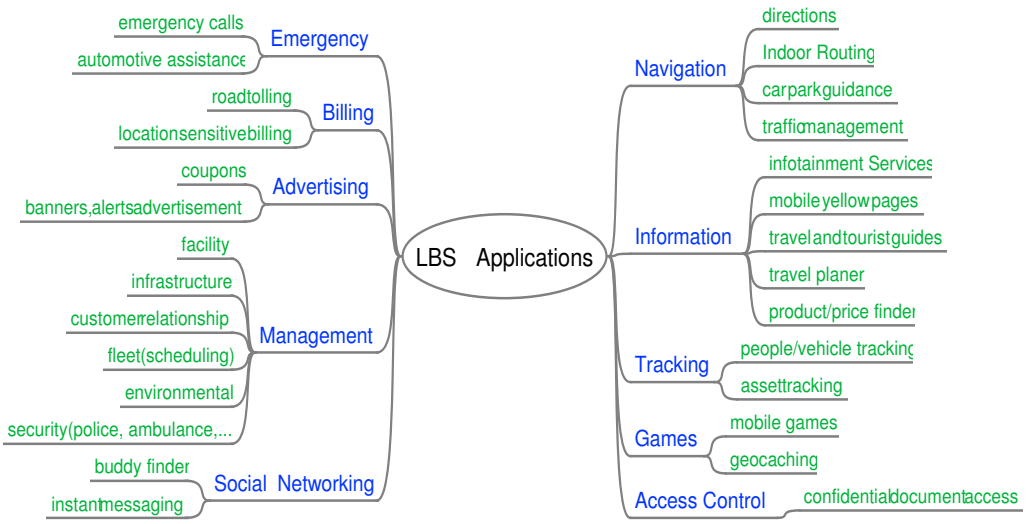


Figure 1.2. LBS applications.

requester is critical in emergency services. The most well-known emergency assistant system is the Enhanced-911 (E911) system. E911 works in North America when the emergency telephone number 911 is called. It automatically associates a physical address with the calling party's telephone number, and routes the call to the most appropriate Public Safety Answering Point (PSAP) for that address [14]. The Federal Communications Commission (FCC) has adopted rules that apply to wireless network operators aimed at improving the effectiveness and reliability of wireless E911 services. The FCC's accuracy standard requires handset based position determination equipment (PDE) to be able to locate the caller within 50 meters 67% of the time, within 150 meters 95% of the time and for network based PDE, it is 100 meters 67% of the time and 300 meters 95% of the time [15].

- **Information Services** provide the user with information customized by the requested location, time specification and user's preference. Already quite a few information services have been implemented and are available on the market. One popular example is the set of applications that allow mobile users to search, using their mobile device, for locations of interest, such as a restaurant, a store, obtain

reviews for these locations, and navigate the users to that location. People can also get traffic information and weather forecasts on the go. Some other information services are designed for particular venues, such as a museum or a national park, to provide detailed digital maps and information about the exhibitions or the habitat of different plants or animals in the current area of the tourists. Information Services are usually pull services. The accuracy of the location information of the requester is essential to the quality of services.

- **Tracking Services** can be applied to assets, pets, family members for personal use, or it can be used by companies to track vehicles such as ambulances, taxies, or repair engineers, so that dispatching can be done efficiently when there are requests from the customers. The Federal Emergency Management Agency (FEMA), a division of the Department of Homeland Security (DHS) has reached out to Sprint to provide 2,000 Nextel handsets in support of FEMAs nationwide vehicle tracking system. The ability to track anywhere and at any time is crucial for FEMA, especially when the agency is called upon to provide disaster support. These handsets are equipped with Nextel Direct Connect and TeleNav Track, a cell phone-based GPS navigation and tracking service and offer FEMA ease of deployment and cost reductions for their emergency communications. AT&T also announced an asset tracking application for businesses and government agencies to monitor and report the location of their portable assets in field operations with GPS precision. Delivered as software-as-a service (SaaS), TeleNav Asset Tracker from AT&T is a portable, battery-powered GPS tracking device designed to help an organization reduce theft and vandalism of company property such as machinery, equipment, containers and vehicles and improve efficiency in the daily use and dispatch of assets. As another example, dog tracking systems allow the owner to quickly locate their dogs whether they are hunting, or even at home. Typical products include the GPS tracking system from Garmin or radio telemetry tracking collars systems from Marshall Radio Telemetry.

- **Advertisement**. In April 2010, NAVTEQ announced the results of its first

location targeted campaign in Europe by a global brand advertiser, McDonalds. The campaign delivered mobile ads to users of Nokia Ovi Maps when they were within five miles of any of McDonalds 82 restaurants in Finland. The campaign, which promoted a McDonalds cheese burger for1 euro resulted in a 7 percent click-thru rate. Of users who clicked through, 39 percent selected the click-to-navigate option to navigate to the nearest McDonalds location [16]. This is a successful example of location-based advertising where promotion information is sent to mobile users that is in the vicinity of the static vendor. In another business model, which is called "Mobile Location Based Services" (MLS), the vendors may also be moving, e.g. a mobile ice cream truck or a taxicab, and the mobile vendors aim to broadcast their service information to the mobile customers within their service radius. It is more complicated as the service area changes constantly with the mobile vendors. We discuss the challenges and threats forced by this business model and propose solutions for a secure architecture in Chapter 6. Location based advertisement is usually a push service, so deciding which area to deliver the service and enforcing the delivery of the service is important.

- **Location Based Access Control**. The fact that wireless networks and localization techniques are becoming increasingly ubiquitous make it possible for new ways to regulate the access to data or services only when the requester is in the right location, rather than conventionally based solely on user's identity. For example, it may be desirable to ensure a laptop is no longer able to access confidential corporate document when it is taken outside of the company's building. It is rational to extend location based access control to Spatio-Temporal Access Control (STAC) where whether a user can access the services or data is based on his location and accessing time. In this kind of application, it is important to specify the access control policies and to have mechanisms to enforce the policies.

1.3 Security and Trustworthiness in LBS

LBS has a great potential to improve our life in many ways such as providing expedited emergency assistance, convenient information sharing and accessing, efficient communication, and better resource management and access control. However, before an LBS system can be launched to the mass market successfully, the LBS system need to be well designed to ensure that it operates in a secure and trustworthy manner. In this section we first analyze the security and trustworthiness challenges facing an LBS system, and then we define our goal and map out our approaches to achieve the trustworthiness in LBS.

1.3.1 Security and Trustworthiness Challenges

There are many reasons why a location based system may operate in an improper manner. For example, such improper operation could result from hostile outside parties attacking the system, or from an insider entity cheating the system by falsely reporting its position so as to obtain services that it should otherwise not be able to obtain, or it can be that the positioning techniques used by the system are not robust and accurate enough to provide precise and timely location information. It is thus essential to be aware of the security and trustworthiness threats that an LBS may face. We summarize these challenges according to three main categories, as described in the next few paragraphs.

The first challenge is to ensure that location information that are needed in the operation of LBS are reliable and accurate. On one hand, localization techniques do not always meet the accuracy requirement. Although GPS usually works well outdoors with errors less than 5 meters, its accuracy can be affected by a number of factors such as weather conditions, natural barriers to the signal (like mountains or tall buildings), and noise in the radio signal. GPS does not work well in an indoor environment, and further there are situations when GPS is not available. In these situations, additional localization infrastructure is required to localize mobile devices or to assist the existing localization system to get better accuracy. On the other hand, attackers may subvert

the operation of the system by corrupting the location information using methods that exploit the localization infrastructure itself [17]. With the existence of such risks, LBS systems should provide mechanisms to verify that the location claims are the accurate representation of the entities' true locations.

The second is the threats from conventional attacks that targeting at destroying the service during message transmission stage. The adversaries may overhear the communications between entities in LBS systems, interfere or inject malicious messages, or jam the traffic. Such attacks are general problems existing in wireless networks.

The third class of attacks are denial of service (DoS) attacks. Hostile entities may subvert the system by sending out tremendous amount of service requests that can not be handled by the LBS system. This threat can be mitigated by using entity authorization and message encryption.

1.3.2 Providing the Trustworthiness in LBS

The ultimate goal of providing trustworthiness to the operation of an LBS system is to ensure that the *right services* are delivered to the *right users* at the *right time*. Currently, the research into LBS security has primarily focused on the issue of providing secure and robust position information. For example, in [18], the authors listed a few attacks that might affect the correctness of localization algorithms along with a few countermeasures. [19] uses hidden and mobile base stations to localize and verify location estimate. Since such base station locations are hard for attackers to infer, it is hard to launch an attack, thereby providing extra security. [20] uses both directional antenna and distance bounding to achieve security. [21] makes use of the data redundancy and robust statistical methods to achieve reliable localization in the presence of attacks.

Although secure localization is often essential to the well operation of LBS system, it is not sufficient and sometimes not necessary. This thesis targets of providing a holistic protection to the operation of LBS system and map out three steps to achieve this goal.

First, an LBS system should clearly specify the *Security Policies*, which define what services should be delivered to which locations at what time, or what services are allowed to be accessed at which locations at what time and by who. Some applications, such as

emergency services, are intended to be accessed as broadly as possible, whereas some other applications will have more restriction on the service accessibility. For example, in an advertisement application, coupons are intended to be sent to people who are in the vicinity of a store; or, in location based access control applications, the confidential document should be accessed only at the allowed areas at the right time. In Chapter 2, we formalize an LBS security policy model and provide different means to represent the security policies. If policies are poorly defined or enforced, the system or the users may be undermined in different ways, such as the service is not delivered on time (which is especially bad in an emergency service), wrong information is sent to users, or an unqualified user outside of the valid area get confidential information, etc.

Second, after defining security policies, there should be mechanisms to *enforce the security policies*. One obvious methodology is to obtain the location of the user, then to decide what services should be provided to the user based on security policies. In this case, we require that the associated location is accurate and trustworthy. However, challenges exist, and as we mentioned in the first challenge, additional localization infrastructure is required to localize mobile devices or to assist the existing localization system to get better accuracy. In Chapter 3, we study how environmental properties can be used to assist localization system and show its effectiveness through experiments. Further in Chapter 4, we propose a key distribution-based scheme that utilizes an auxiliary network of transmitters to verify whether a location claim is trustworthy or not. Also in this thesis, we take the point of view that the risk of positioning spoofing can be bypassed when the LBS infrastructure does not rely on a localization procedure to enforce the security policies. We show in Chapter 5 that how this is made possible in spatio-temporal access control applications when we use the an auxiliary sensor network as a collection of transponders.

Finally, the whole system itself should be put into a *security framework* to prevent manipulation by unscrupulous entities participating in the service. The framework should take into consideration the conventional trustworthiness requirements such as entity authentication, message integrity and message confidentiality. Additionally, the system needs to integrate the LBS security policies and location accuracy requirement

mentioned above. Chapter 6 examines Mobile Location Based Services (MLS), which is a relatively complicated type of LBS where services are provided by mobile vendors to the clienteles who are within their service radius. We identify attacks and misuses faced by MLS and propose a security architecture and a holistic protocol that provides end-to-end trustworthy services.

1.4 Thesis Organization

In this thesis, we focus on providing technical solutions for the trustworthy operation of an LBS system and take the viewpoint that the eventual successful trustworthy operation of LBS can be accomplished. The structure of the thesis is organized as follows.

In Chapter 2, we capture the essential features of policies that regulate a location based service and present a basic security policy model for LBS. We then provide different ways to represent the security policies. In particular, we show how complex policies where historical information is involved can be specified using automata.

In order to enforce security policies, an obvious methodology is to get accurate location information for the user and then provide services based on the policy at that location. In this case, we require that the location information is accurate and trustworthy. In Chapter 3, we show how the data from embedded sensor networks that monitor environmental properties, such as temperature or ambient acoustic energy, can be used to support localization and position verification and propose a theory for exploiting this spatial variability for localization. Our Spatio-Correlation Weighting Mechanism (SCWM) uses spatial correlation across different phenomena to isolate an appropriate subset of environmental parameters for better location accuracy. We then develop an array of algorithms employing environmental parameters for performing localization. Our experiments provide strong evidence of the viability of using general sensor readings for location-aware applications.

Further, Chapter 4 proposes a generalized location verification model. In this model,

compared to the last method where we used the information extracted from the environment, we push information to the environment through an underlying architecture of low cost transmitters. In this model the transmitters broadcast verification keys to its vicinity. An entity is required to provide a sufficient number of valid keys in order to prove his location claim. We investigate the relationships between the transmitter deployment density, the number of keys received and the location verification accuracy based on statistical models and signal propagation models. These relationships provide a guideline for service providers deploying the transmitters so that they satisfy the location verification requirement.

We also take the point of view that the risk of position spoofing can be bypassed when the LBS infrastructure does not rely on a localization procedure to enforce the security policies. In Chapter 5, we show that how this is possible in spatio-temporal access control applications. In this scheme, the service or resource is protected through encryption. By controlling the distribution of decryption keys only to the valid service area we control the access to the service or resource.

Next, in Chapter 6, we study a practical Mobile Location based Service (MLS) system, and analyze the security threats that such a system might face. Our proposed framework and communication protocols incorporate cryptographic techniques, location verification methods and a reputation system, to achieve holistic security and trustworthiness in an MLS system. Finally, we conclude this thesis in Chapter 7 and present future research directions.

Chapter 2

Policies that Regulate Location Based Services

2.1 Introduction

Service control policies in LBS systems will, on one hand, serve as a guideline to regulate the services so that the trustworthiness of the LBS system is ensured, and on the other hand, help with designing the system infrastructure and deciding what techniques to use. The policies define the rules that control the access (in *pull service*) or distribution (in *push services*) of the services or information in LBS. Security policies are especially important for LBS applications that are more spatio-temporal sensitive. In this chapter, we capture the essential features of policies that regulate a location based service and present a basic security policy model for LBS. We then provide different ways to represent the security policies.

2.2 LBS Security Policy Components

A LBS security policy model involves four basic elements: *users*(USERS), *objects*(OBS), *operations*(OPS) and *spatio-temporal regions*(STRGNS). Figure 2.1 illustrates the core components of the LBS security policy model. A set of LBS *security policies* are defined above these components that specify what permissions/privileges may be granted/rejected based upon a user's access attempt. More specifically, the central idea of LBS service regulation is to define the *operations* of a *user*'s *operation* on an *object* based on the *spatio-temporal region* the user is in, according to these *security policies*. *Service Control mechanisms* are the collection of techniques used to enforce this process. We now specify each of these components in more detail.

User. A *user* is the end-user in LBS who seeks access to objects or is the intended

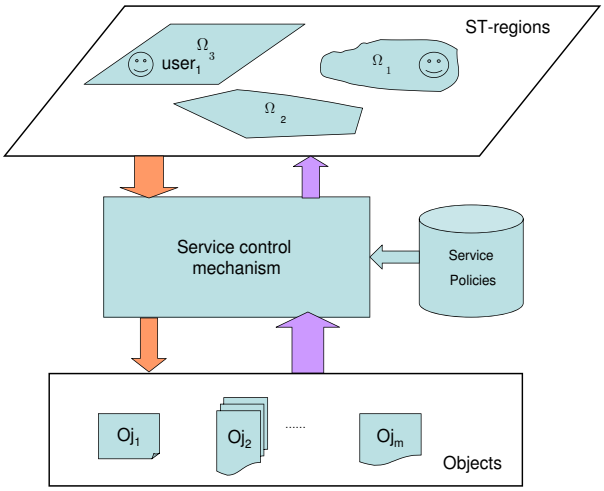


Figure 2.1. An conceptual picture of LBS security policy enforcement model

receiver for objects.

Object. An *object* can be a piece of information which is enclosed into an information container, such as a file, or an exhaustible system resource (e.g. wireless connections or CPU cycles). When an object is involved in the LBS system, it is endowed with temporal character, and can be generally categorized into static and streaming cases. A streaming object continually evolves with time, such as a movie being broadcast to an entire network of users, or live scores from a sporting event being transmitted to the audiences in a sports arena. On the other hand, a static object does not change with time (except for the possible exception of version revision, as often occurs for software objects). Since streaming objects are time-varying, their corresponding policies will inherently become more complicated to express.

As an example, consider an object O_j that is a streaming object. We may partition this object into subobjects according to time intervals, such as

$$O_j = \{O_j[t_0, t_1), O_j[t_1, t_2), O_j[t_2, t_3)\}$$

. Here O_j has been broken down into three pieces according to the time intervals $[t_0, t_1)$, $[t_1, t_2)$, and $[t_2, t_3)$ respectively. These subobjects may be further decomposed, and such a decomposition naturally raises the issue of the maximum amount that an object can be decomposed. We refer to the smallest constituent piece of a larger object as the

object atom. The size of an object atom is determined by the *temporal resolution* of a LBS system.

Operation. An *operation* is a function that a user can execute on an object. The types of operations depend on the types of applications and systems. For example, when considering access to a file, the operations could either be simply access/no-access, or they can be multilevel such as read, write or execute. For the simplicity of discussion, we shall restrict our attention to binary operations. We note, however, that multilevel cases can be converted to the binary cases by defining a set of new objects, one for each operation, that substitute for the original one. For example, given an object *f1* with 3 operations read, write or execute, we define 3 new objects *f1read, f1write, f1exec* that substitute for the original object *f1*, and each of these new objects becomes a binary case. Throughout this thesis, we shall represent access privileges for binary operations by a 1 corresponding to service approval, while 0 corresponds to service rejection.

Spatio-temporal region. A useful concept for specifying LBS objects in the policy model is the notion of a *spatio-temporal region*. A spatio-temporal region, denoted by Ω is defined as a set of 3-tuple, $\Omega = \{(x, y, t) : \text{valid areas in space and time}\}$, where (x, y) represents a spatial location and t represents an arbitrary time instance, and hence each (x, y, t) is a point in 3-dimensional spatio-temporal space. The spatio-temporal regions, which we shall refer to as ST-regions for brevity, are set up by the system according to the service policies, such as which object can be accessed where at what time period. Thus, it is often useful to visualize a ST-region as a continuous region in 3-dimensional space, instead of a set of discrete points. For an object *ob* and an operation *op*, a ST-region is called the *secure ST-region of (ob, op)* if the operation *op* is allowed to be performed on the object *ob* at this ST-region. In the case that the operation is binary, as we focus on in this thesis, we simply refer to such a region as the *secure ST-region of ob*. In some LBS applications, such as emergency services, the secure ST-region of the service is the whole 3-dimensional space, which indicates that the service is usable anywhere at anytime. Some services are only provided to a specific area, for example, in Figure 2.2, the ST-region Ω_0 is limited to the shaded area, and the spatial region that is constantly specified from time 0 to time t. This is can be

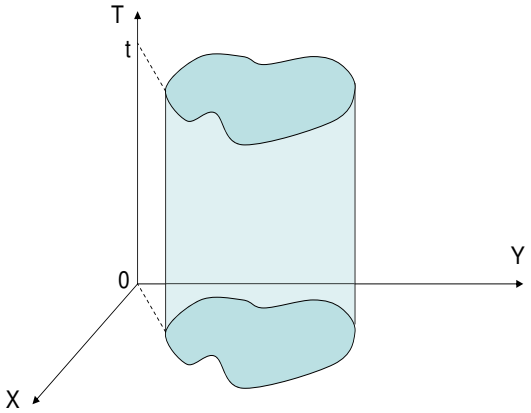


Figure 2.2. Example of a ST-region Ω_0 that is spatially constant.

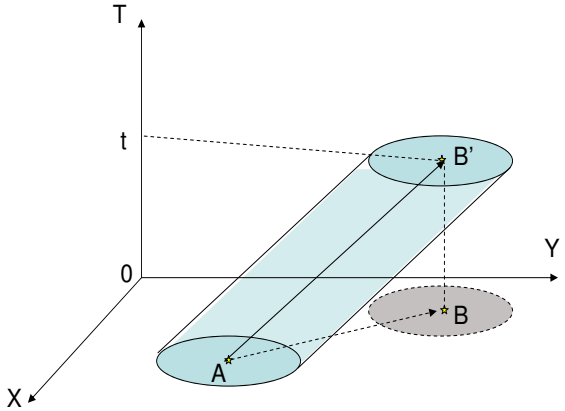


Figure 2.3. Example a spatio-temporal regions Ω_1 and Ω_2.

the case that information only provided to the specific area, such as the local weather forecast, information in a national park, etc. Figure 2.3 shows a more complicated case, where Ω_1 indicates a ST-region in a mobile location based service, where a vendor moves from location A to location B in the physical location, and the vendor would like to its service information to be received by customers within the specific radius around his location. So the spatial region varies with time in Ω_1. This shape of the ST-region, however, if associated with spatio-temporal access control service, would require that a user must move in a specific manner in order to maintain access privileges to an object.

Table 2.1. Using Access Control Matrix to Represent LBS Security Policies

		S1	**S2**	**F1**	**Mv**			**...**	Ojm	
		-	-	-	Mv_1	Mv_2	Mv_3	...	Ojm_1	Ojm_2
Ω_1	-	1	1	0	0	0	0	...	0	0
Ω_2	Ω_{21}	0	1	1	1	0	0	...	0	0
	Ω_{22}	0	1	1	0	1	0	...	0	0
	Ω_{23}	0	1	1	0	0	1	...	0	0
	Ω_{24}	0	1	1	0	0	0	...	0	0
...	...	...	...	...	...	...	...	...	...	...
Ω_n	Ω_{n1}	1	0	0	1	1	1	...	1	0
	Ω_{n2}	1	0	0	1	1	1	...	0	1

2.3 Service policies and their representations

2.3.1 Basic policies and their representations by an access control matrix

Security policies outline the rules and regulations for appropriate service access or distribution. A basic access policy is a 3-tuple $A = \{(\Omega; op; O_j;)\}$, where $O_j \in OBS$, $op \in OPS$, $\Omega \in STRGNS$, which is interpreted to mean that within the ST-region Ω, the operation op on object O_j is approved. Access control matrices can be used to represent such policies. In the access control matrix, the columns correspond to objects, the rows to ST-regions, and the cell where the column and row intersect specifies the operation privilege. Table 2.1 shows an example of an access control matrix for LBS, where the operations on objects are binary.

In this table, for example, object $S1$ can be accessed in the spatio-temporal location Ω_1, but not in Ω_2 (which has been further decomposed into smaller spatio-temporal regions). As a more involved example, we can decompose object Mv (which might correspond to a movie) down into sub objects (e.g. first 10 minutes, second 10 minutes, etc.). By similarly decomposing region Ω_2 into smaller regions Ω_{21}, Ω_{22}, Ω_{23} and Ω_{24}, we may now describe a more refined LBS service policy, where Mv_1 is accessible at location Ω_{21} but not Ω_{22}, and thus the user must move its physical location with an appropriate time period to Ω_{22} in order to access Mv_2, and hence resume access to Mv.

2.3.2 Complex security policies and their representations by FA

The example of object Mv described above gives insight that security policies for LBS can be powerful and flexible. In particular, a LBS system can perform complex access control by decomposing objects into *object atoms* and suitably associating smaller *region atoms* with these objects. By doing so, it is possible to grant or deny a user access to an object not only based on his current location, but also based on his previous spatio-temporal behavior. For example, we might require (for additional security), that a user have the ability to access object ob_1 at location l_1 and time t_1, and then be at location l_2 at time t_2 in order to access object ob_2. That is, without having been to (l_1, t_1) and having accessed ob_1, the user would not have the requisite access control information needed in order to access ob_2 at location (l_2, t_2).

Such a form of access control is stateful, and is unwieldy for representing with access control matrices. Notice in Table 2.1, for example, that although object Mv_1 can be accessed at Ω_{21}, Mv_2 can be accessed at Ω_{22}, and Mv_3 can be accessed at Ω_{23}, there is no information contained in the matrix that specifies that the user had to be at Ω_{21} before proceeding to Ω_{22} in order to access Mv_2.

In order to represent these more advanced forms of LBS access policies, we must employ a syntactical framework that facilitates the representation of state. We now show how automata theory can be employed to describe such complex polices. First, though, we provide a brief review of automata theory to facilitate our later discussion. A *finite automaton* is denoted by a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of *states*, Σ is a finite *input* alphabet, q_0 in Q is the *initial* state, $F \subseteq Q$ is the set of *final* states, and δ is the *transition function* mapping $Q \times \Sigma \rightarrow Q$ [22]. A string x is said to be *accepted* by a finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ if $\delta(q_0, x) = p$ for some $p \in F$. The *language accepted by M*, designated $L(M)$, is the set $\{x|\delta(q_0, x)\}$.

We may represent an access policy using an automaton $M = (Q, \Sigma, \delta, q_0, F)$ and capture the user's history for consideration in the access policy. To do so, we let the input alphabet $\Sigma = STRGNS \cup OBS$. A string x that is accepted by M is a sequence mixed by ST-regions that a user is required to locate and the objects that the user is

required to access. By properly designing the δ and the set of states Q, the desired access policy can be expressed.

In order to illustrate how, we now provide an example. Suppose that we have a movie that is divided into 3 parts—Mv_1, Mv_2 and Mv_3. Further, suppose that the access policy we want to describe is that in order for a user to be able to view a later part of the movie, he must have finished viewing the part(s) that came before that part. Further, suppose that we have the additional spatio-temporal requirement that Mv_1 can only be accessed at location l_1 at time t_1, Mv_2 can only be accessed at location l_2 at time t_2, and Mv_3 can only be accessed at location l_3 at time t_3.

A finite automaton M_1 for this policy is defined as follows. $M_1 = (Q, \Sigma, \delta, q_0, F)$, where $Q = \{q_0, q_1, q_2, q_3, p_1, p_2, p_3\}$; $\Sigma = \{\Omega_1, \Omega_2, \Omega_3, Mv_1, Mv_2, Mv_3\}$; $\Omega_1 = (l_1, t_1)$, $\Omega_2 = (l_2, t_2)$, $\Omega_3 = (l_3, t_3)$; $F = \{p_1, p_2, p_3\}$. Here, in our specification, there are two classes of states q_j and p_j. The states q_j correspond to a description of the user being at the spatio-temporal contextual state needed to access the movie object Mv_j, where as state p_j corresponds to a description that the user has been in state q_j, and then performed the required accessing of the movie object Mv_j (and consequently, can move to the next ST-region Ω_{j+1}). If we examine the transition diagram depicted in Figure 2.4, we can describe the transition mapping as follows:

$$\delta(q_0, \Omega_1) = q_1$$

$$\delta(q_1, Mv_1) = p_1$$

$$\delta(p_1, \Omega_2) = q_2$$

$$\delta(q_2, Mv_2) = p_2$$

$$\delta(q_2, \Omega_3) = q_3$$

$$\delta(q_3, Mv_3) = p_3.$$

We now walk through this transition mapping. The user starts in a null state q_0, and if it has moved to location Ω_1 at time t_1, it is described as being in state q_1, and hence has

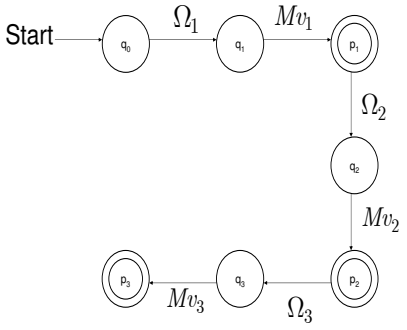


Figure 2.4. The transition diagram for M_1

the ability to access Mv_1. Once the user has watched the movie portion Mv_1, its state transitions to p_1, and the user may now move to location Ω_2 (by time t_2). The process continues, as the user moves to a new location, its state changes so it can access the content, then having accessed the content the user can now move to the next location. Formally, we may state the accepted language of this automaton M_1 as the collection of valid strings: $L(M_1) = \{\Omega_1 Mv_1, \Omega_1 Mv_1 \Omega_2 Mv_2, \Omega_1 Mv_1 \Omega_2 Mv_2 \Omega_3 Mv_3\}$.

