CAP : A *C*ONTEXT-*A*WARE *P*RIVACY PROTECTION SYSTEM FOR LOCATION-BASED SERVICES

by

ANIKET PINGLEY

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON
August 2008

To my late grandmother Sushila, and my parents Anita and Santosh, whose blessings are always with me.

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Nan Zhang for guiding me through this research project *and* for providing me with financial support. My association with him has been an invaluable experience for me, *and* has given me a new perspective towards research. I am grateful to him for his patience *and* for inspiring me to overcome my failures.

I am grateful to Dr. Matthew Wright and Dr. Donggang Liu for their interest in my research *and* for taking time to serve on my thesis committee. I am grateful to the research collaborators Dr. Xinwen Fu, Dr. Wei Zhao and Dr. Wei Yu without whom this project would not have been a success.

I also thank all the other members of iSEC lab at UTA for their help and support. I would like to thank all my cousins and friends for their support and encouragement. I would especially like to thank Jaideep Padhye, for his invaluable help during this project. I wish to thank Manali Kide, who has always helped me in relieving stress during tough times.

Most importantly, I would like thank my parents, whose belief in my abilities has been most vital for what I am today. I deeply value all the sacrifices they made for me *and* their never ending support. A very special thanks to my sister Aishwarya, who is the most dear person to me.

July 10, 2008

ABSTRACT

CAP : A *C*ONTEXT-*A*WARE *P*RIVACY PROTECTION SYSTEM FOR LOCATION-BASED SERVICES

Aniket Pingley, M.S.

The University of Texas at Arlington, 2008

Supervising Professor: Dr. Nan Zhang

Location Based Services (LBS) are information services that provide users with customized contents, such as the nearest restaurants/hotels/clinics, retrieved from a dedicated spatial database. They make use of technologies such as Global Positioning System (GPS), triangulation/triliteration etc. to get the geographical position of the user. Since the queries on spatial database include the user's current location, LBS may raise serious concerns on the user's location privacy. If disclosed, a user's location information may be misused in many ways by a malicious adversary who has access to the LBS server or even by the LBS provider. Therefore, our aim in this thesis is to provide an user with a system to protect her location privacy, without impeding the LBS.

In this thesis, we address issues related to privacy protection for location-based services (LBS) without trusted-third parties (e.g., anonymizers). There are two critical challenges to such a system. First, the degree of privacy protection and LBS accuracy depends on the context, such as population and road density, around a user's location e.g. for an user in rural area, we use more perturbation than for an user in downtown to achieve the same level of privacy protection and LBS accuracy. Second, location privacy may be breached through not only an LBS query, but also via the network traffic that carries the query payload, leading to a dual requirement on data privacy and communication anonymity. In order to address these challenges, we introduce CAP, a *C*ontext-*A*ware *P*rivacy-preserving LBS system with integrated protection for data privacy and communication anonymity. For data privacy, we propose a projection-based location data perturbation algorithm, called Various-size-grid Hilbert Curve (VHC) - mapping, which provides universal guarantees on privacy protection and LBS accuracy for all locations with diverse context. VHC-mapping is designed to require minimal storage and computational cost. For communication anonymity, CAP uses a revised version of Tor. In this revised version, we address the issue of QoS degradation due to Tor's random routing protocols. By exploiting the dual requirement with data privacy, we propose a set of new routing algorithms with significantly enhanced QoS. We have implemented a prototype of CAP which can be readily integrated with an existing LBS. Our theoretical analysis and experimental results validate CAP's effectiveness on privacy protection, LBS accuracy, and communication QoS.

TABLE OF CONTENTS

Chapter

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Location information is a set of data describing an individual's geographical location (e.g. latitudinal and longitudinal co-ordinates). This information can be used in many ways to provide information and entertainment services to the user who has the proper resources and infrastructure to avail such services. Today, most of the hand held devices, such as PDAs, have the necessary hardware (e.g. Global Positioning System), inbuilt. This combined with the device's ability to connect to the Internet from any geographical location has resulted in immense amount of information being made available to the users by the location based services, merely at a click of a button.

Location Based Services (LBS) are information services that provide users with customized contents, such as the nearest restaurants/hotels/clinics, retrieved from a dedicated spatial database based on the user's current location. The LBS can obtain user's geographical position/location by making use of technologies such as Global Positioning System (GPS), triangulation/triliteration etc.: For example, the *My Location* feature in the Mobile Google Maps can obtain the approximate geographical location of a mobile device by using the technology of triangulation/triliteration [17]. Thus, the main advantage of LBS is that a user does not have to manually specify location identifiers, such as zip codes, for using it. Not only as information services, but LBS are also being used as entertainment services [4]. Thus, due to prolific development and application of Geographical Information System (GIS), spatial databases, and mobile devices that have access to both the Internet and the GPS, the popularity of LBS is rapidly growing. LBS not only serves individual mobile users, but also plays an important role in public safety, transportation, emergency response, and disaster management. As such, the number of GPS-enabled LBS subscribers is predicted by the ABI research report [39] to reach 315 million in 5 years.

Privacy is a fundamental human right which can be broadly classified into four categories: Information privacy, Bodily privacy, Privacy of communications and Territorial privacy [37]. This thesis talks about *location privacy*, a particular type of information privacy, which was defined in [2] as the *ability to prevent other parties from learning one's current or past location.* In this thesis we have extended this definition from the point of view of a LBS server/provider or a malicious adversary. We define location privacy as *the probability with which a LBS server/provider or a malicious adversary can guess the presence of a user at a particular location, landmark or area.* Thus, we state that if the LBS server/provider or the malicious adversary has more locations/landmarks to choose from where the user can be present, then location privacy is better protected.

With the advent of pervasive computing, LBS are making a variety of information available to the user. While such services make life easier for the users, they are raising serious concerns over the location privacy of the user. For example, a user may feel threatened to know that her visits to a hospital and then a pharmacy store for past few weeks are being learned of. Thus, the problem of location privacy being compromised, which was not seen as threat until recently, is being viewed as a major concern today.

Generally, a request for LBS can be considered as a query on the server's spatial database. The query has the user's current location specified in the selection conditions. For example, using SQL-like top-k query syntax, a query for the nearest hotel can be expressed as:

SELECT TOP 1 FROM *Hotel*

ORDER BY DISTANCE(*Hotel.Location*, *userLoc*) ASC;

where *userLoc* is the user's current location, and ASC indicates the ascending order.

Since the user's current location is included in the query sent to the server through the Internet, LBS may raise serious concerns on the user's (location) privacy. If disclosed, a user's location information may be misused in many ways by a malicious adversary who has access to the LBS server *or* by the LBS provider: For example, the adversary may learn a user's political and religious affiliations based on the locations the user visits. There have been several reports on the abuse of LBS by individuals and companies to intrude others' privacy [33, 44]. Thus, it is critical to protect a user's location privacy without impeding the LBS.

The problem of preserving location privacy has received growing attention from the research community. In [23, 3, 12, 18, 29], a k-anonymity [45] based framework was proposed to protect location privacy using a trusted third-party service called the *anonymizer*. With this framework, a user sends the location to the centralized anonymizer, which subsequently generates a *cloaking region* which covers not only this user, but also other $k-1$ users. Then, the anonymizer transmits the cloaking region to the LBS and forwards the query result to this user.

This framework can be transparent to the LBS server/provider, and prevents an adversary with access to the LBS server/provider from distinguishing a user from at least other $k-1$ users. Nonetheless, it may be difficult to find such a trusted third-party anonymizer in practice. Even if the anonymizer existed, it would create a single point-of-attack/failure and is usually the bottleneck for system efficiency. Furthermore, to be effective, this framework requires a large number of users to subscribe to the anonymizer service. However, a user's location may be compromised if the LBS server/provider colludes with other $k-1$ users in the system.

A recent work based on private information retrieval (PIR) has been shown to remove the requirement of a trusted third-party [15]. However, this approach is no longer transparent to the LBS server/provider, and thus cannot provide privacy protection if the LBS does not support the PIR protocol. Moreover, a well-known problem of PIR is its high computational and communication overhead [42]. It was shown that PIR may incur even longer overhead than an oblivious transfer of the entire server-side database [42]. Such a cost may be prohibitive for mobile devices and LBS servers, which need to process a large number of queries in real-time.

In this thesis, we consider systems without a trusted third-party middleware/anonymizer. Instead, we focus on a simple user-driven privacy-preserving technique, with which a user perturbs her location locally before transmitting it to the LBS server/provider. Advantages of this scheme include the following -

- *Transparency:* The LBS server/provider is not aware about who issued the request.
- *Universality:* The scheme can be readily integrated with existing LBS systems e.g., Google Maps, Mapquest etc. Additionally, our scheme works equally well irrespective of the type of location a user is in e.g., rural area or a downtown.
- *Customizability:* A user can flexibly control the tradeoff between privacy protection and accuracy for LBS. This essentially means that our scheme does not require users to have high computational power or large storage space. For example, mobile phones have much lesser computational power and storage space than a desktop computer.

However, to such systems there are three major challenges as follows -

- *Context-Aware Heterogeneous Perturbation:* The perturbation must be customized by the "context", such as road or population density, of a user's location. The intuition behind this is twofold i) There are more landmarks (e.g., coffee shop, shopping mall, hospital, bookstore etc.) inside a downtown, than inside a rural area. Here, note the fact that a user is present inside a particular landmark *or* near it, may reveal some information about user to the LBS server/provider *and* ii) The amount of perturbation affects the accuracy of the scheme. Intuitively, to achieve the same level of privacy protection and LBS accuracy, a user should perform a "stronger" perturbation in the country than in downtown. A naïve solution is to access a topology map in real-time and decide the granularity of perturbation. However, this may lead to excessive computational and/or storage cost. Thus, the critical challenge is how to perform context-aware perturbation in an efficient manner.

- *QoS of Anonymous Communication:* Since a user's location may be derived from her IP address [14], location privacy may be breached through not only an LBS query, but also via the traffic that carries the query payload. To achieve communication anonymity, many researchers have proposed to adopt anonymous communication services provided by anonymous communication networks such as Tor [10]. Unfortunately, our real-world experiments showed that Tor and other similar anonymous communication networks often suffer from serious QoS degradation (see Chapter 6), which must be properly addressed for a practical LBS system.
- *Dual Requirements on Data Privacy and Communication Anonymity:* Since the level of privacy disclosure is determined by the "shorter board" of data privacy and communication anonymity, the overhead introduced by communication anonymity services should be the minimum necessary to "match" the protection level provided by data perturbation. This calls for an effective integration of location perturbation (for data privacy) and communication anonymity.

In this thesis, we introduce *CAP*, a *C*ontext-*A*ware *P*rivacy-preserving LBS system with integrated protection for both data privacy and communication anonymity.

- To achieve context-aware perturbation, we propose a location perturbation component based on road density information. In particular, we introduce Various-size-grid Hilbert Curve (VHC)-mapping, a locality-preserving mapping from any 2-d geographical map to a 1-d space, such that every location within the 1-d space has equal road density and thus can be perturbed in a homogeneous manner. VHC-mapping is designed to be computed offline with minimal storage and retrieval cost. With VHC-mapping, the perturbed 2-d location is derived by mapping the result of homogeneous perturbation in the 1-d space back to the original 2-d space.

- To achieve enhanced QoS, we propose an anonymous routing component which dynamically customizes the routing strategy of Tor based on the result of data perturbation. We found that Tor suffers from serious QoS degradation because of its random routing protocols. To satify the dual requirement, we propose a set of new routing algorithms, with which Tor achieves the same degree of protection on location privacy as the location perturbation component. Our anonymous routing component provides significantly enhanced QoS on Tor.

To the best of our knowledge, CAP is the first real system that provides a comprehensive solution for both data privacy and communication anonymity, and can be readily applied to existing location-based services. It is also the first that measures and addresses the QoS concerns for privacy-preserving LBS systems. We have implemented a prototypical system and expect to release it to the public in the near future. We have also performed theoretical analysis of the quantitative impact of CAP on privacy protection and LBS accuracy, and present a comprehensive set of experiments that demonstrate the effectiveness of CAP.

This thesis is organized as follows: In Chapter 2 we provide an overview of the CAP, and define the accuracy and privacy guarantees provided by it. Chapters 3 and 4 describe in the detail the two central components of CAP, namely location perturbation component and anonymous routing component. In Chapter 5, we deal with the design of

CAP and discuss in details about the implementation of the two components. Chapter 6 discusses about the experimental setup and results. In Chapter 7, we mention in brief about the related work and conclude in Chapter 8.

CHAPTER 2

SYSTEM OVERVIEW

In this chapter we present an overview of CAP, our context-aware privacy-preserving LBS system with integrated protection for both data privacy and communication anonymity. We will first introduce the system architecture and its major components. Then, we will define the performance measures for CAP. The detailed design of the components will be presented in Chapter 3 and 4.

2.1 System Architecture

Figure 2.1 illustrates the baseline architecture of CAP. As we can see, CAP involves four (types of) entities: positioning devices, mobile clients, anonymous network, and LBS server. We introduce each entity respectively as follows-

- *Positioning Device:* It senses the user's location (e.g., latitude and longitude). The passive positioning techniques such as GPS do not expose the user's location to the positioning infrastructure. On the other hand, it is extremely difficult, if not impossible, to hide location from an active positioning infrastructure (e.g., triangulation of cellular signal) [40, 34]. For the purpose of this thesis, we do not consider privacy disclosure through the positioning infrastructure.
- *Mobile Client:* It obtains the location (e.g., latitude and longitude) from the equipped positioning device. Then, it employs two main components:
  - The *location perturbing component* perturbs the location and sends the result, along with the LBS query, to the anonymous routing component. This achieves location privacy.

