2007

Exploring historical location data for anonymity preservation in location-based services

Ge Toby Xu
Iowa State University

Exploring historical location data for anonymity preservation in location-based services

by

Ge (Toby) Xu

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Ying Cai, Major Professor
Wensheng Zhang
Ahmed Kamal

Iowa State University

Ames, Iowa

2007

UMI Number: 1447584

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

We present a new approach for K-anonymity protection in Location-Based Services (LBSs). Specifically, we depersonalize location information by ensuring that each location reported for LBSs is a cloaking area that contains K different footprints—*historical locations* of different mobile nodes. Therefore, the exact identity and location of the service requestor remain anonymous from LBS service providers. Existing techniques, on the other hand, compute the cloaking area using *current locations* of K neighboring hosts of the service requestor. Because of this difference, our approach significantly reduces the cloaking area, which in turn decreases query processing and communication overhead for returning query results to the requesting host. In addition, existing techniques also require frequent location updates from all nodes, regardless of whether or not these nodes are requesting LBSs. Most importantly, our approach is the first practical solution that provides *K-anonymity trajectory protection* needed to ensure anonymity when the mobile host requests LBSs continuously as it moves. Our solution depersonalizes a user's trajectory (a time-series of the user's locations) based on the historical trajectories of other users.

CHAPTER 1. Introduction

A major concern of the large-scale deployment of location-based services (LBSs) is the safeguards of the client location data collected by service providers. To use an LBS, a user needs to submit her location information. A person's whereabouts, however, may imply sensitive private information. For example, physical destinations such as medical clinics may indicate a person's health problems. Likewise, regular stops at certain types of places may be linked directly to one's lifestyles or political associations. Although the service users may be informed of the policies regarding to the collection and distribution of their location data, the execution of these policies is typically beyond the users' control and relies solely on the service providers.

To prevent potential abuses of location data, a viable solution is to allow users to preserve their anonymity in requesting LBSs. For this purpose, however, simply using a pseudonym is not sufficient because a user's location itself may reveal her real-world identity. For example, if a reported location belongs to a private property, it is likely that the user is the owner of the property. Once a subject is identified, all her visits to other locations may be disclosed. This problem has motivated a series of research efforts on location depersonalization (e.g., by Gruteser and Grunwald (11), Gedik and Liu (10), and Mokbel *et al* (21)). Existing techniques employ a central anonymity server as a trusted proxy between mobile nodes and the providers of LBSs. The anonymity server tracks the movement of mobile nodes. When a node requests an LBS, the server computes a *cloaking* box that contains the client node and $K-1$ other nodes, and then uses this box as the client's location to request the LBS. Converting an accurate position into a cloaking box prevents a reported location from being linked to a subject with a certain degree of anonymity protection – a cloaking box with K nodes inside provides K-anonymity protection to the service user.

Reducing location resolution has been shown effective in depersonalizing location information. However, existing techniques compute a node's cloaking box based on the position of its current neighbors. Therefore, these techniques share several limitations. First of all, *all* mobile nodes, regardless of whether or not they request LBSs, must report their location frequently to the anonymity server in order for it to track their latest position. In reality, nodes not needing LBSs may not be willing to spend their resources to help others maintain their anonymity. Excessive location updates from a large number of mobile nodes also present overwhelming communication and processing bottlenecks on the server side. In addition to the practicality and scalability issues, another problem is that the sizes of cloaking boxes produced by the existing approaches are highly dependent on the network density. When a node is in an unpopulated area, its cloaking box can be very large since it needs to contain the node itself and at least $K-1$ other nodes to provide K-anonymity protection. A fine cloaking resolution is critical for the quality of an LBS. An LBS server can retrieve and send back all query results pertaining to a cloaking box, but this would incur additional computation and communication costs.

The above limitations are not the most serious problem that arises from using neighbors' position for cloaking. The most serious problem, in our opinion, is that this strategy is feasible only for depersonalizing an individual location instance, but not a time-series sequence of them. Indeed, existing techniques are practically impossible to support anonymity for *continuous* LBSs such as GM's OnStar services (4), or general processing continuous nearest queries (40) (41) and continuous range queries (42) (43) (44). Continuous LBSs require frequent location updates from their clients. Simply ensuring each reported location is a cloaking box containing at least K nodes does not give a user K-anonymity protection. This is due to the fact that a time-series sequence of cloaking boxes forms a trajectory that may reveal the real identity of the user if, for instance, it links to the user's home and office. It may first appear that one can confuse a trajectory by associating each cloaking box with a different pseudonym. Unfortunately, using different pseudonyms, or simply not using identifiers at all, may not be effective. Since successive location samples are highly correlative, they could be re-linked based

on a common trajectory using trajectory tracking methods (e.g., Multi-Target Tracking (22)) without the need to know any identifiers.

To preserve a node's anonymity from its trajectory, one solution is to choose K nodes in the first cloaking box, and then ensure these K nodes are included in all future cloaking boxes generated for the user. Since the trajectory covers the moving paths of K different nodes, this approach supports *trajectory K-anonymity*. It, however, does not work in many cases. Since the K nodes may move in different directions, future cloaking boxes will become increasingly large, and eventually unacceptable for any meaningful LBSs. Ultimately, it is untenable to preserve a node's future anonymity based on its current neighbors.

In this thesis, we investigate location depersonalization from a new perspective, aiming at addressing the above limitations. Public areas like parks and highways are naturally depersonalized spatial regions – they are not private property like house and office which can reveal a subject's identity; and such areas are characterized by a large number of visits by different people at different times. In light of this observation, we propose using users' *footprints*, instead of their current positions, for cloaking. A footprint is defined to be a user's location sample collected at some time point. The more footprints a spatial region has from different people, the less likely it can be correlated to identify a subject successfully. Thus, we can cloak a node's position based on its nearby footprints left by other people. Cloaking with footprints makes it possible to prevent users not engaged in LBSs from having to report their location. This strategy can also significantly improve the cloaking resolution, since a spatial region can be exported as long as it has been visited by a certain number of different people. In particular, it provides a practical means to support trajectory K-anonymity: Given a user's expected route, the anonymity server can cloak it with $K - 1$ historical trajectories collected from other mobile nodes.