Finite automata are fairly flexible and able to represent stateful access control policies rather easily. As another example, the finite automaton M_1 can be modified easily to support similar but slightly different policies. For example:

- If we add two transition functions to M_1. $\delta(q_0, \Omega_2) = q_2$, $\delta(q_0, \Omega_3) = q_3$, as shown in the transition diagram for the finite automaton M_1' in Figure 2.5, then the access policy takes on a different interpretation. Now, the access policy does not explicitly require that the user must have accessed the prior content Mv_{j-1} before accessing Mv_j. Instead, all that is required is that the user is at Ω_j in order to access Mv_j. This scheme corresponds to the movie access policy specified in the access control matrix in Table 2.1.

- We note that it is also possible to specify policies that have no involvement with the object, but instead only require the user to go through a particular spatio-temporal path. Such policies, as depicted in the transition diagram in Figure 2.6 for an automaton M_1'', might not be directly interesting from an access control

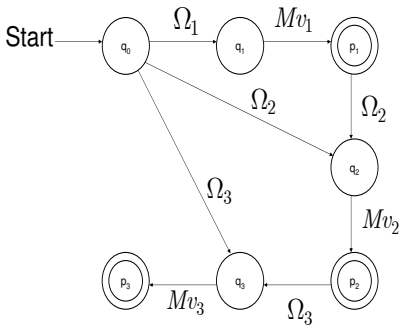


Figure 2.5. The transition diagram for $M_1^{'}$

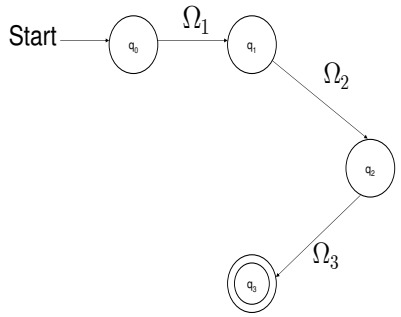


Figure 2.6. The transition diagram for $M_1^{''}$

point of view, but they can be useful as building blocks for more complicated access policies. For example, one can easily envision requiring that a user go to a succession of different locations, prior to being able to access content.

2.4 Related Work

In the area of developing formal access control models, most of the efforts in the literature have focused on general context-sensitive access control models, such as presented in [23]. Here, a formal model is presented which consists of context, policy, request, and algorithm sets. Context information is represented as a six dimensional vector, which includes *time* and *location* as two of the dimensions. This work provides some high-level outlines of authorization and access control protocols that can be based upon

their model. Similarly, in [24], the Context Sensitive Access Control framework was presented, which provided a comparison of different access control mechanisms and context verification mechanisms. Two other related efforts were presented in [25] and [26]. In [25], a comprehensive RBAC model is presented that employs location information in its formal model, while [26] employs temporal information in its model.

Comparing with the existing work, our work focuses on time and location, and our proposed object and ST-region decomposition, comprehensive policies and their representation is new to the literature.

2.5 Enforcement of the Policies

We have proposed method to define and represent service policies to regulate LBS system. Specifically we defined the components that involves in LBS security policies and our proposed concepts of streaming objects, ST-regions, objects and ST-regions decomposition make it easier to define comprehensive policies. We provided different ways to represent security policies.

Once an LBS system has defined its security policies, the next step towards providing a trustworthy service is to ensure that these security policies are enforced. There are different possibilities regarding how the policies can be enforced. In Chapter 5, we show how the policies can be enforced through an underlying network of sensor nodes and without having to acquire uses' location information. However, such a non-interactive infrastructure has restrictions. It is only suitable for applications where the policies are only related to time and location and are independent from user to user. A generic way to enforce the security policies is to obtain the user's location first, and then apply the policies that corresponding to the requesting location, time, and probably the user's identity, preference, historical data. So in the generic methodology, it is important to ensure the accuracy of the location information. We focus on providing trustworthy and accurate location information in Chapter 3 and Chapter 4.

Chapter 3

Improving Localization Accuracy Through Embedded Sensor Networks

3.1 Introduction

Obtaining accurate location information related to an end-user is of critical importance to the success of an LBS system, no matter whether the user is a requester or a target. A straightforward approach to enforce the security policies is to get the user's location and then look up the policies to decide if the user is qualified to get the services he is requesting, or, in the reverse way, what services should be sent to the user. In this chapter we aim to study whether and how the environmental parameters gathered by embedded sensor networks can be used to support localization and position verification.

Although the data associated with sensor readings in sensor networks might be intended to drive specific applications, e.g. the remote monitoring of temperature, this wealth of data may also be dual-used for additional purposes. In particular, since the purpose of a sensor network is to provide sampling of a physical phenomena across a wide geographic/spatial distance, the close link between sensor data and location may be used to assist in applications involving localization and position verification.

In this chapter, we propose the use of spatially varying environmental properties to support localization, without requiring the deployment of a localization infrastructure and additional access points (i.e. landmarks, with known locations). We present the problem of localization using general spatial information fields. We examine the use of physical properties, such as temperature and ambient acoustic/RF energy, and explore whether the inherent spatial variability may be used to localize the position of a mobile entity.

Each physical parameter has its unique spatial characteristics relative to the environment. For computational savings, it is desirable to use the most efficient subset of environmental properties that captures spatial variability. We propose a scheme, Spatial Correlation Weighting Mechanism (SCWM), which can guide in parameter selection by determining the parameter combination with the strongest discriminative characteristics needed for localization.

In our localization model, an array of sensors has been initially deployed for environmental monitoring. These sensors are stationary and the deployment information, such as locations of the sensors, is known to the system. The data collected is used as a baseline database, and a user reports the physical readings at its location. We further developed a set of algorithms, employing environmental properties, to determine a user's position. A two-level approach is taken: first, we perform parameter selection, and then we take measurements to localize. More specifically, (1) from the perspective of how the subset of parameters should be chosen, we present the *Flexibly choosing Environmental Parameters (Flex-EP)* algorithm that tries to find a globally-optimal parameter subset; further, the *Progressive Flexibly choosing Environmental Parameters (Prog-Flex-EP)* algorithm is presented, which is a sequential algorithm that locally customizes the best set of parameters for each user, and (2) we implemented different schemes in mapping the selected set of parameters to a physical location.

To validate our approach we collected an array of environmental parameters at one hundred locations in a real building environment. Using these data for our experiments, we found that SCWM is highly effective in evaluating and selecting the parameter subset with the highest discriminative power. Further, we observed that environmental parameters have localizing capability. By using environmental readings plus the Received Signal Strength (RSS) from one access point, we can achieve qualitatively the same performance as traditional localization schemes employing RSS with at least four access points [27]. Moreover, under the assistance of additional environmental parameters, the localization performance can be refined and improved over traditional approaches.

In summary, our contributions in this work include:

- A localizing mechanism that uses existing sensor network readings and does not

need additional localization infrastructure.

- An environmental parameter selection method that optimizes the subset of parameters for localization.
- An approach that uses these environmental readings to refine conventional localization results.

The remainder of this chapter is organized as follows. In Section 3.2, we provide an overview of our problem. We then formulate the theoretical model of exploiting environmental properties for localization and propose SCWM in Section 3.3. Next, in Section 3.4 we present a spectrum of algorithms that we developed. Section 3.5 shows our experimental methodology and our evaluation results. In Section 3.6, we discuss in depth about two issues, given the result of our experiments. We then place our work in the context of the broader localization literature in Section 3.7. Finally, we conclude in Section 6.6.

3.2 Problem Overview

Wireless sensor networks typically record environmental readings corresponding to underlying physical information fields for applications to utilize. For instance, temperature, humidity, and ambient acoustic energy are common environmental parameters under constant monitoring. The fact that these data are measurements of the environment in a specific area and at a specific time interval suggests that they can be used for spatial localization, as well as their original purpose.

Our proposed model for utilizing the environmental readings for localization and position verification is built upon existing wireless sensor networks, as presented in Figure 3.1. In the area of interest, there are sensors deployed to perform environmental monitoring. Sensors periodically report environmental readings back to Base Stations. The sensors are stationary and their locations are known to the Base Station. The reported environmental information is stored in a database, associated to the sensors' location information, in real time for retrieval by the upper-level applications. A management entity containing data processing and analysis capabilities, the

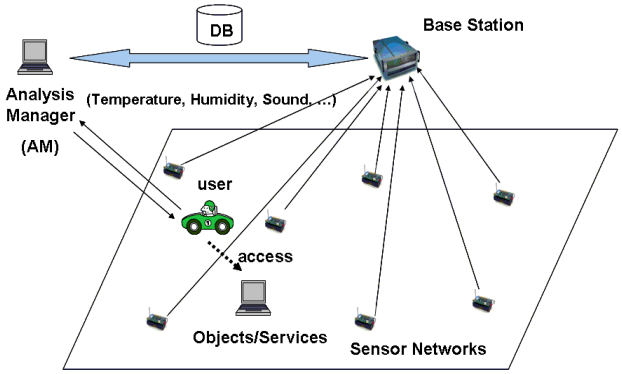


Figure 3.1. Using environmental properties for spatial determination

Analysis Manager(AM), calculates a user's location. The *AM* can be combined with the base station or operate alone in a centralized manner or run with multiple distributed instances. If the *AM* is operating by itself, the *AM* should be able to access the environmental readings stored in the database shown in Figure 3.1.

A user, when it wants to get its position, first sends a request message to *AM*. After receiving the request, the *AM* asks the user to provide environmental readings observed at that time by the user. By running localization algorithms utilizing environmental properties, AM then compares the user's readings to the environmental data (provided by sensors) stored in the database and estimates the user's location.

The traditional approach for localization is to deploy enough landmarks, which measure the received signal strength, with known positions in an area of interest to assist in localization. There maybe cases, however, where there are not enough landmarks in the area of interest (e.g. due to cost limitations), to actually localize. Further it is not always possible to deploy more landmarks due to environmental constraints. In addition, for certain applications, such as position verification in Spatial Access Control [**?**, 28], very high location accuracy results are not needed, so additional landmarks would be wasteful. Thus it is advantageous to use environmental properties from sensor networks to help determine and verify positions without requiring the infrastructure of additional landmarks.

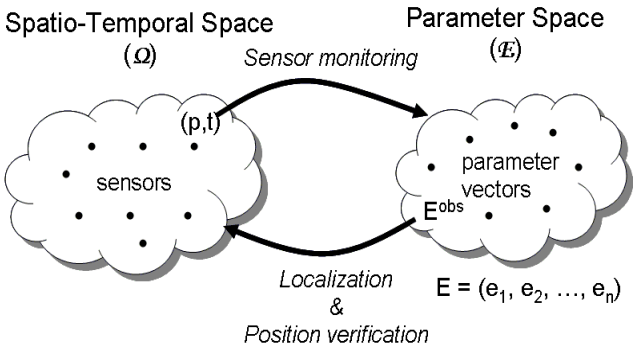


Figure 3.2. Theoretical model: physical domain vs. environmental properties domain

3.3 Theoretical Approach

In this section, we present the theoretical underpinnings behind using environmental properties for localization. We first propose a generalized measurement model, and then provide rules to evaluate each parameter's localizing capability. Finally, we present mechanisms for parameter selection to assist in localization and position verification.

3.3.1 A Generalized Measurement Model

Let $E = (e_1, e_2, ..., e_n)$ denote the vector of environmental properties that are monitored by the sensors, where e_i is the value of the i_{th} environmental parameter. These parameters have the property that they are recorded in the spatio-temporal domain, which means that they may vary with location and time. Thus the value of the parameter vector at position p and time t can be expressed as:

$$E_{p,t} = \left[\; e_1(p,t), \quad e_2(p,t), \quad \ldots, \quad e_n(p,t) \; \right]. \tag{3.1}$$

Here p is a spatial position, which can be one-, two-, or three-dimensional. In this study, we focus on p in a two-dimensional space. More generally speaking, p can represent a point (x, y) or a region. Let $\Omega = P \times T$ be the spatio-temporal region [28] that we are interested in, and $\mathbf{E}$ be the domain of environmental parameter values, then there exists a mapping $f : \Omega \longrightarrow \mathbf{E}$ that takes the physical position p and maps it to an environmental parameter reading $E_{p,t}$ as presented in Figure 3.2. $f(p,t) = E_{p,t}$ represents the environmental readings recorded at the spatio-temporal location (p,t).

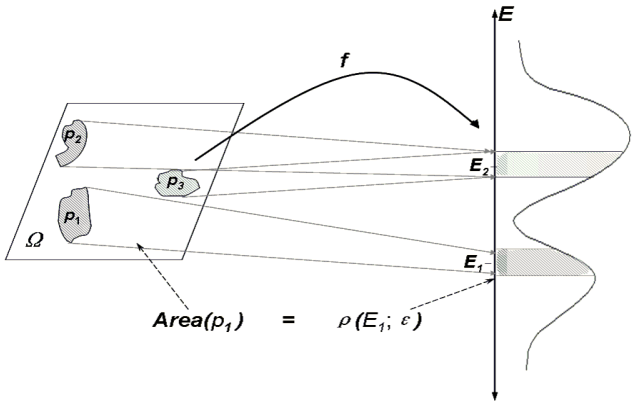


Figure 3.3. Function f induces a probability density function ρ in measurement domain $\mathbf{E}$.

The inverse mapping from $\mathbf{E}$ to Ω enables localization and position verification from environmental properties.

At a fixed time t, the function $f_t(x, y) = f(x, y, t)$ induces a probability density function ρ on $\mathbf{E}$. We further define the function $\rho(E; \epsilon)$ to be the probability of having value $f_t(x, y) \in (E; \epsilon)$ as presented in Figure 3.3, when a position (x, y) is chosen randomly from an uniform distribution from $X \times Y$. Here $(E; \epsilon)$ denotes the ϵ-ball around E. Note that $\rho(E; \epsilon)$ is the integral of the probability density function over the region $(E; \epsilon)$. Given an environment measurement reading E, in order to find the corresponding physical position, we want to find the region $p \subset \Omega$ such that $(E; \epsilon) \subseteq \mathbf{E}$ is mapped back to p. In other words, we want to find the inverse mapping $p = f_t^{-1}(E; \epsilon) = \{(x, y) : f(x, y) \in (E; \epsilon)\}$. Usually Ω has finite area, and we can normalize it to have $Area(\Omega) \triangleq 1$. It is clear that $Area(p) = \rho(E; \epsilon)$, as shown in [**?**].

We also note that, in order to localize a user in a two-dimensional space, simply using a single environmental parameter is generally not sufficient. Thus multiple environmental parameters are desirable for localization and position verification. However, using all the available environmental parameters for localization may result in high computational complexity and cost. We would like to choose subsets of parameters that consist of enough parameters to provide reasonable localization accuracy. Next, we provide an analysis to evaluate environmental parameters and derive methods for effective parameter selection.

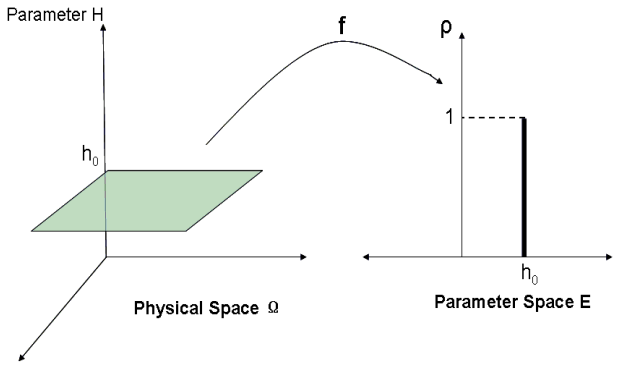


Figure 3.4. An illustration of a "bad" environmental parameter that does not contribute to localization.

3.3.2 Parameter Evaluation

Different parameters have different characteristics in terms of value changes across various physical locations and time. For certain parameters, the values may vary largely across different locations. The physical phenomena reported by these kind of parameters can be utilized to distinguish between locations. We define such parameters as having large *discriminative power*. On the other hand, the values of some parameters may vary little within the area of interest. Figure 3.4 is an illustration of a parameter H belonging to this category. It has the same value h_0 throughout the physical region, thus, in the parameter space $\mathbf{E}$, $\rho(h_0) = 1$. Such a parameter does not have the ability to distinguish between physical locations and thus has poor localization accuracy.

It is important to choose a parameter subset so that the combination of the parameters in the subset have enough discriminative power to support localization. Carelessly choosing a parameter subset may even result in localization errors as shown in Figure 3.3, where two far-away regions p_2 and p_3 in the physical space have the same environmental readings $(E_2; \epsilon)$. The inverse mapping would result in $f_t^{-1}(E_2; \epsilon) = p_2 \bigcup p_3$. This indicates that the subset of parameters in this case is not sufficient for localization.

To summarize, in order for an environmental parameter or a subset of environmental parameters to contribute in localization and position verification, they should have the following characteristics:

1. The subset of parameters should have large discriminative power. We found

that a parameter that contributes largely to location accuracy must have large variance across the environment. On the other hand, a parameter with large variance may not necessarily help to improve its localization capability. Contrary to intuition, the correlation between parameters is not an important factor in location accuracy.

2. The parameter readings with similar values should map to sensors positioned close to each other. This will eliminate large localization errors.

3. There must exist a spatial correlation for the parameter readings so that similar values for parameter readings will result in locations close in the physical domain.

3.3.3 Parameter Selection

Next, we develop a series of approaches to help select environmental parameters, that when combined, have greater capability for localization and position verification.

3.3.3.1 Parameter dispersion

Conceptually, for an environmental parameter e_i, the more disperse the values are, the better the discriminative power is for this parameter. In statistics, there are several ways to measure dispersion of a parameter, such as *range*, *variance*, *standard deviation* and *average absolute deviation*.

However, none of these measurements are complete, since they only look at the data itself and neglect the spatial relationships between the data and the physical environment. For example, if the data readings from two different environmental parameters have the same distribution, their dispersions are about the same. However they may result in very different accuracy if used for localization. Suppose both of them have a subset of readings with the same value, but for one parameter, the same-value readings are clustered, while for the other parameter, the same-value readings are scattered among the region. The latter parameter will generate larger errors when applied in localization.

Further, we found that the cross-parameter relationship, covariance, does not heavily contribute to localizing capability. Instead, the spatial relationship dominates localization accuracy. This leads us to look for metrics that take into consideration of the spatial correlation when evaluating an environmental parameter, in addition to the parameter dispersion.

3.3.3.2 Data Normalization

Different from traditional localization methods [**?**, **?**], the data sources in our problem are from different kinds of environmental parameters, such as temperature, humidity, ambient acoustic energy, and etc. Different environmental parameters have different units and different ranges of values. For example, in our experiments the temperature readings range from 65.2F to 77.3F, whereas Received Signal Strength values range from -59.8dBm to -99dBm. In order to choose a subset of environmental parameters working together for localization purpose, we need to compare and calculate the contribution of each parameter directly. We normalize the data using the classical statistical approach: $e_i^{norm} = \frac{e_i - \mu_i}{\sigma_i}$, where μ_i and σ_i are the mean and standard deviation of the parameter e_i. We then work with the normalized data e_i^{norm} for the rest of our study.

3.3.3.3 Spatial Correlation Weighting Mechanism

A natural approach for selecting parameters is to slice the domain of the environmental parameters **E** into equal-sized bins. Then for parameter readings falling within the same bin, we calculate their corresponding distances in the physical space for every pair of sensors. Based on our criteria for parameter evaluation (item 2 and item 3 in Section 3.3.2), among all the possible combinations of fixed-size parameter subsets, a parameter subset that results in the smallest averaged physical distance per bin is the optimal parameter subset with the highest combined discriminative power for that size of parameter subsets.

The size of the bin and the number of bins are the two critical factors. We found that different parameter subsets behave differently on different bin sizes. In order to give a thorough evaluation of parameters, we need to vary the bin size and come up

with a way to combine the results from different bin sizes to make a fair judgment. In addition, the disadvantage of this approach is that the number of bins (and hence computational complexity) needed to increase exponentially with the number of environmental parameters in a parameter subset. Thus, the computational complexity increases exponentially with the size of the parameter subset. It is not desirable to use a metric that has potentially high computational complexity. To solve these problems, we have developed a method whose complexity does not change much with the size of parameter subset, and inherently solves the different bin-size problem. We now present this method, the Spatial Correlation Weighting Mechanism (SCWM).

The important factor to take into consideration when performing parameter selection is to analyze how far away two positions can be in the physical domain $\mathbf{P}$, given a distance in the environmental-parameter domain $\mathbf{E}$. The SCWM calculates a sum $W(K)$ of pairwise weighted distances in $\mathbf{P}$, which gives larger weight to similar parameter readings in $\mathbf{E}$ and smaller weight to more different parameter readings. If we define K to represent a set of parameter indices chosen to form the parameter subset, $W(K)$ is defined as follows:

$$\begin{aligned} W(K) &= \sum_{p_i,p_j,\, i\neq j} w_{i,j} \cdot d_{i,j} \\ &= \sum_{p_i,p_j,\, i\neq j} w_{i,j} \cdot \|p_i - p_j\|^2 \\ with\ \ w_{i,j} &= \frac{1}{1+\tau \cdot \|e_{k\in K}(p_i) - e_{k\in K}(p_j)\|^2} \end{aligned} \tag{3.2}$$

where τ is a scaling factor. We call $w_{i,j}$ the *parameter weight*, which takes values from $(0, 1]$. The computational complexity does not dramatically increase with the number of parameters in a parameter subset when using SCWM for parameter selection.

Figure 3.5 illustrates how SCWM helps to choose the parameter subset with the highest discriminative power. We describe three typical scenarios during SCWM calculation. The first scenario is shown with position pair p_2 and p_3. The two positions are close to each other and they have similar parameter readings. The contribution of the parameter weight $w_{2,3}$ is large, close to 1. But the resulting $(w_{2,3} \cdot d_{2,3})$ is small because

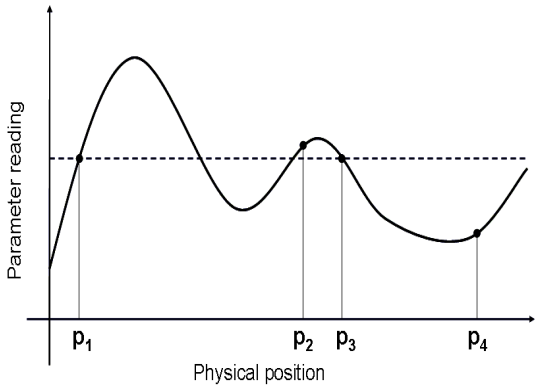


Figure 3.5. Three scenarios under SCWM calculation: (1) position pair $\{p_2, p_3\}$; (2) position pair $\{p_1, p_4\}$; (3) position pairs $\{p_1, p_3\}$ and $\{p_1, p_2\}$. The relationship is: $(w_{1,3} \cdot d_{1,3}) > (w_{1,2} \cdot d_{1,2}) >> (w_{1,4} \cdot d_{1,4})$ *and* $(w_{2,3} \cdot d_{2,3})$.

$d_{2,3}$ is very small. Next, position pair p_1 and p_4 is farther away from each other and their parameter readings are very different. In this case, the contribution of the $w_{1,4}$ is very small and much less than 1. The above two scenarios satisfy the theoratical requirements of better location accuracy described in Section 3.3.2. Finally, we look at a poor scenario with position pairs $\{p_1, p_3\}$, and $\{p_1, p_2\}$. Their parameter readings are the same or very similar, but they are farther away from each other. The contribution of the parameter weight is large, especially for $w_{1,3}$ which reaches its maximum, equaling to 1. Both $(w_{1,3} \cdot d_{1,3})$ and $(w_{1,2} \cdot d_{1,2})$ are also large because the distances are farther away.

For a fixed number of parameters, SCWM calculates all the pairwise weighted distances over all the possible combination of parameters. The parameter subset with most of its readings following the patterns described in the first two scenarios will result in the final value of $W(K)$ to be small. While the parameter subset having most of readings similar to the third scenario we presented, the calculated value of $W(K)$ will be large. The parameter subset that results in the minimum value of $W(K)$ is the optimal parameter combination that contains the highest discriminative power for performing localization. SCWM can sort all the possible combination of parameters under a fixed size parameter subset in the descending order from the highest discriminative power to the lowest discriminative power for localization. In Section 3.5, we present experimental results utilizing SCWM.

Algorithm	Variant	Description
Flexibly choosing Environmental Parameters (Flex-EP)		
Flex-EP-Dist	Basic	Finds the location of the sample point that is closest to the user's reading in the parameter space.
Flex-EP-Dist	Avg	Finds the centroid of the top K locations of sample points that are closest to the user's reading in the parameter space.
Flex-EP-Prob	Basic	Finds the location of the sample point with the highest probability of correctness.
Flex-EP-Prob	CM	Finds the center of mass of the top K locations as determined by the threshold β, weighted by their probability of correctness.
Progressive Flexibly choosing Environmental Parameters (Prog-Flex-EP)		
Prog-Flex-EP-Dist	Basic	In each round, progressively chooses the most effective parameter, excludes candidate positions (that are far from the user's reading in the current subspace) based on the selection ratio γ, and returns the position that is closest to the user's reading among those candidates left in the last round.
Prog-Flex-EP-Dist	Avg	In each round, progressively chooses parameters, selects candidate positions, and returns the centroid of the top K locations of candidates remaining in the last round.
Prog-Flex-EP-Prob	Basic	In each round, progressively chooses the most effective parameter, excludes candidate positions with a low probability of correctness based on either the cumulative probability confidence α or the individual probability threshold τ, and returns the position with the highest probability among the candidates remaining in the last round.
Prog-Flex-EP-Prob	CM	In each round, progressively chooses parameters, candidate positions, and returns the center of mass of the top locations remaining in the last round as determined by the threshold β, weighted by their probability of correctness.

Table 3.1. Summary of the algorithms employing environmental properties.

3.4 Algorithms

In this section, we present our algorithms for using environmental parameters, assisted by SCWM, for localization.

3.4.1 Overview

There are two important aspects when developing algorithms using environmental parameters for localization: one is how to choose the subset of parameters given a fixed number of parameter set, e.g., choose all at once or choose one at a time; and the other is to derive a mapping function from environmental parameters to a physical location. Taking into consideration of these two aspects, we used a two-level approach and developed an array of algorithms employing environmental properties.