  - The *anonymous routing component* uses the received (perturbed) coordinates to configure the routing protocol of anonymous communication network. This achieve communication anonymity and QoS.

  With these two components, a mobile client then sends the LBS query to the LBS server via the anonymous communication network.

- *Anonymous Communication Network:* It relays the query to prevent the LBS server from tracing back to the mobile client and discovering the user's network address and location. For example, Tor is a popular anonymous communication network that consists of thousands of donated computers distributed around the world as Tor routers. Tor uses onion routing to relay user traffic to and from the LBS server [10, 9].

- *LBS Server:* It receives the LBS query and processes it against the spatial database. Then, the server returns the results to the mobile client through the anonymous communication network. Note that the presence of anonymous communication network is transparent to the LBS server.

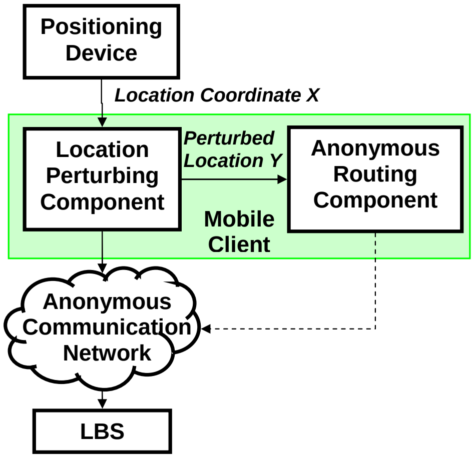

Positioning Device
Location Coordinate X
Location Perturbing Component
Perturbed Location Y
Anonymous Routing Component
Mobile Client
Anonymous Communication Network
LBS

Figure 2.1: Baseline Architecture of CAP

2.2 Performance Measures

Clearly, the performance of a privacy-preserving LBS system should be measured in terms of location privacy, LBS query accuracy and (communication) QoS of the entire system. We define these three measures respectively as follows:

2.2.1 Privacy Measure

Location privacy is subjective to the user. Information that is extremely sensitive to one user may be divulge-able to another. For example, for some users, an adversary knowing that they are in Dallas may mean a complete breach of their location privacy, but some might be insensitive to this fact (being in a particular city and adversary knowing about it). What might be important for some users is the fact that the adversary does not know about their exact location e.g., Starbucks Coffee shop in the campus of UT Arlington. This means that the preciseness of the location information required to breach location privacy depends on what a particular user is impervious to. For example, an adversary may be able to infer that the user is of Hindu religion but the user may not be sensitive to divulging it. Consider the following two cases-

- *Case 1:* Instead of advertising only one location (user's real location in Arlington city), a user advertises three more locations from three different cities, e.g., Chicago, Baltimore and San Jose. In this case the adversary would have 25% confidence that the user is in either of the locations *and* either of the cities.
- *Case 2:* Instead of advertising the real location, user advertises a fake location (real *and* the fake locations are in Arlington city). Thus the adversary has 100% confidence that the user is in Arlington but only 1% confidence that user is inside a particular building in Arlington.

Users who consider that their location privacy is breached if the adversary knows the city in which they were *or* are, would find Case 1 to be a better approach. The same might not be true for a user whose exact location (particular building in Arlington) being disclosed means a complete breach of his location privacy, and thus the user would find Case 2 to be a better approach. Thus there is no single technique/dimension by which privacy can be measured. In addition to this, since we no longer require the presence of a trusted third-party anonymizer, the traditional definition of k-anonymity cannot be directly applied. Thus we define a (ϵ, N)-privacy guarantee as follows :

Definition 2.2.1. *A privacy-preserving mechanism satisfies* (ϵ, N)*-privacy guarantee if and only if for any given user and any geographical area with population less than* N*, the confidence the LBS server has on the presence of the user in the area is always less than* ϵ*.*

Note that "population" in the definition is loosely defined. For example, it can refer to the number of residents living/working in a geographical area, the number of users currently located in the area, or the number of users who have a historic footprint in the area. All these three instances are strongly correlated - A resident-dense area is also likely to have more users/historic footprints. The traditional k-anonymity definition [18] and its time-series extension [55] can be considered as adopting the second and the third instances, respectively. In this thesis, we mainly focus on addressing the first instance.

Also note that a privacy-preserving scheme may simultaneously satisfy multiple privacy guarantees, essentially defining a *privacy skyline*. For example, k-anonymity for location privacy can be considered as achieving $(1/k, 1), (2/k, 2), \ldots, (1, k)$-privacy guarantees. Figure 2.2 defines the corresponding privacy skylines for $k = 5$ and $k = 10$. As we can see, for two given (different) skylines A (e.g., $k = 10$) and B (e.g., $k = 5$), if the ϵ-value of A is less than or equal to B for all possible values of N, then A provides better privacy protection than B.

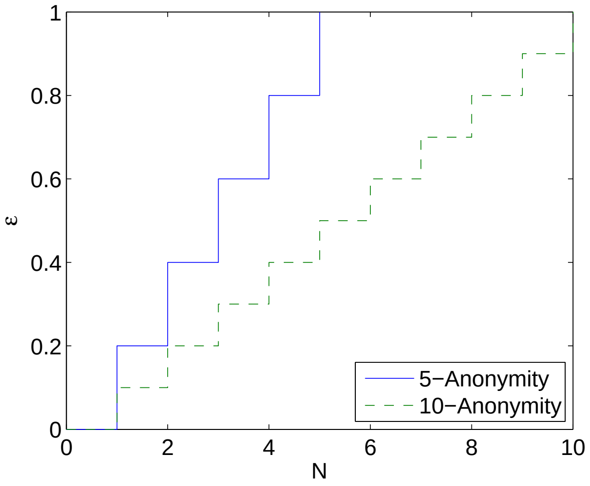


Figure 2.2: Privacy Skyline with (ϵ, N)-Privacy Guarantee

2.2.2 Accuracy Measure

Throughout the thesis, our discussion revolves around the creation, transmission and processing of LBS queries. We start by defining the simplest LBS query instances. Consider an LBS server which holds a spatial database T of n tuples, each of which is corresponding to a *point of interest* (POI) such as restaurants, etc. Let us follow a common assumption that an LBS query is a nearest neighbor (NN) query [15] which returns the POI(s) closest to a user's current location. Using top-k query syntax, we can describe such an LBS query as

q: SELECT TOP k FROM T

ORDER BY DISTANCE(T.Location, *userLoc*) ASC

For the purpose of this thesis, we consider the simplest case of $k = 1$. However, note that all results in the thesis can be easily generalized to all possible values of k.

Suppose that $q(x)$ is the LBS query answer when *userLoc* $= x$. We define an accuracy measure as follows:

Definition 2.2.2. *The degree of LBS accuracy of a privacy-preserving mechanism which perturbs userLoc from x to $R(x)$ is*

$$l_{\mathrm{r}} = \Pr\{q(R(x)) = q(x)\}. \tag{2.1}$$

Intuitively, l_{r} is the probability that the privacy-preserving mechanism returns the correct answer of the LBS query. An interesting fact to be noticed here is that even accuracy is subjective to the user as well to the query. This can be explained with following two examples-

- *Example 1:* Consider the perturbation distance (distance from x to $R(x)$) to be 1 mile. This means that the user might have to cover an additional 1 mile before reaching the nearest location for which she had queried. A user who will be walking to the nearest location may consider an an inaccuracy of 1 mile to be very high while this might not be true for a user who will be driving.
- *Example 2:* Consider a user issuing a query for nearest restaurant. Instead of the correct option *Subway* restaurant, a user is returned with *Chipotle* restaurant. If a user had requested for nearest *Subway*, then the result would have been accurate provided there is not another *Subway* very near to the one which is being returned as the result. This means that the preciseness of the query (specifying the restaurant) is directly related to the accuracy of the result.

In this thesis we do not address the issue of change in accuracy being subjective to the user *or* to the query. Thus developing an interactive application to provide better accuracy to a user is a part of our future work.

2.2.3 QoS Measure

In an LBS system, since a mobile client may be moving around, the LBS queries must be answered in a timely fashion. Unfortunately, our real-world experiments presented in Chapter 6 show that Tor may suffer from serious QoS (e.g., throughput) degradation due to its random routing protocol. In order for a privacy-preserving LBS system to be practical, we need to propose routing mechanisms that achieve enhanced QoS for Tor.

CHAPTER 3

LOCATION PERTURBING COMPONENT

In this chapter, we introduce the location perturbing component of our CAP system. This component provides data privacy to the user by making use of a technique called VHC-mapping which generates a fake location (latitude and longitude) to be sent to the LBS server. We will first describe the basic ideas, then the technique of VHC-mapping, and later present the detailed algorithms. We will also provide theoretical analysis on the degree of LBS accuracy and privacy guarantees achieved by the location perturbing component.

3.1 Input to Location Perturbing Component

The objective of location perturbation is to make a proper tradeoff between privacy protection and LBS accuracy. Both measures depend on the context, such as the population and road density, of a user's current location. The privacy guarantee defined in Definition 2.2.1 clearly depends on the population density. The same applies to LBS accuracy, as an error of few blocks in downtown with high road density is more likely to change the nearest POI than in a rural area.

Thus, location perturbation must be performed based on contextual information such as population and road density. Fortunately, economic studies show that these two densities are strongly correlated, following (approximately) a linear relationship [16]. Thus, we only need to consider one type of density information. In this thesis, we use road density as input to the location perturbing component. The information about road density is provided by Topological Integrated Geographic Encoding and Referencing

(TIGER) system published by the US Census Bureau. This information is not exactly in the form of road density but is in the form of latitude and longitude of a road *or* a part of it from which road density can be calculated.

3.2 Key Idea

To achieve context-aware location perturbation, we must perform "heterogeneous" perturbation to locations with different road density. The intuition behind this is twofold-

- There are more landmarks (e.g., coffee shop, shopping mall, hospital, bookstore etc.) inside a downtown, than inside a rural area. Here, note that the fact that a user is present inside a particular landmark *or* near it, may reveal some information about user to the LBS server/provider.
- The amount of perturbation affects the accuracy of the scheme. For example, if a user requests for the nearest coffee shop (inside same downtown as she is), then the perturbation of just few blocks will affect the accuracy.

Intuitively, from the point of view of a LBS server or a malicious adversary, there is less possibility of presence of a user inside *or* near a particular landmark/location when in downtown than in a rural area. Thus the key idea is to make the LBS server believe that even if the user is in a rural area, her presence is possible inside *or* near more number of landmarks/locations. Since it is not possible to alter the number of landmarks/locations, a user should instead perform a "stronger" perturbation in the rural area than in downtown. Detailed analysis on privacy guarantees and LBS accuracy is presented in subsection 3.6.

Now we present two simple but naive techniques which the users can employ to achieve location privacy. These techniques are as follows-

- The most naive technique is to have user add random perturbation to her current location. However, this may result in highly inaccurate results. This also puts additional responsibility on user to manually perform context-aware perturbation.
- A very simple method is to have each user dynamically compute her location's road density based on a locally-stored topology map, and then perform the corresponding perturbation. However, this is inefficient in terms of space and time. For example, the topology map provided by US Census Bureau requires 11MB for Travis County, Texas only. This already exceeds the capacity of many mobile devices such as cellular phones.

To address these challenge, our key idea is to pre-compute a locality-preserving mapping from the original 2-d space (of latitude and longitude) to a new space, such that all points in the new space have equal road density. Here locality-preserving means that spatial proximity is preserved: i.e., two nearby points after the mapping should have high probability to be close before it (and vice versa). For simple illustration, Figure 3.1 depicts an example for a 1-d original space. In the original space, the road density near B, C, or D are higher than A, E, or F. It is easy to observe that the mapped space has constant density and the mapping is locality-preserving.

With such a mapping, our location perturbing component first maps a user's original location (i.e., latitude/longitude) to the new space. Since all points in the new space have equal road density, they can be perturbed homogeneously with random noise of the same distribution. After the noise is added to 1-d projected space and a new point is generated, we map that point back to the original latitude/longitude space and output the result as the perturbed location. Consider Figure 3.1 for a simple illustration of the idea on 1-d data. As we can see, a location in a high-density area of the original space will receive less perturbation than one in a low-density area. This is consistent with our intuition discussed above.

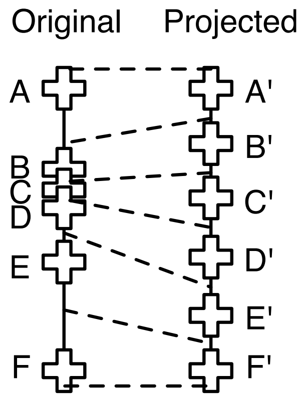


Figure 3.1: Mapping between original and projected space

3.3 VHC-Mapping

We now introduce the detailed design of our locality-preserving mapping, the *V*arious-size grid *H*ilbert *C*urve based Mapping (VHC-mapping). VHC-mapping is designed to map the original 2-d space to a 1-d constant-density space in a locality-preserving manner. We first outline the basic steps for the construction of VHC-mapping, and then explain the intuition behind it.