The main contributions of this work are as follows. We propose to depersonalize location information using footprints, instead of the current positions of mobile nodes. We consider both single location cloaking and trajectory cloaking. For the former, which depersonalizes a user's current position, we propose a polynomial-time algorithm that can find the minimal

bounding circle that bounds the user and at least $K-1$ others. Up to date, there is only one technique that considers minimizing cloaking area, and its complexity is exponential. Trajectory cloaking deals with depersonalizing a user's time-series of location samples. To our knowledge, no practical solution can be found for this purpose in literature. We give a formal definition of K-anonymity trajectory (KAT) and develop efficient algorithms for computing such trajectories with cloaking resolution that is as fine as possible. The effectiveness of our techniques is studied under various conditions using location data synthetically generated based on real road maps.

The rest of the thesis is organized as follows. In Section 2, we give an overview of our system model. We present techniques for single location cloaking and trajectory cloaking in Section 3 and 4, respectively. The proposed techniques are evaluated in Section 5. We discuss more related work in Section 6 and conclude the thesis in Section 7.

CHAPTER 2. Background and Related Work

2.1 Location-based Service

The continuous advances in wireless technologies and positioning systems have created a large number of online mobile appliances that are location-aware (2). The huge customer base has attracted a strong commercial interest in location-based services (LBSs), which tail information services according to their clients' current location. According to the report in *IT Roadmap to a Geospatial Future* (1), LBSs are expected to seamlessly and ubiquitously integrate into our life. Some examples of LBSs are as follows:

- Traffic management: A user's position information is tied to a map in order to offer navigation and directions to specific addresses, gas stations, restaurants, and hospitals.
- E-alert: A user may receive a notification of sales from nearby shopping center, or a warning of traffic jam ahead of his route.
- Safety guard: The administrator can monitor tourists traveling in dangerous terrain for fast emergencies reaction.
- Gaming and entertaining: It is possible to provide some location-based ICQ services. For example, a number of virtue treasures, each associated with a physical location, can be created for a treasure-hunting game.

FCC's Phase II E911 requires wireless carriers to provide precise location information within 50-100 meters for emergency handling. Today, many wireless carriers have leveraged such information and provided their clients some kinds of LBSs such as transportation assistance. In addition to the carriers, a large number of third parties have also started to offer LBSs

such as GM's OnStar (4) and NextBus (3). While there is no doubt about their values, LBSs presents unique and heightened threats to individuals.

2.2 Location Privacy

Location privacy has been a significant concern since early 1990s (34) (35) (36) (37) (38). Existing research addresses this issue from several fronts. On the legislation front, the legal standards governing the collection and distribution of personal location data have been or are in the process of being enacted in some countries. The European Union, for instance, has introduced Directive 2002/58/EC (28), which requires explicit consent from users in order to have their information collected. On the technical front, the Internet Engineering Task Forces's Geopriv (29) working group has developed a set of protocols and APIs for secure storage and transferring of location information. On a personal level, location management was investigated in (30) (31) (32) (33). The proposed frameworks allow users to control when and to whom their location information can be released. Nevertheless, these approaches generally do not work when location data are subject to risks such as potential misuse by insiders, unintentional or mistaken disclosure, and access by unauthorized individuals.

The problem of location anonymity was first studied by Gruteser and Grunwald (11). As an extension of the traditional K-anonymity model (27) (26) (20) (18) (19), they proposed to reduce the accuracy of a user's location information along spatial and/or temporal dimensions for a certain level of anonymity protection. Specifically, spatial cloaking is used to ensure that every location reported to a service provider is a cloaking area that contains at least K nodes. If the resolution of a location is too coarse for quality services, temporal cloaking is applied, i.e., delaying a user's service request. When more mobile nodes come near to the user, a smaller cloaking area can then be computed. This basic concept has inspired a series of research publications. In (10), Gedik and Liu considered allowing users to specify their own value of K and minimizing the size of the cloaking areas, a factor critical for the quality of LBSs. The proposed *CliqueCloak* algorithm, however, incurs high computation overhead and is appropriate only when the value of K is small. The techniques proposed in (21), (16) and (39),

by Mokbel et al, Kalnis et al and Cheng et al, respectively, also support customization of K,+ but do not minimize the size of cloaking areas. An important contribution of these two works is their consideration of query processing, i.e., how a location-dependent query can be processed with a location of reduced resolution. While these techniques rely on a central anonymity server which functions as a middleware in between service users and service providers, the work in (9) by Chow et al assumes a decentralized mobile peer-to-peer environment. To request a service, a user broadcasts a message to find $K-1$ peers in its vicinity, and then uses the region that contains all these K nodes as its cloaking area. Despite their differences, all these techniques cloak a node's position based on its neighbors' location and guarantee K-anonymity protection under the assumption that each location report is an independent event.

In their pioneering work (6), Beresford and Stajano investigated the challenges of hiding a user's moving trace and proposed an innovative concept called *mix zone*. A mix zone is defined to be a spatial region in which a mobile node does not report its location. When there are multiple nodes inside a same mix zone, they exchange their pseudonyms. After exiting the mix zone, these nodes start to use new pseudonyms in location updates, making it hard for an adversary to link incoming and outgoing paths of these nodes. While this approach relies on a set of pre-defined spatial regions for pseudonym exchanging, the path confusion algorithm proposed by Hoh and Gruteser in (14) allows mobile nodes to switch their pseudonyms when their paths are close to each other, say, within some threshold. These approaches are not designed to support anonymous uses of LBSs, since they report users' true position to service providers. Another technique designed for trace hiding was using false dummies investigated in (17). In this scheme, each location submitted to a service provide is accompanied by $K-1$ false dummies, which are generated to simulate the movement of mobile nodes. By making $K-1$ faked traces, the trace of a service user is under K-anonymity protection. This approach, however, cannot support anonymous uses of LBSs either. The adversary can identify the dummy trace as a fake if a sample false dummy is located inside a non-residential region (e.g. a lake), or the trace passes through multiple spatial regions that exclusively belongs to different users. Under these circumstances, the user's anonymity is compromised. The problem

of trajectory anonymity is also investigated in (8) and (15). Their basic idea is to ensure all cloaking boxes generated for a user include a common set of $K-1$ other users covered by the first cloaking box. As mentioned early, this strategy will make future cloaking boxes larger and larger and eventually meaningless for LBSs.