First, we propose two basic algorithms, namely *Flexibly choosing Environmental Parameters (Flex-EP)* and *Progressive Flexibly choosing Environmental Parameters (Prog-Flex-EP)* for selecting parameters. Given a set of parameters Γ that is chosen from all the available environmental parameters using SCWM, *Flex-EP* chooses the subset of parameters in one shot based on the results from SCWM, while *Prog-Flex-EP* progressively chooses one parameter at a time until reaching the number of parameters defined by the parameter subset. Thus, *Flex-EP* tries to find a global-optimal parameter subset, while *Prog-Flex-EP* locally customizes the best set of parameters for each testing point.

Next, we consider how position estimation is carried out. We implement the basic algorithms using two types of mapping functions, denoted by *-Dist* and *-Prob*. *-Dist* utilizes nearest neighbor matching in the parameter space, whereas *-Prob* employs a statistical maximum likelihood estimation approach. We derived four algorithms to perform location estimation and position verification: *Flex-EP-Dist*, *Flex-EP-Prob*, *Prog-Flex-EP-Dist*, and *Prog-Flex-EP-Prob*.

In addition, we developed variants of the above algorithms extending from their basic ideas. *-Avg* is a variant for the *-Dist* function, which finds the centroid of the top k returned locations. The *-CM* variant is derived from the *-Prob* function that returns

---

input $E_{\Gamma}^{obs}(p,t)$, Γ
output
min dist in the parameter domain
closest sensor position in the physical domain
initialize
$minDist = maxNum$
$sensorPosition = empty$

loop through information reported by sensors
 for each set of information from a sensor **begin**
  $dist = \|E_{\Gamma}^{obs}(p,t) - E_{\Gamma}^{sensor}(p,t)\|$

  if $dist < minDist$
  then $minDist = dist$ and $sensorPosition = sensor$
 end for
end loop
return $minDist$, $sensorPosition$

Figure 3.6. The Flex-EP-Dist-Basic algorithm

the center of mass of the top locations satisfying a threshold probability rule. The summary of the algorithms and their variants are presented in Table 3.1. We discuss the details of each algorithm in the following subsections, and later present experimental results.

3.4.2 *Flex-EP*

Flex-EP-Dist: *Flex-EP-Dist* finds the minimum distance in the domain of environmental parameters between the observed readings E_{Γ}^{obs} reported by the user and the collection of measurements recorded by the sensor network, which are stored in the database, as shown in Figure 3.1. *Flex-EP-Dist* reports the position of the closest sensor as the location estimate of the user.

We use *Flex-EP-Dist-Basic* to represent the basic version of *Flex-EP-Dist*. Figure 3.6 presents the pseudo-code that implements *Flex-EP-Dist-Basic*. Further, its variant *Flex-EP-Dist-Avg* chooses the top k sensors that are closest to the user's reading in parameter space and returns the centroid of the k locations, with $k > 1$.

Flex-EP-Prob: Instead of calculating the distance in parameter space between a user's measurement and each sensor's reading as in *Flex-EP-Dist*, *Flex-EP-Prob* takes

the approach of calculating the probability that the user is at each position and choosing the one with the maximum likelihood as the estimation of the user's location. In order to support a probabilistic formulation of our localization problem, we employ several practical assumptions. First, we assume that the sensor network provides accurate sampling of the environment and, specifically, that a sensor at position p_j will measure the value of the i-th environmental parameter as e_i. On the other hand, in order to account for measurement errors relative to the sensor nodes, we assume that a user's measurements at a position p_j will be distributed in a Gaussian manner about the sensor's measurement at that position. That is, a user's measurement at position p_j will have mean $\mu_{I,j} = e_i(p_j, t)$, with a standard deviation σ_i. Additionally, for tractability of our formulation, we assume that the user's environmental parameters are independent of each other and thus at position p_j, the vector random variable $E_\Gamma = (E_1, E_2, \cdots, E_k)$, where k is the cardinality of the parameter subset Γ, follows a multivariate Gaussian density:

$$f_{E_\Gamma, p_j}(\tilde{e}_1, \ldots, \tilde{e}_k) = \prod_{i=1}^{k} f_{E_i, p_j}(\tilde{e}_i)$$
$$= \frac{1}{(2\pi)^{k/2} \prod_{i=1}^{k} \sigma_i} \exp\left(-\frac{1}{2} \sum_{i=1}^{k} \frac{(\tilde{e}_i - \mu_{i,j})^2}{{\sigma_i}^2} \right) \quad (3.3)$$

with $\tilde{e}_i$ being the user's measured values. Given a user's observed parameter vector $E_\Gamma^{Obs} = (\tilde{e}_1, \tilde{e}_2, \cdots, \tilde{e}_k)$, we can calculate the probability of being at position p_j by using Bayes' rule:

$$P\left((p_j, t)|E_\Gamma^{obs}\right) = \frac{P\left(E_\Gamma^{obs}|(p_j, t)\right) \times P(p_j, t)}{P(E_\Gamma^{obs})}. \quad (3.4)$$

Further, we may assume a uniform distribution over possible user locations, i.e. $P(p_j, t) = \frac{1}{N}$, $\forall\, j$, where N is the total number of possible locations. Also we note that $P(E_\Gamma^{obs})$ is a constant, thus we have

$$P\left((p_j, t)|E_\Gamma^{obs}\right) = c \cdot P\left(E_\Gamma^{obs}|(p_j, t)\right). \quad (3.5)$$

Given the fact that the user must be localized to one of the N sample locations, by using the density function from above, we can calculate the probability of being at each location:

$$P\left((p_j, t)|E_\Gamma^{obs}\right) = c \cdot f_{E_\Gamma,j}\left(E_\Gamma^{obs}\right), \tag{3.6}$$

where $c = 1/\sum_{j=1}^{N} f_{E_\Gamma,j}\left(E_\Gamma^{obs}\right)$. Finally, the basic version of *Flex-EP-Prob*, i.e., *Flex-EP-Prob-Basic*, chooses the position with the highest probability as the location estimation.

Moreover, the variant *Flex-EP-Prob-CM* sorts the probabilities in descending order, and then picks the top k locations $(p_{i_1}, ...p_{i_k})$, such that the sum of their probabilities $\sum_{m=1}^{k} p_{i_m} \geq \beta$, where β is an adjustable threshold. *Flex-EP-Prob-CM* returns the *center of mass* of these locations (with high probabilities) as the location estimation:

$$(\hat{x}, \hat{y}) = \left(\sum_{m=1}^{k} p_{i_m} x_{i_m}, \sum_{m=1}^{k} p_{i_m} y_{i_m}\right). \tag{3.7}$$

3.4.3 *Prog-Flex-EP*

Prog-Flex-EP is a successive refinement localization method. Instead of choosing all the parameters of a parameter subset at once (as described in *Flex-EP* algorithms), *Prog-Flex-EP* sequentially chooses one parameter at a time from the parameter subset. In each round, *Prog-Flex-EP* picks a parameter E_i which is most effective, i.e., with high discriminative power when combined with parameters already chosen from the previous rounds based on SCWM. Further, according to the combined parameters $E_\Gamma = (E_1, E_2, ..., E_i)$, in every round, the location candidates are refined by choosing a subset from the candidate locations $\widetilde{P}$ from the previous round. This procedure repeats until it finds a solution with high confidence or reaches the maximum number of parameters given a fixed size of the subset of parameters.

Figure 3.7 illustrates how the successive refining location candidates is performed. Suppose at round $i-1$, the candidate location areas are Ω_1, Ω_2, and Ω_3 as shown in Figure 3.7, while the true location is shown as a star residing within Ω_2. At round i, based on the newly combined parameter subset, we sort the candidate locations

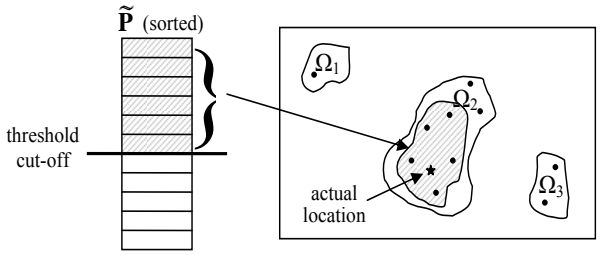


Figure 3.7. The location estimation is refined by sequentially choosing new parameters, leading to successfully smaller subset of locations.

according to the appropriate criteria and select the ones that are within the threshold, as the new candidate location set. We note that the threshold is a general concept: it could indicate the cumulative probability confidence α or the individual probability threshold τ in the algorithm variants of *Prog-Flex-EP*, which will be discussed shortly. As illustrated in Figure 3.7, the new candidate set excludes areas Ω_1 and Ω_3, and includes a partial area of Ω_2 (shown as a shaded region) that contains the true location.

Therefore, in the approach of *Flex-EP* we select the overall best K environment properties beforehand and always use that same set of properties for location estimation. Whereas in *Prog-Flex-EP*, the next selected parameter is subject to change based on the results from the former rounds. In reality, it could be that at different locations, the environmental properties that best distinguish this location from others are different. For example, in one location, the three best parameters are RSS, barometric pressure and light, while in another location, they could be temperature, RF energy and humidity. Thus, the approach of *Prog-Flex-EP* gives the localization process a chance to locally customize the sets of parameters used.

The key challenge for *Prog-Flex-EP* is how to evaluate the candidate locations at each round and decide on the candidates for the next round. We further developed *Prog-Flex-EP-Dist* and *Prog-Flex-EP-Prob* with regard to their different criteria used for ranking the candidate locations and setting the threshold cut-off.

Prog-Flex-EP-Dist: *Prog-Flex-EP-Dist* ranks the candidate locations at each round based on the Euclidean distance between the observed readings and the readings at each location in $\widetilde{P}$ in the parameter space using the current parameter set E_Γ. Then

initialize
$\Gamma = empty$ //Selected environmental parameters
$sensorVectors = list\ of\ all\ sensor\ vectors$
$\gamma = selection\ ratio$

while $|\Gamma| < totalSubParams$ and $|sensorVectors| > 1$
for each $param_i$ not in Γ
calculate $W_i(\{\Gamma, param_i\})$ based on SCWM
find the minW_i, newParam = $param_i$ that makes minW_i
$\Gamma = \{\Gamma, newParam\}$
for each sensor location $\in sensorVectors$
calculate its distance from E^{obs} in the parameter subspace
$dist_\Gamma = \|E_\Gamma^{obs} - E_\Gamma^{sensor}\|$
end for
sort the remaining $sensorVectors$ by increasing distance
select the top $\lceil \gamma \cdot |sensorVectors| \rceil$ entries from the sorted $sensorVectors$ as the new $sensorVectors$
end while

return the first entry of $sensorVectors$

Figure 3.8. The Prog-Flex-EP-Dist-Basic Algorithm

it selects $|P_{new}| = \lceil \gamma \cdot |\widetilde{P}| \rceil$ as the number of candidates for the next round. Here γ is an adjustable *selection ratios* with $0 < \gamma \leq 1$ (e.g. $\gamma = 0.3$), and $|\widetilde{P}|$ denotes the number of locations in $\widetilde{P}$. Figure 3.8 presents the pseudo-code that implements *Prog-Flex-EP-Dist-Basic*. We note that different *selection ratio* will result in different localization results. Smaller ratios allow us to refine our candidate list faster, thus permitting more effective parameter selection. However, it could also result in eliminating too many candidate locations, which increases the risk of throwing out the correct candidates prematurely. Thus, choosing an appropriate selection ratio γ will allow *Prog-Flex-EP-Dist* to be robust.

Prog-Flex-EP-Prob: Similar to *Flex-EP-Prob*, *Prog-Flex-EP-Prob* calculates the probability that the user is at each sample location, but based on the current progressively obtained parameter set Γ instead of the whole parameter subset. Further, we define a *confidence*, α, as the threshold cut-off level. Given a confidence α, we pick the set of most likely locations such that the sum of their probabilities is larger than α.

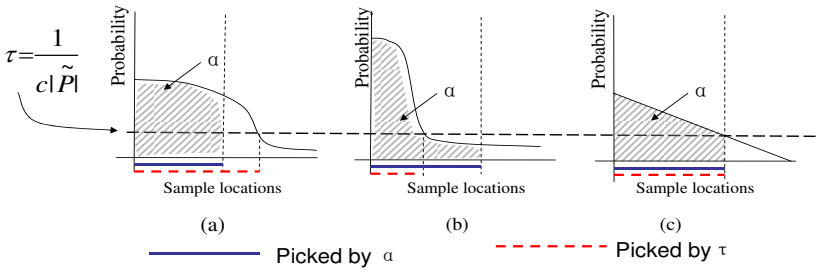


Figure 3.9. Prog-Flex-EP-Prob: Comparison of candidates selection when using cumulative probability confidence α and individual probability threshold τ.

We note that in this algorithm α represents a cumulative threshold from each round: the larger the α, the more candidates are chosen for the next round. Ideally, a threshold would keep the sample locations that have obviously high probabilities to the observed readings, and eliminate those candidates that have much lower probabilities. In some cases, a cumulative threshold may not provide the desired effect. Figure 3.9 presents three scenarios with different probability distributions. In each scenario, the shaded area shows the probabilities that accumulate to the confidence α, as an example $\alpha = 75\%$, and the solid blue line on the bottom shows the sample locations picked by this cumulative threshold. τ is an individual probability value used as threshold. Using τ as the cut-off, every sample location with probability greater than τ is picked to enter the next round and the others are thrown away. The red dotted line shows the sample locations picked by using τ as the threshold. In either Figure 3.9 (a) or (b), there is a significant drop in each of the probability distribution. Keeping the locations before the drop and eliminating the rest would be desirable. But in (a), we accumulated enough probability before the drop so some "good" sample locations still with high probabilities are prematurely eliminated, whereas in (b), many unlikely candidates are unnecessarily included since not enough probability is accumulated before the drop. These low probability points may become the reason to cause confusion in the next round. In these two cases, however, an individual threshold τ does a better cut-off. The individual threshold τ is related to the number of current candidates. We define the threshold τ to be $\frac{1}{C \cdot |\widetilde{P}|}$ where C is an adjustable constant. Figure 3.9 (c) presents

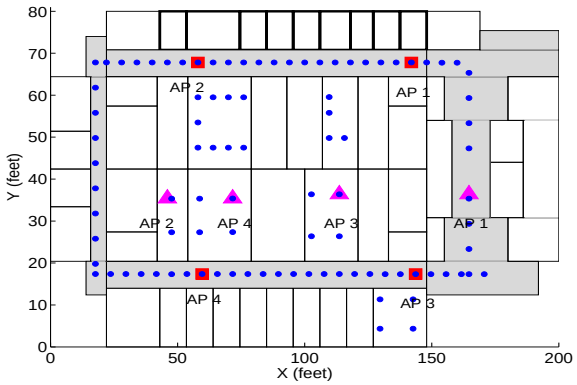


Figure 3.10. Layout of the experimental floor.

that under a normal situation the selection results of using confidence α and threshold τ are the same.

Moreover, in order to compare to the *Flex-EP* approaches, two variants out of the basic algorithms: *Prog-Flex-EP-Prob-Avg* and *Prog-Flex-EP-Prob-CM*, are developed based on the current progressively obtained parameter subsets Γ.

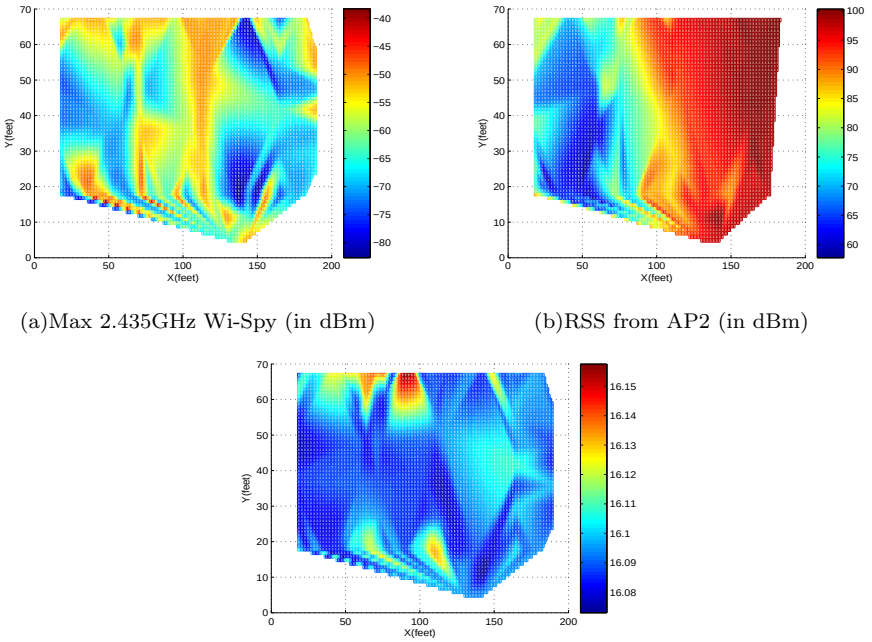
3.5 Experimental Evaluation

In this section we present our experimental evaluation results.

3.5.1 Experimental Methodology

In order to study the effectiveness of using environmental properties for localization and position verification, we conducted experiments in a real office environment, the 3rd floor of the Computer Science building at Rutgers University, as shown in Figure 3.10. For over one hundred locations on the floor, shown as small blue dots, we collected environmental readings at these locations over a one-week period of time. This simulated the setup of a sensor network consisting of one hundred sensors.

The environmental parameters that we studied are temperature, humidity, acoustic noise, spectrum energy, and Received Signal Strength (RSS). The RSS readings are collected from two 802.15.4 (ZigBee) networks, each with four Access Points (AP) deployed across the floor. One network has four APs deployed horizontally across the floor shown as triangles, and the other has four APs distributed in a rectangular pattern



(a)Max 2.435GHz Wi-Spy (in dBm)

(b)RSS from AP2 (in dBm)

(c)Ambient noise (daytime)(in dB)

Figure 3.11. Sample data maps of individual environmental parameters

shown as squares. The access points used were Telosb motes.

To simulate a scenario where only one base station is available in the area of interest, we will choose only one RSS reading (in dBm) when forming the parameter subset. Further, we used a Wi-Spy spectrum analyzer [**?**] to record the spectral energy at each location. It records the signal amplitude (in dBm) versus the frequency from 2.400GHz to 2.485GHz. At each testing location, we picked two frequencies (2.435GHz and 2.465GHz), and calculated their maximum and average amplitude respectively over the recording period. Note that the RSS is the received signal from a beacon packet, while the spectrum energy is the ambient RF energy corresponding to a specific frequency range.

For acoustic noise, our intuition is that the behavior of the parameter can vary largely during daytime and night time. Thus we collected readings of ambient noise (in dB) for both day and night. Moreover, we measured the humidity (in percentage) using a digital hygrometer and temperature (in Fahrenheit) using a thermometer respectively at each location. Table 3.2 is a summary of the parameters and the devices that we used to conduct experiments.

Parameter		Index	Measuring Device
Temperature		1	Thermometer
Humidity		2	Digital hygrometer
Acoustic Noise	Daytime	3	Microphone and Dell laptop
	Night time	4	
Spectrum Energy	2.435GHz Max	5	Wi-Spy Spectrum Analyzer
	2.465GHz Max	6	
	2.435GHz Avg	7	
	2.465GHz Avg	8	
Received Signal Strength (RSS), Colinear	AP 1	9	Telosb motes and Dell laptop
	AP 2	10	
	AP 3	11	
	AP 4	12	
Received Signal Strength (RSS), Rectangular	AP 1	13	Telosb motes and Dell laptop
	AP 2	14	
	AP 3	15	
	AP 4	16	

Table 3.2. Summary of Environmental Parameter Measurement

3.5.2 Evaluation of Individual Parameters

We first study the dispersion of individual environmental parameters through parameter variance. Table 3.3 presents the results of the variance for each individual parameter. We found that the maximum value of the spectral energy and the RSS have large variance across the area of interest, while the average value of the spectrum energy, temperature, humidity, and ambient noise do not vary much across the experimental floor. Both daytime and night time readings of ambient noise have smaller variance compared to other environmental parameters. For the rest of the chapter, we will use the ambient noise data collected at night. The sample maps of spectral energy at 2.435GHz, RSS from AP2, and ambient noise are shown in Figure 3.11. The irregular shape of signal maps is due to the limitation of our data collection. We can see that the sample readings of ambient noise do not change much across the whole floor, while both the maximum values of spectrum samples at 2.435GHz and the RSS readings from AP2 present large variance indicating high discriminative power to describe the uniqueness of each location in the floor.

Parameters and Their Variance			
Temperature	**Humidity**	**Acoustic noise** daytime	**Acoustic noise** night time
4.15	9.3	0.01	0.0012
Spectrum energy			
2.435GHz Max	2.465GHz Max	2.435GHz Avg	2.465GHz Avg
84.36	88.21	2.09	0.08
Received Signal Strength (RSS), Colinear			
AP1	AP2	AP3	AP4
211.63	136.65	123.31	127.27
Received Signal Strength (RSS), Rectangular			
AP1	AP2	AP3	AP4
120.48	142.35	126.76	125.24

Table 3.3. Results of single-parameter dispersion

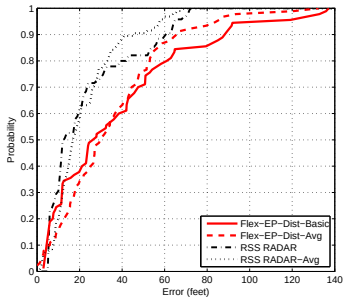
# of Parameters in a subset	Evaluation	Parameters: in index	SCWM calculation
1	Good:	12	20641548.6
	Bad:	8	370356046.8
2	Good:	4, 12	595033.7
	Bad:	2, 7	23284151.1
3	Good:	4, 6, 12	140758.4
	Bad:	1, 2, 8	940833.1
4	Good:	1, 4, 6, 10	72365.1
	Bad:	1, 2, 4, 7	201856.6
5	Good:	1, 4, 5, 8, 12	55198.0
	Bad:	1, 2, 4, 7, 8	112585.6

Table 3.4. Evaluation of SCWM with different size of parameter subsets

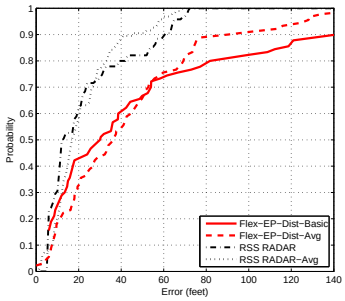
3.5.3 Effectiveness of Parameter Selection

In this section, we present the results of parameter selection using SCWM. We then evaluate the effectiveness of SCWM by comparing the cumulative distribution function (CDF) of localization errors under different sizes of parameter subsets with traditional localization methods.

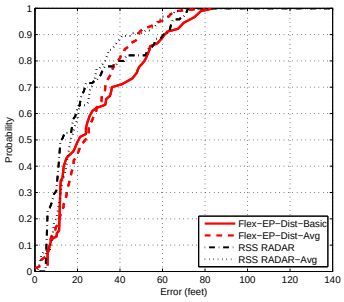
Parameter Selection Using SCWM: Table 3.4 presents the results of $W(K)$ calculated from SCWM with the size of K equal to 1, 2, 3, 4, and 5 respectively. We have shown a representative subset of parameters in Table 3.4 with "good" and "bad" indicating that the value of $W(K)$ is smaller or larger. As we described in



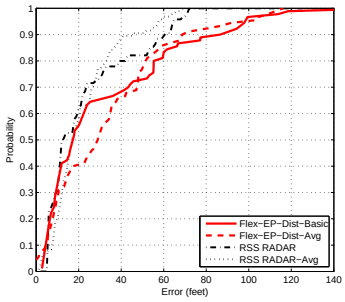
(a) two parameters (good): index 4, 12



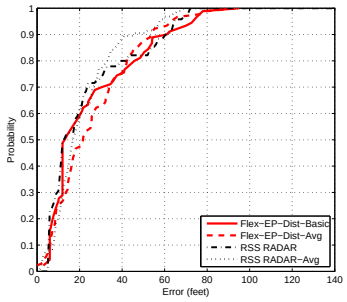
(d) two parameters (bad): index 2, 7



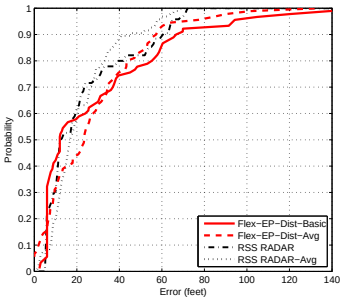
(b) three parameters(good): index 4, 6, 12



(e) three parameters (bad): index 1, 2, 8



(c) four parameters (good): index 1, 4, 6, 10



(f) four parameters (bad): index 1, 2, 4, 7

Figure 3.12. Comparison of localization errors using cumulative distribution function (CDF)

Section 3.3.3, the smaller the value of $W(K)$ for a parameter subset, the higher the discriminative power the parameter subset has. From our experimental results, we found that the parameter subset containing all the RSS parameters will result in the minimum value of $W(K)$. This is because the parameters of RSS readings have the largest variance, also have high spatial correlation, and thus can uniquely describe the physical variability across the experimental floor. However, since we also interested in the situations where there is no localization infrastructure available, we may rely on the additional environmental properties to assist in localization and position verification. Thus, the parameter subsets displayed in Table 3.4 only involved at most one RSS

parameter in the subset.

Localization Using Environmental Parameters: Based on the parameter selection results obtained from SCWM, we further conducted localization with these parameter subsets utilizing the *Flex-EP-Dist-Basic* and *Flex-EP-Dist-Avg* algorithms. In order to compare the performance of our approach, we need to compare with a performance benchmark in the current localization research. The traditional RADAR algorithm [?] and its corresponding variants are used for our comparison, which utilize the RSS readings collected from four APs in our ZigBee network with the horizontal AP deployment.

Figure 3.12 presents the CDFs of localization errors for *FLEX-EP* with the size of the parameter subset set to 2, 3, and 4 respectively. The localization results using RADAR are presented as a comparison. In each figure, the results from RADAR are presented using thinner black dash-dot lines (for regular RADAR) and black dotted lines (for RADAR-Avg algorithm), and the results from our algorithms are showed in thicker red lines. Figures 3.12(a) and 3.12(d) are the results using two parameters in the parameter subset. The localization results when using *RSS from AP4* and *acoustic noise* are better than using *humidity* and *2.435GHz Avg*. This is because the parameter, *RSS from AP4*, has large variance and better spatial correlation across the experimental floor. Thus, the parameter subset, {*RSS from AP4*, *acoustic noise*}, has smaller SCWM value than the set of {*humidity*, *2.435GHz Avg*}. The performance of $Flex-EP$ using two parameters is not as good as the performance of RADAR.