Without loss of generality, consider the original 2-d latitude/longitude space as a square. VHC-mapping involves a repeated partitioning of the square into various-size grid cells. Each cell is either partitioned into 4 equal-size square cells, or not (further) partitioned (i.e., becomes a base cell). A *main idea* of VHC-mapping is to enforce the following rule during the partitioning process:

Min-Density Rule: *Partition a cell into 4 square sub-cells iff the total road length (in the original space) covered by every sub-cell is more than μ times the edge length of the sub-cell, where $\mu > 0$ is a pre-determined granularity ratio.*

After the partitioning process, VHC-mapping constructs the mapped 1-d space as a (variation of) Hilbert space-filling curve [35] which connects all base cells in the original space. Figure 3.2 depicts a simple illustration of such a Hilbert curve, while Figure 3.3 depicts a real example on the map of Baltimore, MD with granularity ratio $\mu = 40$.

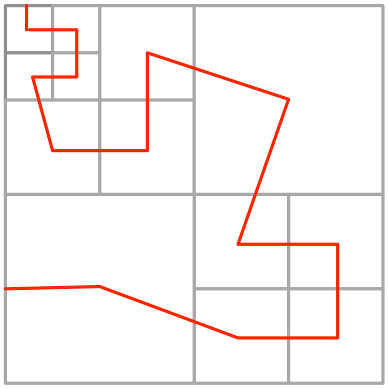


Figure 3.2: Illustration of Various-size grid Hilbert Curve

The mapping is then designed as follows: A 2-d point in the original space is mapped to the (geometrically) nearest point to it on the Hilbert curve. A 1-d point is mapped back to the original space by randomly choosing a 2-d point corresponding to the 1-d point.

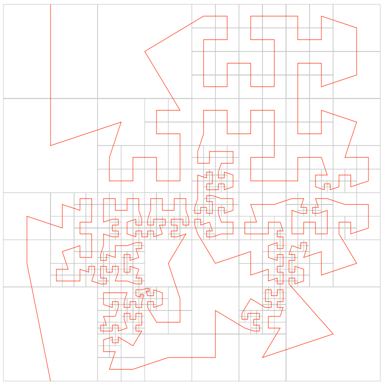


Figure 3.3: VHC-mapping for Baltimore County, Granularity ratio = 40

We now explain the rationale behind the design of VHC-mapping. Recall that there are two requirements for the mapping: 1) the projected space should have constant density, and 2) the mapping should be locality-preserving. For the constant-density requirement, there are two key observations:

- Due to the min-density rule, the total road length covered by each base cell is at least, and approximately the same as, μ times the edge length of the cell.
- Due to the property of Hilbert curve, the length of the Hilbert curve covered by a cell is at least, and approximately the same as, the edge length of the cell.

As we can see, every point in the projected space (i.e., on the Hilbert curve) can be considered as corresponding to (approximately) μ points on roads in the original space. Thus, the road density is (approximately) constant for all points in the projected space. This satisfies constant-density requirement. On the other hand, note that a well-known property of Hilbert curve is locality preservation in that two points nearby each other in the projected space is likely to be close in the original space [32]. Thus, VHC-mapping also satisfies the locality-preserving requirement.

3.4 Efficiency

Recall that our *key idea* is to provide a user with pre-computed map file. For this we propose to store the VHC-mapping in a 4-tree data structure *and* later statically store the state of this data structure into a file using *Binary Serialization*. This file will be used by the user to re-construct the identically ordered 4-tree in memory. In order to support efficient transformation between the original space and the new space (1-d projected space), we propose to store each node in the 4-tree as either 0 (i.e., base cell) or 4 children. Figure 3.4(b) depicts an example of this 4-tree for the VHC-mapping in Figure 3.4(a). Each leaf node of the tree is a base cell (i.e., 1 to 13). Each layer of the tree corresponds to a level of granularity for the partition of the original space. As we can see, the leaf nodes can be at different layers of the tree.

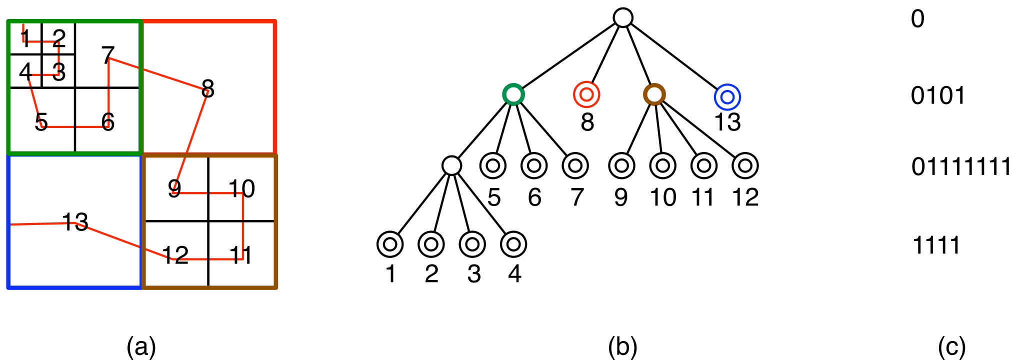


Figure 3.4: (a) Illustration, (b) 4-Tree data structure, (c) Static storage

Based on the 4-tree, we can efficiently perform the mapping by a search of the tree for $O(\log n)$ time, where n is the number of leaf nodes. Recall that we also need to map a location in the new space back to the original space. Such inverse mapping can be done with $O(\log n)$ time as well through a binary search on all leaf nodes.

Note that the 4-tree can be stored in a space-efficient data structure. Since each node in the tree either is a leaf node or has 4 children, we only need to store 1-bit information for each node to indicate whether it is a leaf. Figure 3.4(c) shows an example of such encoding scheme for the tree in Figure 3.4(b). Since a 4-tree with n leaf nodes has at most $4n/3$ (total) nodes, the space taken by the 4-tree is at most $4n/3$ bits. Our scheme has been able to reduce a file size of 240 MB for entire Texas State to 4 KB (using the Granularity Ratio = 20). An important fact to be noticed here is that to re-construct the 4-tree, only the pre-computed binary map file needs to be parsed instead of the original map file which is very large in size (parsing a bit instead of a string/stream).

Thus along with satisfying the requirement of having minimal storage, our scheme is capable of being used by users with minimal computational power.

3.5 Detailed Algorithms

We now present the detailed algorithms for our approach. There are two algorithms: One is the offline construction and storage of VHC-mapping. The other is the online retrieval of the mapping and the perturbation of a user's locations.

Algorithm 1 Offline Construction of VHC-Mapping

```
Require: Map, $C$ as the (rectangle) boundary of the map
Store $C \| \textsc{BuildTree}(C)$ as the HC-mapping file.
function $\textsc{BuildTree}(C)$
    if total road length in $C \geq \mu \cdot$ edge length of $C$ then
        Partition $C$ equally into $C_{\text{nw}}$, $C_{\text{ne}}$, $C_{\text{se}}$, $C_{\text{sw}}$.
        for $i = \text{nw}, \text{ne}, \text{se}, \text{sw}$ do
            return $0 \| \text{BuildTree}(C_i)$
        end for
    else
        return 1
    end if
end function
```

Algorithm 1 depicts the offline construction and storage of VHC-mapping. In the algorithm, we use || to represent the concatenation operation. We partition the original map based on the min-density rule and store the 4-tree into a bit stream.

Algorithm 2 Online Location Perturbation

```
Require: Pre-computed VHC-mapping file $hcFile$
1: Load a 4-tree $T$ of the partition from $hcFile$.
2: Build a Hilbert curve that connects all leaf nodes in $T$.
3: Wait until receiving an original 2-d location $X$.
4:         Map $X$ to 1-d $F(X)$ based on the Hilbert curve.
5:         Generate random noise $r \sim N(0, \sigma^2)$.
6:         Output $R(X) = F^{-1}(F(X) + r)$.
7: Goto 3
```

Algorithm 2 depicts the online retrieval of VHC-mapping and perturbation. Given a 2-d location X, we map it to 1-d point $F(X)$, add a homogeneous noise r, and use $F^{-1}(F(X)+r)$ as the perturbed location. Note that r is generated from a pre-determined distribution. In this thesis, we consider normal distribution with mean 0 and variance σ^2. We will analyze the setting for σ in the next subsection.

Suppose that the storage of boundary C takes 16 bytes (for two latitudes and two longitudes). The VHC-mapping file requires at most $16 + n/6$ bytes to store, where n is the number of leaf nodes in the tree. We will show in the experiments that the VHC-mapping for a real-world map is extremely small.

Algorithm 1 is executed offline and has computational complexity of $O(n)$. The computational complexity of Algorithm 2 is $O(n)$ for the retrieval of HC-mapping file (i.e., Steps 1 and 2) and $O(\log n)$ for the perturbation of each location. Note that the retrieval part only needs to be executed once for multiple locations in one VHC-mapping file (e.g., a county).

3.6 LBS Accuracy and Privacy Guarantees

First, we will demonstrate that how the same value of σ will result is different values of noise r in Algorithm 2. Precisely, the value of noise r will be more in case of a rural area than a downtown area. Consider a square with x as the length of its side. Thus the length of hilbert curve for this square would be equal to x. Recall that in our scheme a square cell is either partitioned into 4 equal-size square cells, or not partitioned at all. Suppose a main square cell is partitioned into T equal-size square cells. In this case the length of hilbert curve for a single cell would be equal to $\frac{x}{\log_2 T}$ and $\frac{Tx}{\log_2 T}$ for the main square cell. Thus it can be seen that the length of a hilbert curve increases with the number of partitions. Recall that in our scheme an area with higher road density is partitioned more.

Consider two square areas D and R with same x (length of side), and thus same area, where D is downtown area and R is the rural area. Thus the hilbert curve, which in our case is 1-d realization of the 2-d geographical space, will be greater in length for D than for R. Let m be the starting point of the hilbert curve and m' be the ending point for both D and R. In case of R, the distance between m and m' is d or if put in the context of our scheme, d is the amount of noise added to m to generate m'. Now we use the same value of d in the case of D, which is downtown area. In this case adding d amount of noise to m will not generate m' which is the ending point of the curve. Rather it would generate a point p which is somewhere on the hilbert curve but still far from m'. It can be clearly seen that the euclidean distance between m and m' is greater than the euclidean distance between m and p. Thus adding same amount of noise in case of D results in less perturbation than in case of R.

We now analyze the performance of the location perturbing component in terms of the degree of LBS accuracy l_{r} and the (ϵ, N)-privacy guarantees. We will first prove that our approach satisfies the objective of achieving the same degree of LBS accuracy for all locations in the map. Then, we will derive the privacy guarantees based on the noise parameter σ.

Theorem 3.6.1. *With Algorithm 2, if all POIs are i.i.d. random locations uniform over all points on the roads, the expected degree of LBS accuracy is constant for all locations in the original map.*

Proof. Consider the simplest case with only two POIs. The generic case can be proved in analogy. In Figure 3.5, A and C are the POIs, and X and $R(X)$ are the original and perturbed locations, respectively. Clearly, LBS remains accurate after the perturbation iff C falls outside the red part of the circle. That is,

$$l_{\mathrm{r}} = \Pr\{D(R(X), B) \geq D(R(X), A)\} = 1 - \frac{\beta}{\pi}. \tag{3.1}$$

where $D(\cdot, \cdot)$ is the distance function.

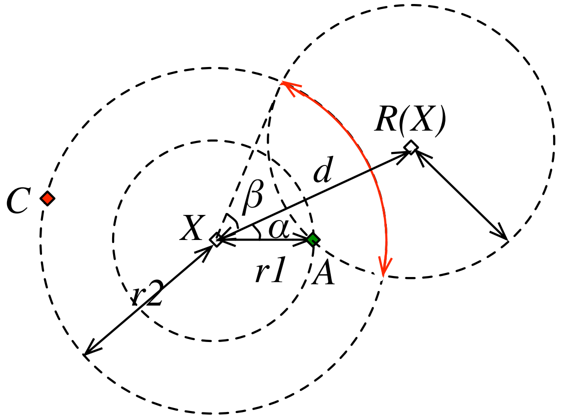


Figure 3.5: Nearest Neighbor

We have

$$\beta = \arccos \frac{r_2^2 - r_1^2 + 2dr_1 \cos\alpha}{2dr_2}. \tag{3.2}$$

Consider the base cell of the original space that covers X. Let ρ be road density (i.e., the total length of roads divided by the area) of the cell. Note that when the POIs are distributed uniformly at random on the roads, both r_1 and r_2 are inversely proportional to ρ^2. Also note that d is inversely proportional to ρ^2 as well, while α is independent of ρ. Thus, the expected value of β remains constant for all road density. That is, the degree of LBS accuracy l_{r} remains constant for all locations in the original map. □

For privacy guarantees, we have the following theorem. Recall from Section 3.1 that there is a constant ratio between population and road density. Let such ratio be λ.

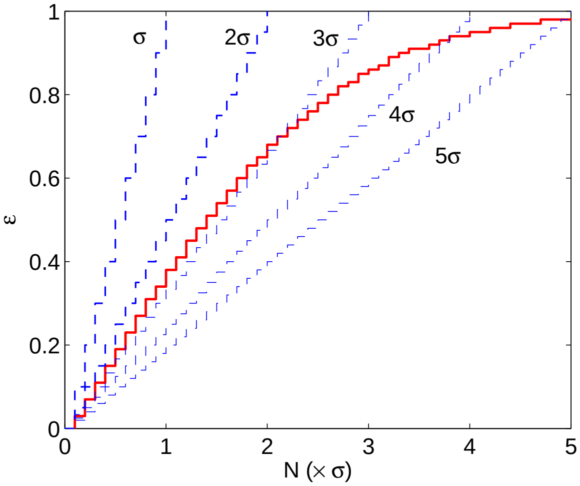


Figure 3.6: Privacy Skylines

Theorem 3.6.2. *(Privacy Guarantees) For any $\epsilon \in (0,1)$, Algorithm 2 achieves (ϵ, N)-privacy guarantee for all locations in the original map if*

$$N \leq 2 \cdot \sigma \cdot \mu \cdot \lambda \cdot \Phi^{-1}(1 - \frac{\epsilon}{2}). \tag{3.3}$$

where $\Phi^{-1}(\cdot)$ is the probit function [47].