In (12), Gruteser and Liu considered the problem of hiding sensitive areas (e.g., a night club) visited by users from adversary. This work classifies areas as either sensitive or insensitive. When a node is in a sensitive area, its location update is suppressed. Otherwise, it reports its location, but reduces the location accuracy when moving close to some sensitive area. Specifically, when the node is in a region around sensitive areas, it reports an area that contains at least K sensitive areas as its current location. In this way, an adversary cannot find which sensitive area the user actually enters. This work protects a user's location privacy – an adversary may be allowed to know a subject's identity, but not the sensitive areas it visits. This is different from the aforementioned location anonymity which protects user identity by ensuring each cloaking area contains at least K users, rather than K sensitive areas.

CHAPTER 3. System Overview

Figure 3.1 illustrates the overall system architecture, where an anonymity server is used as a proxy agent for mobile nodes to receive LBSs. For LBSs that require user authentication (e.g., for service charges), we assume anonymous authentication such as those proposed in (13)(25)(23) is used. These schemes apply the concept of blind signature and allow a service provider to verify a user's legitimacy without having to request her true identity. Like other existing techniques, we assume that the anonymity server is part of trusted infrastructures managed by some cellular service provider, through which mobile users have access to wireless communications.

The cellular service provider offers anonymization services as a value-added feature to their clients, and supplies the anonymity server with the initial footprint database for cloaking. The location samples in the database may be collected from clients' regular phone calls. If such an initial database does not exist, we assume a location sampling phase, during which mobile nodes report their location periodically to the anonymity server. Unlike existing techniques, such periodic location update is no longer needed after the sampling phase, which may last only a short time period (e.g., a few days). More location data can be obtained from mobile nodes in their requests of LBSs and will be subsequently added to the database to improve cloaking resolution. Hereafter, we will use terms location sample and footprint interchangeably. Recall that a trajectory is a time-series sequence of footprints collected from a same user. Thus, the database can be considered as a trajectory repository.

Today's localization technologies allow cellular service providers to determine the position of a caller within a radius of 50 to 300 meters. In contrast, a GPS-enabled mobile device can detect its own position more precisely, up to 10 meter accurate. Due to this imperfect

positioning, we use a spatial region, a circular region in particular, to represent each location sample. A rectangle can also be used to represent a location sample. However, rectangles of different shapes can have the same area, making it less desirable for cloaking.

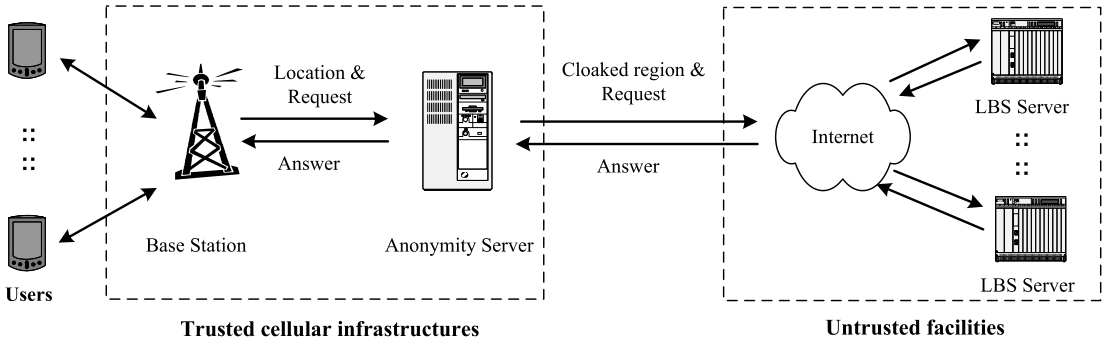

Figure 3.1 System architecture

For efficient retrieval of location data, we index the footprint database using a simple grid-based approach, as illustrated in Figure 3.2. We partition the network domain recursively into cells in a quad-tree style. Unless a cell is already at its minimal size (our implementation sets each cell to be at least 500×500 $meter^2$), it is split if the number of users who have footprints inside it exceeds some threshold. For each cell, we maintain a cell table, which stores a list of pointers that link to the trajectories which have at least one footprint that overlaps with the cell. Specifically, each tuple of a cell table is a record of $(uid, tlink)$, where uid is the ID of a mobile node which traverses this cell, and $tlink$ is a pointer that links to the node's trajectory information. Thus, given a cell, we can efficiently retrieve the trajectories that pass through the cell.

Supporting an instant LBS: To request an instant LBS, a mobile node reports its current location c and a desired anonymity level K to the anonymity server. In response, the server computes a circular region that contains c and $K - 1$ footprints, each from a different user, and exports this region to the provider of the LBS. Based on this location information, the provider delivers the requested services (e.g., query results) to the anonymity server, which then forwards to the service user.

Supporting a continuous LBS: To receive a continuous LBS, a user reports the anonymity

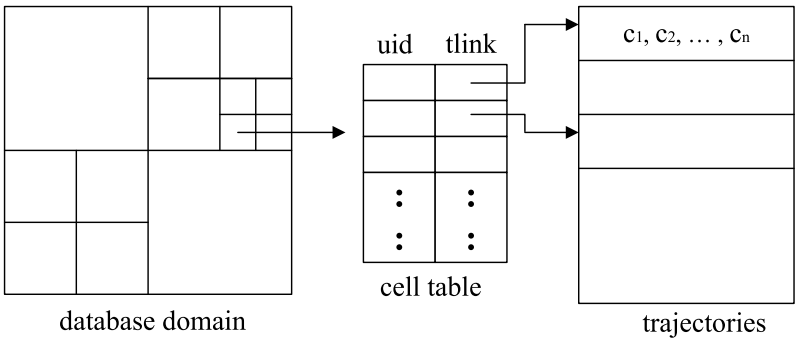

Figure 3.2 Footprint Database

server a *base* trajectory $T_0 = \{c_1, c_2, ..., c_n\}$, where c_i is a location sample on the trajectory along which the user will move. For better quality of services, the user may choose to have more location samples on the base trajectory. The user may also let the anonymity server generate the trajectory by giving a starting position and a final destination. Given an anonymity level K and a base trajectory T_0, the server selects from the footprint database $K-1$ other users' trajectories, each having at least n footprints, and uses them to cloak T_0. The cloaking procedure will generate a *K-anonymity trajectory* (KAT) $T = \{C_1, C_2, ..., C_n\}$. By covering T_0 and footprints from at least other $K-1$ nodes, T can provide the user K-anonymity protection. A formal definition of KAT will be given later. After computing T, the server contacts the provider of the requested LBS to start a service session. As the node moves along the base trajectory T_0, it reports to the server whenever it arrives at c_i. In response, the server exports the corresponding C_i to request the service on behalf of the user. When the service session terminates, the location data reported by the service user is added to the footprint database for future cloaking.