Next, Figures 3.12(b) and 3.12(e) show the error CDFs when using three parameters in the parameter subset. We added one more parameter with high discriminative power, *2.465GHz Max*, into the "good" parameter set of two parameters shown in Figure 3.12(a). Figure 3.12(b) presents the results of using {*acoustic noise*, *2.465GHz Max*, *RSS from AP4*}. We found that when using two environmental parameters with high discriminative power (*2.465GHz Max* and *RSS from AP4*) and one parameter with low discriminative power (*acoustic noise*), the performance of *Flex-EP* is qualitatively similar to traditional RADAR algorithms, which uses four RSS parameters. In Figure 3.12(e) each parameter in the parameter subset {*temperature*, *humidity*, *2.465GHz*

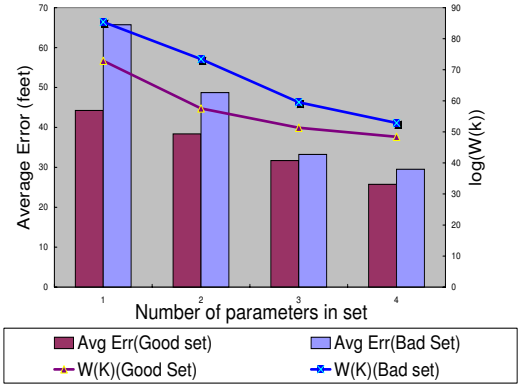


Figure 3.13. Summary of the efficiency of SCWM across different parameter subsets.

Avg} has low discriminative power and results in a larger SCWM value as shown in Table 3.4. Hence, using these properties for localization yields slightly worse performance than RADAR.

Further, we examined the localization error CDFs when using four environmental parameters in Figures 3.12(c) and 3.12(f). In Figure 3.12(c), we still use two environmental parameters containing high discriminative power, *2.465GHz Max* and *RSS from AP4*, while *acoustic noise* and *temperature* do not vary much across the experimental site. Again, we observed the performance of *Flex-EP* is about the same as the RADAR and its variants. Moreover, under the assistance of two environmental parameters with low discriminative power, the performance is slightly improved over the three-parameter subset case as shown in Figure 3.12(b).

These results indicate that choosing two environmental parameters containing high discriminative power is enough to produce comparable performance to the traditional localization approaches employing RSS with at least four access points. On the other hand, as shown in Figure 3.12 (f), simply adding environmental parameters with low discriminative power into a parameter subset does not significantly improve the localization performance.

Figure 3.13 shows the efficiency of SCWM in one picture. Given a set of parameters K, the point on the curve shows the calculated value of $W(K)$ according to SCWM, and the bar below the point shows the actual average error distances using the corresponding

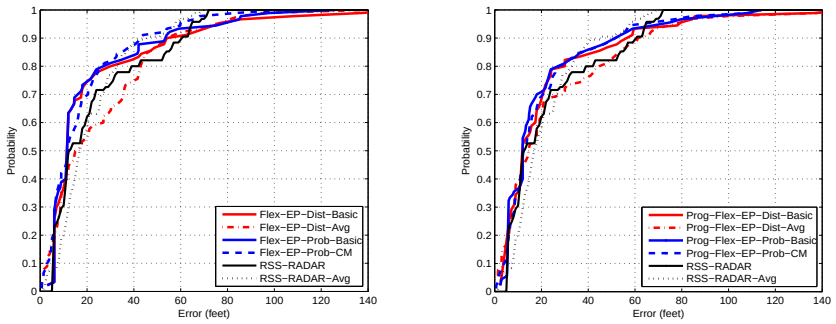
set K. According to the theory of SCWM, we predict that the sets on the blue curve with squares, which is denoted as $W(K)$(Bad set), will perform worse than the sets on the purple curve with triangles, which is the $W(K)$(Good set) curve, because they have larger values of $W(K)$. The fact that the bars of Avg Err(Good set) are always lower than the bars of Ave Err(Bad set), demonstrate that our SCWM consistently predicts the performance of parameter subsets for localization.

Since when using RSS for localization, the performance across a broad spectrum of algorithms was found to be about the same [?], we conclude that utilizing $SCWM$ for parameter selection and our algorithms are effective and can achieve similar performance to a broad array of traditional localization algorithms. The similar performance is very encouraging as it indicates utilizing environmental properties can effectively determine the position of a user.

3.5.4 Algorithm Performance Comparison

Flex-EP and its variants: The performance results for *Flex-EP-Dist-Basic*, *Flex-EP-Dist-Avg*, *Flex-EP-Prob-Basic*, and *Flex-EP-Prob-CM* are presented in Figure 3.14 (a). The selection ratio γ is set to 0.5 for *Flex-EP-Prob* algorithms. In both sets of the experiments, we use four parameters, in which only one is RSS. We observed that *Flex-EP* and its variants perform similarly. When using parameter set $\{1, 4, 6, 16\}$ (see Table 3.2), the results of *Flex-EP-Prob*, as shown in blue lines, are always higher than the black lines, which are the results of RADAR, thus we observed *Flex-EP-Prob* outperforms *RADAR* in this setting. The result of the *Flex-EP-Prob-CM* algorithm is shown under threshold $\tau = 0.95$. We also tested on $\tau = 0.99$ or 0.75, and the results are similar.

Prog-Flex-EP and its variants: The performance results of using *Prog-Flex-EP-Dist-Basic*, *Prog-Flex-EP-Dist-Avg*, *Prog-Flex-EP-Prob-Basic*, and *Prog-Flex-EP-Prob-CM* are presented in Figure 3.14 (b). The results of *Prog-Flex-EP-Prob* algorithms are obtained with cumulative probability confidence α set to 0.95. In each test, we allow the use of only one RSS parameter in addition to all the other parameters. The program will determine which parameters to use as it runs. Figure 3.14 (b) shows that the progressive



(a) Flex-EP (b) Prog-Flex-EP

Figure 3.14. Comparison of localization performance: (a) Flex-EP-Dist and Flex-EP-Prob using parameter set $\{1, 4, 6, 16\}$ with the selection ratio $\gamma = 0.5$, and (b) Prog-Flex-EP-Dist and Prog-Flex-EP-Prob with the cumulative probability confidence $\alpha = 0.95$, using one RSS from Rectangular AP4.

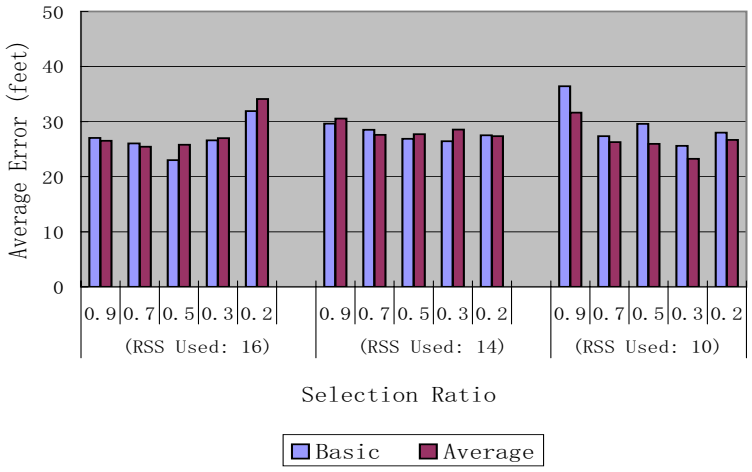


Figure 3.15. Prog-Flex-EP-Dist: performance comparison when using different selection ratio γ. The size of the parameter subset is four and only one RSS is used in each experiment.

methods can achieve localization performance equal to or better than RADAR.

Determining Threshold: Figure 3.15 shows the average error distance over all the testing points when using different RSS parameters and different settings of the selection ratio γ.

The results show that usually the best performance happens when the selection ratio γ is set to 0.5 or 0.3. This trend agrees with our analysis in Section 3.4, which indicates smaller ratios are more effective in eliminating the "noise" locations, and thus improves the result; but with a too small ratio (e.g. 0.2), the error distances increase because correct locations are thrown out prematurely. We also observed that there is

RSS used	Best Average Error	
	Prog-Flex-EP-Dist	Prog-Flex-EP-prob
16	23.009	19.719
14	26.447	23.411
10	23.255	22.152

Table 3.5. Best average errors for *Prog-Flex-EP-Dist* and *Prog-Flex-EP-Prob* on different parameter settings.

no obvious advantages to use the basic methods over the ones employing averaging.

Figure 3.16 shows the average error using *Prog-Flex-EP-Prob* with different settings of the cumulative probability confidence α and using different RSS parameters. We found the obvious trend that the larger the α the more accurate the localization results. The average error of using the individual probability threshold method is also shown in Figure 3.16 indicated by τ. The individual probability threshold means that the algorithm looks at each testing point individually. If it passes the threshold, the testing point goes to the next round. We found that in general using the *individual probability threshold* method achieves better performance than using the cumulative probability confidence. We also notice that *Prog-Flex-EP-Prob-CM* always performs better than *Prog-Flex-EP-Prob-Basic* under the same setting. Further, under the same input parameter sets, table 3.5 shows the best average error of *Prog-Flex-EP-Dist* and *Prog-Flex-EP-Prob* respectively among all the threshold settings we tested on. We observed that *Prog-Flex-EP-Prob* always performs better than *Prog-Flex-EP-Dist*.

3.6 Discussion

3.6.1 Refining Localization

In this section, we discuss how conventional localization results can be refined using *Flex-EP* algorithms. In a four-parameter subset, we further increased the number of parameters with high discriminative power to three by adding an additional RSS parameter into the parameter subset. Figure 3.17 (a) presents the corresponding error CDFs. We found that by utilizing three parameters with high discriminative power in a four-parameter subset, the localization performance is further refined and is almost

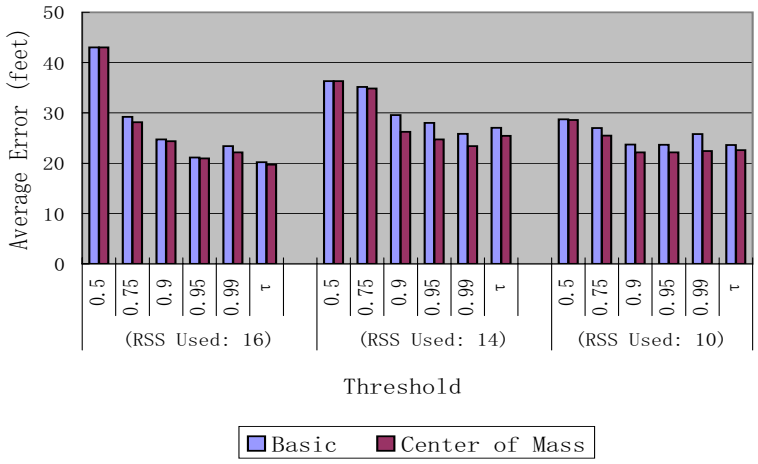
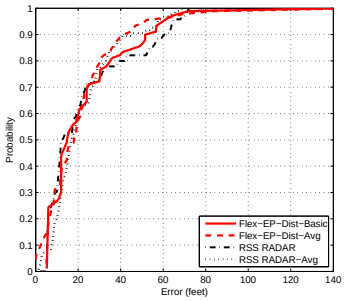


Figure 3.16. Prog-Flex-EP-Prob: performance comparison of threshold values when α is set to 0.5, 0.75, 0.9, 0.95, and 0.99 respectively and $\tau = \frac{1}{2|\widetilde{P}|}$. The size of the parameter subset is at most four and only one RSS is used in each experiment.

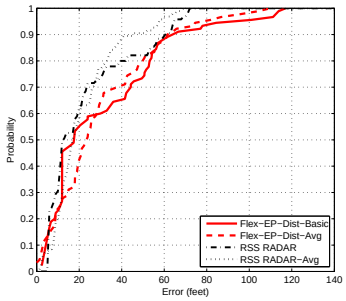
exactly the same as RADAR.

Further, we explored the parameter subset with five parameters. Figure 3.17 (b) and (c) show the localization error CDFs utilizing five parameters. The parameter subset in Figure 3.17 (b) still contains only two parameters with high discriminative power (*2.435GHz Max* and *RSS from AP4*), the same as the four-parameter case in Figure 3.12 (e), and three other parameters with low discriminative power (*temperature*, *ambient noise*, and *2.465GHz Avg*). We observed that the localization capability is about the same as in Figure 3.12 (e) for the four-parameter case. This is in line with our previous observation, adding more environmental parameters with low discriminative power does not help much in improving the localization performance.

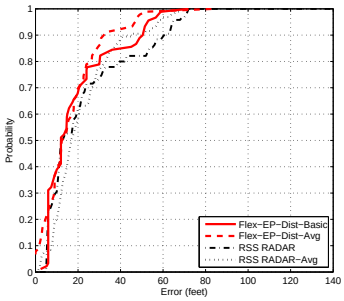
Turning to examine Figure 3.17 (c), which has three parameters (*2.435GHz Max*, *RSS from AP2*, and *RSS from AP3*) with high discriminative power in a five-parameter subset, interestingly, the localization performance has a 10% increase compared to the traditional RADAR algorithms, especially for $Flex - EP - Avg$, which gained over 20% performance improvement. In this case, only two RSS parameters are used, which means that under the assistance of other environmental parameters, only two access points are needed to achieve a better localization performance than the traditional localization algorithms employing RSS using at least four access points. This provides strong evidence that utilizing environmental properties for localization can both achieve



(a) four parameters: index 1, 5, 10, 11



(b) five parameters: index 1, 4, 5, 8, 12



(c) five parameters: index 1, 4, 5, 10, 11

Figure 3.17. Using environmental properties to refine localization results.

similar performance to the traditional approaches, as well as refine conventional localization results.

3.6.2 Comparison of *Flex-EP* and *Prog-Flex-EP*

In this section, we provide a discussion about *Flex-EP* and *Prog-Flex-EP* algorithms in terms of computational cost, parameter selection strategy, and missing sensor readings.

Computational Cost: We first look at the computational cost of the algorithms. *Flex-EP* calculates the $W(K)$ for every possible parameter combination. If we have N available parameters and want to choose a subset of K parameters to use, the computational cost is $\mathrm{C}_N^K = \frac{N!}{K!(N-K)!}$. In our case, it is C_{16}^4. Whereas for *Prog-Flex-EP*, it chooses one parameter at a time from the current available parameters, thus, the computation cost for parameter selection is at most $\mathrm{C}_N^1 \cdot \mathrm{C}_{(N-1)}^1 ... \mathrm{C}_{(N-K)}^1 = \frac{N!}{(N-K-1)!}$. Note that we say at most is because sometimes the algorithm stops before reaching the maximum K parameters if the desired confidence is achieved. *Prog-Flex-EP* refines the candidate locations and calculates W(K) based on new parameter sets at each round.

However the *Flex-EP* decides the parameter set off-line and calculates the location estimation in one round. *Flex-EP* runs faster than *Prog-Flex-EP* in reality.

Parameter Selection Strategy: Examining the parameter selection, under a given parameter size, *Flex-EP* finds the overall best parameters to use for any localization tasks. On the other hand, *Prog-Flex-EP* is a locally customized method. The parameters chosen by *Prog-Flex-EP* vary based on the environment surrounding each testing point (i.e., each user).

Table 3.6 presents a part of our running records using the *Prog-Flex-EP-Prob* algorithm as an example. It shows for a testing point which parameter is used in each round and how many points are kept as candidates after each round. We set the maximum number of rounds to 4. We found that the parameters used for different testing points vary largely. The first round is the same for every test because at that time we have no information about the environment around the testing point. Thus the algorithm chooses the parameter with the strongest discriminative power. In some cases, the algorithm stopped before the total 4 rounds as it has already refined to one candidate location, such as the testing points 46, 49, and 50. *Prog-Flex-EP* takes the advantage of the idea that the optimal parameter set is location-dependent, thus giving it the potential to outperform *Flex-EP*, which uses a globally optimized parameter subset.

However, the "greedy" nature of *Prog-Flex-EP* drags down its performance. A "bad" parameter may be chosen at an early stage, which results in excluding the "good" location candidates prematurely and leads to a larger localization error. Looking at testing point 49 in Table 3.6, it only uses 3 parameters and has already achieved the sufficient confidence; however, the error distance is as large as 114 feet. This is an example where "good" location candidates are eliminated too early. Our experimental results, in Figure 3.18 show that *Prog-Flex-EP* and *Flex-EP* perform quantitatively the same.

Dealing with Missing Environmental Readings: *Prog-Flex-EP* can deal with the situation when not all environmental properties are available at all locations and times. *Prog-Flex-EP* is a sequential localization algorithm: it does not require to make the localization decision in one shot with all the environmental readings being responded

Test #		Round				Error (feet)	
		1	2	3	4	-Dist	-Avg
46	Param used	16	2	5		12.0	12.0
	# of pts left	17	6	1			
47	Param used	16	8	5	1	36.0	39.698
	# of pts left	14	13	6	2		
48	Param used	16	1	3	6	12.0	12.0
	# ofpts left	19	4	2	1		
49	Param used	16	8	5		114.0	114.0
	# of pts left	16	13	1			
50	Param used	16	2	6		6.0	6.0
	# of pts left	23	7	1			

Table 3.6. Sample testing output from *Prog-Flex-EP-Prob* test, using parameter set {1,2,3,5,6,7,8,16}, $\tau = 0.95$

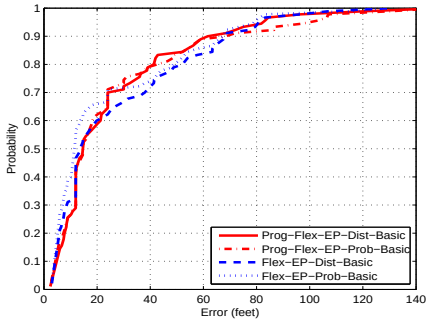


Figure 3.18. Comparison of *Flex-EP-basic* using parameter set {1,3,6,14} and *Prog-Flex-EP-basic* with 4 parameters chosen from parameter set {1,2,3,5,6,7,8,14} at each testing point.

by the user. Instead, it can be used in a *challenge-response* based manner for location refinement. Such an algorithm first asks the user, "tell me the temperature at your location", and based on the user's answer, the algorithm localizes the user within some regions and then chooses the next parameter to challenge the user. Based on the subsequent answers from the user, the localization results are further refined.

By using this approach, *Prog-Flex-EP* is flexible, especially when dealing with the situation when some environmental properties are not available at specific locations and times. This lack of availability can happen to both the sensor network side and the user's side: (1) On the sensor's side, it is possible that at a particular region, some sensor readings are unavailable or the sparse network density in that region can not provide sufficient readings, e.g., RSS is too weak when the sensor is far away from the

AP. *Prog-Flex-EP* can avoid using such insufficient parameters, and will instead choose other parameters to ask the user. (2) On the user's side, if the user cannot provide the parameter requested, the algorithm is flexible to accept other parameters that the user can provide to continue the localization process.

Flex-EP can be modified to deal with the situation where the user cannot provide all the parameters by applying SCWM after getting the user's vector of available parameters. However, in the case of the sensor's side, it is possible that some parameters are not available in one region and some other parameters are not available in another region, In this case, *Flex-EP* has to totally discard the use of all these deficient parameters, which will affect the localization performance.

3.7 Related Work

In this section, we first discuss research efforts in using spatio-temporal information in wireless sensor networks (WSN). Then we overview the active research in wireless localization and describe the work that are mostly related to ours.

By utilizing the radio on sensor nodes, it is possible to invert the role of sensor networks, and allow sensor nodes to actuate the environment. [28] utilized sensor networks in an inverted fashion to facilitate new forms of access control that are based on whether a user is located at the right place at the right time. Moreover, [**?**] pointed out that sensor observations are highly correlated in the spatial domain. They proposed a theoretical framework to capture the spatial and temporal correlations in WSN and enable the development of efficient communication protocols in WSN utilizing these information. In this work, we explore the possibility of utilizing the physical phenomena monitored by WSN to assist in wireless localization and position verification.

Localization of nodes in WSN has become increasingly important. Localization techniques can be categorized along several dimensions. [**?**] used infrared methods and [**?**, **?**] employed ultrasound as the basis for a localization infrastructure. On the other hand, in spite of its several meter-level accuracy, using RSS [?, ?, 29] is attractive because it can reuse the existing wireless infrastructure. Dealing with ranging methodologies,

range-based algorithms involve distance estimation to landmarks using the measurement of various physical properties like RSS [?, ?], Time Of Arrival (TOA) [**?**] and Time Difference Of Arrival (TDOA) [?]. Range-free algorithms [**?**, **?**] use coarser metrics to place bounds on candidate positions. [**?**] combines the range-based and range-free algorithms. Another method of classifying localization algorithms involves examining the strategy used to map a node to a location. Lateration approaches [**?**, ?, ?, **?**], use distances to landmarks, while angulation uses the angles from landmarks. Scene matching (or fingerprint matching) strategies [**?**, **?**, ?, ?] use a function that maps observed radio properties to locations on a pre-constructed radio map or database. Finally, another dimension of classification extends to aggregate [?, **?**, **?**] or singular algorithms.

The same type of physical properties is required to be used in each of the above methods to ensure the appropriate functioning of the mechanism. Our work is unique in that our localization approach is generic, i.e. we are not restricted to examining a single type of physical property. The closest works to our work are [?, **?**]. [?] developed a localization mechanism measuring the minimum Euclidean distance in the signal space, and only deals with the physical property of RSS. [?] proposed a GSM signal strength fingerprinting-based localization system to determine the current floor of a user. It addressed the problem that certain physical sources may not contribute to localization accuracy by developing a set of feature selection techniques. However, these feature selection techniques did not track the performance of each possible combination in parameter subsets and might contain "bad" physical sources to start with. Also, [?] only deals with one type of physical property, signal strength. By handling all kinds of physical properties, our work is broader than [?, ?], and our SCWM algorithm for parameter selection is more general than the feature selection approaches in [?]. In addition, our method is novel in that we utilize an existing sensor network to assist in localization, rather than requiring the deployment of a localization infrastructure or additional access points (or landmarks) in the area of interest.

3.8 Conclusion

In this work, we proposed to use the inherent spatial variability in physical phenomena recorded by sensor networks to support wireless localization and position verification. We formulated the problem using a theoretical measurement model to quantify the localizing capability of environmental properties. For parameter evaluation and selection, we proposed a scheme to evaluate the environmental parameters' ability to capture the physical variability, the Sptio-Correlation Weighting Method (SCWM), which can find the optimal parameter subset with the highest discriminative power for localization under a given size of the parameter subset. Moreover, we developed a spectrum of algorithms to perform localization and position verification utilizing parameter subsets obtained from SCWM.

To evaluate the generality of our approach and the effectiveness of SCWM, we conducted experiments in a real office building by collecting various environmental parameters including temperature, humidity, ambient noise, spectrum energy, and RSS over one hundred locations. We found that choosing two environmental parameters containing high discriminative power is enough to produce comparable performance to the traditional localization approaches employing RSS with at least four access points. By increasing the number of parameters with high discriminative power in a subset, we can further refine the localization accuracy and obtain better performance than conventional localization results. Thus, our experimental results provide strong evidence of the feasibility of utilizing environmental properties to assist in localization and the effectiveness of our approach by using SCWM and environmental parameter based algorithms. Note that there is a trade off between the localization performance, which is related to the size of the parameter subset, and the computational cost. SCWM can help select appropriate parameter subsets which achieve the localization performance based on application requirements.

3.9 Appendix

Theorem: There is no continuous injective function $f : \mathbb{R}^2 \to \mathbb{R}$

Proof: Consider any line l in $\mathbb{R}^2$. It is trivially simply-connected, and so by the Simply-Connected Theorem, its mapping under f is a simply-connected subset of $\mathbb{R}$. But since f is injective, this means that it maps to some non-zero interval on $\mathbb{R}$. Let z be a non-boundary point along that interval. Then there exist points p_1 and p_2 on l such that $f(p_1) < z < f(p_2)$. Now consider any point p_3 in $\mathbb{R}^2$ not on l. Assume WLOG that $f(p_3) \geq z$, and let m be the line determined by p_1 and p_3. By the Intermediate Value Theorem, $[f(p_1), f(p_3)]$ is contained in the image set $f(m)$. So, there exists a point q on m such that $f(q) = z$. But some points on l also map to z, and the only common point of l and m is p_1. So we have found two distinct points that both map to z under f. Thus f is not injective, which is a contradiction.

Chapter 4

Actuating the Environment to Verify Location Claims in LBS

4.1 Introduction

For large scale LBS that operate on cellular networks, location verification faces several challenges. First, for cellular networks, the service area is very large, which makes it impractical to use fingerprinting methods because fingerprinting methods rely on an offline training procedure [27]. Second, the location claimants (targets or requesters in LBS) are not trusted, since they are the ones to be verified. We cannot rely on the readings that the claimants report, such as received signal strength, because we should assume they have the intention and ability to modify the data. We thus propose a new location verification method, which we call Key Distribution-based Location Verification (KDLV). KDLV makes use of an auxiliary network consisting of high a density of transmitters. The transmitters in the auxiliary network can be access points in neighboring WiFi networks or a collection of transponders. We note that the deployment of auxiliary networks to assist cellular networks is a topic that has received wide spread attention by the research community [30–34]. In this chapter, we first present our Key Distribution-based Location Verification model in Section 4.2. Next, in Section 4.3 we present our method of analyzing the performance of KDLV based on statistical and signal propagation models. Particularly, we investigate the relationship between the transmitter density and the location verification accuracy. Then in Section 4.4, we show our simulation methodology and results. Last, we conclude this work in Section 4.5.

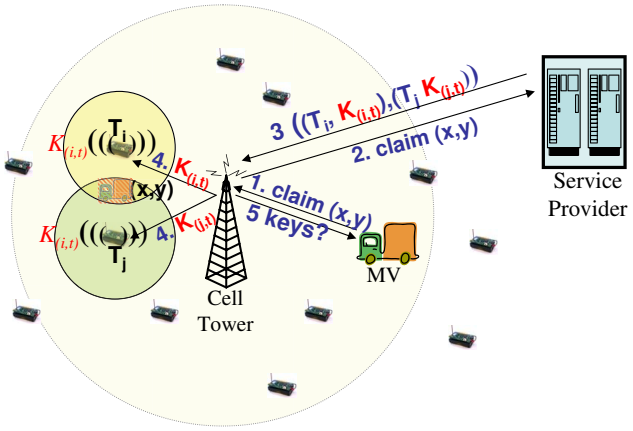


Figure 4.1. Key Distribution-base Location Verification with keys assigned on demand.

4.2 Key Distribution-based Location Verification Method

We assume the transponders in the auxiliary network have the ability to communicate with the network operator and that the locations of transmitters are known by the network. Each transmitter in the auxiliary network broadcasts a time-varying verification key at regular time intervals. The keys are properly scheduled and evolve with time. If a location claimer is at the location and the time it claims, it should be able to receive the keys transmitted by the transmitters whose transmission range covers the claimed location. We verify a location claim by verifying the keys that the location claimer receives.