Proof. First consider the perturbation applied to the 1-d projected space. Since the additive noise follows Gaussian distribution with variance of σ^2, the confidence interval of ϵ has width $2 \cdot \sigma \cdot \Phi^{-1}(1-\epsilon/2)$.

Now consider the projection from the 1-d back to the 2-d space. Note that the length of the VHC in any given cell is equal to the edge length of that cell. Thus, when $\mu = 1$, there is a one-one mapping between every point in the 1-d space and every point *on the road* in the 2-d space. More generally, the confidence interval of width $2 \cdot \sigma \cdot \Phi^{-1}(1-\epsilon/2)$ in the 1-d space is corresponding to road length of $2 \cdot \sigma \cdot \mu \cdot \Phi^{-1}(1-\epsilon/2)$ in the 2-d space.

Recall that the ratio between population and road density is λ. Thus, the confidence interval is corresponding to a population of $2 \cdot \sigma \cdot \mu \cdot \lambda \cdot \Phi^{-1}(1-\epsilon/2)$. Thus, (ϵ, N)-privacy guarantee is satisfied if $N \leq 2 \cdot \sigma \cdot \mu \cdot \lambda \cdot \Phi^{-1}(1-\epsilon/2)$. □

CHAPTER 4

ANONYMOUS ROUTING COMPONENT

For the anonymous routing component, the main design challenge is how to limit the overhead introduced by anonymous communication to the minimum degree necessary for matching the level of protection provided by the location perturbing component (through data privacy). CAP utilizes Tor, a practical and running anonymous communication network on the Internet, for anonymous routing. In this section, we first provide a brief overview of Tor, and then introduce our approach to address the above design challenge.

4.1 Tor Network

Tor [9] is an overlay network on the Internet providing anonymous communication. Figure 4.1 depicts the basic architecture of the Tor network. Before introducing our approach, we first briefly review the functionalities of Tor network [9]. Readers familiar with Tor can skip to the next subsection.

There are four entities within the Tor network-

- *Client*: A client uses an *onion proxy* (*OP*) installed on her local machine and connects to services through the Tor network, e.g.,the mobile client in Figure 2.1.
- *Server*: In context of CAP, it is the LBS server.
- *Tor routers (Onion Router)*: The special proxy which relays the application data for the client to the server. TLS connections are built between Tor routers for link encryption.
- *Directory servers*: Servers holding information regarding the Tor routers. The client downloads the router information from the directory servers and caches it locally.

Functions of onion proxy, onion router and directory server are integrated into the same software package. A user can edit a configuration file and configure a computer to have any combination of those functions. A volunteer donating her network bandwidth installs the package and configures her computer as an Tor router or directory server. Tor routers can be configured using a leaky bucket mechanism to constrain the donated bandwidth. A client installs the same software package but configures her computer as a Tor client to utilize the Tor network.

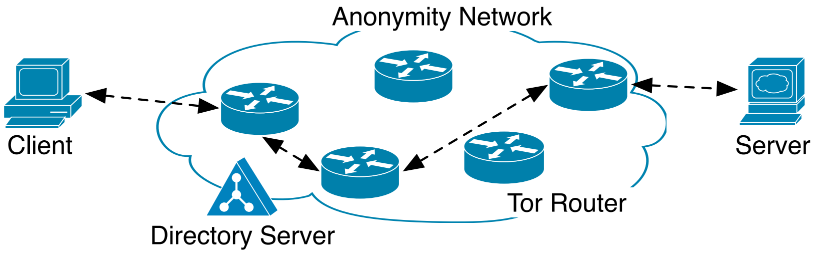


Figure 4.1: Tor Network

The basic operation of Tor is illustrated in Figure 4.1. Tor uses a type of source routing. To access the LBS server in Figure 4.1 while hiding the connection, the client chooses a series of Tor routers from the cached Tor router directory. We refer to the sequence of ordered Tor routers as the *path* of the client's communication through the Tor network to a server. The number of Tor routers is the *path length*. The default path length is 3. The client first negotiates session keys with the three chosen routers one by one using the *Diffie-Hellman* key exchange protocol. The first Tor router is called the *entry guard* while the last one is the *exit node*. The client packs application data into cells encrypted like onions transmitted over the Tor network. Because of the special layered encryption, Tor's routing is called *onion routing*.

4.2 Input to Anonymous Routing Component

The key design principle for anonymous routing in CAP is the “pail law”: water flows out from the shortest board of a pail. Similarly, the location privacy that an LBS system can achieve depends on the weakest “link”, either on the data privacy or communication anonymity. Since the level of data privacy achieved by the location perturbing component is already determined by the intended degree of LBS accuracy, there is no need for the anonymous routing to achieve a “better” privacy protection than the location perturbing component.

Consider the data privacy provided by the location perturbing component. We assume that the adversary who has access to the LBS server knows the perturbing algorithm. Therefore, from the perturbed location coordinate $\vec{Y}$, the adversary can infer the location of the mobile to be within certain confidence interval. For example, suppose that the 95% confidence interval is area $[\vec{Y} - \vec{\Delta}, \vec{Y} + \vec{\Delta}]$. To locate the mobile client, the adversary can then guess the base station that the mobile client connects. As we know, the map of base stations is publicly available [1, 13]. The adversary can first infer the list of base stations within $[\vec{Y} - \vec{\Delta}, \vec{Y} + \vec{\Delta}]$. The number of base stations in this list is denoted as M. To locate the mobile client, the adversary can then launch an *observation attack* by deploying accomplices at the M target base stations to discover the location of the mobile client. For example, by using timing attacks to correlate messages from mobile clients and messages arriving at the LBS server, the adversary can identify the mobile client who communicates with the LBS server [57]. As we can see, if there are l accomplices to be deployed by the adversary, the probability that the mobile client can be identified is l/M. In other words, l/M is the upper bound of location privacy that the location perturbation can achieve through data privacy discussed in Section 3.1.

The adversary may also break the communication anonymity by tracing the communication through the Tor network from the LBS server back to the mobile client.

Once the mobile client is discovered, we assume that location privacy of mobile client is exposed as well. There are various attacks against anonymous networks [11]. Since Tor uses donated nodes as Tor routers, we believe that the most feasible attack to break Tor is those using the malicious entry nodes. For example, using the flow marking technique [51], the adversary can interfere with the mobile client traffic at the entry guard and marginally vary packet interarrival times to embed a secret signal pattern into the traffic. The embedded signal pattern is carried along with the traffic from the entry node to the LBS server, so the adversary can recognize the communication relationship between the mobile client and LBS server. Therefore, the location of mobile client will be exposed despite the use of anonymous communication networks. For example, by using marking techniques communication relationship can be confirmed, thus the location of a mobile client can be exposed as well [56]

Recall that one objective of CAP is to optimize the QoS of the whole system. To this end, assume that the mobile client chooses one of S Tor nodes as the entry node. Assume that there are K malicious nodes controlled by the adversary in these S Tor entry nodes. The probability of choosing one of the K malicious nodes is K/S. To guarantee the minimum location privacy (l/M) that the location perturbation can achieve, we need to choose an appropriate S such that

$$\frac{K}{S} \leq \frac{l}{M}. \tag{4.1}$$

That is

$$S \geq \left\lceil \frac{KM}{l} \right\rceil, \tag{4.2}$$

where $\lceil x \rceil$ refers to the minimum integer $\geq x$. Therefore, in order to improve the communication performance, we can select entry routers with high donated bandwidth to build paths through the Tor network as long as equation (4.2) is met.

From the discussion above, we know that the input to the anonymous routing component should include (i) $R(X)$, the perturbed location, and M, the number of base stations within the region the adversary may do the brute force search; (ii) user specified QoS requirements such as throughput. Note that the precise value of M may not be needed. Instead, we could simply compute an upper bound on it based on common knowledge on the density of cellular towers and/or wifi hotspots.

4.3 Design principle

The basic idea of designing the anonymous routing component is that, given the location privacy requirements, we optimize the throughput of Tor, which often suffers from serious performance degradation [27]. From (4.2), we know that the anonymous routing component should design algorithms to select a route from a list of N entry nodes determined by location privacy requirements. Recall that Tor is an overlay network and Tor routers use donated bandwidth from users, who may limit the donated bandwidth using the leaky bucket mechanism. A set of sequential TCP connections are used to relay packets from the source to the destination and the end-to-end throughput will be limited by the bottleneck segment [25]. Hence, it is reasonable to assume Tor routers in Figure 4.1 are bottlenecks for TCP throughput and that the client and server nodes are not.

We find that if we use fewer low-bandwidth Tor routers, we can improve overall TCP throughput on the Tor network. The Tor network could be partitioned into classes of Tor routers with high or low donated bandwidth. We can also exclude low bandwidth entry routers as long as equation (4.2) is met. Paths drawn from the class of high-bandwidth routers can provide better performance, while those from the class of low-bandwidth routers have throughput no lower than unpartitioned paths, lower throughput are more probable. This fact is derived in Theorem 4.3.1 [36].

Theorem 4.3.1. *In a Tor network, there are n Tor routers, and the path length is m. Assume that the bandwidth of Tor routers forms a set $\{B_1, \cdots, B_l, B_{l+1}, \cdots, B_n\}$ and $B_1 > \cdots > B_l > B_{l+1} > \cdots > B_n$. If only the first l Tor routers are used, the average of the path TCP throughput increases. Denote $B(,)$ as the average TCP throughput bound, that is,*

$$B(l, m) > B(n, m), \textit{where } l < n. \tag{4.3}$$

The bound in Theorem 4.3.1 refers to the case where the sequential TCP connections of a Tor path have exclusive access to the Tor bandwidth. The bandwidth available for a TCP connection may be much lower than this bound because the links along the path may be shared by cross traffic. Therefore, partitioning routers based on bandwidth paves the way to improved performance by supporting differential QoS in the Tor network. In particular, the routers could be organized into multiple classes based on their bandwidth and chosen for flow requests based on a request's priority. In this way, higher priority flows (e.g., LBS query request and response) will obtain high bandwidth and low priority flows will obtain lower bandwidth. So long as user requirements can be met with differential QoS, this will make more effective use of bandwidth.

4.4 Detailed Algorithms

The anonymous routing component in Figure 2.1 will control Tor's routing in order to achieve differential QoS for Tor clients. By default, a Tor client selects Tor routers based on a number of factors, but primarily relies on a *weighted random algorithm*. The algorithm uses the onion routers advertised bandwidth as the weight so that these routers with a higher advertised bandwidth have a higher probability to be selected. Our research is focused on improving this algorithm to see whether better performance could be achieved. We have implemented the two different atomic path selection algorithms in favor of differential QoS in the Tor network.

The first algorithm is shown in Algorithm 3 which provides the differential routing with two priorities. The high priority means that a user chooses Tor routers with bandwidth greater than or equal to $MinBW$. The low priority means that a user chooses Tor routers with bandwidth smaller than $MinBW$. Since our location privacy preserving LBS, CAP, is sensitive to communication QoS, it uses the high priority service.

Algorithm 3 Differential Routing (*Diff*)

```
Require: (i) Perturbed location; (ii) User specified path throughput capacity $MinBW$
Randomly select an entry router from a pool of entry nodes satisfying location privacy constraint determined in (4.2).
Build a pool of Tor routers whose bandwidth is greater or equal to $MinBW$.
Use the weighted random algorithm and build a circuit (i.e. path) through the pool. Record used Tor nodes in existing circuits and future circuits will not use those used Tor nodes.
```

Algorithm 3 can only partially guarantee the bound of the average path throughput. The actual path throughput will be much lower because of congestion on the Internet as numerous flows share the Tor nodes. Because of this, we may achieve poor paths under Algorithm 3. To overcome this problem, the second routing algorithm (Diff/CA in short) we propose considers the congestion avoidance as shown in Algorithm 4. Recall that Tor can create circuits proactively and wait for user connections. To avoid congestion, Diff/CA creates circuits proactively and measure the path throughput [21, 26] until the throughput requirement is met. This incurs a delay in circuit creation. Our experimental results show that the delay is within a reasonable range. Algorithm 4 can also be adapted to meet the path latency requirement by measurement [6].

```
Algorithm 4 Differential Routing with Congestion Avoidance (Diff/CA)
Require: (i) Perturbed location; (ii) User specified minimum path throughput capacity
    $MinBW$ and tolerable throughput $TolBW$.
1: Randomly select an entry node from a pool of entry routers satisfying the location
   privacy constraint (4.2).
2: Build a pool of Tor routers whose bandwidth $\geq MinBW$.
3: Use the weighted random algorithm and build a circuit through the pool. Measure
   the circuit throughput until it is greater or equal to $TolBW$. Record used Tor routers
   in existing circuits and future circuits will not use those used Tor routers.
```

CHAPTER 5

CAP PROTOTYPE

In this chapter we present the implementation details of our prototype. First we present the prototype's interface followed by description of different components. In the prototype's interface section, Graphical User Interface, we first discuss about the user options and later present some snapshots. In the components section we discuss in detail about the implementation of two important components, namely location perturbation component and anonymous routing component. In the last section we discuss few major implementation issues.