CHAPTER 4. Single Location Cloaking

Existing techniques depersonalize a mobile node N's location by converting it into a spatial region that contains N and at least $K-1$ other nodes. To our knowledge, (10) is the only one that considers minimizing cloaking area, a factor that is critical for quality LBSs. The proposed algorithm, *CliqueCloaking*, needs to compute the clique graph and therefore is NP-hard. Because of high computation cost, this algorithm is feasible only when K is small. In this thesis, we present a novel polynomial-time algorithm that can find the minimal bounding circle (MBC) that bounds N and at least $K-1$ other nodes. To facilitate our discussion, we use C_{min} to denote this MBC, C_a a bounding circle that contains N and at least $K-1$ other nodes, and C_b the circle centered at N with a radius that is two times of that of C_a. Also, given a circle C, we denote its radius as $C.R$. These notations are illustrated in Figure 4.1. Our algorithm of searching C_{min} is based on the following observation:

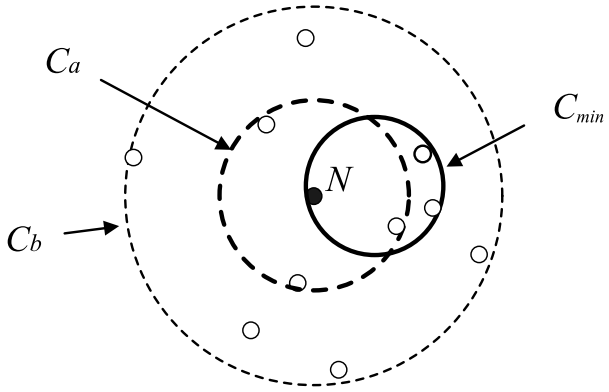

Figure 4.1 C_{min} must be inside C_b ($K = 4$)

Theorem 1 *C_{min} must be bounded by C_b.*

Proof 1 *By its definition, C_a contains K nodes including N. Since C_a is a candidate of C_{min},*

C_{min} must be no larger than C_a, i.e., $C_{min}.R \leq C_a.R$. Since both C_{min} and C_a contain N, the distance between any point in C_{min} and N's position must not be larger than $2 \cdot C_a.R$. As a result, C_{min} must be inside C_b.

The problem is how to find a C_a with a small radius. This can be done in different ways, depending on how the positions of mobile nodes are indexed. For instance, if R-tree is used, we can find N's $K-1$ nearest neighbors and use the MBC that bounds N and these $K-1$ nodes as C_a. Existing techniques (e.g., (24)) can find KNN at a cost of $O(K \log K)$. If a quad-tree is used (e.g., see Figure 3.2), the following simple approach can be used to find a C_a. First, we find the cell where N locates and mark this cell as the searching box. If the number of nodes inside the searching box is less than K, we expand the searching box by including its adjacent cells. This process is repeated until the searching box contains at least K nodes. Among these nodes, we find $K-1$ nodes that are nearest to N and set C_a to be the MBC that bounds these $K-1$ nodes and N. This approach costs $O(K)$

After locating a C_a, we then determine C_b and retrieve all nodes inside C_b. Let S be the set of these nodes and $|S|$ the number of them. As the area of C_b is 4 times of that of C_a, the number of nodes inside C_b is $O(K)$. Given C_b and the set of nodes inside it, we now construct the candidates for C_{min} and then select the one that has the smallest radius as C_{min}. Since C_{min} is the minimum circle that contains N and at least $K-1$ other nodes, there must have at least two nodes on the circle line of C_{min}. Thus, we can classify C_{min}'s candidates into two categories.

A candidate in the first category has exactly two nodes on its circle line. In this case, the two nodes must form a diameter of the candidate. Such candidates can be enumerated by considering all possible pairs of the nodes inside C_b. Given a pair of nodes, we construct the circle with the two nodes as its diameter. The circle is a valid candidate if it contains N and at least $K-1$ other nodes. Among all valid candidates, we find the one that has the smallest diameter. Let this candidate be C. Given a set of nodes S, there are totally $\binom{|S|}{2}$ different pairs of nodes. Thus, the computational cost in this step is $O(K^2)$.

A candidate in the second category has at least three nodes on its circle line. Note that any three of nodes can form a triangle and in a two-dimension domain (as long as they are not on the same line), and a triangle can form only one circumscribed circle. Thus, we can enumerate all possible triple nodes in S. For each triple, we construct the circumscribed circle formed by the three nodes. If the circle contains N and at least $K-1$ other nodes, it is a valid candidate. Again, among all valid candidates, we find the one that is smallest. Let this candidate be C'. Since the number of possible triples is $\binom{|S|}{3}$, the computation cost in this step is $O(K^3)$.

Finally, we compare C with C', and the smaller one is C_{min}. Since the total cost of the entire process is $O(K) + O(K^2) + O(K^3) = O(K^3)$, the above algorithm finds C_{min} in a polynomial time. Its pseudo code is given in Algorithm 1.

Algorithm 1 FindMBC(N, K)

$S_a \leftarrow N$'s $K-1$ nearest neighbors
$C_a \leftarrow$ MBC of all nodes of S_a
$C_b.centre = N.pos$
$C_b.R = 2 \cdot C_a.R$
$S_b \leftarrow$ all nodes inside C_b
$C_{min} = C_a$
for any two nodes X and Y in S_b **do**
    $C \leftarrow$ the circle of which $\overline{XY}$ is diameter
    if C contains N and at least $K-1$ other nodes in S_b, **and** $C.R < C_{min}.R$ **then**
        $C_{min} \leftarrow C$
    end if
end for
for any three nodes in S_b as X, Y, Z **do**
    $C \leftarrow$ the circumcircle of triangle XYZ
    if C contains N and at least other $K-1$ nodes in S_b, **and** $C.R < C_{min}.R$ **then**
        $C_{min} \leftarrow C$
    end if
end for
return C_{min}

CHAPTER 5. Trajectory Cloaking

For continuous LBSs, a user needs to report a base trajectory $T_0 = \{c_1, c_2, ..., c_n\}$. In response, the anonymity server will compute a new trajectory $T = \{C_1, C_2, ..., C_n\}$ that can provide the user K-anonymity protection. For this purpose, T must cover T_0. In addition, it must also cover footprints from at least $K-1$ trajectories (from different users), which we will refer to as *additive* trajectories. Let these trajectories be T_1, T_2, ..., T_{K-1}, and $T_j = \{a_{[j,1]}, a_{[j,2]}, \cdots, a_{[j,m_j]}\}$, where $1 \leq j \leq K-1$ and m_j denotes the number of footprints in T_j. We give a formal definition of *K-anonymity trajectory* (KAT) as follows.