In order to reduce the energy consumption of the transmitters, an alternative on-demand scheme can be used, in which the transmitters only broadcast the keys when they are needed to verify a claimed location that is close to them. Figure 4.1 depicts a typical KDLV procedure with the on-demand scheme. A location claimant, i.e. the MV in the figure, claims to be at location (x, y), which is marked by the shaded van. This location claim is routed to the Service Provider (SP) through cell towers. The SP is responsible for choosing the transmitters whose transmission range cover the claimed location and for assigning the keys to the transmitters. The infrastructure then sends the challenge to the location claimer "What keys did you receive?". The location claimant replies with a list of keys it is able to hear.

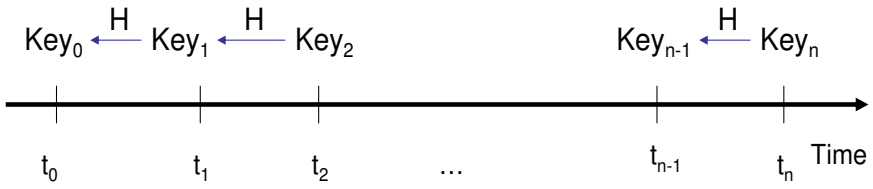


Figure 4.2. One-way chain of encryption keys

In both a constant transmitting scheme or an on-demand KDLV scheme, the keys must be changed over time, so that an entity who got the keys in the area at a previous time would not be able to reuse these keys at a later time when it has moved out of that area. However, there is significant overhead associated with frequent updates of keys and in order to reduce this overhead, we make use of a chain of one-way hash functions to generate and store keys. A one-way key chain $(Key_0; ...; Key_n)$ is a collection of values such that each key Key_i (except the last value Key_n) is a one-way function of the next value Key_{i+1}. In particular, we have that $Key_i = H(Key_{i+1})$ for $0 \leq i < N$. Here H is a one-way function, and is often selected as a cryptographic hash function. For setup of the one-way chain, the generator chooses at random the root or seed of the chain, i.e., the value Key_n, and derives all previous values Key_i by iteratively applying the hash function H as described above. By employing the hash chain, the service providers SP need only send the anchor seed Key_n and the times at which the transmitters should change keys in the case that the transmitters constantly broadcast keys. In the case of an on-demand scheme, the working key is changed when a command from the SP is received. When the keys are used up, the SP will repeat the process.

Given a location claim that is to be verified, the SP is able to figure out what keys the location claimant should be able to hear at the claimed location, based on the transmitters' transmission powers and the underlying propagation model. However, there can be cases where the location claimant missed some keys even if it is at the claimed location, or reversely, an entity can still get the verifying keys even it is certain distance away from the claimed location. We study the relationship between the transmitter deployment and verification accuracy and provide performance analysis in the next section.

Table 4.1. Notations used in KDLV

Notation	Description
S	The overall area of interest (e.g. a cell in mobile network), supposed to be a circular area.
T	A transmitter.
N	Number of transmitters in the area S.
r	Radius of S.
P_T	The transmission power of each transmitter T (assume to be the same for all transmitters).
ρ	The probability of receiving the signal from a transmitter T.
$P_{thre}(\rho)$	The receive power threshold needed for a receiver to receive from T with probability ρ.
$d(P_T, \rho)$	The distance that, with probability ρ, the receiver can receive from T who transmits with power P_T. The value of d depends on $P_{thre}(\rho)$.
A	The circle area with radius d, centered at the claimed location.
$N(A)$	The number of transmitters in A.
λ	Intensity parameter of the spatial Poisson process.
α	The confidence that the deployment satisfies the desired accuracy. $0 < \alpha < 1$.
k	The least number of keys must be received at any location for a certain level of location accuracy.
p_k	The probability of getting at least k keys at any location.
$eArea(k, \alpha)$	The size of the area a location claimer can be located within with k keys, with confidence α.
$eMag$	$= \sqrt{eArea}$, magnitude of the estimation.

4.3 Analysis of Key Distribution-based Location Verification

In this section, we analyze the performance of our Key Distribution-based Location Verification Method. We investigate the relationship between the transmitter density and the location verification accuracy. Our analysis can serve as a guideline for service providers to deploy the transmitters that satisfy the location verification requirements.

4.3.1 Analysis Overview

The notations used in this chapter are summarized in Table 4.1. In this analysis, we focus on studying the transmitter deployment required in order to keep the error distance within a certain level with confidence α. We assume all the transmitters

are identical in terms of functionality and the same power settings. We model the deployment of the transmitters as a spatial Poisson process.

A **Spatial Poisson Process** is a random set of points in R^2 such that in any measurable subset of R^2 the number of points is distributed as a Poisson random variable with parameter λ. We assume the transmitters are distributed over the area that the MLS service covers. To investigate the impact of the transmitter density, we analyze a unit area, for example, a cell in the mobile network and use S to denote this area. Further, we assume S to be circular with radius r. Let $\mathbb{A}$ be the family of subsets of S. For all $A \in \mathbb{A}$, let $|A|$ denote the area of A and $N(A)$ denote the number of transmitters in A.

By modeling the deployment of transmitters as a spatial Poisson process, the distribution of the transmitters follows the Poisson Postulates [35]:

1. Only nonnegative integers are assumed by $N(A)$ and the probability $P(N(A) > 0) \in (0, 1)$ if $|A| > 0$.

2. The probability distribution of $N|A|$ depends on the set A only through its size $|A|$.

3. If $A_1, A_2, ..., A_n$ are disjoint regions, then $N(A_1), N(A_2), ..., N(A_n)$ are independent random variables and

$$N(A_1 \cup A_2 \cup ... \cup A_n) = N(A_1) + N(A_2) + \ + N(A_n).$$

4. $$\lim_{|A| \to 0} \frac{N(A) \geq 1}{N(A) = 1} = 1.$$

Further, we have

$$P(N(A) = n) = \frac{e^{-\lambda|A|}(\lambda|A|)^n}{n!}. \tag{4.1}$$

The expected number of transmitters in A is thus

$$E[N(A)] = \lambda|A|. \tag{4.2}$$

Given a set of location-keys that a claimant receives at its current location, the SP will map the keys into the transmitters that were transmitting the corresponding keys at that moment, and use this information to decide whether the claimant is in the area where these transmitters' transmission ranges overlap. Hence, the claimant's location is estimated as the overlap area, rather than as a point. We define the size of the overlap area as $eArea$, which indicates how precisely we can locate the user. We further define $eMag$ as the square root of the $eArea$, which represents the magnitude of the precision of location estimation. In order to determine the impact of transmitter density $N(S)$, i.e., the number of transmitters in S, on the accuracy of the location verification $eMag$, our study is performed in the following two phases. Note, for simplicity, we use N to denote $N(S)$ for the rest of our analysis.

1. $k - N$ relationship: In order to ensure a claimant receives at least k keys at any location, with confidence α, how many transmitters are required to be deployed in S?

2. $k - eMag$ relationship: If a claimant submits k valid keys, how accurate we can estimate its position in terms of $eMag$?

4.3.2 $k - N$ Relationship Study

When a user resides at a location, shown as a star in Figure 4.3, we consider a circular area A centered at the claimant's location with radius d. The radius of the circle d is the distance at which a receiver can receive packets from a transmitter with a given probability ρ, which is based on the propagation model and the transmitters' transmission power. We first calculate the probability of having n transmitters located inside this circle. These transmitters can be heard with a probability ρ. For instance, there are three transmitters T_1, T_2, T_3 in A as shown in Figure 4.3. The transmitters' transmission range is usually irregular in shape [36], e.g., the transmission range of the transmitter T_3 is outlined by the dash-dot lines to illustrate its irregularity. Intuitively, the higher the density of transmitters, the higher the probability of receiving at least k keys at a location. We use a confidence level α to balance the trade off between

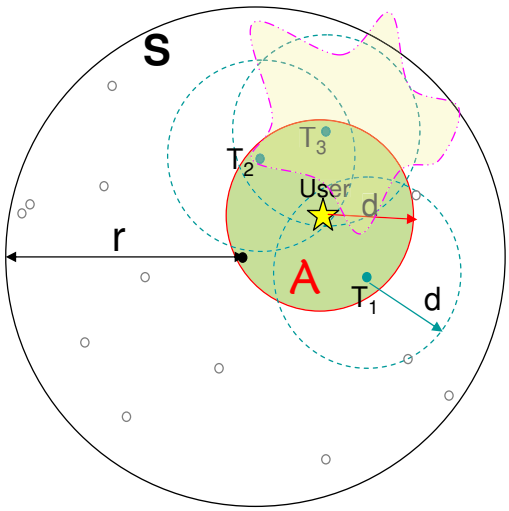


Figure 4.3. Illustration

the requirement of needing a minimum number of received keys and the deployment density of transmitters. We then compute the minimum number of transmitters N in S required to ensure at least k keys to be received with confidence α by using the following steps:

1) Pick a ρ: ρ represents the probability of receiving the transmission signal from a transmitter T.

2) $\rho \rightarrow P_{thre}$: For each ρ, finding the corresponding receive power threshold P_{thre}. $P_{thre}(\rho)$ is the threshold for a receiver to receive from a transmitter T with probability ρ. Figure 4.4 shows real data using a CC1100 radio that was measured in our lab for the probability of losing a packet due to ambient effects at different received signal strengths. In general, the weaker the receiving power, the more chance the packet will be lost. Making use of these data, we can find the received signal strength threshold P_{thre} for a certain receiving percentage $\rho = 1 -$ (miss percentage).

3) $(P_{thre}, P_T) \rightarrow d$: Calculating the distance d that a receiver can receive signals from a transmitter T with probability ρ, based on a generic log-distance path loss model [37]. This distance is related to the transmission power P_T of the transmitter. Thus, d is a function of ρ and P_T:

$$PL(d)[dB] = P_0 + 10\gamma \log_{10}(\frac{d}{d_0}) + X_\sigma \tag{4.3}$$

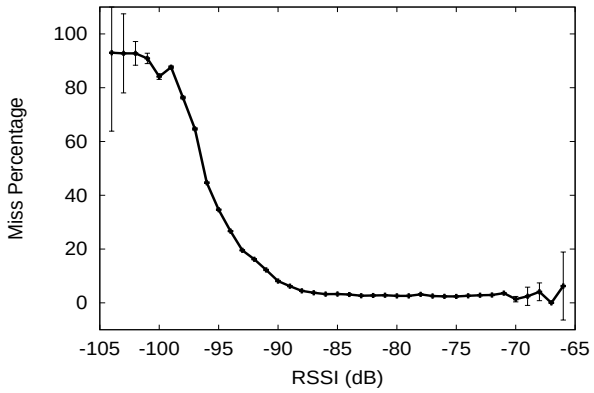


Figure 4.4. The package loss percentage verses RSSI in a typical environment.

where PL is the path loss measured in Decibels (dB), d is the length of the path, i.e. the distance to the transmitter, $P_0 = 20\log_{10}(4\pi d_0/\tau)$ is the path loss at the reference distance d_0 in dB, d_0 is the reference distance, γ is the path loss distance exponent, and X_σ is a random variable with zero mean, reflecting the attenuation caused by flat fading. In the case of no fading, this variable is 0.

The received power at a location that is d meters from a transmitter is then $P_T - PL(d)$. In order to receive a transmitter's signal with probability no less than ρ, the following must hold:

$$P_T - PL(d) \geq P_{thre}(\rho). \tag{4.4}$$

Using equations (4.3) and (4.4), we are able to get an upper bound of d for a given P_T and ρ:

$$d(P_T, \rho) = d_0 10^{\frac{P_T - P_{thre}(\rho) - P_0}{10\gamma}}. \tag{4.5}$$

4) $(N, r, d) \rightarrow \lambda|A|$: For a given total number of transmitters N in the whole area S (which we shall assume has radius r), based on the spatial Poisson distribution model, the expected number of transmitters in A can be obtained as $\lambda|A|$. According to the properties of a spatial Poisson process, the number of transmitters in an area A follows the Poisson distribution with intensity parameter $\lambda|A|$, where

$$\lambda|A| = N\frac{d^2}{r^2}. \tag{4.6}$$

5) $\lambda|A| \rightarrow p_k$: Calculating the probability p_k of getting at least k location-keys at any location. From the last step, we have a $\lambda|A|$ and N relationship, so we can get p_k as a function of N, i.e. $p_k(N)$. Since we assume, for the n transmitters, the probability of receiving from each of them is ρ, and the n transmitters are independent, the probability of receiving k keys from the n transmitters, denoted by $q_{k,n}$ follows a binomial distribution:

$$q_{k,n} = \begin{pmatrix} n \\ k \end{pmatrix} \rho^k (1-\rho)^{(n-k)}. \tag{4.7}$$

The probability of receiving at least k keys from the n transmitters in A is the sum of the probability of receiving $k, k+1, ..., n$ keys, and then sum over all the possible n, which are all the integer between k and N.

$$p_k = \sum_{n=k}^{N} (p(N(A) = n) Q_{k,n}) \tag{4.8}$$

where

$$Q_{k,n} = \sum_{j=k}^{n} q_{j,n}. \tag{4.9}$$

We have derived the relation between p_k and $\lambda|A|$ through Equations (4.8), (4.9) and (4.1). Using equation (4.6), we get the relationship between p_k and N.

Finally, we have $p_k(N) > \alpha$. Solving this discloses the relationship between k and N. The above procedure obtains the estimated relationship between N and k. We note that the relationship between k and N is a function of ρ and P_T. We will study the effects of these parameters in the next section.

4.3.3 $k - eMag$ Relationship Study

We now examine the problem: given the k keys reported by the claimant, how large is the location verification error in terms of $eMag$?

Since the deployment of the transmitters follows a spatial Poisson process, the $N(A)$ transmitters are uniformly distributed in A, which is a direct conclusion from the fact that for any $B \subset A$, $P(N(B) = 1|N(A) = 1) = \frac{|B|}{|A|}$. Further, we assume the $N(A)$

transmitters in the circle A are received with the same probability ρ. Thus, the k location keys received by the user can be modeled as uniformly distributed in the circular area A. Note that this is an approximation used in our model, since in reality, the closer a transmitter is to the center of A, the greater the chance it will be heard by a user located at the center of A. This approximation is necessary in order to make the technical analysis of the tradeoffs tractable.

In our study, we generated k locations (following a uniform distribution) in A. These are the locations of the transmitters that the claimant receives keys from. For each set of k locations, we use Monte Carlo sampling to determine the area where a user can also receive the same k keys with probability greater than α.

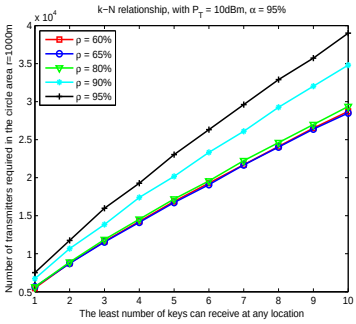
At each sampling location, the probability of receiving the key sent by the i^{th} transmitter (out of k transmitters) is Pr_i and is calculated based on the distance between this sampling location and the transmitter using equation (3). Since each transmitter operates independently, and is assumed to be deployed independent of other transmitters, the probability of receiving from all k transmitters, Pr, is the product of these probabilities: $Pr = Pr_1 \times Pr_2 \times ... \times Pr_k$. The localization area of the user is:

$$eArea = \text{(Sampling area)}\frac{\text{(no. of sampling locations with}Pr > \alpha)}{\text{no. of sampling locations}}.$$

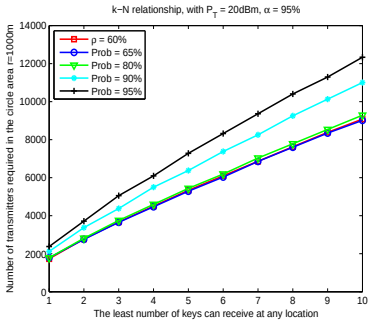
We repeatedly generated k transmitter locations, and calculated $eMag = \sqrt{eArea}$ for each of these deployments. The distribution of $eMag$s gives insight into how accurately k keys can verify a location claim. When we vary the number of keys, we discover how $eMag$ evolves with the number of keys.

Table 4.2. The values of P_{thre} v.s. ρ.

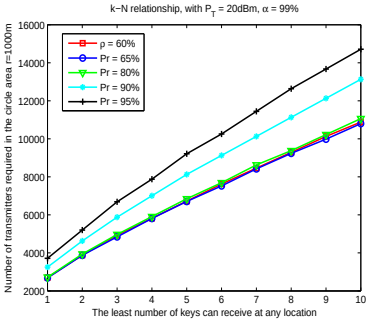
$\rho(\%)$	60	65	80	90	95
$P_{thre}(dB)$	-95.63	-95	-93	-90.5	-89



(a)$P_T = 10dBm, \alpha = 95\%$



(b)$P_T = 20dBm, \alpha = 95\%$



(c)$P_T = 20dBm, \alpha = 99\%$

Figure 4.5. k-N relationship, r = 1000m

4.4 Simulation

4.4.1 Simulation Methodology

In our simulation, we have chosen $\rho = 60\%, 65\%, 80\%, 90\%, 95\%$. The corresponding threshold powers P_{thre} according to the empirical result [38] are shown in Table 4.2.

We chose typical outdoor environmental parameters [37] for the propagation model: $\tau = 0.06m$, $d_0 = 1m$, $\gamma = 4$, and for simplicity, chose X_σ as 0. For P_T, we choose the typical powers for active transponder tags (e.g. active RFID tags), $P_T = 20$ dBm and 10dBm.

4.4.2 Results for the $k - N$ Relationship

In our simulations, we consider the circular region S with radius $r = 1000m$, and assume the transmitters follow a spatial Poisson distribution. We calculated the minimum number of transmitters that must be deployed in order to ensure that a user at any location is able to receive at least k keys with confidence α. k varies from 1 to 10. Figure 4.5 (a) shows the results of our simulation when the transmitters transmission power P_T is 10dBm, and the confidence is 95%. The different curves correspond to different ρ. Ideally, the true value for N should not change with ρ, but because there is approximation in our method and different ρ result from this approximation, these lines give different values for N.

We observed that when $k = 1$, $\rho = 60\%$ requires at least 5480 transmitters, and 5653 when $\rho = 65\%$, and 7520 when $\rho = 95\%$. When k increases, the number of required transmitters increases as well. When considering these numbers, recall that the region of interest has a 1 kilometer radius. When $k = 3$ the estimated N ranges from 11542 when $\rho = 65\%$ to 15980 when $\rho = 95\%$. When $k = 10$, the number becomes quite large, with range from 28502 to 39009. Figure 4.5 (b) shows that when we increase the transmission power of each transmitter to 20dBm, the required number of transmitters drops. When $k = 1$, N ranges from 1733 to 2378; and when $k = 10$, N ranges from 9013 to 12336. If we increase the confidence to 99% (but keep $P_T = 20dBm$), then more transmitters will be needed, as shown in Figure 4.5 (c). Overall, we can see that for any fixed settings of P_T, α and ρ, k and N follows a nearly linear relationship. In addition, the smaller the transmission power, the larger the number of transmitters that are needed, and a higher confidence requires more transmitters to achieve this confidence– all of which are inline with our intuition.

Discussion: In our method of evaluation, we made some approximations.

1. We only considered the transmitters within the ρ circle (i.e. A), and ignored the situations that the user may also receive keys from the transmitters outside of the ρ circle. For example, when we consider $\rho = 65\%$ circle, there may be transmitters that are just outside of the circle and can be received with a 64% likelihood. By

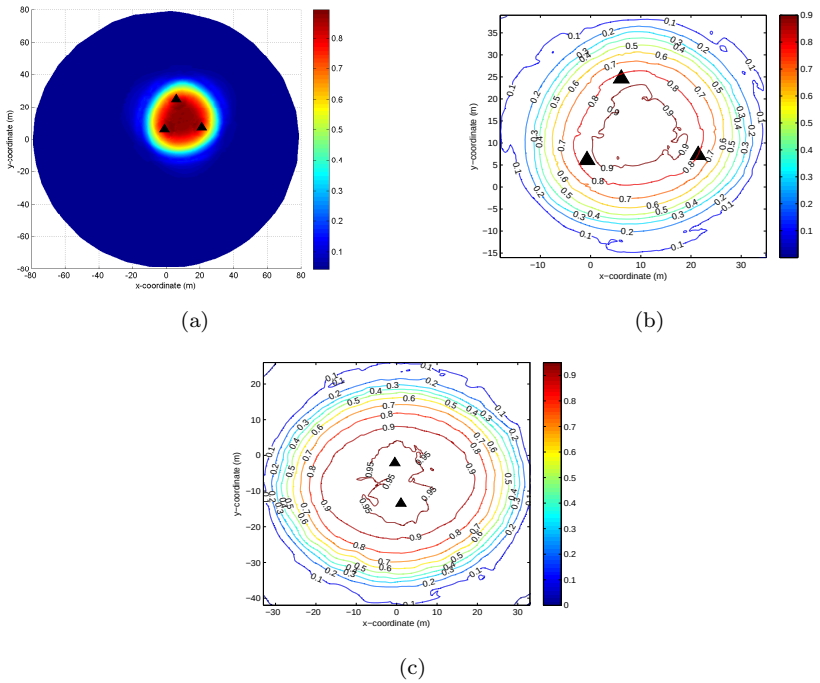


Figure 4.6. Probability map of receiving signals from all the transmitters (marked as triangles). (a) and (b) are of the same development where there were 3 transmitters $P_T = 10dBm, \rho = 60\%$. (c) corresponds to two transmitters, $P_T = 10dBm, \rho = 95\%$.

ignoring such situations, the number of keys a user can receive in reality is larger than what we estimated, and thus the required numbers of transmitters N we obtained are in fact conservative results. Notice that the larger ρ is, the more transmitters are ignored, and thus a larger N results. In Figure 4.5, the lines corresponding to bigger ρ are above the lines corresponding to smaller ρ, which is inline with the analysis.

2. Within the circular area A, all the transmitters have equal probability (ρ) to be received. In reality, it should be that the closer the transmitter is to the user, the larger the probability it will be heard. Again, such an approximation only makes the evaluation of N larger than necessary, and hence our conservative estimate of N is safe to use in guiding real application decisions.

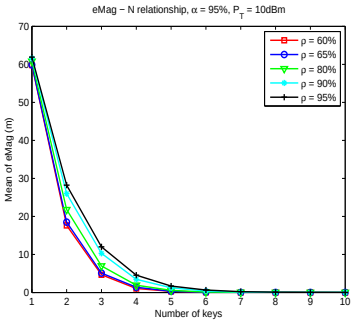
Thus in Figure 4.5 each curve corresponding to a different ρ gives an upper bound for N. Hence, the lowest edge of these curves is also an upper bound of N and is more strict. We only need to look at the lowest curves in each figure.

4.4.3 Results of $k - eMag$ Relationship

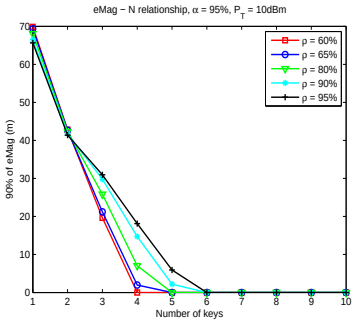
For each k and ρ, we ran 10000 trials. In each trial we randomly generated k transmitter locations within the ρ-circle, and sample 10000 locations within the sampling area to obtain $eMag$. Instead of using S as the sampling area, we sampled the circular area with radius $d(P_T, \rho) + d(P_T, 0.07)$, where 0.07 corresponds to the lowest receiving percentage in the experiments in Figure 4.4. This is because the locations farther than that distance are not able to hear from all of the transmitters with a probability greater than α.

Figure 4.6 presents typical probability maps for receiving signals from all transmitters. Figure 4.6 (a) and (b) are from the same trial where there are 3 transmitters with transmission power $P_T = 10dBm$. Figure 4.6 (b) is a partially enlarged graph of Figure 4.6 (a) which shows more detail. The most inner contour outlines the area with $Pr > 90\%$, which is approximately within a 20 meters by 20 meters square. In this trial, however, there are only 25 out of 10000 sample points with $Pr > 95\%$, which is too small to show in the figure. In Figure 4.6 (c), there are only two transmitters, thus the area that is able to hear from both the transmitters with probability greater than confidence α is increased. Outlined by the most inner contour is the area with $P_r \geq 95\%$, which is the $eArea$ when $\alpha = 95\%$.

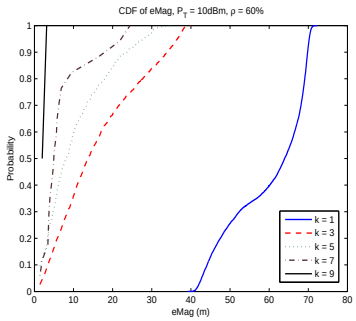
Figure 4.7(a) shows the median of the $eMag$ and (b) shows the 90% of $eMag$ among our 10000 trials in the simulation, with k varying from 1 to 10. Both of the two graphs show similar trends that $eMag$ drops dramatically as k increases. The curve with higher ρ tends to have larger $eMag$. This is because higher ρ corresponds to a smaller ρ-circle. When we deploy the same number of keys, the keys in the smaller ρ-circle tend to be closer to each other, thus there is more chance at a given location to hear from all the k transmitters. Figure 4.7 (c) shows the Cumulation Distribution Function (CDF) of $eMag$s for $\rho = 60\%$, where we removed those deployment trials where no sample location was able to hear from all the transmitters with confidence α. It shows clearly that when $k = 1$, the $eMag$s ranges from 39 meters to 72meters. But $eMag$s drop rapidly as k increases. When $k = 3$, the $eMag$ ranges from 1 meter to 39 meters; when $k = 9$, there are only two trials with $eMag > 0$, and the largest $eMag$ is 3.1meters.