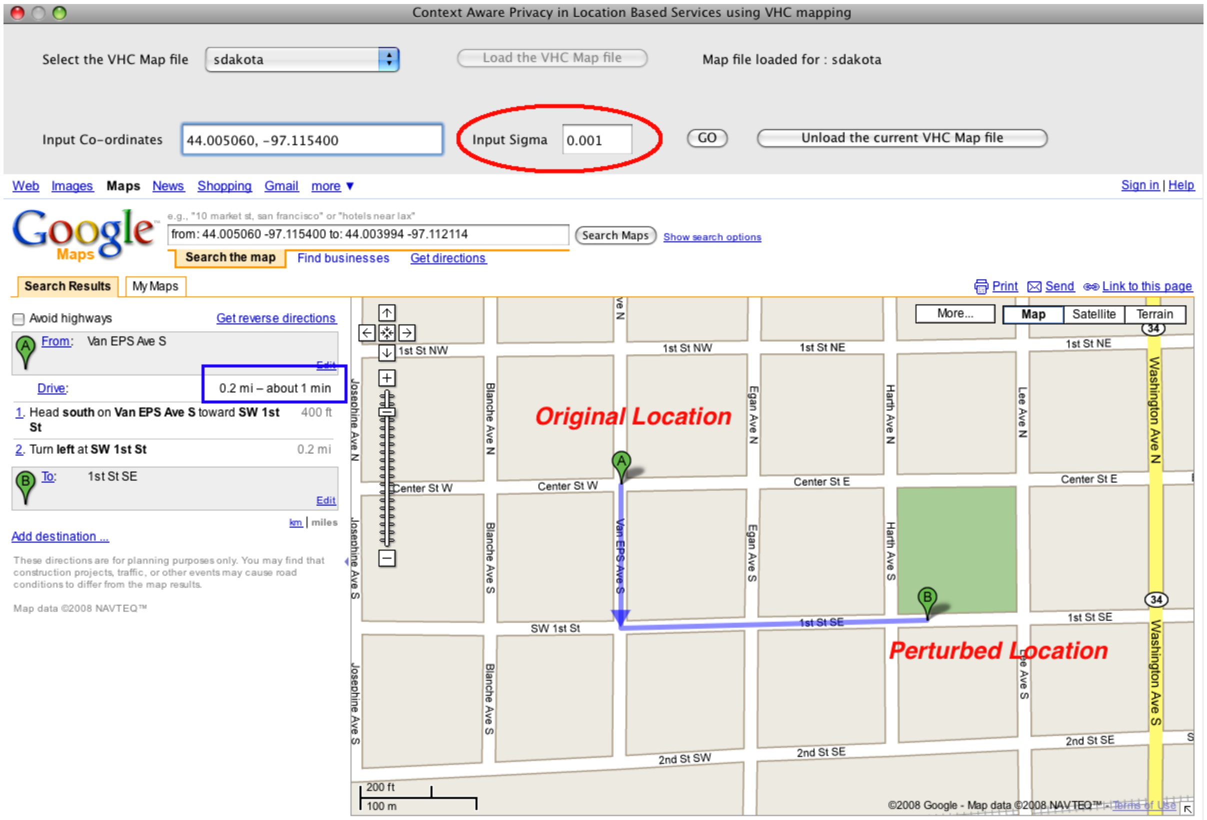
5.1 Graphical User Interface

The prototype's GUI was developed on Mac OS X v10.5.3 using Qt [49] and C++. Our aim was to present a user with a very user-friendly solution to achieve complete privacy. Thus, the prototype is a graphical user interface (GUI) integrated with Google Maps. With the current features of the prototype, users can -

- Select the desired map file from a drop down list, where map files belong to particular state in the US.
- Manually input perturbation factor by which a user can make a tradeoff between privacy and LBS accuracy. This option can viewed in Figure 5.1 with a red oval around it. If a user skips to put in a value here, a default value will be used.
- Manually input the latitude and longitude of the location to be perturbed. This is highlighted with a red oval in Figure 5.3

- View the original and perturbed location in Google Maps which is embedded inside the GUI (refer to Figure 5.1).

In this section we present three snapshots of the prototype. In figure 5.1, which is an illustration of a part in Madison, South Dakota, we can see the original location A is being perturbed to new location B, which is few blocks away. The exact distance in miles, which is provided by Google Maps, can be viewed in the blue colored box. Also, driving directions can be clearly viewed between A and B.



Context Aware Privacy in Location Based Services using VHC mapping

Select the VHC Map file: sdakota

Map file loaded for : sdakota

Input Co-ordinates: 44.005060, -97.115400

Input Sigma: 0.001

from: 44.005060 -97.115400 to: 44.003994 -97.112114

□ Avoid highways

From: Van EPS Ave S

Drive: 0.2 mi – about 1 min

1. Head **south** on **Van EPS Ave S** toward **SW 1st St** 400 ft
2. Turn **left** at **SW 1st St** 0.2 mi

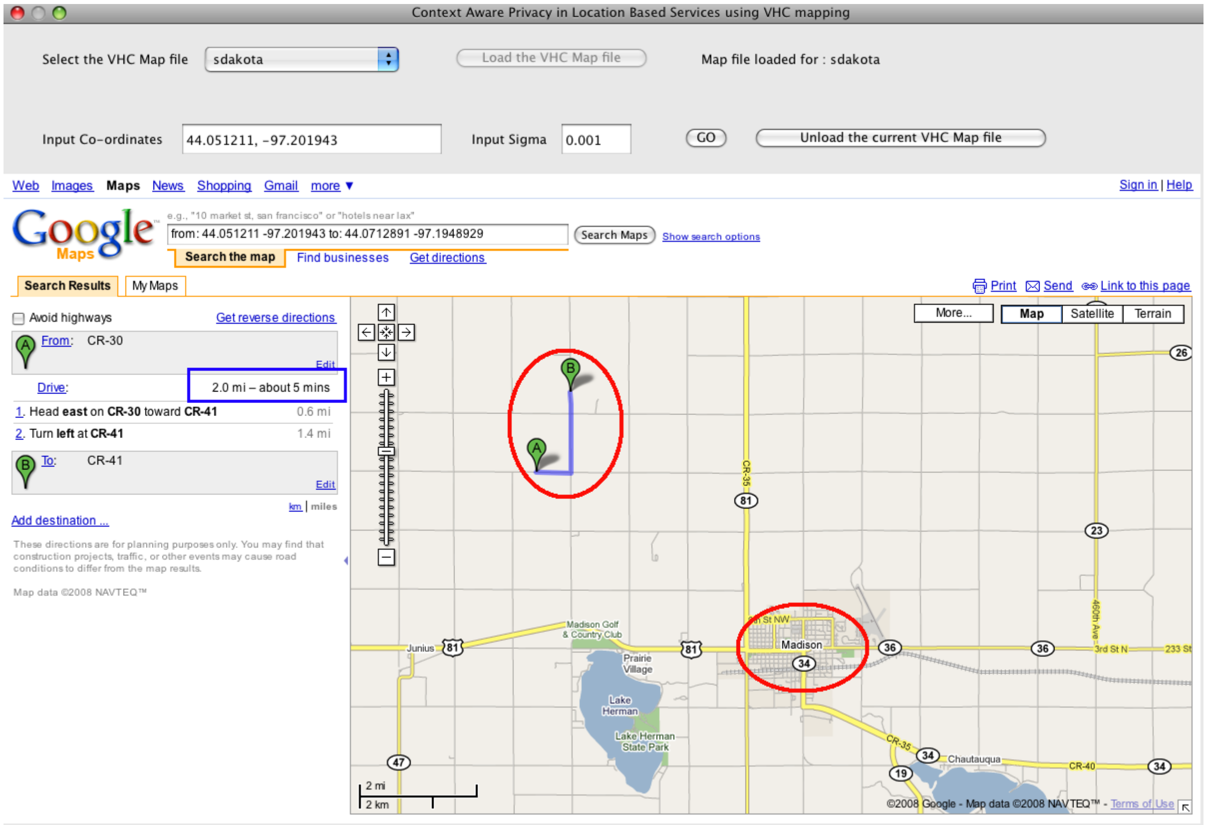
To: 1st St SE

These directions are for planning purposes only. You may find that construction projects, traffic, or other events may cause road conditions to differ from the map results.

Map data ©2008 NAVTEQ™

Figure 5.1: Inside Madison, South Dakota: Perturbation factor = 0.001

In the next figure, Figure 5.2, the distance between A and B is much higher (2 miles as viewed in the highlighted blue box) as compared to perturbation distance in Figure 5.1. This is because the original location is inside a rural area. Thus we achieve context-aware perturbation i.e. stronger perturbation in rural areas as compared to downtown. The upper red oval highlights the distance between original and perturbed location. The lower red oval highlights the Madison downtown. It can be seen that the perturbation distance is almost equal to the entire span of the downtown, even when the value in *Input Sigma* box remains the same.



Context Aware Privacy in Location Based Services using VHC mapping

Select the VHC Map file: sdakota

Map file loaded for : sdakota

Input Co-ordinates: 44.051211, -97.201943

Input Sigma: 0.001

from: 44.051211 -97.201943 to: 44.0712891 -97.1948929

□ Avoid highways

From: CR-30

Drive: 2.0 mi – about 5 mins

1. Head **east** on **CR-30** toward **CR-41** 0.6 mi
2. Turn **left** at **CR-41** 1.4 mi

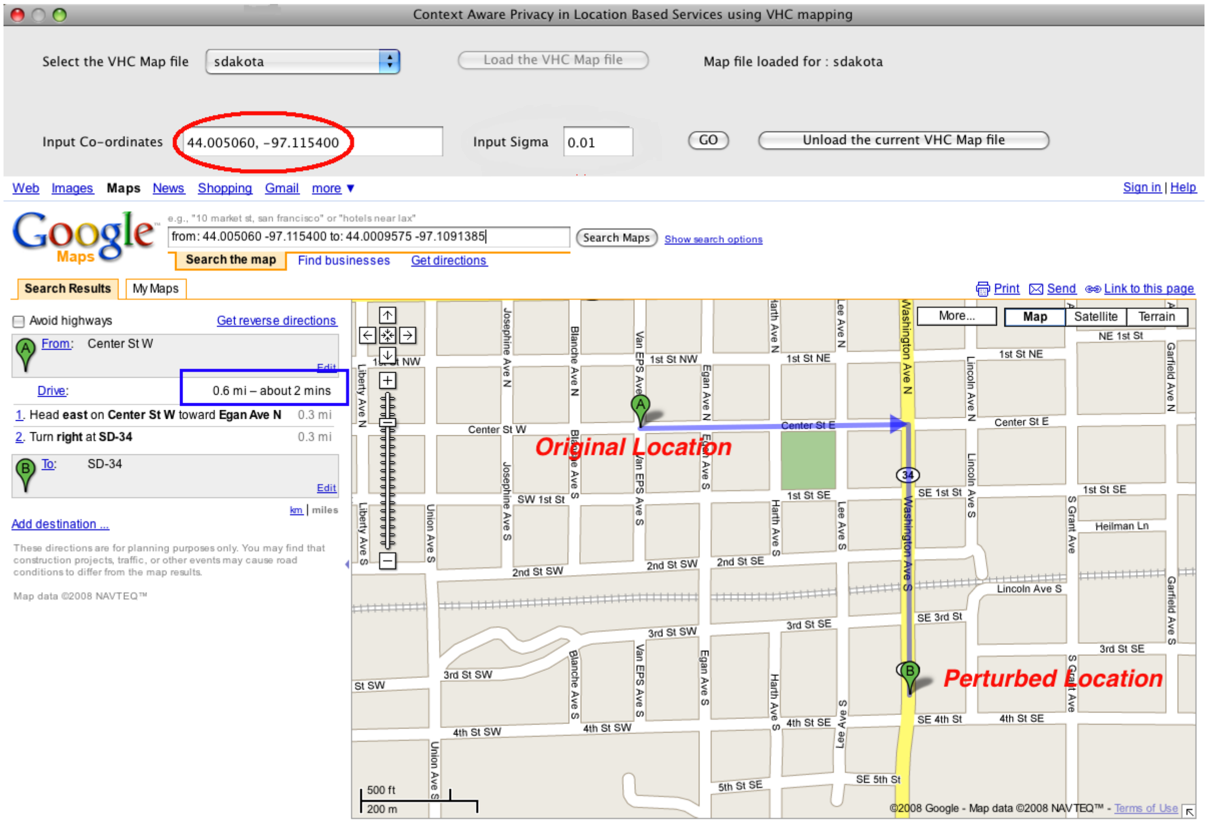
To: CR-41

These directions are for planning purposes only. You may find that construction projects, traffic, or other events may cause road conditions to differ from the map results.

Map data ©2008 NAVTEQ™

Figure 5.2: Higher perturbation in a rural area, Lake county, South Dakota

In the next case, Figure 5.3, we demonstrate that when the perturbation factor (value in the *Input Sigma* box) is higher, more perturbation will be achieved even if the original location is exactly the same. As compared to case of Figure 5.1, the perturbation achieved is thrice (0.6 miles in the highlighted blue box). Here, note that even if higher perturbation is achieved, the perturbed location still is in downtown. This will result in less LBS inaccuracy (refer section).