Definition 1 *T is a* KAT *of T_0, iff for each circle C_i in T, the following conditions are satisfied:*

1. *C_i covers c_i in T_0, i.e., $c_i \subseteq C_i$;*
2. *C_i covers at least one footprint in each additive trajectory;*
3. *For any C_i and C_{i+1}, there exist two footprints $a_{[j,x]}$ and $a_{[j,y]}$ in each additive trajectory T_j such that $a_{[j,x]} \subseteq C_i$, $a_{[j,y]} \subseteq C_{i+1}$, and $x < y$.*

The first two conditions ensure that each circle in T covers at least K location samples, each in a different trajectory. Given an additive trajectory T_j, it is not necessary to have all of its footprints covered by T in order to provide K-anonymity protection to T_0. Instead, we just need to make sure that T covers at least n footprints that are in the same order as they appear in T_j. The third condition in the above KAT definition is to guarantee this requirement. Figure 5.1 illustrates an example of KAT, where $K = 3$.

Given a trajectory $T = \{C_1, ..., C_n\}$, we define its resolution to be $|T| = \frac{\sum_{i=1}^{n} Area(C_i)}{n}$, where $Area(C_i)$ denotes the area of spatial region C_i. For quality of services, a KAT's resolution

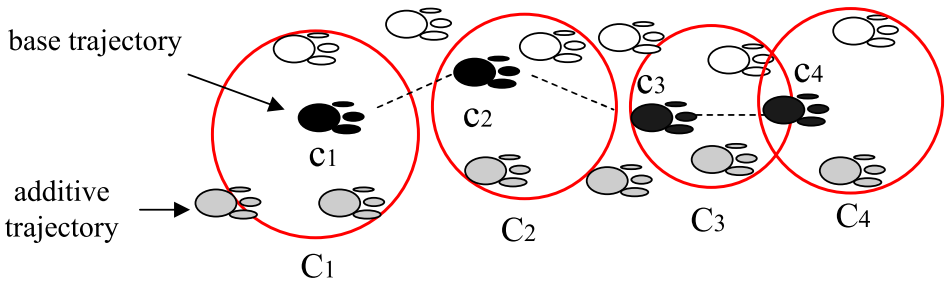

Figure 5.1 An example of K-anonymity trajectory

needs to be as fine as possible. Given a database of N trajectories, there are $\binom{N}{K-1}$ different trajectory sets with cardinality $K-1$. For each of these sets, its $K-1$ trajectories can be used as the addictive trajectories to cloak base trajectory T_0. Given a set of $K-1$ additive trajectories, different orders of cloaking will also result in different KATs. Enumerating all possible combinations allows us to find the KAT with the best cloaking resolution, but this would require intensive computation. In the following subsections, we first discuss how to cloak T_0 with one trajectories, and then apply the proposed algorithm to cloak T_0 with a set of $K-1$ trajectories. Finally, we discuss how to select a small set of trajectories for cloaking from a potentially large number of trajectory candidates.

5.1 Cloaking One Additive Trajectory

Consider cloaking T_0 with an additive trajectory T_a. Let $T_0 = \{c_1, c_2, \cdots, c_n\}$, $T_a = \{a_1, a_2, \cdots, a_m\}$, where $n \leq m$, and $T = \{C_1, C_2, \cdots, C_n\}$ be the cloaking result. For each circle C_i in T, it needs to contain c_i and at least one footprint in T_a. Thus, to minimize cloaking area, we can set C_i to be the minimum bounding circle (MBC) that contains b_i and some footprint in T_a. When a footprint in T_a is selected to create the MBC for C_i, we call this footprint C_i's pivot. Because of the ordering constraint of KAT, not every footprint in T_a can serve as C_i's pivot. To circumvent this problem, we can create a set of pivots by selecting n footprints from T_a and using them as pivots based on their index number as follows. Let this set of n footprints be $\{a_{p_1}, a_{p_2}, ..., a_{p_n}\}$, where $p_1 < p_2 < ... < p_n$; then for all $1 \leq i \leq n$, a_{p_i} is used as C_i's pivot. The cloaking trajectory generated by this approach must be a KAT.

The first two conditions are satisfied because C_i is the MBC that bounds c_i and its pivot, a footprint selected from T_a. The third condition is also satisfied because the pivots included in T are in the same order as they appear in T_a.

The challenge is how to select a set of pivots that can result in the best cloaking resolution. Given a set of pivots $\{a_{p_1}, a_{p_2}, ..., a_{p_n}\}$, we have $T = \{MBC(c_1, a_{p_1}),\ MBC(c_2, a_{p_2}),\ ...,\ MBC(c_n, a_{p_n})\}$, where $MBC(c_i, a_{p_i})$ denotes the minimum bounding circle that bounds c_i and a_{p_i}. To find T with the best resolution, we can find all different sets of pivots, and for each set, compute the corresponding T's resolution. Since there are totally $\binom{m-1}{n-1}$ different sets of pivots, such exhaustive search may not be feasible in practice. To address this problem, we develop a simple yet effective approach to generate pivots for each C_i, starting from $i = 1$. For C_1, we select its pivot from the following $m - n + 1$ candidates: a_1, a_2, ..., and a_{m-n+1}. For each candidate, we compute the MBC that bounds this candidate and c_1. The candidate that results in the smallest MBC is then selected as C_1's pivot a_{p_1}. Let a_{p_1} be the footprint selected as C_1's pivot, where $1 \leq p_1 \leq m-n+1$. Then, we select C_2's pivot from the following $m - n + 2 - p_1$ candidates: a_{p_1+1}, ..., and a_{m-n+2}. Again, for each of these candidates, we compute the MBC that bounds this candidate and c_2, and then select the one with the smallest MBC as C_2's pivot. Suppose a_{p_2} is selected as C_2's pivot, where $p_1 + 1 \leq p_2 \leq m - n + 2$. We then select C_3's pivot from the following $m - n + 3 - p_2$ candidates: a_{p_2+1}, ..., and a_{m-n+3}, based on their corresponding MBCs (with c_3). The same procedure is used to select the pivot for each of the rest of the circles in T. The complexity of this heuristic algorithm is $O(m)$.