(a)$P_T = 10dBm, \alpha = 95\%$, median of eMag



(b)$P_T = 10dBm, \alpha = 95\%$ 90% eMag

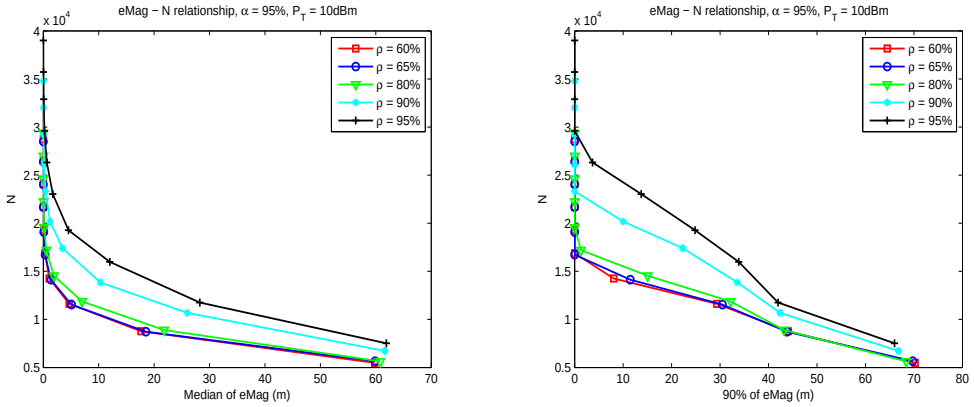


(c) eMeg CDF, $\rho = 60\%$

Figure 4.7. k-eMag relationship, r = 1000m

4.4.4 Results for the $eMag - N$ Relationship

Combining the results from $k-N$ relationship and $k-eMag$ relationship, we are able to infer how the density of transmitters affects the resulting location verification accuracy, and further the required number of transmitters we should deploy if we want to keep the accuracy at a certain level. In Figure 4.8, the transmission power P_T is set to be 10dBm, and our confidence level is set to be 95%, (a) shows the relationship between the median of the $eMag$s and the total number of transmitters required in the circular area with radius $r = 1000m$, and (b) shows the relationship between the 90% of the $eMag$s and N. As stated in the discussion part of Section 4.4.2, every curve in the figure corresponding to different ρ is an upper bound for N, so the lowest curve is also a upper bound. In Figure 4.8 (a) and (b), the curves correspond to $\rho = 60\%$ and $\rho = 65\%$ are almost identical and are the lowest, and the curves corresponds to $\rho = 80\%$ are also very close to them. These are the curves which give the best estimate. Our result shows that in order to keep the median of $eMag$ less than 17.5m, 8715 transmitters are required to



(a)$P_T = 10dBm, \alpha = 95\%$, median of $eMag$ (b)$P_T = 10dBm, \alpha = 95\%90\%$ $eMag$

Figure 4.8. eMag-N relationship, r = 1000m

be deployed in S, which corresponds to using 2 location keys for location verification. If we increase N to 11,542, then the median of $eMag$ is within 5 meters. Examining the 90% eMag in Figure 4.8(b), we see that 5653 transmitters will ensure that 90% of time, $eMag$ is less than 70 meters. However, 8715 transmitters will guarantee that 90% of $eMag$ is less than 42.3 meters. This corresponds to an average density of $\pi r^2/N = 360.48m^2$ per transmitter. If the application requires higher accuracy, then 11,542 transmitters keeps the 90% $eMag$ within 20 meters, and 14,133 transmitters makes the 90% $eMag$ level less than 2 meters.

4.5 Conclusion

Our proposed Key Distribution-based Location Verification method to improve the trustworthiness of location information takes advantage of auxiliary networks enabled by the increasingly wide deployment of wireless technologies and the fact that there will be a high density of access points or other wireless transmitters in the future. We first proposed a constant key transmitting scheme, then in order to reduce the transmission cost, we proposed an on-demand KDLV scheme. To validate our approach, we derived an analytical model for our location verification schemes to study the relationship between the number of nodes in the auxiliary network and the number of required "location-security" keys received by users, which are used to verify position claims. Our

extensive simulations results show that our proposed approach is effective and provide useful insights about how to utilize auxiliary networks to facilitate trustworthy mobile services.

Chapter 5

A Noninteractive Method to Enforce LBS Policies

5.1 Introduction

The problem of localization and location verification can be bypassed in some LBS applications by properly designed architectures. In this chapter, we describe the use of inverted sensor networks (i.e. a sensor network that is used as a network of transponders) to support spatio-temporal access control (STAC). In the basic inverted sensor network scheme, we assume that sensor networks are deployed in a regular (lattice) pattern, and that the sensors employ constant power levels when transmitting keys. This leads to a coverage issue, and further raises the issue of how precisely we may cover a specified spatio-temporal region Ω using the basic configuration. Since the general sensor network will not be deployed in a regular pattern, and since sensor nodes can employ variable power levels across the network, we next examine the issue of adjusting the coverage region to support a spatial region. Finally, we examine issues related to key deployment/management and the frequency of key announcement in order to support a desired time resolution for the region Ω.

5.2 Overview of Inverted Sensor Networks

Traditionally, wireless sensor networks have been used for a variety of monitoring applications, ranging from military sensing to industry automation and traffic control. A common feature to all of these different sensor applications is that they *pull* data from the environment, i.e. they collect data and route this data (or process locally) for external applications to utilize. This pull-oriented formulation is the original, intuitive usage of sensors nodes. However, since sensor nodes consist of radios that they use for

communication, it is also possible to turn the the role of the sensor node upside down and make the sensor *push* information into the environment.

This notion of an *inverted sensor network* is contrary to the traditional application of sensor networks where the sensors themselves merely gather data that another device can use to actuate and make decisions with. In the inverted model, however, the environment is changed using the radio, and a new information field is created in RF space that may be monitored using other devices. Due to the low-power nature of sensor node transmissions, the information injected into the environment by a sensor node is inherently localized.

The localized effect that a sensor node can have on the RF environment makes it possible to support access control based on the spatio-temporal location of a node that monitors the environment. Spatio-temporal access control allows for objects to be accessed only if the accessing entity is in the right place at the right time. STAC may thus be supported by having the environment locally convey the cryptographic information (keys) needed by the entity in order to access enciphered objects. To achieve STAC, then, a sufficiently dense network of low-power radios, as exists in traditional sensor nodes, is needed, and each sensor node can transmit a time-varying schedule of assigned cryptographic information in support of STAC.

While conventional access control is based on the user's identity, there are many scenarios where identity-based authentication is not only inconvenient but also unnecessary, and instead user spatio-temporal contexts are more desirable for basing access control upon:

1. A company may restrict its commercially confidential documents so that they can only be accessed while inside a building during normal business hours.

2. Network connectivity may be provided only to users who are located in a specific room (e.g. a conference room) during a specified meeting time.

3. Devices, such as corporate laptops, can be made to cease functioning if it leaves the building.

4. During sporting events, a sports service may transmit value-add information, such as live scores and player information, during the game, but only want those within the stadium to be able to access this information.

5. Movies or entertainment may be made to be accessible only to vehicles that are located on a specific road.

The above examples illustrate cases where both objects and services may be restricted based on location. Although the implementation of STAC to services, such as network connectivity, might not necessitate use of encipherment, we shall take the viewpoint in this chapter that all objects/services that we wish to control access to can be suitably protected through encryption and appropriate key management. The extension of the ideas provided in this chapter to more general cases, such as controlling access to network connectivity, can be done through simple resource allocation mechanisms, and we thus consider general cases a straight-forward application of the techniques detailed in this chapter.

Throughout the rest of this chapter, we shall refer to the STAC communication model depicted in Figure 5.1. In this model, we have an object that is protected through encryption with a set of keys. These keys may, for example, encrypt different portions of the object. Assisting in the STAC process is a broad array of sensor nodes, spread out over the region of interest, that each hold a set of decryption keys. These sensors can be configured to emit specific keys at specific times with specific power levels. As a result, only users that are near the right sensor at the right time can witness keys and access objects. The schedule of key transmissions, and their corresponding power levels, can either be pre-loaded, or controlled by an external entity connected to the sensor network, as depicted in the figure. Additionally, in the STAC model depicted in the figure, we present two different means by which the user can obtain a protected object: first, it can be broadcasted via a central entity (much like a television broadcast); or it can downloaded, locally stored on the user's device, and then accessed when the device witnesses the appropriate keys.

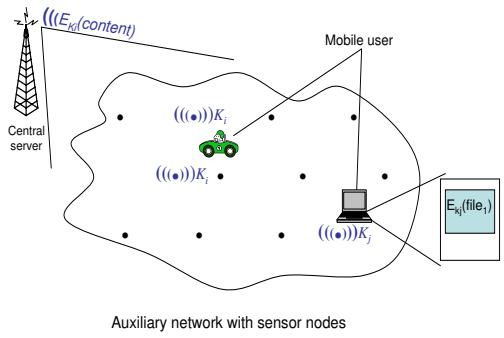


Figure 5.1. Our basic architecture for spatio-temporal access control consists of a central entity that supplies encrypted content, an auxiliary network of sensor nodes that emit keys, and mobile users that desire to access content based on their spatio-temporal context.

5.3 Inverted sensor network infrastructure

There are three basic components involved in the inverted sensor network approach to STAC, as shown in Figure 5.1. The first component is a centralized content distributor that does not need to have any interaction with the user. Instead, the content distributor might broadcast or supply objects for download that have been enciphered with a set of keys known by the central server.

The second component is the auxiliary network (the inverted sensor network), which consists of sensor motes that have been deployed to cover a region of interest. These sensor motes will transmit a schedule of encryption keys that vary with time. Here, we assume that time is broken down into intervals, and that during any given interval, the corresponding key will be repeatedly transmitted at regular intervals. Any entity within the radio range of the sensor mote can, should it wish, acquire the key that is announced by that sensor at that time. We note that the keys that are transmitted by the sensor motes must be initially distributed to these nodes for use in supporting STAC.

Finally, the third component is the mobile user itself, which must move around the region of interest in an appropriate manner in order to acquire the keys transmitted by the sensor nodes.

In order to explore the properties of the inverted sensor network, we assume that

sensor nodes are deployed in a regular, hexagonal fashion across the region of interest. It is well known that a regular hexagonal lattice is more efficient than either rectangular or quincunx sampling in two-dimensions. Hence, we assume the coverage area is partitioned into hexagon cells of the same size, and that a single sensor node is placed at the center of each hexagon. Prior to initiating the STAC enforcement, we assume that each sensor has been assigned a schedule of keys (for simplicity, we shall say that the keys have been distributed by a centralized key distribution center). During STAC operation, each sensor emits a key according to its schedule, which can be observed by any other entity within radio range of the sensor.

If we let a be the length of an edge of one of the regular hexagons, and assume isotropic radio propagation then we may assume that each radio coverage is a circle with radius r. There are two natural choices for how the hexagonal lattice is deployed: either we deploy the sensor nodes so that the radio ranges do not share any overlap (and hence the circular regions would touch each other only at tangent points), or we can assume that we have deployed the sensors so that there is some overlap between the radio regions. Since the first option implies gaps in the radio coverage of the sensor grid, we assume that the deployment is of the second type. In this case, we have a deployment such as the one depicted in Figure 5.2, and hence the radius of radio coverage is $r = a$.

Given a ST-region of an object that needs to be protected, we assume that the content distributor knows the key schedule of all of the sensor nodes (perhaps it is also the key distributor and generated the key schedule in the first place), and that the object has been encrypted with the symmetric encryption key corresponding to the key that is being transmitted by a specific sensor node at a specific time. Exploring this idea further, reveals a some system requirements. If we need a STAC region that is larger than a single radio region of a sensor node, then we need all sensor nodes within the ST-region to transmit the same key. This requires that either the key distribution center has a priori knowledge of the ST-regions (perhaps corresponding to one or more objects that need to be protected) that need to be enforced for spatio-temporal access control, or that the key distribution center has a means to adapt the key transmission schedule

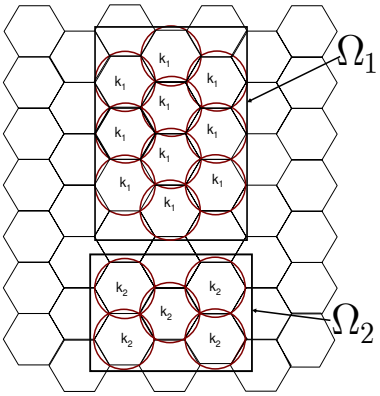


Figure 5.2. An example of how the keys may be assigned in order to cover two ST regions Ω_1 and Ω_2 for an inverted sensor network support spatio-temporal access control.

of the sensor network. As an example, Figure 5.2 illustrates the key distribution scheme for two secure ST-regions Ω_1 for object O_1 and Ω_2 for object O_2. Here, we suppose k_1 and k_2 are the decryption keys for O_1 and O_2 respectively at a particular time. Therefore, it is necessary that k_1 has been assigned to all sensors whose radio discs are inside the rectangle Ω_1, and k_2 to the sensors whose radio discs are inside the rectangle Ω_2.

5.4 Improving the coverage

As seen in Figure 5.2, the regions of Ω_1 and Ω_2 are not fully covered by the keys sent by sensors. In particular, the concept of an *approximating ST-region* naturally arises.

Definition 1: An *approximating ST-region* $\overline{\Omega}$ of a ST-region Ω given an STAC mechanism Σ is the spatio-temporal region that the object is actually accessible under Σ.

In practice, the approximating ST-region is not likely to be the same as the original, desired ST-region, and in fact we are only able to protect an approximation of the original ST-region. If we restrict our attention to just the spatial portion of a ST-region (which we shall denote by R), then we want the approximating ST-region to cover as large of a portion of the desired ST-region as possible. Before we proceed onto exploring how to optimize the approximating ST-region, we note that our discussion

has centered around approximating a ST-region from the inside, and that it is possible to consider approximating regions that are *larger* than the ST-region that they are meant to represent. In such a case, we include sensors whose radio regions share any overlap with the desired ST-region. For this case, we would aim to reduce the amount of extra area covered by the ST-region.

For this work, though, we focus on approximating a ST-region from within, and hence we would like to minimize the amount of blank area between a desired ST-region and the approximating ST-region. Although we have deployed our sensors in a regular fashion, we may adjust the transmission powers of the sensor nodes in such a way so as to improve the amount of area covered by the radio regions.

To formalize this, let us define the region of interest to be G, and define $S = \{s_j\} \in G$ to be a set of sensor nodes. Here, we use the notation s_j to refer to the spatial position of node j. In the discussion that follows, we do not require that the sensor positions fall on a lattice, but instead the positions can be more general. For a given region $R \subseteq G$ we may specify what it means to cover (or fill) R from within.

Definition 2: A *cover of R from inside*, denoted C, is a set of circles C_j centered at s_j such that the union of the circles is fully within the inside R, that is $C = \bigcup(C_j) \subseteq R$. A cover from inside C is a function of a subset of the sensor nodes that are selected, and the corresponding transmission powers assigned to each of these sensor nodes. Hence, if we denote $\underline{P} = \{p_1, p_2, ...\}$ to be the power allocation vector for nodes $\{s_j\}$, then we may represent the cover as $C(\underline{P})$.

A cover from the inside will typically not completely cover the region R in the formal topological sense, and thus we are interested in measuring how accurately a cover from the inside approximates R. To capture this notion, we introduce the blank region and its corresponding area.

Definition 3: For a cover from inside C of a region R, a *Blank Region* is the set $B(C, R) = R \setminus \bigcup(C_j')$. A measure of the magnitude of the blank region is the *Blank Area*, $BA(C, R) = Area(R \setminus \bigcup(C_j'))$.

In an access control system, it is desirable to minimize the blank area. We note that it is easily possible to minimize the blank area by covering areas outside of R. This,

however, implies that users who are not in the restricted area can access protected content, and hence should be considered as security weakness. Consequently, we take the viewpoint that it is desirable to from a security point-of-view to sacrifice some area between the boundary of the STAC region and the actual approximation region so long as we do not have any key information being leaked outside the desired access control region.

In order to accomplish this, we may adjust the transmission powers in such a way to best cover from the inside a particular region R. We now present Algorithm 1, which describes a greedy algorithm for constructing an approximation of a region R from within, with the objective of minimizing the blank area $BA(C, R)$. In this algorithm, the input is the collection of sensor node locations $\{s_j\}$, as well as the desired region R. Additionally, we provide a constraint m which describes the maximum allowed transmission radius for each sensor node. For example, m might be determined by policy or by hardware restrictions. In this algorithm, we assume that there is a direct way to relate the actual transmission power to a corresponding coverage radius p_i (for example, by employing a propagation model). Hence, rather than explicitly define the algorithm in terms of assigning wattage to different nodes, we are instead formulating the problem in an equivalent manner using distances. We use the notation $d(s_i, s_j)$ to denote the distance between the nodes s_i and s_j.

The algorithm basically starts by assigning powers to ensure that the maximal coverage region for each sensor node is as large as possible, while remaining inside of the region R. Then, the algorithm proceeds to remove redundant nodes or power assignments. Using Algorithm 1, it is possible to achieve a more efficient coverage pattern for an arbitrary region R than using a default deployment pattern where every node has the same power assignment. We present an example power configuration that results in Figure 5.3.

In order to further illustrate the advantages of adapting the power levels, we conducted a simulation study where the ST-region to be protected is a square with sides of length d, and deployed sensors in a uniform hexagonal tiling where the distance between sensor nodes is r. We varied the ratio d/r, and measured the blank area for both

Data: Spatial Region R, the set of sensor node locations $s_1, s_2, ..., s_n$ and the maximal radius constraint m
Result: An cover from inside C, such that $BA(C, R)$ is minimal.
for *Every* s_i *inside of* R **do**
  Draw an inscribed circle centered at s_i, r_i denotes the radius of the inscribed circle.;
  $p_i = \min(\ m, r_i)$;
end
Sort the sensors by the decreasing of their inscribed circle's radius. $s_{(1)}, s_{(2)}, ..., s_{(n)}$;
for $i = 1$ *to* n **do**
  if $p_{(i)}$*!=0* **then**
    for $j = i + 1$ *to* n **do**
      if $r_{(j)} + d(s_{(i)}, s_{(j)}) < r_{(i)}$ **then**
        $p_{(i)} == 0$;
      end
    end
  end
end

Algorithm 1: An algorithm for finding a near minimal blank area given a set of sensor locations $\{s_j\}$ and a desired region R to cover from the inside.

the uniform coverage and adaptive coverage resulting from Algorithm 1 using Monte Carlo sampling techniques. We report the results in Figure 5.4, where the x-axis is parameterized via d/r and the y-axis corresponds to the ratio of the blank area to the area of the total square. In this figure, we see that at small d/r, it is not possible for the uniform deployment to cover the square (going beyond the boundary of the square is not allowed), but the power allocations assigned by Algorithm 1 adapts the transmission power to cover square without going outside the square. As we increase d/r, we allow more sensor nodes to fall within the square, and we see that the adaptive power allocation algorithm consistently results in less blank area than uniform power allocation.

Finally, we note that Algorithm 1 is flexible and can be applied to address the power allocation for any placement of sensor nodes beyond the regular lattice patterns that we have focused our discussions on.

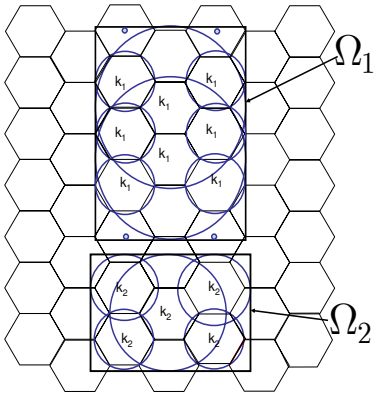


Figure 5.3. An example of the key distribution coverage pattern after the power allocations of each sensor node have been adjusted using Algorithm 1.

5.5 Dynamic Encryption and Key Updating

Spatio-temporal access control not only involves access based on an entity's spatial location, but also implies that there might be important temporal contexts that affect the ability of a user to access content. There are many cases where we can specify a STAC policy for an object (such as an entire movie) that has the requirement that access changes with time. One particularly important example of an object that would have such a policy is a streaming object that varies with time. For this case, it is necessary to decompose an object into smaller object atoms (as defined earlier in Chapter 2), and then treat each of these smaller object atoms as individual objects that are protected over a spatial region that is fixed over a smaller time interval. As an example, we can consider an object with a formal ST-region Ω_2, but would have to approximate Ω_2 via a collection of smaller and simpler ST-regions with temporal resolution τ, as depicted in Figure 5.5. Here, in this figure, region Ω_2 is thus approximated as the union

$$\Omega_2 = \bigcup_{j=1}^{4} \Omega_{2j}.$$

Each of these ST-atoms Ω_{2j} will be enciphered by a corresponding key k_j, and hence the encryption of such an object will be dynamic in the sense that the key will vary with time. For streaming objects, if we were to not employ dynamic encryption, then

it would be possible for an adversary to observe the key in a valid ST-region (e.g. Ω_{21}) at a valid time, and then access the rest of the content from a region not allowed by the spatio-temporal access control policy.

As a further issue, we note that dynamic encryption is only meant to protect the content during the initial access period for that content. By this we mean that once a user has recorded a STAC-protected object and has satisfied the spatio-temporal requirements to access the object, that user has effectively unlocked the object for him/her to access at any later time. Essentially, once a user has the file and the keys, access is granted from that point on. We note that dynamic encryption is only meant to protect dynamically evolving content/objects. If it is desired to strictly limit a user to accessing content during a specific time and not after that time, then it is necessary to employ the use of additional security mechanisms, such as trusted operating systems and secure containers which would guarantee that key information is stored and accessible to the user only during a specified time.

Dynamic encryption and the decomposition of objects into smaller, more refined ST-regions requires that the each ST-region is associated with different keys, and hence the problem of managing the keying information in an inverted sensor network becomes important. Although one could envision that a key distribution center might be able to manage and frequently update keys within the sensor network by issuing updates to each sensor node (e.g. through a gateway between the sensor network and the broader Internet) every time the key needs to change, such a scheme is impractical. Rather, we should reduce the frequency at which the key distribution center interacts with each sensor node by having the center distribute a single set of keys corresponding to a keying schedule.

In order to do so, the KDC must either initially install a large set of keys prior to deployment, or the KDC can communicate as needed with the sensor nodes through a set of keys shared between the KDC and each sensor, i.e. $K_{s_i,KDC}$. There is significant overhead associated with frequent updates of keys, and in order to reduce this overhead, we make use of a chain of one-way hash functions to generate and store keys.

A one-way chain $(V_0, ..., V_n)$ is a collection of values such that each value V_i (except

the last value V_n) is a one-way function of the next value V_{i+1}. In particular, we have that $V_i = H(V_{i+1})$ for $0 \leq i < N$. Here H is a one-way function, and is often selected as a cryptographic hash function. For setup of the one-way chain, the generator chooses at random the root or seed of the chain, i.e., the value V_n, and derives all previous values V_i by iteratively applying the hash function H as described above, yielding a chain as in:

$$V_0 \leftarrow V_1 \leftarrow V_2 \leftarrow \cdots \leftarrow V_{n-1} \leftarrow V_n.$$

By employing the hash chain, the entity responsible for key distribution need only send the anchor seed and the times at which the sensor node should change keys. For example, if we let the last key be the key seed, i.e. $V_n = K_n$, then the KDC simply performs

$$KDC \rightarrow SN : E_{K_{(s_i,KDC)}}(K_n, t_0, t_1, ...t_n).$$

The sensor then can derive K_1, K_2, all the way up to K_{n-1} locally by applying H. When the keys are used up, the central server will repeat the process.

One necessary system requirement for STAC, though, is that all sensor nodes maintain synchronization with each other and the server so as to guarantee that keys are transmitted during the correct time period.

5.6 Discussion on the operation of inverted sensor networks

The use of the inverted sensor network for spatio-temporal access control achieves several advantages when compared to a centralized scheme: first, it reduces the risk of a privacy breach; second, it is naturally resistant to location spoofing attacks; and third, it facilitates new classes of applications that can easily be implemented.

5.6.1 Reduced Contextual Privacy Risk

Privacy is the guarantee that information, in its general sense, is observable or decipherable by only those who are intentionally meant to observe or decipher it. The phrase "in its general sense is meant to imply that there may be types of information besides

the content of a message that are associated with a message transmission. When you access an ATM at a bank, this action is observable, and some contextual information regarding your actions is revealed to anyone observing you. An adversary that witnesses you going to a bank should naturally conclude that you likely have withdrawn money, and he does not need to launch any sophisticated cryptographic attacks to acquire your money (he simply robs you as you walk away from the bank).

In this bank example, a user's contextual information is revealed. More generally, the issue of contextual privacy has come up in other scenarios, such as database access and location-services. Generally speaking, there is a risk of a privacy breach when any entity A contacts another entity B asking for some service. For example, in the case of database privacy, when an entity A requests information from B, B's sole function should be to provide the service or answer to the query. It might not be desirable for B to know the details of the query or the specific answer [39–41].

In access control systems, there is always the risk of a privacy breach. When the user requests a service, this request can be logged by the entity providing the access control. In Kerberos, for example, all of the emphasis is placed on secure, authenticated exchanges but it is possible for the servers that administer service granting tickets to record which user has made a request for which service, thereby tracking a user's usage patterns or preferences. Similarly, in a centralized spatio-temporal access control, when the user attempts to prove that it is in a specific location, this can be recorded by the centralized entity, and used to infer the user's activities– it becomes possible to not only infer the user's current position, but also by accumulating the user's position over time it is possible to discover the user's habits.

In order to avoid this privacy risk, what is needed is a technique for access control that does not pass through a centralized entity. In our use of inverted sensor networks, the user (or users) are supplied content through some external means. For example, the content may be streaming content that is broadcast, and any entity that wants to access the content is free to do so by simply being at the right place at the right time to acquire the keys transmitted by the inverted sensor network. Since the users don't have to interact explicitly with any entity, there is no information revealed as to whether a

specific user is accessing specific content, and there is no information revealed about the user's spatio-temporal profile.

5.6.2 Resistant to Positioning Spoofing

Any scheme that requires that a user prove that it is in a specific location becomes susceptible to attacks that might be launched against the localization infrastructure [42] (such as an adversary trying to prove that it is in a location that it is not, which would allow the adversary to access content he is not intended to access). Although secure localization schemes can mitigate this threat, there might be a limit to the effectiveness of these techniques against non-cryptographic attacks [?]. By using inverted sensor networks, however, this issue is bypassed. There is no reliance on an entity proving its location to another entity. We do note, however, that inverted sensor networks can be physically attacked by adversaries destroying nodes, capturing and reprogramming nodes, or by simply covering sensors so that their transmissions are blocked. These issues however, are common to all wireless networks, and must be addressed through careful deployment of the sensor nodes (e.g. placing the nodes on the ceiling in a building might make it harder for vandalism). Lastly, we note that we are implicitly assuming a limitation on the antennas being used by a user's device. More expensive hardware may lead to an expansion of radio ranges, thus enlarging spatial regions covered by a key.

5.6.3 Support of Applications with Little Effort

Taking advantage of a sensor network in an inverted fashion in order to facilitate spatio-temporal access control represents what we feel is an exciting opportunity to develop new classes of location-based applications. Programming STAC applications becomes relatively easy with the assistance of an inverted sensor network: sensor nodes can be deployed in support of STAC and then loaded with their key schedule; while a user's application simply needs to receive broadcast content, listen for the appropriate key, and then decrypt the content.

One promising new style of application that we envision is a spatio-temporal scavenger hunt, which might be an interesting paradigm for educational applications. In the scavenger hunt application, the user receives content and can only access it at the right place and right time. This content, once opened, might give the user a puzzle to solve describing where the user must next go to in order to advance to the next stage of the scavenger hunt. If the user solves the puzzle and makes it to the next location within a specified time limit, then the user would get the next puzzle. The process can continue until the user achieves the final objective.

There are many variations that are possible to the basic scavenger hunt. The objective of the scavenger hunt game can be specified by constructing a suitable transition diagram, which would detail the rules and paths allowed for a user to traverse the game. For example, we may create a cut through in the transition diagram (similar to the cut-throughs in Figure 2.5) which would allow for a user to bypass the requirement of going to a certain area and accessing a certain object. This could correspond, for example, to a user solving a more challenging puzzle and being allowed to advance further in the game. Or, as another variation, if a user solves a puzzle and moves to the next location in a short amount of time, then the user might receive a different decryption key than if it had taken longer to solve the puzzle, thus allowing the user to access a different puzzle when it is in this location. Overall, we believe that inverted sensor networks can easily facilitate new classes of applications.