Context Aware Privacy in Location Based Services using VHC mapping

Select the VHC Map file: sdakota

Map file loaded for : sdakota

Input Co-ordinates: 44.005060, -97.115400

Input Sigma: 0.01

from: 44.005060 -97.115400 to: 44.0009575 -97.1091385

□ Avoid highways

From: Center St W

Drive: 0.6 mi – about 2 mins

1. Head **east** on **Center St W** toward **Egan Ave N** 0.3 mi

2. Turn **right** at **SD-34** 0.3 mi

To: SD-34

These directions are for planning purposes only. You may find that construction projects, traffic, or other events may cause road conditions to differ from the map results.

Map data ©2008 NAVTEQ™

Figure 5.3: Inside Madison, South Dakota: Perturbation factor = 0.01

5.2 Components

In this subsection we will present are implementation details of the Location Perturbation component and Anonymous Routing component-

5.2.1 Location Perturbation Component

This component was developed in C++ on a Linux kernel v2.6.22 using boost libraries [5]. This component is dynamically loaded into the memory when the user selects the map file from the GUI and clicks the *Load Map* button. This component is the 4-tree data structure which is constructed from the binary map file. We will discuss this component in three parts namely *Binary Map file*, Constructing the component and Perturbing the location

- *Binary Map file:* A binary map file is a statically stored state of the 4-tree data structure that is constructed from the original map file. Such original map files are available for download at [50]. The reason behind creating a binary map file is threefold-
  - This file will be used to re-construct the 4-tree data structure, which is our location perturbation component.
  - This enables extremely fast processing because the decision to partition is simply based on the bit value, rather than parsing the entire original map file (this is done while creating the binary map file from the original map file).
  - The size of the original map file is significantly reduced. The exact size of the binary map file, which is not more than few kilobytes, depends on the area under consideration (e.g., state or county) and more importantly, the granularity ratio. Refer to section 3.3 for details.

  Since the prototype components also need the bounding co-ordinates of the area under consideration, we provide a separate file for it, whose size will not be more than 43 bytes.
- *Constructing the component:* The root node of the 4-tree represents the outer bounding co-ordinates of the map area (e.g., Lake county, SD). Each bit in a binary map file is used as a decision to partition a node into 4 children or no children at all (leaf node). The mechanism for constructing this component is very similar to the depth-first-search technique. Once 4 children nodes are created, we climb down the 4-tree to make decision on partitioning of those children nodes. The decision is

not made simultaneously for all children, rather we keep climbing down the 4-tree till we reach a leaf node on a particular branch of the tree. Once a leaf node is encountered, we climb up the tree to make decisions for remaining children. Thus, we stop when we reach the last child (4^{th}) of the root node, while climbing up the tree.

- *Perturbing the location:* This component perturbs the 2-d location in two steps i) The corresponding 1-d location is retrieved from the 4-tree data structure, and it is perturbed to a new value. ii) For the new 1-d value, the corresponding 2-d value is obtained, which is the perturbed 2-d location.

  The concept of 1-d mapping is better explained by demonstrating how an area is partitioned using VHC-mapping technique. Figure 5.4 is a graphical realization of the leaf nodes in the 4-tree data structure of the Lake county in South Dakota. The granularity ratio μ is 3. Each gray colored box in Figure 5.4 is represented by a leaf node in the 4-tree and the red line is the hilbert curve. This hilbert curve is our 1-d space where each point on the curve corresponds μ (granularity ratio) number of 2-d points on the road, contained within a particular gray box. Thus a corresponding 1-d value for a 2-d location (some point in a gray box) is first obtained and noise is added to it. The new 1-d thus obtained belongs to that part of curve (red) which resides in some other gray box and the corresponding 2-d value of the perturbed 1-d value become the 2-d perturbed location. In the Figure 5.4, it can be clearly seen that in few areas the hilbert curve is clustered more than other. The more clustered areas represent the towns where road density is much higher than the rural areas. A very distinctly seen cluster is in the center of of Figure 5.4 representing the Madison city which is the largest city in Lake county. Figure 5.5, a graphical realization for Texas state, provides complete credibility to our scheme as it perfectly demonstrates the outline of Texas and areas with higher road density such as DFW Metroplex, Houston, San Antonio etc. are clearly visible as being more clustered than rest of the parts. As in the case of binary map file, the exact size of the memory footprint of the 4-tree data structure depends on the area under consideration (e.g., state or county) and more importantly, the granularity ratio. Refer to section 3.3 for details.

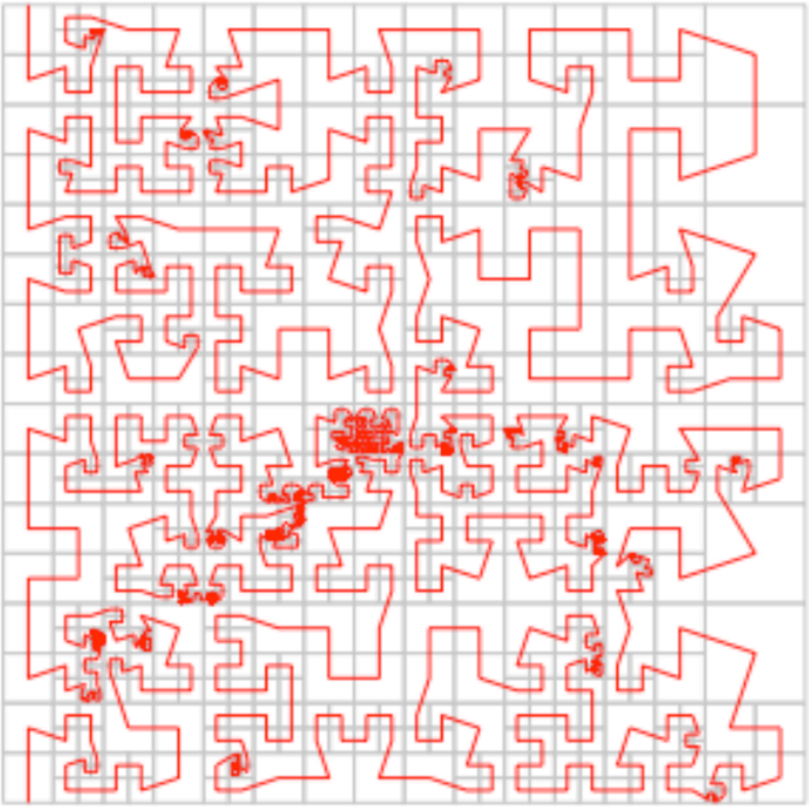


Figure 5.4: Various-size-grid Hilbert Curve for Lake county, South Dakota

5.2.2 Anonymous Routing Component

We have used Tor for anonymous communication which is widely available at [48]. Once Tor is installed, we set *Web Proxy (HTTP)* and *Secure Web Proxy (HTTPS)* as specified in the guidelines. Thus in our prototype, when a request/query is sent over the Internet, it goes through the anonymous communication network. We make few changes to the configuration file of Tor in order to achieve desired QoS.

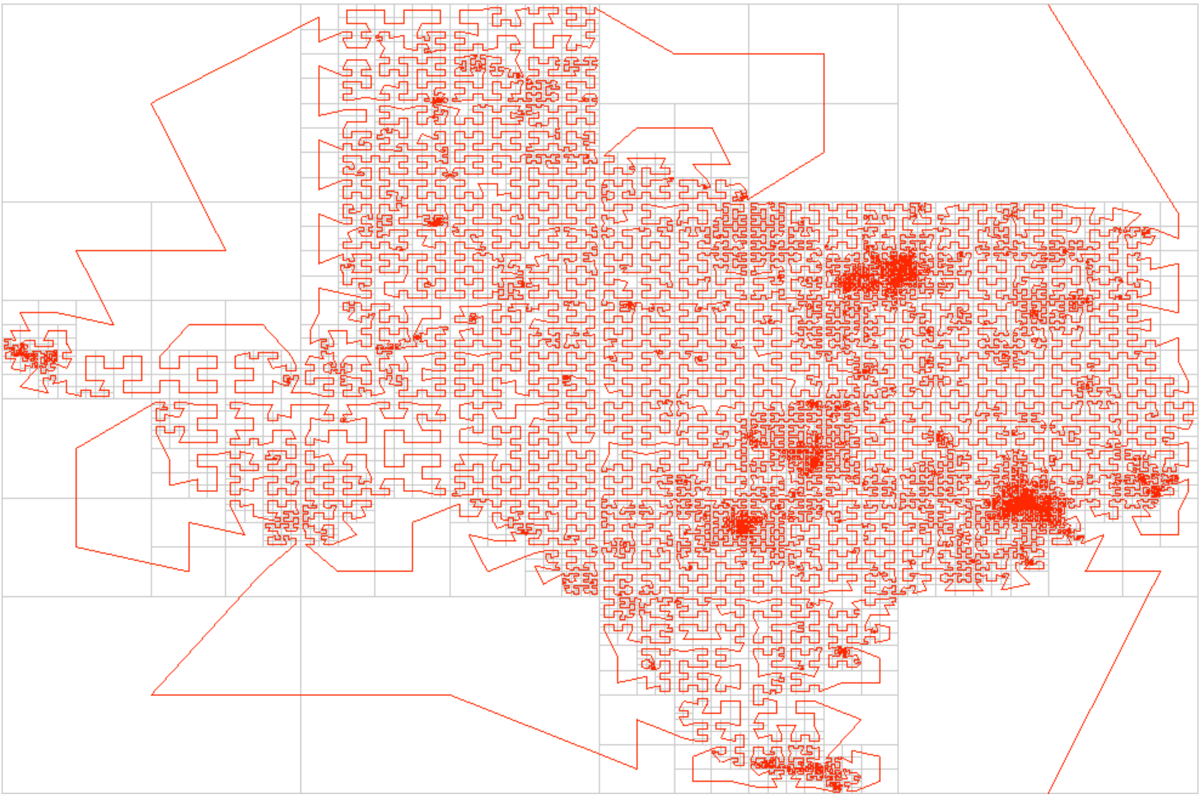


Figure 5.5: Various-size-grid Hilbert Curve for Texas state, Granularity ratio = 20

5.3 Implementation issues

We have used heap memory instead of stack memory, for the location perturbation component to reside in, to avoid stack overflows. However, stack overflows can also be caused due to recursive programming, and thus the location perturbation component is constructed using iterative programming only. For creating a binary map file, it is not possible to write a bit value into a file (the smallest size that can be written to a file is a byte). Thus we stuff each bit into a byte using logical shift operators and then write the value of that byte (sequence of 8 bits) to the binary map file.

CHAPTER 6

EXPERIMENTAL RESULTS

6.1 Experiment Setup

We ran our experiments on the map of Lake county, South Dakota, USA. The map was retrieved from the 2006 second edition of the Topological Integrated Geographic Encoding and Referencing (TIGER) system published by the US Census Bureau. It contains geographic and cartographic information including the topology of all roads in the county. The map can be downloaded as a zipped TIGER/Line file from http://www2.census.gov/geo/tiger/tiger2006se/SD. The size of the zip file is 910KB. For our experiments we have used only the data of starting and ending co-ordinates of a road or a part of it. Since we focus on road density only, we do not require data about any landmarks/locations in the county.

We collected 20 POIs in the Lake county such as restaurants, hotels, clinics, and supermarkets. Most of them are located in Madison, the largest city in the county. To test our revised routing algorithm, we tested as payload the map image of Lake county, South Dakota from TIGER. The intention was to emulate the download of driving directions, etc, as the returned result of LBS query. The downloading software was the command line utility *wget* with appropriate proxy configuration in order to use Tor.

6.2 Evaluation of Location Perturbing Component

Recall from Section 3.5 that the location perturbing component has two input parameters: σ, the standard deviation of inserted noise, and μ, the granularity ratio. We tested the performance of location perturbing component while changing the two parameters. To test against locations with diverse road density, we define the *road density index* of a location as the level of the leaf node that contains this location (root has level 1). The depth of the tree is 12 when $\mu = 3$, which is used in most experiments. For the Lake county, that we are testing for, all the leaf nodes in the tree, which deal with co-ordinates for Madison city, have higher road density index in range 10, 11 and 12. Also, nodes dealing with the co-ordinates of smaller towns like Ramona and Wentworth (in Lake county) too have higher road density index as compared to the rural areas. Generally, the road density increases in exponential order with the road density index.

Figure 6.1 depicts the relationship between the average 2-d perturbation distance $D(X, R(X))$ and the noise standard deviation σ for locations with various road density. We measured the 2-d distance by the Euclidean distance between the original and perturbed locations. We also tested with Manhattan distance [24] and obtained similar results. As we can see, the 2-d perturbation distance on a rural location is much larger than that on a downtown location. This confirms our discussion in Section 3.6.

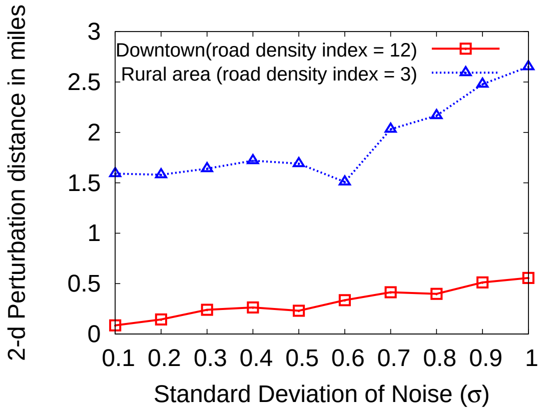


Figure 6.1: 2-d Perturbation Distance vs. σ

Figure 6.2 depicts the relationship between the degree of LBS accuracy l_{r} and the standard deviation of noise σ for locations with various road density. Recall that l_{r} is the probability that a nearest neighbor query returns the precise result. We can make two observations: First, l_{r} decreases with the increase of σ. Second, there is no significant difference for LBS accuracy between locations of different road density indices. This coheres with our findings in Theorem 3.6.1.