When determining a pivot, it is possible that multiple candidates result in the same smallest MBC. In this case, the one with the smallest index is chosen as the pivot. This would give more candidates choices when selecting the next pivot. It is worth mentioning that the above procedure selects each pivot from a certain range of footprints in T_a. For C_1, its pivot is selected from T_a's first $m - n + 1$ footprints. For all $i > 1$, C_i's pivot is selected the range from $a_{p_{i-1}+1}$ to a_{m-n+i}. The pseudo code of the cloaking procedure $Cloak(T_0, T_a)$ is given Algorithm 2. To illustrate this process, we use an example shown in Figure 5.2. T_0 and T_a have 4 and 9 location samples, respectively. For C_1, its pivot can be selected from a_1 to a_6.

Since $MBC(c_1, a_2)$ is the smallest, a_2 becomes C_1's pivot. For C_2, we can then select its pivot from a_3 to a_7. After selecting a_4 as C_2's pivot, we proceed to select C_3's pivot, which has four candidates ranged from a_5 to a_8. Note that a_6 is chosen as the pivot although $MBC(c_3, a_6)$ and $MBC(c_3, a_7)$ have the same size. As a result, C_4 can have two candidates, a_7 and a_8, to select its pivot.

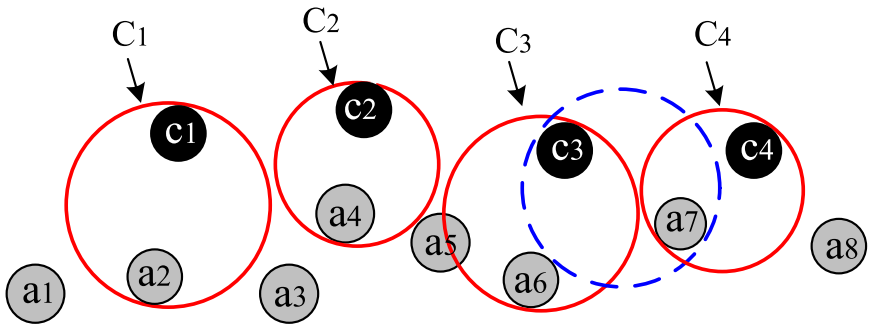

Figure 5.2 Example of cloaking T_0 with T_a

Algorithm 2 Cloak(T_0, T_a)

```
1: p ← 0
2: for 1 ≤ j ≤ n do
3:     M ← ∞
4:     for p < i ≤ m − n + j do
5:         if M > Area(MBC(c_j, a_i)) then
6:             M ← Area(MBC(c_j, a_i))
7:             p' ← i
8:         end if
9:     end for
10:    C_j ← MBC(c_j, a_p')
11:    p ← p'
12: end for
13: T ← {C_1, C_2, ··· , C_n}
```

5.2 Cloaking $K-1$ Additive Trajectories

With $Cloak(T_0, T_a)$ in place, we now consider how to generate a KAT for T_0, given a set of additive trajectories S. Let $S = \{T_1, T_2, ..., T_s\}$, where $s \geq K-1$, and let $T_i = \{a_{[i,1]}, a_{[i,2]}, \cdots, a_{[i,m_i]}\}$, where $1 \leq i \leq s$ and m_i denotes the number of footprints in T_i. To

generate a KAT for T_0, we need to cloak T_0 with $K-1$ additive trajectories. Clearly, choosing different additive trajectories can have vastly different cloaking results. Even with a fixed set of $K-1$ additive trajectories, the order of cloaking can also affect the cloaking resolution of the cloaking results.

To avoid exhaustive search, we propose two heuristic approaches, *Linear* and *Quadratic*. The former incurs less computation costs, but the latter can lead to better cloaking results. Linear works as follows. For each trajectory T_i in S, it calls $Cloak(T_0, T_i)$ to generate a cloaking trajectory, which we will denote as T_i'. If T_i' has a better resolution than T_j', we say T_i is closer to T_0 than T_j. The trajectories in S are then sorted based on their distance to T_0 in ascending order, and the first $K-1$ trajectories (which are closest to T_0) are selected as T_0's additive trajectories. Let these sorted trajectories be T_1'', ..., T_{K-1}'', where T_i'' is closer to T_0 than T_j'' for all $1 \leq i < j \leq K-1$. The $K-1$ trajectories are then used to cloak T_0 one by one recursively. Specifically, T_0 is first cloaked with T_1''. The cloaking result is considered as a new base trajectory and cloaked with T_2''. The new cloaking result is then cloaked with T_3'' and so on so forth until all $K-1$ trajectories are added. We call this algorithm Linear as it calls $Cloak(T_0, T_i)$ $s+K-1$ times. Its pseudo code is given in Algorithm 3.

Algorithm 3 Linear(T_0, $\mathbf{S}$)

```
1: {S = {T_1, T_2, ..., T_s}}
2: for 1 ≤ i ≤ s do
3:     T_i' ← Cloak(T_0, T_i)
4:     calculate |T_i'|
5: end for
6: S' ← Sort S in ascending order based on distance to T_0
7: T ← T_0
8: {Suppose S' = {T_1'', T_2'', ··· , T_s''}}
9: for 1 ≤ i ≤ K − 1 do
10:    T ← Cloak(T, T_i'')
11: end for
12: return T
```

In Linear, additive trajectories are selected based on their distance to T_0. The distance also determines the order of cloaking. This simple strategy falls short in some cases because it does not consider the spatial relationships among the additive trajectories. This problem is

addressed by Quadratic at a higher computation cost. This scheme also has $K-1$ iterations, and in each iteration, it selects a new additive trajectory to cloak the trajectory, say T, which is generated in the previous iteration. However, the selection of the new additive trajectory is based on its distance to T, instead of T_0. Initially, T is set to be T_0. In each iteration, it calls $Cloak(T, T_j)$ for each T_j in S. Among all generated trajectories, the one with the best resolution is set to be T, and the corresponding T_j is removed from S. After repeating this cloaking and selecting process $K-1$ times, T is output as T_0's KAT. In the above approach, procedure $Cloak(T_0, T_a)$ is called $(K-1)\cdot(s-\frac{K-2}{2})$ times. The pseudo code for Quadratic is given in Algorithm 4.