5.7 Related Work

The conventional literature on access control models can be broken into several main categories: identity-based access control, role-based access control, and context-based access control. Location-based access control may be considered as a specialized form of context-based access control [43, 44].

Research into supporting location-based access control has primarily focused on the issue of providing secure and robust position information. In [18], the authors listed a few attacks that might affect the correctness of localization algorithms along with a few

countermeasures. SecRLoc [45] employs a sectored antenna, and presented an algorithm that makes use of the property that two sensor nodes that can hear from each other must be within the distance $2r$ assuming r is fixed in order to defend against attacks. [19] uses hidden and mobile base stations to localize and verify location estimate. Since such base station locations are hard for attackers to infer, it is hard to launch an attack, thereby providing extra security. [20] uses both directional antenna and distance bounding to achieve security. Compared to all these methods, which employ location verification and discard location estimate that indicates under attack, [46] and [?] try to eliminate the effect of attack and still provide good localization. [?] makes use of the data redundancy and robust statistical methods to achieve reliable localization in the presence of attacks.

There has been less work devoted to developing the remaining components needed for spatio-temporal access control, and there has only been a few efforts that have tried to develop location-based access control systems. In [47], the objective is to provide location-based access control to a resource(in this case, the wireless links). The authors propose a Key-independent Wireless Infrastructure (KIWI) where, during the handshake period, KIWI challenges a client who is intended to access the resource with a set of nonces. Only when the client send back the proof that it correctly received all the nonces, can it complete the authentication handshake. In [48, 49], the PAC architecture is proposed in order to provide location-aware access control in pervasive computing environments. Although PAC preserves user anonymity and does not expose a user's exact location, one drawback is that it uses only coarse geographical location areas.

One important issue related to STAC is user privacy, as techniques that acquire the user's exact location also can expose this information to unwanted parties [50]. Location privacy has recently been studied in the context of location-based services [51, 52]. In [51, 52], a distributed anonymity algorithm was introduced that serves to remove fine levels of detail that could compromise the privacy associated with user locations in location-oriented services. For example, a location-based service might choose to reveal that a group of users is at a specific location (such as an office), or an individual is located in a vague location (such as in a building), but would not reveal that a specific

individual is located at a specific location. Duri examined the protection of telematics data by applying a set of privacy and security techniques [53].

5.8 Conclusion

As wireless networks become increasingly prevalent, they will provide the means to support new classes of location-based services. One type of location-oriented service that can be deployed are those that make use of spatio-temporal location-information to control access to objects or services. The approach that we explored, which was the primary focus of this chapter, involved inverting the role of a sensor network. Specifically, an appropriately deployed sensor network can consist of nodes that locally transmit a schedule of keys, thereby facilitating access to enciphered objects. We then examined issues of optimizing the region covered by the sensor network in support of spatio-temporal access control by providing an algorithm for optimizing sensor node power allocation.

We believe that spatio-temporal access control in general, and inverted sensor networks in particular, represent a promising paradigm for the development of new location-oriented applications. The techniques outlined in this chapter represent the beginning of a larger effort to develop spatio-temporal applications using inverted sensor networks.

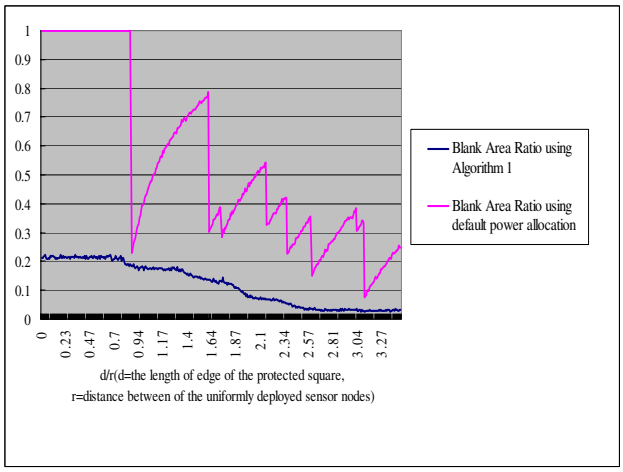


Figure 5.4. A comparison of the blank area for a uniformly (hexagonal) deployed sensor network with fixed transmit powers with the same network where the transmit powers are adjusted using Algorithm 1. The x-axis corresponds to varying the . The y-axis is the ratio of the blank area to total area of a square with sides d.

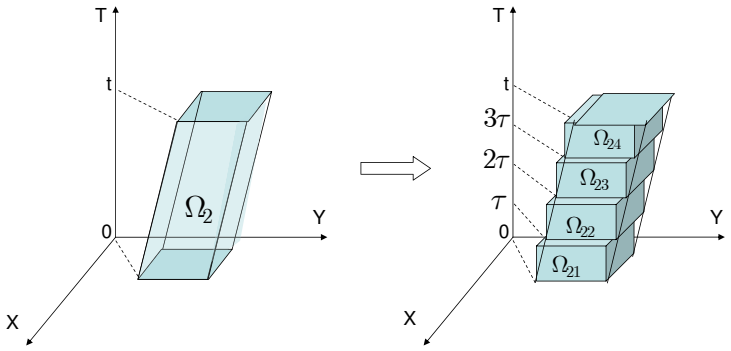


Figure 5.5. A ST-region Ω_2 is originally specified in a continuous, smooth manner. However, in practice it is necessary to decompose the region into ST region atoms Ω_{2j}, and correspondingly decompose the object into smaller object atoms.

Chapter 6

A Security Architecture and Protocols for Mobile Location Based Service

6.1 Introduction

Computing and networking are shifting from the static model of the wired Internet toward the new and exciting "anytime-anywhere" service model of the mobile Internet. Mobile Location-based Services (MLS), whereby a user obtains certain goods or services from a moving vendor by requesting the goods or services based on the proximity of vendors to that user, represents a new form of business that is enabled by the wireless mobile Internet. MLS will require the ability for potential customers in the service area of a *mobile vendor* to be notified of potential services. In particular, emerging MLS will help to eliminate missed business opportunities by making customers and mobile vendors more aware of each other.

More importantly, in order for this emerging application to be realized, it is necessary to ensure that the MLS operates in a secure and trustworthy manner–in spite of the apparent vulnerabilities associated with wireless networks and mobile devices [54], [55]. In particular, since wireless devices have become increasingly affordable and programmable, they may also represent an ideal means to subvert an MLS. For example, an adversarial customer may reprogram its device to lie about its location (e.g. by not reporting correct GPS location provided by the GPS receiver on a mobile phone), and as a result attract a mobile vendor to a false location where the customer does not reside. Similarly, a mobile vendor may claim that it is at a location that has more business opportunities rather than reporting its true location. Further, a malicious mobile vendor may collect users' information and infer personal behaviors associated with some users. These kind of adversarial behaviors are particularly harmful to the application of MLS

as they will not only waste the time and energy of both mobile vendors and customers, but also lead to lost business opportunities.

Thus, to realize the broad business opportunities enabled by mobile wireless services, there is an urgent need to design MLS that integrate security into their design. In this chapter, we propose a security architecture for MLS that can be integrated with new or existing MLS to provide trustworthy services. In addition to identifying different attacks and misuse faced by MLS, we take the viewpoint that auxiliary networks enabled by the increasingly wide deployment of wireless technologies (e.g., WiFi hotspots that may be run by the same company deploying wide-area wireless services, such as cellular and WiMax) can be used to facilitate position verification, which is a critical step in providing trustworthy location information.

In particular, we propose to use two levels of location verification schemes, namely, *History-based Consistency Check* and *Key Distribution-based Verification*. Our *History-based Consistency Check* scheme performs coarse-grained position verification based on location claims over time, whereas the *Key Distribution-based Verification* as we presented in Chapter 4 is a fine-grained position verification scheme. Further, we develop a secure MLS framework that provides end-to-end trustworthiness (entity authentication, message integrity, and message confidentiality) for mobile services.

The reminder of this chapter is organized as follows: In Section 6.2, we provide an overview of MLS by presenting its basic architecture. We then identify various attacks that can undermine the application of MLS in Section 6.3. We propose our position verification methods of *History-based Consistency Check* and *Key Distribution-based Verification* in Section 6.4. We next put our approaches in a broader context by developing an end-to-end secure MLS framework in Section 6.5, and finally, we conclude in Section 6.6.

6.2 Mobile Location-based Services

A current business model in several industries involves offering goods or services to customers by means of a mobile business unit, such as a car, truck or van. Unlike

traditional brick and mortar businesses where a potential customer must travel to the location of a business in order to receive goods or services, mobile business units have the flexibility to travel to a customer. Mobile ice cream trucks and taxicabs are two familiar examples of such mobile business services. Mobile services benefit the client by providing convenience and potentially lowering costs since the overhead of maintaining a brick and mortar presence is no longer needed. It is clear that the implications of mobile services are great.

However, a significant problem may arise for this service model because of missed business opportunities. For example, consider how frequently someone misses the opportunity to hand off a package to a parcel delivery service because they missed the pickup time by a few minutes. In general, some causes for the missed business opportunities may be summarized as: (1) a mobile vendor does not know at which location its goods or services are needed at any given time; (2) the mobile vendor is limited in how the arrival of its goods or services may be advertised to the potential customers. The development of wireless networks and, in particular, the next generation of mobile phones provide a means to address the communication gap between mobile vendors and customers without the assistance of a human dispatcher.

In this chapter, we examine an architecture that can support Mobile Location-based Services, whereby mobile vendors are connected to potential clients. Our solution is well-suited for cellular networks, where each vendor defines a service area, which is a circular region around that vendor's current location. The vendor's service information is periodically sent to customers subscribed to this MLS service within the vendor's service area. As the mobile vendor moves, a customer may move out of the vendor's service range, in which case the customer then is not able to view the vendor's service information.

6.2.1 Basic MLS Architecture

Here we present our basic MLS system architecture. The system consists of three entities: the mobile vendor (MV), the User (U) and the service provider (SP):

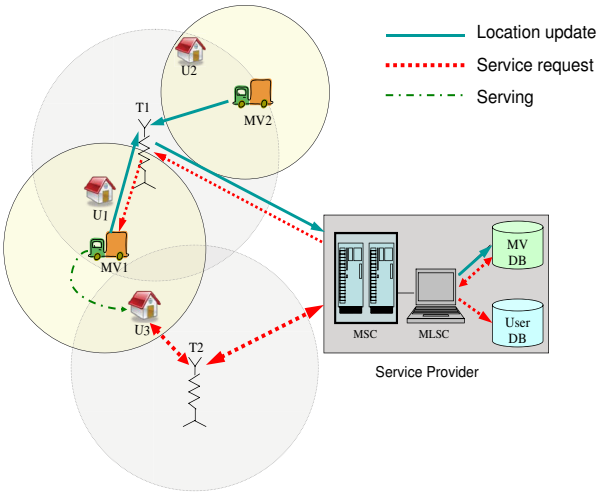


Figure 6.1. The architecture and working flow of a basic MLS

- Service Provider (SP): This is any network provider, such as a cellular network provider. The SP provides the Mobile Location-based Service to its network users and vendors. This SP keeps a database of subscribed mobile vendors and users.

- Mobile Vendor (MV): The MV provides goods or services to a requesting user. Mobile Vendors are able to move around to satisfy requests. The MV is equipped with a device that can locate itself and send its location to the service provider. This device may, for example, be a cell phone or PDA with GPS functions enabled.

- User (U): The user has a device (e.g. a cell phone) that displays the information about mobile vendors received from the service provider. In this basic architecture, we assume the device can locate itself and report its location to the SP in order for the SP to determine the list of MVs that are close to the user.

Figure 6.1 illustrates a sample MLS architecture for a cellular system. In this architecture, an MLS module in the SP called *Mobile Location-based Service Center* (MLSC) connected to mobile switching center (MSC) is in charge of the MLS service and the MLSC keeps databases for the MVs and the Users. Each MV sets a radius as to how far from its current position it would like its service information to be advertised to potential users, e.g. 2 miles. The service areas are shown in Figure 6.1 as circles centered

at the $MV1$ and $MV2$ respectively. The service information (vendor's name/brand, goods/services,...) as well as the service radius are sent to the SP at the time this MV subscribed to the service.

T_1 and T_2 are cell towers. Every cell tower communicates with the MSC. The communications between a user and the SP, or a MV and the SP, could for example be through the Short Messaging Service (SMS), or other data communication method provided by the SP. The messages go though the closest tower or access point from/to a user or MV to/from the SP. The solid lines in the figure show the message flow when the MVs update their locations to the SP. The location information for a MV is sent from the MV through the closest cell tower to the SP and is stored to the MV database in MLSC. The dotted lines shows the message flow when a User requests an MLS service. The request is initialized from a user, and the detailed system working flow is described as follows:

1. The U starts the MLS program on its device, which initializes a request for the list of vendors whose service areas cover its location.

2. The SP sends a request to the customer's device for its GPS coordinates.

3. The U's device automatically finds its GPS coordinates and sends them back to the SP. This step is transparent to the user.

4. The SP searches the MV database, and sends the list of vendors whose service areas cover the customer. The updated list is sent to the customer periodically, e.g. every 2 minutes, until the customer exit the MLS program.

5. The user chooses an MLS service MV and sends a request to the service provider with the requesting service and maximum waiting time.

6. SP forward user's request to the vendor. The MV moves to the location of the requester U, should he accepts the service request.

6.3 Security Threats Analysis

In the real world, one or more entities in an MLS may behave harmfully for various reasons. For example, competition may cause vendors to grab business opportunities through malicious practices, or entities may be compromised by hostile parties, or users may change their minds and neglect to cancel service requests. It is thus critical to be aware of the security threats that a system may face. In this section, we analyze the security threats facing an MLS system.

6.3.1 Attack models

Attacks can be categorized into three groups according to which type of entity initiates the attack: mobile vendor attacks, user attacks and outsider attacks. It is reasonable to assume the service provider is a trustworthy entity [56].

Mobile Vendor Attacks: Mobile vendors are motivated to obtain as many business opportunities as possible by having its service information heard by as many as potential customers as possible. Furthermore, it wants a user to choose its business over other MVs that provide the same type of business and are available to the user. This can prompt an MV to act maliciously to its own ends. Some related attacks include:

- *False Location Claims*: The MV reports to the SP a location other than its true current location. For example if a MV is at location P_1, but from its experience there are usually more customers at location P_2 (e.g. a downtown location), it could then claim to be at P_2 so that the customers around P_2 can get its service information before it actually moves into that area. In the basic MLS system, a MV's location is sent to the SP directly by the application. An MV could send a falsified location if the application software is specifically modified. False location claims may also be carried out by other means under other system designs. For example, in the system where the SP locates the MV through signal strength, a falsified location may be calculated if the MV modifies its transmission power. When a user chooses the cheating MV's service, it may have to wait longer than it would if it had chosen another MV. This is unfair to other vendors who may

lose this business opportunity.

- *Sybil Attacks [57]*: One single MV claims multiple locations ($P_1, P_2, ...$) to the SP, so that its business information gets broadcasted to all the users in the multiple locations. Again, this attack may result in longer waiting time for the users and it breaks fair business competition relationships among the MVs.

- *Service Failures*: The MV accepts a service request, but fails to provide the service. There can be many reasons that result in the failed service, for example, traffic blocks the MV from arriving on time, the MV changes its mind, or is out of goods or service. In any case, the MV should notify the SP about it being unable to make the transaction at the earliest time, so that the SP can notify the user. If the MV fails to do so, this harmful behavior should be recorded.

User Attacks: Users may intentionally or unintentionally hurt the system. The following describes several threats that users may pose:

- *False request*: The user denies the service when the MV that is being requested arrives with the service. This may happen unintentionally when the customer changes its mind or is unsatisfied with the service, or intentionally when the customer is hostile. False requests waste the MV's resources (such as time and gas), and may cause MV to lose other business opportunities. It also reduces the chances of other customers to get the services, because this user occupies the service resource.

- *False location claim*: The user claims to be at a falsified location. When the MV comes to the claimed location, the user is not present. This will result in the same harmfulness to the MV and other users as false request.

Outsider Attacks: Outsiders may intend to attack the system to destroy the service. The following are examples of such attacks:

- *DOS attack:* Outsiders pretend to be users and send tremendous amount of requests that cannot be handled by MVs. This threat can be mitigated by using

entity authorization and message encryption, as describes in Section 6.5, in which each MV or user is assigned a unique ID and public key and private key pair and every message is encrypted and authenticated. An outsider would have no way of pretending to be other MVs, nor Users unless it becomes an insider to attack the system, in which case, the problem is converted to the MV or the user attacks.

- *Signal interference:* Attacks such as jamming are general problems existing in wireless networks.

  In this chapter, we focus on solving the security problems that are specific to the MLS systems, and leave the solution to these attacks to the works address on them.

6.4 Key Technological Approaches

In this section, we present the key approaches that we believe should be used to secure an MLS system: location verification, methods to prevent Sybil attacks and reputation systems. We will provide a brief description of current methods in the literature related to reputation systems and for preventing Sybil attacks. Since these technologies are mature, most of the research discussion throughout this chapter shall focus on location verification approaches.

6.4.1 Location Verification

Location verification can detect false location claims, which may occur at both the MVs and users' sides. In the discussion that follows, we present two levels of location verification methods. Consistency checks are simple, easy to carry out but the verification is coarse. Key Distribution-based location verification (KDLV), on the other hand, are finer-grained and make use of an auxiliary network to give more reliable verification results.

Our location verification scheme used in MLS is a two-step procedure. When the SP receives a location claim Loc_{cur} from an MV or a user , it first conducts consistency checks. If the location claim fails any of the consistency checks, it is put to the second

step of location verification, Key Distribution-based Location Verification. By using the two-step procedure, we save the cost of applying strict and complex location verification methods directly to every location claim.

Coarse-grained Methods: Consistency Checks. Consistency checks are used as filters that find suspicious location claims before sending them to a stricter location verification scheme. Our consistency checks employ two types of information:

1. Consistency checks based on historical location claims: Compare the location claim with the last recorded location. Given a threshold speed $MaxSpeed$, if the claimant (MV or user) is calculated to have velocity more than $MaxSpeed$, then the claim is suspicious. Let Loc_{cur} be the current location claim at time t_{cur}, and Loc_{old} be the location at t_{old} as recorded in the SP. If

$$\frac{||Loc_{cur} - Loc_{old}||}{(t_{cur} - t_{old})} > MaxSpeed,$$

   then we suspect that this location claim may be false.

2. Consistency checks using cell tower information: In mobile networks, such as GSM and IS95, the subscribers' information is saved in mobile switching center, particularly, in Home Location Register (HLR) and Visitor Location Register (VLR) if the user is roaming [58]. Aside from other information such as the subscriber's profile, the current cell tower, or more specifically, the Local Area Code (LAC) that the subscriber is associated is also recorded in HLR. We can use this information to do a rough judgement. If the location claim is not in the cell tower's coverage area, that is $Loc_{cur} \notin Area(Cell_t)$, then this location claim is suspicious. Although it is possible that the user is in the transition from one cell tower to another cell tower, we send the location claim to the next step for more careful check.

Fine-grained Method: Key Distribution-based Location Verification. We use the Key Distribution-based Location Verification method presented in Chapter 4 to do a fine-grained location verification.

6.4.2 Anti Sybil Attack Methods

Formally, a Sybil attack is any attack where a single entity illegitimately claims several identities. Such attacks are often used in distributed systems as a means of consuming system resources. [59] classifies different types of Sybil attacks in terms of three orthogonal dimensions, and proposes several techniques to defend against Sybil attacks. Among the classification dimensions proposed in [59], "fabricated vs. stolen identities" says that the additional illegitimate identities can be gotten in one of two ways. It can be simply fabricated by the attackers or are stolen from a legitimate identity. In our context, we use this classification to help designing and testifying our anti Sybil attack mechanisms. We note that the following methods could be employed: i) Initially, each MV or user needs to register its mobile service with the service provider, and is assigned a unique ID, and establishes a public key and private key pair. The ID is associated with the MV or user's device. When an MV or user claims its location, it has to submit its ID, and encrypt the message with its private key. By using this registration and key predistribution mechanism we prevent one device to fabricate multiple locations. By encrypting the messages, we prevent malicious party from stealing another entity's ID. ii) In the case that one entity gets multiple identities either by having multiple devices and registering all of them or by stealing other entities' IDs through offline resources, [60] proposed a technique to detect that two devices are always located together. So even though the one entity has multiple identities, its attempt of claiming to be at two different locations will not succeed.

6.4.3 Reputation System

A reputation system is used in our secure and trustworthy MLS system to help filtering suspicious location claims, to discover bad behavior users or MVs, and to encourage the entities to be more trustworthy. At the end of each transaction, the involved MV and user is asked to evaluate the other party, or report bad behavior to the service provider. A bad transaction behavior, such as false request, would cause a drop of the user's reputation score. Similarly, false location claims would also be monitored by the system, and

reputations would also be used to assist location verification schemes. As we mentioned before, fine-grained location verification methods are costly compared to coarse-grained location verification methods in terms of waiting time and communication cost. When a user or MV fails the coarse-grained check, we use its reputation as reference, if it has low reputation, we put this location claim into fine-grained verification.

Reputation systems are widely used in decision support for Internet mediated service provision [61, 62]. There have been many research efforts studying how reputation systems should work and could work better [63,64]; the security threats that reputation system faces and solutions to maintain robust reputation systems [65, 66]. In this chapter, we do not place our focus on the details of building reliable reputation systems, but merely propose that they be used in an off-the-shelf manner. Readers who are interested are encouraged to read related works in this area.

6.5 A Security Framework for MLS

In this section, we propose a framework that uses the position verification methods proposed in Section 6.4 as part of a broader security solution that addresses issues of entity authentication, and message authentication, integrity, and confidentiality so as to protect the Mobile Location-based Services from the threats discussed in Section 6.3.

The framework consists of three phases, namely *Setup*, *Location Information Update*, and *Service Request*. The notations used to describe security protocols and cryptographic operations are stated in Table 6.1. In the *Setup* phase, an MV or a user initializes its registration with the SP for the MLS service and signs in to the service. The *Location Information Update* phase deals with the location information updating messages from an active MV or user. The *Service Request* phase describes the procedure of a user requesting a service until it gets the service.

6.5.1 Phase I: Setup Phase

We assume that there exists a trusted Certificate Authority (CA) that creates digital certificates, for example, X.509-certificates [67] . We rely on the certificate as the means

Table 6.1. Notations used in security protocols

Notation	Description
M	Mobile vendor, MV
S	Service provider, SP
U	User
CA	The Certificate Authority
ID_X	The identification number of the entity X. X can be M, S or U.
t_{X_n}	The n_{th} time stamp issued by the entity X.
KU_X	The public key of the entity X
KR_X	The private key of the entity X
K_{SM}	The session key shared between SP and MV
K_{SU}	The session key shared between SP and U
K_{SM}^{mac}	The secret key shared between SP and MV to compute a message authentication code (MAC)
K_{SU}^{mac}	The secret key shared between SP and U to compute MAC
$E_K(Msg)$	The message Msg encrypted by key K
$Cert_X^{CA}$	The entity X's digital certificate created by CA
Msg_{X_i}	The i_{th} message sent by X

for initial trust. Each entity X needs a certificate to join the MLS service. A certificate contains, among other fields, the certifying entity's ID and the public key, and is signed by the CA's private key:

$$Cert_X^{CA} = E_{KR_{CA}}[ID_X, KU_X].$$

The CA's certificate is assumed to be publicly available. An entity X can get the CA's public key from the CA's certificate and verify that the certificate is issued by the CA [68]. The SP has its digital certificate $Cert_S^{CA}$ issued by the CA, and a public and private key pair (KU_S, KR_S). In the *Setup* phase, we discuss three main steps: Registration, Sign-In and Sign-Out.

6.5.1.1 Registration

When an MV decides to start its mobile business and use the SP's MLS system, it registers with the SP. The SP, after getting the MV's registration request, will verify

(probably off-line) the MV's validation of providing the services and its identity, etc, and create an ID for the MV, ID_M. As stated in Section 6.4, in order to prevent Sybil attacks, each MV is assigned a unique ID which is associated with the MV's device. The SIM card number of the MV's mobile device is a good option to serve as the index of the MV's ID. The SP also sends to the MV its certificate $Cert_S^{CA}$ if the MV is valid. The MV gets its certificate $Cert_M^{CA}$ and public and private key pair (KU_M, KR_M) from the CA. The SP creates an entry for the MV in its MLS database and in the reputation system. The SP stores ID_M, $Cert_M^{CA}$ and KU_M in this MV's entry in the database.

On the user's side, the MLS service can either come with the user's mobile service as an optional service, or the user can download the application software onto its cell phone and install it. When the user starts the program, it first registers user's cell phone to the MLS service with the SP, and the application helps the user to get a certificate $Cert_U^{CA}$ and its public and private key pair (KU_U, KR_U). The SP also sends the user its certificate $Cert_S^{CA}$. Similarly, the SP sets up an entry in the database and the reputation system for the user (e.g., using its SIM card number as the index) and associates $Cert_U^{CA}$ and KU_U with it.

Further, since the MVs will typically be low-powered and computationally constrained devices, public key encryption [68] should not be used for every message exchange. Hence, in our framework, the communication between an MV and the SP or a U and the SP is encrypted using symmetric-key encryption by using session keys and keys for message authentication codes (MAC) [69], [70]. For an MV, a session is defined as the time between when it signs in and when it signs out; while for a user, it is defined as the time from when the user starts this application on the mobile phone till when it exits the program.

6.5.1.2 Sign-In

When an MV starts its service in a new session, it sends a sign-in message Msg_{M_0} to the SP. Msg_{M_0} contains the MV's certificate $Cert_M^{CA}$ and a "sign in" message, with its ID, and a time stamp t_{M_0} which is encrypted by the SP's public key and signed by

KR_M:

$$ReqSI = (ID_M, t_{M_0}, SignIn)$$

$$Msg_{M_0} = (Cert_M^{CA}, E_{KU_S}(ReqSI, E_{KR_M}(ReqSI))).$$

The time stamp is used to prevent replay attacks [56] as well as used in location verification. Upon receiving the sign-in message, the SP extracts the MV's public key from $Cert_M^{CA}$ and decrypts the message with its private key. The SP uses KU_M to verify that the message is originally from the MV, and not from anyone else who has the certificate of the MV. Upon successful verification, the SP generates and sends to the MV a unique session key for starting up the new session and a key for calculating the MAC shared between the SP and the MV. Then for every message that is sent between this MV and the SP in this session, a message authentication code (MAC) will be attached to the message and the whole message is encrypted using the session key K_{SM}

$$SesKey = (ID_S, ID_M, t_{S_0}, K_{SM}, K_{SM}^{mac})$$

$$Msg_{S_0} = (Cert_S^{CA}, E_{KU_M}(SesKey, E_{KR_S}(SesKey))).$$

6.5.1.3 Sign-Out

When a MV signs out, the MV sends to SP a special sign-out message:

$$ReqSO = (ID_M, t_{M_n}, SignOut)$$

$$Msg_{M_n} = E_{K_{SM}}(ReqSO, MAC_{K_{SM}^{mac}}(ReqSO)).$$

When receiving the sign-out message, the SP expires the use of K_{SM} and K_{SM}^{mac}. The message flow between the MV and the SP during the *Setup* phase is presented in Figure 6.2.