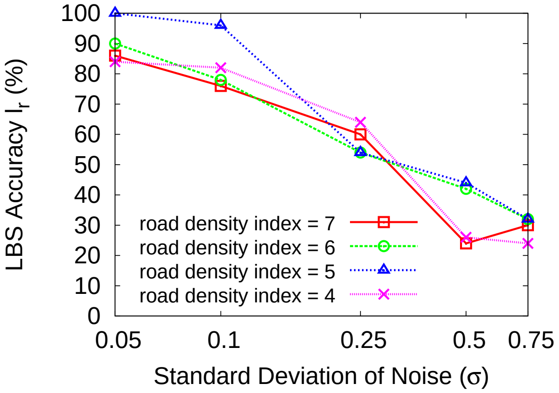


Figure 6.2: LBS Accuracy l_{r} vs. σ

To test Theorem 3.6.1 more thoroughly, we considered l_{r} against a full spectrum of road density indices. Figure 6.3 depicts the results. An interesting observation is that, although l_{r} appears approximately constant for indices from 4 to 8, the value of l_{r} is substantially lower for locations with extremely high road density. This seems to contradict Theorem 3.6.1.

We investigated this issue on the map and found the following reason: Since we manually collected the POIs, all of them locate near the Madison city area. On the other hand, many high-density locations in the county locate in the downtown area of other (smaller) cities. This is not consistent with our assumption in Theorem 3.6.1 (and

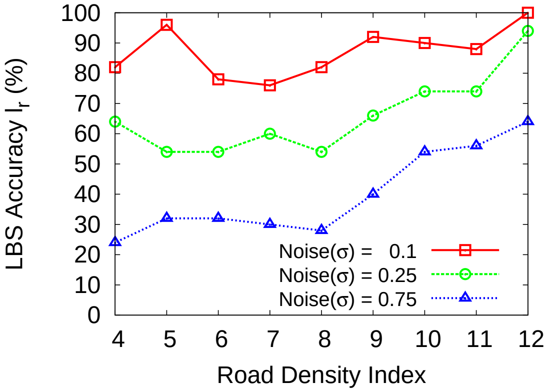


Figure 6.3: l_r vs. Road Density for Manual POIs

the real-world scenario) that the nearest POI to a downtown location should be closer than that to a rural one. To address this problem, we randomly generated 100 POIs as locations uniformly distributed on all roads in the county. The results, as shown in Figure 6.4, are consistent with Theorem 3.6.1.

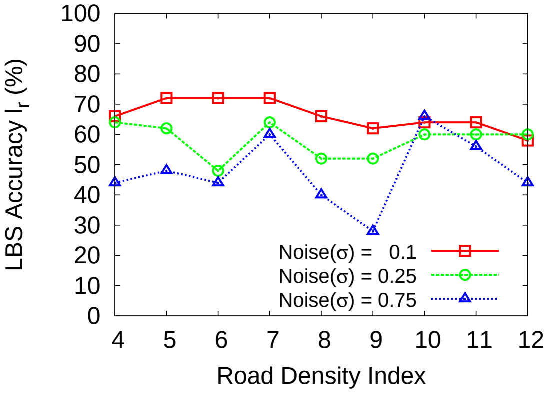


Figure 6.4: l_r vs. Road Density for Random POIs

Recall that the granularity ratio μ controls the size of the 4-tree (i.e., the VHC-mapping). Figure 6.5 depicts the relationship between the storage cost of the 4-tree and the granularity ratio μ. As we can see, the storage cost decreases exponentially when μ increases. In particular, when $\mu = 5$, we only need 400 bits to store the 4-tree. This is much smaller than the size (910KB) of the original TIGER/Line map.

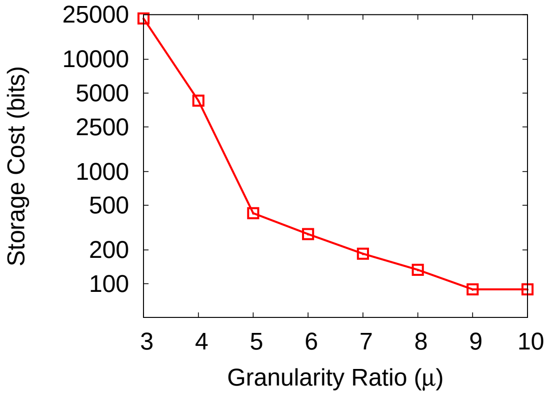


Figure 6.5: Storage Cost vs. Granularity Ratio μ

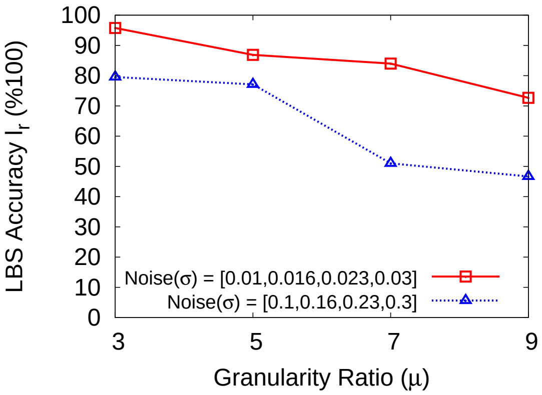


Figure 6.6: l_{r} vs. Granularity Ratio μ

Figure 6.6 depicts the relationship between the degree of LBS accuracy l_{r} and the granularity ratio μ in order to achieve the same privacy guarantees. As we can see, l_{r}

decreases when μ increases. Nonetheless, the decrease is much slower than the decrease of storage cost.

6.3 Evaluation of Anonymous Routing Component

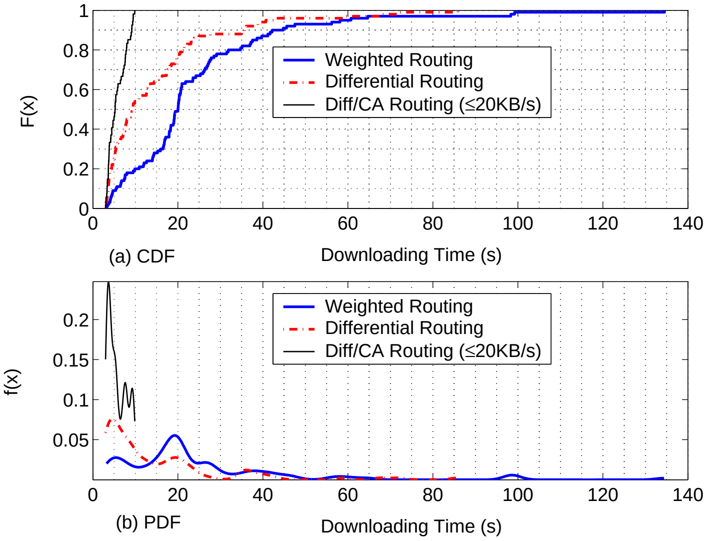
We also evaluated the QoS achieved by the anonymous routing component of CAP. We set $S = 79$ as the input from the data perturbing component. Figure 6.7 depicts the cumulative distribution function (CDF) and probability density function (PDF) of time downloading the map image of 208,310 bytes from TIGER under the anonymous routing algorithms we proposed in Section 4.4. Diff/CA (≤20KB/s) refers to differential routing with congestion avoidance whose tolerable throughput is 20KB/s. Table 6.1 gives the mean, median and confidence interval (95%) of the downloading time for different Tor routing algorithms.

Table 6.1: Downloading Time Comparison (unit: seconds)

	Weighted Routing	Diff	Diff/CA (≤5KBs)	Diff/CA (≤10KB/s)	Diff/CA (≤20KB/s)
Median	20.0422	9.2733	8.9566	7.3709	5.2343
Mean	24.3192	15.0296	12.0749	8.6298	5.711
Lo	19.9743	12.9606	9.8527	6.9204	5.1213
Up	30.0927	18.4387	14.6869	10.6894	6.4669

We have a few observations from Figure 6.7 and Table 6.1.

1. The performance of Tor's default routing algorithm, weighted routing, can be intolerable for performance sensitive service such as LBS. The largest downloading time of the map image is 134.49s.
2. The differential routing and the differential routing with congestion avoidance can significantly improve Tor's performance. With Diff/CA(≤20KB/s), the median downloading time is 5.23s compared with the weighted routing's 20.04s.



(a) CDF

(b) PDF

Figure 6.7: Download Time

From the experiments, we can see that because Tor uses donated computers with limited donated bandwidth, its performance varies from time to time. In the design of a privacy-preserving LBS system such as CAP, we could reduce the communication load through Tor in order to further improve the overall system performance.

CHAPTER 7

RELATED WORK

In the following, we have divided the review of existing research into two sections, namely location privacy *and* network anonymity. The schemes mentioned in the location privacy section are classified as trusted third-party based *and* which do not require trusted third-party. At the end of this section, we will review existing work on a highly customizable solution for location privacy. In the next section, network anonymity, existing schemes are classified as hiding user's network-identities *and* anonymizing network traffic.

7.1 Location Privacy

7.1.1 Trusted third-party middleware based schemes

Most of the existing work in preserving location privacy in LBS involves trusted third party based schemes [12, 19, 28, 2, 3, 18, 55] . The introduction of *k-anonymity* [45] found its usage in preserving location privacy. This scheme can safeguard that a user is indistinguishable from other $k - 1$ users. For example, Gruteser *et al.* in [19] studied the k-area cloaking schemes in which the space is divided into a set of zones where each zone has at least k-sensitive areas. Therefore, the adversary cannot identify which area that the user visits. Gruteser *et al.* in [18] defined k-anonymity in LBS and proposed an algorithm to adjust location resolution based on anonymity requirements. In this scheme, the trusted middleware performs spatial and temporal cloaking, before sending the data to LBS provider. Spatial cloaking is a process where the request from a user is generalized to a sufficiently large area for enough objects to inhabit it, to satisfy the anonymity constraint. Temporal cloaking is a process of delaying the request until k visitors have visited that area. This scheme assumes a system-wide static k value for all messages [12], which is unrealistic in practice as mobile users might have varying privacy protection requirements. It has been shown that when users are in the dense area and move to different directions, the algorithms for k-anonymity become very complicated and the desired anonymity may be degraded.

A customizable framework was thus proposed in [12] where each message can be specified with its preferred spatial (acceptable decrease in the spatial resolution) and temporal (tolerable delay) tolerance and the k value. As in [3], this scheme too requires a large number users to subscribe with the middleware for a single user to gain sufficient anonymity. The *CliqueCloak* algorithm, proposed in this scheme, finds clique that satisfy the anonymity constraints of all messages included in the clique. Since Clique problem is NP-complete, *CliqueCloak* may be computationally expensive. A recent work proposed an interesting extension of the k-anonymity model to include the historic footprints of users rather than using current user locations [55].

Another scheme which uses third party middleware/anonymizer was proposed in [30]. This scheme consists of two components - anonymizer and privacy-aware query processor. The query processor is embedded inside the database server and deals with cloaked spaces instead of exact location. A user specifies her privacy profile, which includes k (k-anonymity) value *and* minimum area within which she wants to hide her location information to the location anonymizer. The location anonymizer component is the trusted third party which accepts the exact location from the user along-with the privacy profile, and obfuscates it using cloaking, to be delivered to the server. The privacy-aware query processor does not return exact value but a candidate list of answers. In this scheme, specification of strict privacy requirements, incurs in high transmission time because a candidate list larger in size, is returned to the user. Another drawback of this scheme is that it involves client side processing overhead of extracting desired information from the obtained candidate list.

7.1.2 Schemes not requiring trusted third-party middleware

As mentioned earlier, it is difficult to find a trusted third party in practice, and thus few schemes were introduced which are user based. One such scheme was proposed in [42], which removes the requirement of trusted third-party by using private information retrieval (PIR) technique. In this scheme, a user sends an encrypted request to the server and is returned with a list of areas/regions. The user then finds the region that locates her and then uses PIR to get all the point of interests within that area/region. Since this scheme uses PIR, it is the first to provide guarantees against the correlation attack (e.g. a user continuously issuing query while on the move, can be identified with a high probability by observing the different cloaked regions of which the user is a part). This scheme is computationally as PIR is widely criticized for being a computationally expensive technique. In another effort to relax the trusted third-party assumption, Mokbel *et al.* in [28] studied a scheme that leverages the peer-to-peer concept. However, the management of trust relationships among autonomous peers in LBS remains an open issue.

Another scheme, which we do not classify as third-party *or* independent of it, was proposed in [43]. This scheme worked towards providing a highly customizable solution for user to achieve location privacy. This includes setting up a policy custodian, where the user formulated policies would be stored, and a policy custodian directory which would direct the location provider to the policy custodian. However a user's privacy can be compromised if an attack is made on the policy custodian directory or the policy custodian. An attacker can obtain important information about the user if user policies get exposed. Another possible concern would be the ability of an average cell phone user to formulate a policy.