Algorithm 4 Quadratic(T_0, **S**)

```
1: {S = {T_1, T_2, ..., T_s}}
2: T ← T_0
3: for 1 ≤ i ≤ K − 1 do
4:     for all T_j ∈ S do
5:         T'_j ← Cloak(T, T_j)
6:         calculate |T'_j|
7:     end for
8:     compare |T'_j| for all T_j ∈ S
9:     T'' ← the trajectory that is closest to T
10:    T ← Cloak(T, T'')
11:    S ← S − T''
12: end for
13: return T
```

5.3 Selecting Additive Trajectory Candidates

In both Linear and Quadratic, the entire set of trajectories S is scanned in the process of selecting $K-1$ additive trajectories. Since the number of trajectories recorded in the footprint database can be very large, it is necessary to create a small set of additive trajectory candidates before starting a cloaking process. Obviously, only those trajectories close to the base trajectory should be considered as the candidates. In our implementation, we use the following approach to build a set of additive trajectory candidates given a base trajectory T_0. We first find out all cells that overlap with T_0's location samples. These cells are marked as

searching boxes. According to their cell tables, we then retrieve the trajectories that traverse through all of these cells. If the total number of these trajectories is less than $K-1$, we expand the search scope by merging each searching box and its adjacent cells together as a new searching box. For the new searching boxes, we retrieve the set of trajectories that pass through them. This process is repeated until the cardinality of the trajectory set is at least $K-1$, which are then chosen as the additive trajectories to generate KAT.

CHAPTER 6. Performance Study

For performance study, we have implemented a prototype of the proposed system. For single location cloaking, we compare the performance of the proposed technique using *footprints* and the traditional scheme using *real-time locations*; For trajectory cloaking, we evaluate the proposed two trajectory cloaking approaches, namely *Linear* and *Quadratic*. For comparison purpose, we have also implemented a native approach, referred to as *Baseline* hereafter, which uses the current position of mobile nodes for cloaking. This scheme sets a node N's first cloaking circle to be the MBC that contains the node and at least $K-1$ others. Among the nodes in the circle, Baseline selects $K-1$ nodes that are nearest to N as N's companies. From then on, each time N makes a location update, Baseline finds the MBC that contains N and these $K-1$ companies and reports this MBC as N's cloaking circle.

We modify the *Network-based Generator of Moving Objects* (7) to generate mobile nodes and simulate their movement on the real road map of Oldenburg, Germany, a city about $15 \times 15 km^2$ (Figure 6.1). We extract four types of roads from the road map, primary road (interstate expressway), secondary road (state road), connecting road and neighborhood road as defined in census TIGER/Line (5). In our simulation, mobile nodes change their speeds at each intersection based on a normal distribution determined by the road type. The mean speeds and the standard deviations of moving speeds for each road type are listed in Table 6.1. We generate a footprint database that contains a certain number of trajectories with randomly assigned user IDs. These trajectories are indexed using the grid-based approach discussed in the system overview section.

We are mainly interested in the potential impact of a cloaking technique on the quality of LBSs. For this purpose, we select *cloaking range*, defined to be the average radius of cloaking

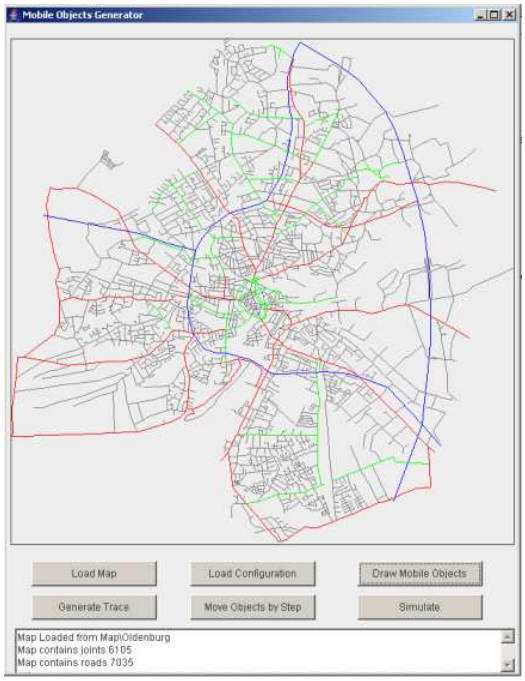

Figure 6.1 The map of Oldenburg loaded in generator

circles in a KAT, as our performance metric.

Table 6.1 Traffic parameters

Road type	Mean speed	Standard deviation
Primary	$100km/h$	$20km/h$
Secondary	$60km/h$	$15km/h$
Connecting	$45km/h$	$10km/h$
Neighborhood	$30km/h$	$5km/h$

6.1 Single location cloaking

For single location cloaking, we compare the performance of two schemes. The proposed scheme that cloaks a service user's position using footprints, which we refer to *footprint-based cloaking* (FC), and the traditional approach based on the actual positions of mobile nodes in the service user's neighboring area, which we refer to *neighborhood-based cloaking* (NC). In each simulation, we generated 5000 mobile users and randomly distributed them in the map.

We randomly selected 200 mobile users, each submits a service request. We varied the value of K from 5 to 100, and investigate the impact of anonymity requirement (i.e., the value of K, as requested by users) on the performance of the two techniques. The performance results are plotted in Figure 6.2. It shows that the neighborhood-based cloaking performs much worse than the footprint-based approach. Furthermore, when K increases, the average cloaking range under the traditional approach increases dramatically, while the average cloaking range under the proposed scheme is much less sensitive to the change of K. For the former, larger K means more mobile users should be included in the cloaking circle. Since the users are randomly distributed in the network, the size of the circle must be larger. For the latter, the number of footprints in an area is much more than the number of people inside. The server always be able to find enough different footprints in a user's neighboring area. As we can see, even when the K increase to 100, the cloaking resolution of the footprint-based cloaking is still finer than the resolution of the neighborhood-based cloaking that $K = 5$. Thus, the footprint based scheme can effectively support the secure use of LBS with high anonymity requirement.