Furthermore, on the user's side, the procedure for the *Sign-In* and *Sign-Out* steps are similar to those of the MV. The *Sign-In* procedure is triggered when a user starts the MLS program on its cell phone, and when the user exits the program, the *Sign-Out*

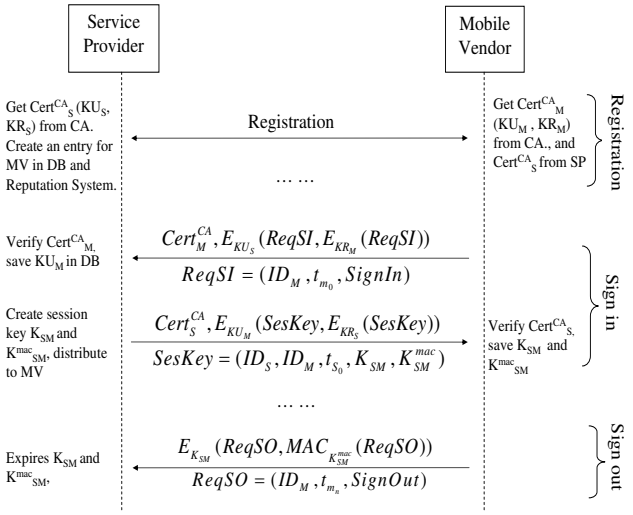


Figure 6.2. The message flow between the MV and SP at the setup phase.

procedure is triggered. After the *Sign-In* procedure, the user will communicate with the SP using the session key K_{SU} and MAC key K_{SU}^{mac}.

6.5.2 Phase II: Location Information Update Phase

After the MV signs in, it periodically updates its location information and other service information in case there is any change to the SP.

$$LocUpdt = (ID_M, t_{m_1}, Loc(x, y))$$

$$Msg_{M_2} = E_{K_{SM}}(LocUpdt, MAC_{K_{SM}^{mac}}(LocUpdt).$$

The SP verifies the reported location $Loc(x, y)$ before it updates the location in the database. Figure 6.3 shows the procedure behind the *Location Information Update* phase, which integrates the location verification techniques we developed in Section 6.4 and the use of a reputation system.

6.5.3 Phase III: Request for Service from an MV Phase

When a user starts the MLS application on the mobile phone, a sign-in procedure is triggered and after this procedure a session key K_{SU} and a MAC key K_{SU}^{MAC} are

input
MV's reported location $Loc(x, y)$
result
MV's location and reputation are updated in SP

if $(Loc(x, y) \in Area(Cell_t))$ **then**
  if $(||Loc(x, y) - Loc(x, y)_{old}||/(t - t_{old})$ ¡ *MaxSpeed*) **then**
    SP updates MV's location in database as $Loc(x, y)$;
  else if (*Reputation(MV)* ¿*ScoreThres*) **then**
    SP updates MV's location in database as $Loc(x, y)$;
  else
    Perform Key-Distribution based Location Verification.
    SP updates MV's location to verified location $Loc(x, y)_{veri}$;
    if (——$Loc(x, y)_{veri} - Loc(x, y)$——¿ MaxError) **then**
      Reputation(MV) –
      Send warnings to MV
    end
  end
else
  Perform Key-Distribution based Location Verification.
  SP updates MV's location to verified location $Loc(x, y)_{veri}$; return $Loc(x, y)_{veri}$.
  if (——$Loc(x, y)_{veri} - Loc(x, y)$——¿ *MaxError*) **then**
    Reputation(MV) –
    Send warnings to MV
  end
end

Figure 6.3. SP's Algorithm for Updating MV's location

established which are used in the communication between the SP and the user in the session. The messages follow the format:

$$data = (ID_X, t_{X_i}, text)$$

$$Msg_{X_i} = E_{K_{SU}}(data, MAC_{K_{SU}^{mac}}(data)),$$

where X is the sender and can be U or S. The messages between SP and a MV has similar format but use K_{SM} and K_{SM}^{MAC} instead. In the following we show the step by step message flow among the three parties during a typical service request. Since the messages use the same encryption format, we only show the *plaintext* for clarity.

1. $S \rightarrow U$: (Req:Location)

   SP asks U its current location.

2. $U \rightarrow S$: (Loc_U)

User's GPS enabled cell phone detects its location and sends it to SP.

3. $S \rightarrow U : (Info(MV_1), Info(MV_2), ...Info(MV_n))$

   The SP looks (in its database) for the MVs whose service range covers the user's current location and sends the information associated with each of the MVs to the user. The information may include the MV's service type (e.g. ice cream truck, postal delivery, etc.), name/brand, distance from the user or reputation, depends on the application. An alternative implementation would have the user request for a specific types of service, and the SP would only return information associated with the MVs who provide that type of service.

4. $U \rightarrow S : (Req : MV_i, MWT)$

   U makes his choice and sends the request for MV_i to the SP, together with his maximum waiting time (MWT) that states how long he can wait for the service to arrive, which helps the MV to decide whether it can make the service.

5. $S \rightarrow M_i : (Req_{ID}, Loc_U, MWT)$

   SP assigns a request ID for this Request and send it together with the user's location and MWT to MV_i.

6. **if** MV_i decides to provide the service, **then**

   $M_i \rightarrow S : (Accept, Req_{ID}, EAT)$

   $S \rightarrow U : (Trans_{ID}, MV_i, EAT)$

   else

   $M_i \rightarrow S : (Reject, Req_{ID})$

   $S \rightarrow U : (MV_i$ not available)

   The MV_i calculates and sends its estimated arrival time (EAT) based on the its distance to Loc_U, its moving speed, the service requests already in his queue. If $EAT \leq MWT$, MV_i can choose if it will accept the request or not, but if $EAT \geq MWT$, the program will automatically reject the request. If it decides to provide the service, it includes the EAT in the response. The SP then assigns a transaction ID $Trans_{ID}$ to this transaction, and sends this to U together with

EAT of MV_i. If the MV_i rejects the request, the message is forwarded to the U, who can start another request.

7. $U \rightarrow S$: (Service Evaluation)

   $M_i \rightarrow S$: (User Evaluation)

   If the MV_i accepted the request, both U and MV_i send the evaluation for the other party to SP. This is intended to support the use of reputation systems to assist vendors and users in identifying entities with a history of unscrupulous behavior.

6.6 Conclusion

Due to the vulnerabilities associated with wireless networks and mobile devices, it is critical to ensure that emerging mobile location-based services (MLS) operate in a secure and trustworthy manner. In this work, we proposed a security architecture for mobile location-based services, which involved providing trustworthy location information to support mobile services, as well as a holistic protocol framework describing the proper interaction between mobile users and mobile vendors. Our approach to improving the trustworthiness of location information takes advantage of auxiliary networks enabled by the increasingly wide deployment of wireless technologies and the fact that there will be a high density of access points or other wireless transmitters in the future. Specifically, we designed a two-level location verification scheme – coarse-grained and fine-grained scheme that facilitate the important step of position verification that is essential to designing a the trustworthy MLS. We then integrated these methods into a traditional network security framework where we describe the sequence of protocol steps that should take place between the user/customer, (wireless) service providers, and mobile vendors. In particular, we used certificates as the basis for initial trust in a setup phase, whereby users and vendors register with the service provider, as well as presenting the corresponding sign-out steps. We described how location information should be updated in such a service, and draw the connection between our location verification schemes and how they are important in updating position information.

Lastly, we described a procedure for requesting and accepting service from a mobile vendor.

Chapter 7

Conclusion

7.1 Thesis Contributions

The development of low-cost, ubiquitous, wireless systems is leading to a future where location will define the next generation of computing applications. Location-based services have shown a great potential to improve our life in many ways. However, before an LBS system can be launched to the mass market successfully, we need to ensure that it operates in a secure and trustworthy way. This thesis explored and proposed methods to provide trustworthiness to the operations of LBS.

We mapped out three steps needed to achieve the security and trustworthiness requirement: First, identify the security policies that regulate the LBS application. Towards this end, we proposed a policy representation model and provided different ways to represent policies. Second, enforce the security policies. We explored the feasibility of using environmental properties for wireless localization and location verification and a Key Distributed-based Location Verification method. We also presented a noninteractive infrastructure that makes it possible to enforce the security policies without knowing users location information. Thirdly, the whole system itself should be put into a security framework to prevent manipulation by unscrupulous entities participating in the service. We presented a security architecture and protocols to provide end-to-end trustworthy communication for mobile location based services.

More specifically, our contributions in this thesis include:

- We proposed a policy model to regulate LBS. Particularly we captured the spatial and temporal character in LBS and proposed the concept of Spatio-Temporal

regions (ST-regions), streaming objects, ST-region decomposition, and object decomposition, which makes it possible to define more flexible and powerful policies. We also proposed the concept of stateful policies where the grant of a user's access is based on his previous spatio-temporal behavior. We showed how to use automata to represent stateful policies.

- We proposed a parameter selection mechanism – Spatio-Correlation Weighting Mechanism (SCWM), which is an effective tool to predict the effectiveness of a subset of parameters when used in localization. Given a set of parameters, SCWM can guide in parameter selection by determining the optimal parameter combination that will have high discriminative power, which will result in good localization accuracy when this subset is used for localization. Our experiments show the effectiveness of SCWM in predicting the performance of parameters.

- We showed the feasibility of utilizing the inherent spatial variability in environmental parameters to assist in wireless localization and position verification. In order to perform localization, we developed the Flex-EP algorithm series, which use a globally-optimal parameter subset all at once, and the Prog-Flex-EP algorithm series which are sequential algorithms that locally customizes the best set of parameters for each user. Under each of these two algorithm categories, we developed different algorithms using closest distance and maximum likelihood as measuring metrics respectively. Through our experiments, we found that using two environmental parameters containing high discriminative power and two with low discriminative power is enough to produce comparable performance to the traditional localization approach RADAR, which employs RSS, with at least four access points. By increasing the number of parameters with high discriminative power, we can further refine the localization accuracy. For example, in our experiments we showed that when we used 3 environmental parameters with high discriminative power in a five-parameter subset, the localization performance has a 10% increase compared to RADAR.

- We proposed the Key Distribution-based Location Verification (KDLV) scheme

which is a generic approach that takes advantage of an auxiliary network of transponders that distribute "location-security" keys to the users' claimed locations and the problem of position verification is resolved by verifying the received keys by users. We derived an analytical model to study the relationship between the node density in the auxiliary network and the location verification accuracy. Our analysis provides useful insights about how to deploy auxiliary networks to facilitate trustworthy LBS.

- We further extended the idea of key distribution and designed a noninteractive LBS infrastructure. In this infrastructure, the content or services are encrypted and broadcasted to the whole area of interests and we enforce the security policies of LBS system by control the distributing of decryption keys. This infrastructure is especially useful for applications such as spatio-temporal access control (STAC), where the enforcement of security access policies are essential to the service. In our noninteractive infrastructure, the enforcement of the security policies does not rely on getting the locations of the users, so the risks caused by location spoofing is bypassed. In addition, users do not need to expose their location information to the third party such as the network operators or service providers, which is a great advantage in protecting users from a potential privacy breach. In addition, in this work we designed an algorithm which optimizes the region covered by the transponders by optimizing power allocation.

- We proposed a security architecture for mobile location based services (MLS), which is a more complicated form of LBS since both the customers and the services are mobile. In our architecture, we took into consideration the particular security threats and attacks that an MLS system might face and integrated two levels of location verification schemes and reputation system into traditional network security framework. We described the sequence of protocol steps, how location information should be updated in such a service and the procedure of requesting and accepting service from a mobile vendor. Our architecture represents an example of how an LBS can be made secure and trustworthy with the

appropriate analysis of the security needs in the services and using a combination of technical tools. It serves as a guideline for further secure LBS development.

7.2 Future Work

Looking forward, the work in this thesis suggests a number of research topics in the future. First, we have mapped out a step by step security architecture and protocols for LBS systems. When we seek to implement such a system in reality, there will be issues that need to be addressed, such as energy consumption, network delay, and interoperability between network operators. Considering these practical issues could represent a large hurdle to address, and our protocol framework would only serve as starting point for the development of a holistic, real-world system.

Second, the issue of privacy. Our noninteractive infrastructure is resistant to privacy breaches because it does not rely on a procedure that acquires users' location information. Such an approach, however, is only applicable to a fraction of LBS applications. In the more general case, we assume the trustworthiness of network operators and service providers. More research is needed to address the details about who owns the location data, how this location data can be shared between participants in the system, while adhering to both access control policies *and* privacy policies.

Third, but not the last, at the application level, we see exciting opportunities to develop new classes of location-based applications. For example the spatio-temporal scavenger hunt mentioned in Chapter 5 might be an interesting paradigm for educational applications. The development of new applications would serve as a means to further drive security and privacy research in Location Based Services.

References

[1] Prosenjit Bose, Pat Morin, Ivan Stojmenovic, and Jorge Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *Wireless Networks*, vol. 7, no. 6, pp. 609–616, November 2001.

[2] Brad Karp, "Gpsr: Greedy perimeter stateless routing for wireless networks," 2000, pp. 243–254.

[3] Young bae Ko and Nitin H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," 1998, pp. 66–75.

[4] Ya Xu, John Heidemann, and Deborah Estrin, "Geography-informed energy conservation for ad hoc routing," in *ACM MOBICOM*, 2001, pp. 70–84.

[5] Xiaoyan Hong, Kaixin Xu, and Mario Gerla, "Scalable routing protocols for mobile ad hoc networks," in *IEEE Network Magazine*, 2002.

[6] Mo Li, Yunhao Liu, and Lei Chen, "Nonthreshold-based event detection for 3d environment monitoring in sensor networks," *IEEE Trans. on Knowl. and Data Eng.*, vol. 20, no. 12, pp. 1699–1711, 2008.

[7] Joseph Polastre, Robert Szewczyk, Alan Mainwaring, David Culler, and John Anderson, "Analysis of wireless sensor networks for habitat monitoring," pp. 399–423, 2004.

[8] David Moore, John Leonard, Daniela Rus, and Seth Teller, "Robust distributed network localization with noisy range measurements," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, New York, NY, USA, 2004, pp. 50–61, ACM.

[9] Yunhao Liu and Mo Li, "Iso-map: Energy-efficient contour mapping in wireless sensor networks," in *ICDCS '07: Proceedings of the 27th International Conference on Distributed Computing Systems*, Washington, DC, USA, 2007, p. 36, IEEE Computer Society.

[10] "3GPP TS 44.031," http://www.3gpp.org/ftp/Specs/html-info/44031.htm.

[11] "The guide to Geographic Information Systems," http://www.gis.com.

[12] Stefan Steiniger, Moritz Neun, and Alistair Edwardes, "Foundations of Location Based Services," University of Zurich.

[13] GSM Association, "Permanent Reference Document SE.23: Location Based Services," http://www.gsmworld.com/documents/se23.pdf.

[14] Wikipedia, "Enhanced 911," http://en.wikipedia.org/wiki/Enhanced_911.

[15] FCC, "Enhanced 9-1-1 - Wireless Services," http://www.fcc.gov/pshs/services/911-services/enhanced911/Welcome.html.

[16] LBS Insight, "NAVTEQ Trials Location-Based Advertising with Exiting Results," http://www.lbsinsight.com/?id=3307.

[17] Yingying Chen, *Securing Wireless Localization Against Signal Strength Attacks*, Ph.D. thesis, Department of Computer Science, Rutgers University, New Brunswick, NJ, 2007.

[18] S. Capkun and J.P. Hubaux, "Secure positioning in sensor networks," Technical report EPFL/IC/200444, May 2004.

[19] S. Capkun and J.P. Hubaux, "Securing localization with hidden and mobile base stations," Proceedings of IEEE Infocom 2006.

[20] L. Lazos, R. Poovendran, and S. Capkun, "Rope: robust position estimation in wireless sensor networks," in *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, 2005, pp. 324–331.

[21] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust Statistical Methods for Securing Wireless Localization in Sensor Networks," in *The Fourth International Conference on Information Processing in Sensor Networks (IPSN)*, 2005, pp. 91–98.

[22] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata theory, languages and computation*, Addison-Wesley Publishing Company, 1979.

[23] W. Han, J. Zhang, and X. Yao, "Context-sensitive access control model and implementation," in *The Fifth International Conference on Computer and Information Technology*, pp. 757–763.

[24] R. J. Hulsebosch, A. H. Salden, M. S. Bargh, P. W. G. Ebben, and J. Reitsma, "Context sensitive access control," in *SACMAT '05: Proceedings of the tenth ACM symposium on Access control models and technologies*, New York, NY, USA, 2005, pp. 111–119, ACM Press.

[25] E. Bertino, B. Catania, M. L. Damiani, and P. Perlasca, "GEO-RBAC: a spatially aware RBAC," in *SACMAT '05: Proceedings of the tenth ACM symposium on Access control models and technologies*, New York, NY, USA, 2005, pp. 29–37, ACM Press.

[26] J. Joshi, E. Bertino, U. Latif, and A. Ghafoor, "A generalized temporal role-based access control model," vol. 17, pp. 4–23, 2005.

[27] P. Bahl and V.N. Padmanabhan, "RADAR: An in-building RFbased user location and tracking system," in *Proceedings of IEEE Infocom 2000*, 2000, pp. 775–784.

[28] S. Chen, Y. Zhang, and W. Trappe, "Inverting Sensor Networks and Actuating the Environment for Spatio-Temporal Access Control," in *Proceedings of the Forth ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, 2006, pp. 1–12.

[29] Y. Chen, J. Francisco, W. Trappe, and R. P. Martin, "A Practical Approach to Landmark Deployment for Indoor Localization," in *Proceedings of the Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, September 2006.

[30] R. Chakravorty, S. Agarwal, S. Banerjee, and I. Pratt, "Mob: A mobile bazaar for wide-area wireless services," in *Proceeding of ACM MOBICOM*, 2005.

[31] H. Luo, R. Ramjee, P. Sinha, L. E. Li, and S. Lu, "Ucan: A unified cellular and ad-hoc network architecture," in *Proceeding of ACM MOBICOM*, 2003.

[32] L.K. Law, S.V. Krishnamurthy, and M. Faloutsos, "Capacity of hybrid cellular-ad hoc data networks," in *Proceeding of IEEE INFOCOM 2008*, 2008.

[33] Y.D. Lin and Y.C. Hsu, "Multihop cellular: A new architecture for wireless communications," in *Proceeding of IEEE INFOCOM 2000*, 2000.

[34] H. Wu, C. Qiao, S. De, and O. Tonguz, "Integrated cellular and ad hoc relaying systems," , no. 10, 2001.

[35] G. Casella and R. Berger, *Statistical Inference (2nd Edition)*, Duxbury Press, 2001.

[36] G. Zhou, T. He, S. Krishnamurthy, and J. Stankovic, "Models and solutions for radio irregularity in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 2, pp. 221–262, 2006.

[37] V. Erceg, L. Greenstain, S. Tjandra, S. Parkoff, A. Gupta, B. Kulic, A. Julius, and R. Bianchi, "An empirically based path loss model for wireless channels in suburban environments," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 7.

[38] B. Firner, P. Jadhav, Y. Zhang, R. Howard, W. Trappe, and E. Fenson, "Towards continuous asset tracking: Low-power communication and fail-safe presence assurance," in *Proceeding of IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON),*, 2009, pp. 1–9.

[39] Y. Gertner, S. Goldwasser, and T. Malkin, "A random server model for private information retrieval or how to achieve information theoretic PIR avoiding database replication," *Lecture Notes in Computer Science*, vol. 1518, 1998.

[40] G. D. Crescenzo, Y. Ishai, and R. Ostrovsky, "Universal service-providers for database private information retrieval (extended abstract)," in *Symposium on Principles of Distributed Computing*, 1998, pp. 91–100.

[41] C. Cachin, S. Micali, and M. Stadler, "Computationally private information retrieval with polylogarithmic communication," *Lecture Notes in Computer Science*, vol. 1592, 1999.

[42] S. Capkun and J. Hubaux, "Secure Positioning of Wireless Devices with Application to Sensor Networks," in *Proceedings of the IEEE INFOCOM*, 2005, pp. 1917–1928.

[43] M. Bishop, *Computer Security: Art and Practice*, Addison Wesley, 2003.

[44] S. Gavrila D. Ferraiolo, R. Sandhu, D. Richard Kuhn, and R. Chandramouli, "Proposed NIST standard for Role-Based Access Control," *ACM Transactions on Information and System Security*, vol. 4, no. 3, pp. 224–274, 2001.

[45] L. Lazos and R. Poovendran, "SeRLoc: Secure range-independent localization for wireless sensor networks," in *Proceedings of the 2004 ACM Workshop on Wireless Security*, 2004, pp. 21–30.

[46] D. Liu, P. Ning, and W. Du, "Attack-resistant location estimation in sensor networks," in *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, 2005.

[47] D. B. Faria and D. R. Cheriton, "No Longterm Secrets: Location-based Security in Overprovisioned Wireless LANs," in *Proceedings of the Third ACM Workshop on Hot Topics in Networks*, 2004.

[48] N. Michalakis, "PAC: Location Aware Access Control for Pervasive Computing Environments," 16 September 2002.

[49] N. Michalakis, "Location-aware Access Control for Pervasive Computing Environments," Master of Engineering Thesis, Department of Electrical Engineering and Computer Science, MIT, 2003.

[50] B. Schilit, J. Hong, and M. Gruteser, "Wireless Location Privacy Protection," *Computer*, vol. 36, no. 12, pp. 135–137, 2003.

[51] M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald, "Privacy-aware location sensor networks," in *Workshop on Hot Topics in Operating Systems (HotOS)*, 2003.

[52] M. Gruteser and D. Grunwald, "Anonymous Usage of Location-based Services through Spatial and Temporal Cloaking," in *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2003.

[53] S. Duri, M. Gruteser, X. Liu, P. Moskowitz, R. Perez, M. Singh, and J. Tang, "Context and Location: Framework for security and privacy in automotive telematics," in *Proceedings of the 2nd international workshop on Mobile commerce*, 2002.

[54] Frank Stajano and Ross Anderson, "The resurrecting duckling: Security issues for ad-hoc wireless networks," *Book Series Lecture Notes in Computer Science*, vol. 1796/2000.

[55] Y.C. Hu, A. Perrig, and D.B. Johnson, "Wormhole attacks in wireless networks," vol. 24, no. 2, pp. 370–380, 2006.

[56] K. 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 2004 Symposium on Cryptography and Information Security*, 2004.

[57] John Douceur and Judith S. Donath, "The sybil attack," in *Proceeding of the 1st International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002, pp. 251–260.

[58] C.S.R. Prabhu, *Mobile Computing: A Book of Readings*, Orient Blackswan, 2004.

[59] James Newsome, Elaine Shi, Dawn Song, and Adrian Perrig, "The sybil attack in sensor networks: analysis & defenses," in *IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks*, New York, NY, USA, 2004, pp. 259–268, ACM.

[60] Jie Yang, Y. Chen, and W. Trappe, "Detecting Sybil Attacks in Wireless and Sensor Networks Using Cluster Analysis," in *Proceedings of the Fourth IEEE International Workshop on Wireless and Sensor Networks Security (WSNS)*, 2008.

[61] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation Systems: Facilitating Trust in Internet Interactions," *Communications of the ACM*, vol. 43, 2000.

[62] Audun Josang, Roslan Ismail, and Colin Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, 2007.

[63] Y. Sun and Y. Yang, "Trust establishment in distributed networks: Analysis and modeling," in *Proceedings of IEEE ICC*, 2007.

[64] Th. G. Papaioannou and G. D. Stamoulis., "Achieving honest ratings with reputation-based fines in electronic markets," in *Proceedings of IEEE INFOCOM 2008*, 2008.

[65] Y. Yang, Y. Sun, S. Kay, and Q. Yang, "Defending Online Reputation Systems against Collaborative Unfair Raters through Signal Modeling and Trust," in *Proceedings of the 24th ACM Symposium on Applied Computing (ACM SAC'09)*, 2009.

[66] Y. Sun, Z. Han, and K. J. Ray Liu, "Defense of trust management vulnerabilities in distributed networks," *IEEE Communications Magazine, Feature Topic on Security in Mobile Ad Hoc and Sensor Networks*, vol. 46, no. 2, 2008.

[67] ITU-T, "The directory: authentication framework," .

[68] W. Trappe and L. Washington, *Introduction to Cryptography with Coding Theory (2nd Edition)*, Prentice-Hall, Inc., 2005.

[69] Mihir Bellare, Ran Canetti, and Hugo Krawczyk, "Keying hash functions for message authentication," in *CRYPTO '96: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, London, UK, 1996, pp. 1–15, Springer-Verlag.

[70] Mathias Bohge and Wade Trappe, "An Authentication Framework for Hierarchical Ad Hoc Sensor Networks," in *Proceedings of the 2003 ACM WiSE*, 2003, pp. 79–87.

Vita

Shu Chen

2000 **B.S. in Computer Science and Engineering,**
Harbin Institute of Technology, Harbin, China

2002 **M.S. in Computer Science and Engineering,**
Harbin Institute of Technology, Harbin, China

2005 **M.S. in Computer Science,**
Rutgers University, New Brunswick, New Jersey, USA

2010 **Ph.D. in Computer Science,**
Rutgers University, New Brunswick, New Jersey, USA

Selected Publications

2006 "Inverting Sensor Networks and Actuating the Environment for Spatio-Temporal Access Control". Shu Chen, Yu Zhang, Wade Trappe. Proceedings of 4th ACM Workshop on Security of Ad-Hoc and Sensor Networks, held with CCS, 2006.

2007 "Exploiting Environmental Properties for Wireless Localization". Shu Chen, Yingying Chen, Wade Trappe. student poster, 13th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), Montreal, Sep. 2007.

2008 "Exploiting Environmental Properties for Wireless Localization and Location Aware Applications". Shu Chen, Yingying Chen, Wade Trappe. IEEE International Conference on Pervasive Computing and Communications (Percom 2008), Hong Kong, Mar. 2008.

2008 "Exploiting Environmental Properties for Wireless Localization". Shu Chen, Yingying Chen, Wade Trappe. ACM SIGMOBILE Mobile Computing and Communications Review (MC2R), Volume 12, Issue 1, Pages 49-51, 2008.

2009 "Inverting Systems of Sensors for Position Verification in Location-Aware Applications". Shu Chen, Yingying Chen, Wade Trappe. IEEE Transactions on Parallel and Distributed Systems (IEEE TPDS), Vol. 20, Issue 12, 2009.