7.2 Network Anonymity

7.2.1 Hiding user's network-identities

In the following two schemes, users assume fake identities (pseudonym) to be sent to LBS providers. Here, the aim is to prevent the correlation of sent messages to the sender. The concept of *Mix Zones*, which is a geographical location/area/zone where the user does not receive or transmits frames/data, was introduced in [2] and extended in [3]. This work essentially targets to provide anonymity to the user on the move. The basic technique in this scheme is to hide the identity of the user from the LBS by using constantly changing pseudonyms. By using this technique the LBS provider can not link the user with her advertised identity(pseudonym). In this scheme the attacker/LBS provider is likely to infer user movements across the mix zone and thus this scheme was refined in [3]. The refined scheme provides a method of measuring and providing feedback of level of anonymity the user experiences. However, for a single user to gain sufficient anonymity, this scheme requires a large number users to subscribe with the middleware, which is a disadvantage. Furthermore, privacy-insensitive nodes may not participate due to constrained communication area (*Mix Zones*) [22].

Jiang, in [22] proposed a scheme for location privacy by hiding users' network identities, such as network address . Their approach obfuscates sender identity, timing of transmission and signal strength. Sender's identity is protected using constantly changing pseudonyms while timing information is protected using opportunistic silent periods in which the user does not receive or transfer frames. Silent periods can thus be said to be similar to using *Mix Zones* [3]. The signal strength can be controlled such that only one access point can hear to transmissions and thus can not collude with other access points. This scheme was implemented at the network layer by modifying the wireless network card's driver. Protecting privacy on the network layer makes it less portable than when it is used at application layer, as the user might not get same network layer services when connected in a different network (e.g. user can not carry the wireless router at his home with him, but a mozilla firefox extension has no issues with portability).

7.2.2 Anonymizing network traffic

The schemes discussed in the following too aim to prevent the correlation of sent messages to the sender, but do not use the mechanism of fake identities. Chaum was the first to propose a system for anonymous communication between the sender and the recipient [7]. Subsequently, few more systems, such as Babel [20], Mix-Master [31], Mixminion [8] etc., have been proposed, and are classified as high-latency systems. Tor, *The Second-Generation Onion Router*, is low-latency system, based on the concept of onion routing [46], which tries to anonymize interactive network traffic [10]. Tor seeks to frustrate attackers from linking communication partners, or from linking multiple communications to or from a single user [10]. There are various attacks against anonymous networks e.g. a digital watermarking based technique was proposed to attack the low-latency anonymous communication systems in [54]. Their technique involves injecting a unique watermark in the inter-packet timing domain, and they show that techniques such as cover traffic, packet dropping, timing perturbation etc. do not necessarily make network flows undistinguishable. Wright, in [52], has investigated attacks by corrupt group members in the network, generalized as *predecessor attack*, that result in the degradation of protocols for anonymous communication. A *predecessor attack* exploits the process of path initialization rather than timing information or size of the packet. Anonymous networks have been target of many other attacks such as timing attack, correlation attack, passive logging attack etc. [11, 53]. In the revised version of Tor that we have used, we do not provide any guarantees against such attacks. Rather, we address the issue of QoS degradation.

Rennhard *et al.* in [38] empirically analyzed the performance of web browsing in a mix network. In this scheme, it is shown that instead of many dummy messages, which might unnecessarily utilize bandwidth, real messages can be used to hide a particular message, by using technique of buffering the real messages for short time. McCoy *et al.* in [27] plainly presented some results of Tor's performance measurement including router geopolitical distributions, circuit latency and throughput. A scheme to tune up the performance of Tor was presented by Snader in [41]. Tor makes use of bandwidth values advertised by Tor routers to create routes to the destination. Since the Internet is volatile in terms of fluctuating values of available bandwidth, the authors state that the dependance of a Tor client on the advertised bandwidths may result in overestimating the traffic carrying capacity of a route. Thus a scheme for bandwidth measurement, called as *Opportunistic bandwidth measurement*, was proposed where technique of probing is used to measure available bandwidth. In the revised version of Tor that CAP uses, the bandwidth values advertised by the Tor routers are used. We do not address any attacks based on advertising false bandwidth value (higher) and simply focus on improving the throughput by reducing the latency of communication.

CHAPTER 8

CONCLUSION

In this thesis, we developed CAP to address two challenging issues in privacy-preserving LBS: protection of users location privacy from both location data and network communication perspectives. We have demonstrated that CAP does not require a trusted third-party middleware. Instead, processing is entirely done on the client-side with minimal computational requirements. We have used authentic data provided by US Census Bureau and efficiently reduced its size so that a client requires extremely small storage space. Our real world experiments demonstrate that CAP seamlessly integrates its location perturbation and anonymous routing components. We measure CAP in terms of location privacy, LBS query accuracy and (communication) QoS of the entire system. Its effectiveness is demonstrated by theoretical analysis, simulations, and experiments with an implemented prototype. Our work is the first end-to-end solution to protect location privacy and improve the accuracy of LBS while taking communication QoS into account.

8.1 Future Work

We intend to develop a web browser embedded tool, such as a mozilla firefox extension, which can be used with popular LBS like Google Maps, Mapquest etc. We also intend to develop a full-fledged CAP system for a variety of mobile clients and release it to the public in the near future.

REFERENCES

[1] Arkasha and Bobzilla. WiGLE - wireless geographic logging engine - plotting wifi on maps. `http://www.wigle.net/`, 2008.

[2] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. In *Proc. of the 2rd IEEE Annual Conference on Pervasive Computing and Communication Workshops*, 2003.

[3] A. R. Beresford and F. Stajano. Mix zones: user privacy in location-aware services. In *Proc. of the 2rd IEEE Annual Conference on Pervasive Computing and Communication Workshops*, 2004.

[4] Blister Entertainment. blisterent. `http://www.blisterent.com/`, 2008.

[5] Boost Community. Boost C++ Library. `http://www.boost.org/`, 2008.

[6] S. Bortzmeyer. echoping home page. `http://echoping.sourceforge.net/`, 2008.

[7] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), February 1981.

[8] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *Proc. of the 2003 IEEE Symposium on Security and Privacy (S&P)*, 2003.

[9] R. Dingledine and N. Mathewson. Tor: An anonymous internet communication system. `http://archives.seul.org/or/talk/`, 2006.

[10] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proc. of the 13th USENIX Security Symposium*, 2004.

[11] freehaven. Selected papers in anonymity. `http://www.freehaven.net/anonbib/topic.html`, 2008.

[12] B. Gedik and L. Liu. A customizable k-anonymity model for protecting location privacy. In *Proc. of the IEEE Internation Conference on Distributed Computing Systems (ICDCS)*, 2005.

[13] General Data Resources, Inc. Antennasearch - search for cell towers, cell reception, hidden antennas and more. `http://www.antennasearch.com/`, 2008.

[14] geobytes.com. Ip address locator tool, 2008.

[15] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, Tan, and Kian-Lee. Private queries in location based services: Anonymizers are not necessary. In *Proc. of ACM SIGMOD*, 2008.

[16] D. R. Glover and J. L. Simon. The effect of population density on infrastructure: The case of road building. *Economic Development and Cultural Change*, 23(3):453–468, 1975.

[17] Google Maps. Google Maps for Mobile, My Location feature. `http://www.google.com/mobile/gmm/mylocation/index.html`, 2008.

[18] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spacial and temporal cloaking. In *Proc. of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2003.

[19] M. Gruteser and X. Liu. Protecting privacy in continuous location-tracking applications. *IEEE Security and Privacy*, 2(2):28–34, 2004.

[20] C. Gülcü and G. Tsudik. Mixing E-mail with Babel. In *Proc. of the Network and Distributed Security Symposium (NDSS)*, 1996.

[21] V. Jacobson. Pathchar. `http://www.caida.org/tools/utilities/others/pathchar/`, 1997.

[22] T. Jiang, H. J. Wang, and Y.-C. Hu. Preserving location privacy in wireless lans. In *Proc. of the 5th International Conference on Mobile Systems, Applications, and Service (MobiSys)*, 2007.

[23] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE Transactions on Knowledge and Data Engineering*, 19(12):1719–1733, 2007.

[24] E. F. Krause. *Taxicab Geometry*. Dover, 1987.

[25] Y. Liu, Y. Gu, H. Zhang, W. Gong, and D. Towsley. Application level relay for high-bandwidth data transport. In *Proc. of the 1st International Workshop on Networks for Grid (GridNets)*, 2004.

[26] B. A. Mah. pchar: A tool for measuring internet path characteristics. `http://www.kitchenlab.org/www/bmah/Software/pchar/`, 2005.

[27] D. McCoy, K. Bauer, D. Grunwald, P. Tabriz, and D. Sicker. Shining light in dark places: A study of anonymous network usage. Technical report, University of Colorado at Boulder, 2007.

[28] M. F. Mokbel and C. Y. Chow. Challenges in preserving location privacy in peer-to-peer environments. In *Proc. of the International Workshop on Information Processing over Evolving Networks (WINPEN)*, 2006.

[29] M. F. Mokbel, C. Y. Chow, and W. G. Aref. The new casper: query processing for location services without compromising privacy. In *Proc. of the International Conference on Very Large Data Base (VLDB)*, 2006.

[30] M. F. Mokbel, C. Y. Chow, and W. G. Aref. The new casper: Query processing for location services without compromising privacy. In *Proc. of the 32th International Conference on Very Large Data Bases (VLDB)*, 2006.

[31] U. Möller and L. Cottrell. Mixmaster Protocol — Version 2. `http://www.eskimo.com/~rowdenw/crypt/Mix/draft-moeller-mixmaster2-protocol-00.txt`, January 2000.

[32] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz. Analysis of the clustering properties of the hilbert space-filling curve. *IEEE Transactions on Knowledge and Data Engineering*, 13(1):124–141, 2001.

[33] Moonbuggy. Man accused of stalking with gps, 2004.

[34] D. Niculescu and B. Nath. VOR base stations for indoor 802.11 positioning. In *Proc. of ACM/IEEE International Conference on Mobile Computingand Networking (MOBICOM)*, 2004.

[35] H.-O. Peitgen and D. Saupe. *The Science of Fractal Images.* Springer-Verlag, New York, 1988.

[36] R. Pries, W. Yu, S. Graham, and X. Fu. On performance bottleneck of anonymous communication networks. In *Proc. of the 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2008.

[37] Privacy International Human Rights Group. Privacy International. `http://www.privacyinternational.org/`, 2007.

[38] M. Rennhard, S. Rafaeli, L. Mathy, B. Plattnet, and D. Hutchison. Analysis of an anonymity network for web browsing. In *Proc. of the IEEE the 11-th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2002.

[39] A. Research. GPS-enabled location-based services (lbs) subscribers will total 315 million in five years, 2006.

[40] K. Römer. The lighthouse location system for smart dust. In *Proc. of ACM Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2003.

[41] R.Snader and N.Borisov. A tune-up for tor: Improving security and performance in the tor network. In *Proc. of the 15th Annual Network and Distributed System Security Symposium*, 2008.

[42] R. Sion and B. Carbunar. On the computational practicality of private information retrieval. In *Proc. of the Network and Distributed Security Symposium (NDSS)*, 2007.

[43] E. Snekkenes. Concepts for personal location privacy policies. In *EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 48–57, New York, NY, USA, 2001. ACM Press.

[44] Sunbeltblog. Cell phone tracking, 2005.

[45] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[46] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. In *Proc. of IEEE Symposium on Security and Privacy (S&P)*, 1997.

[47] A. C. Tamhane and D. D. Dunlop. *Statistics and Data Analysis: From Elementary to Intermediate.* Prentice-Hall, New Jersey, 2000.

[48] The Onion Router. Tor:anonymity online. `http://www.torproject.org/`, 2008.

[49] Trolltech. Qt Cross-Platform Application Framework. `http://trolltech.com/products/qt/`, 2007.

[50] US census bureau. TIGER/Line. `http://www.census.gov/geo/www/tiger/tiger2006se/tgr2006se.html`, 2008.

[51] X. Wang, S. Chen, and S. Jajodia. Network flow watermarking attack on low-latency anonymous communication systems. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy (S&P)*, 2007.

[52] M. Wright, M. Adler, B. N. Levine, and C. Shields. An analysis of the degradation of anonymous protocols. In *Proc. of the Network and Distributed Security Symposium (NDSS)*, 2002.

[53] M. Wright, M. Adler, B. N. Levine, and C. Shields. Defending anonymous communication against passive logging attacks. In *Proc. of the 2003 IEEE Symposium on Security and Privacy (S&P)*, 2003.

[54] S. C. X. Wang and S. Jajodia. Network flow watermarking attack on low-latency anonymous communication systems. In *Proc. of the 2007 IEEE Symposium on Security and Privacy (S&P)*, 2007.

[55] T. Xu and Y. Cai. Exploring historical location data for anonymity preservation in location-based services. In *Proc. of IEEE International Conference on Computer Communications (INFOCOM)*, 2008.

[56] W. Yu, X. Fu, S. Graham, D. Xuan, and W. Zhao. Dsss-based flow marking technique for invisible traceback. In *Proc. of the 2007 IEEE Symposium on Security and Privacy (S&P)*, 2007.

[57] Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao. On flow correlation attacks and countermeasures in mix networks. In *Proc. of Workshop on Privacy Enhancing Technologies (PET)*, 2004.

BIOGRAPHICAL STATEMENT

Aniket S. Pingley was born in Nagpur, India, in 1984. He received his Bachelor of Engineering degree from Nagpur University, India, in Computer Technology in 2005. In 2008, he received his Master of Science degree with a Thesis from The University of Texas at Arlington in Computer Science and Engineering. He has been part of the Information Security(iSEC) Lab at UTA since January 2007. His research interests include Location Privacy in Wireless Networks, Bluetooth Worms and Peer to Peer Networks.