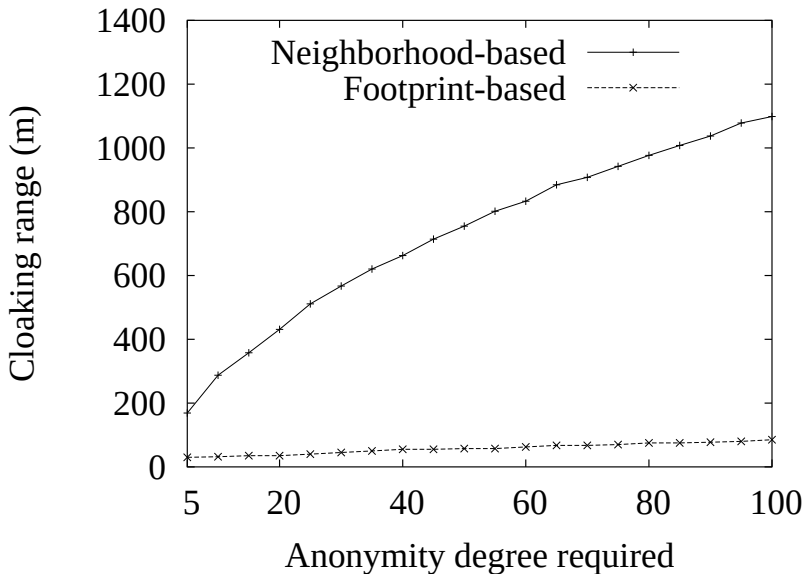

Figure 6.2 Effect of anonymity requirement for single location cloaking

6.2 Trajectory cloaking

For trajectory cloaking, we evaluate the performance of the two proposed techniques and the Baseline. In each simulation, we generate a set of LBS requests. Each request contains a user's ID, the start and destination of a travel plan, and a required anonymity degree. The start and destination are randomly selected from the map, and the fastest path between them is picked as the user's expected route. We select a location sample every 100 meters along the route and these samples form the user's base trajectory. Other parameters used in our study are given in Table 6.2. In the following subsections, we report how the performance of the three techniques is affected by various factors.

Table 6.2 Experiment settings

parameter	range	default	unit
Number of users	5000	5000	*unit*
Anonymity level	10 - 20	15	*unit*
Trajectory database size	$100K - 300K$	$200K$	*unit*
Base trajectory length	$3K - 8K$	$5K$	*meter*
Service request number	200	200	*unit*
Minimum cell size	50×50	50×50	$meter^2$

6.2.1 Effect of anonymity level required

In this study, we investigated the impact of anonymity requirement on the performance of the three techniques. The footprint database used in this study contains 200,000 trajectories. We generated 200 service requests, each having a route of 5000 meters with 500 meters deviation. The value of K is varied from 10 to 20. The performance results are plotted in Figure 6.3. When K increases, the average cloaking range under all schemes increases, as shown in Figure 6.3 (a). However, Baseline always results in the largest cloaking ranges, about 10 times more, as compared to the other two. Given a service user, Baseline needs to ensure that all cloaking circles generated for the user include a common set of K nodes. Since these nodes may move on different directions, the cloaking range becomes increasingly large. When K is

larger, the cloaking results also deteriorate quicker. As for the other two schemes, Figure 6.3 (a) shows that Quadratic always outperforms Linear. This, however, is achieved at a more computation overhead.

Figure 6.3 (b) shows the average cloaking range on different types of roads. The primary and secondary roads are popular. A small space on such roads may have a large number of footprints from different users. Thus, the cloaking range is not very sensitive to the value of K. As the figure shows, the corresponding two curves are almost flat. In contrast, the connecting roads and neighborhood roads are less popular and have a much less number of trajectories passing through them. When K increases, the average cloaking range increases sharply, since a cloaking trajectory may have to cover different roads in order to guarantee a sufficient level of anonymity protection. In reality, a user's route typically covers different types of roads, and a large portion of the route is on highways. Since it is the cloaking circles along these popular areas that dominate the average cloaking range, cloaking with footprints allows users to select a large K for anonymity protection while maintaining good cloaking results.

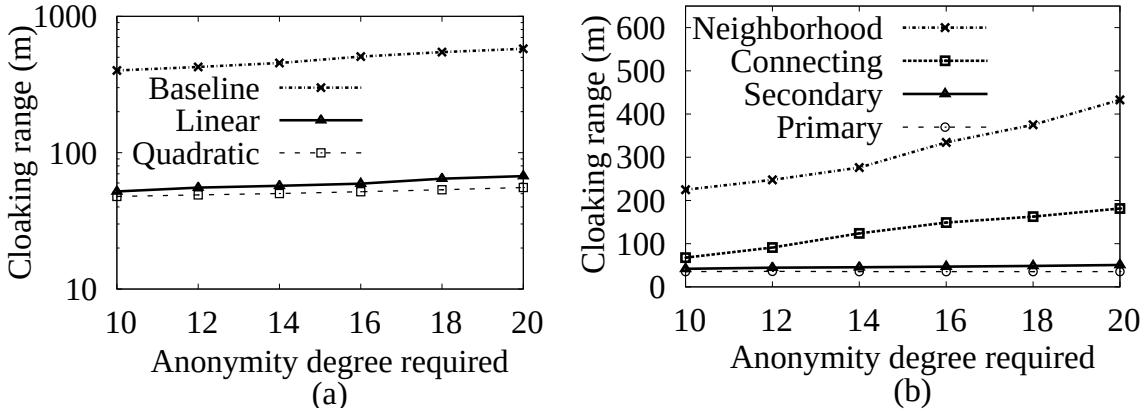

Figure 6.3 Effect of anonymity requirement

6.2.2 Effect of base trajectory length

In this study, we investigated the impact of length of base trajectories on the performance of the three techniques. The footprint database used in this study contains 200,000 trajectories.

In each simulation run, we set $K = 15$ and generated 200 base trajectories. The average length of these base trajectories is varied from 3000 meters to 8000 meters. The performance results are shown in Figure 6.4. Under all three schemes, the average cloaking range increases as the trajectory length increases, as shown in Figure 6.4 (a). However, Baseline performs much worse as compared to its counterparts. It is worth mentioning that the cloaking range under this scheme increases sharply as the base trajectory length increases. This again convinces that cloaking with neighbors' location is untenable for anonymity protection in continuous LBSs. As for Linear and Quadratic, both are little sensitive to the base trajectory length. As explained in the previous study, when a large portion of a user's trajectory is on highways, the cloaking circles on the highways determines the average cloaking ranges. Since our simulation uses the fastest path between a start and a destination as a user's route, when the user's base trajectory becomes longer, the increased portion is most likely on the the highways. Figure 6.4 (a) also shows that Quadratic consistently outperforms Linear. In popular areas, base trajectories and their corresponding additive trajectories usually overlap each other, so the cloaking order does not have much impact on the cloaking results. Figure 6.4 (b) again shows that the average cloaking range on popular roads is much smaller than that on unpopular roads. Also, as base trajectories become longer, the cloaking range increases on unpupular roads, but remains almost constant on popular roads.

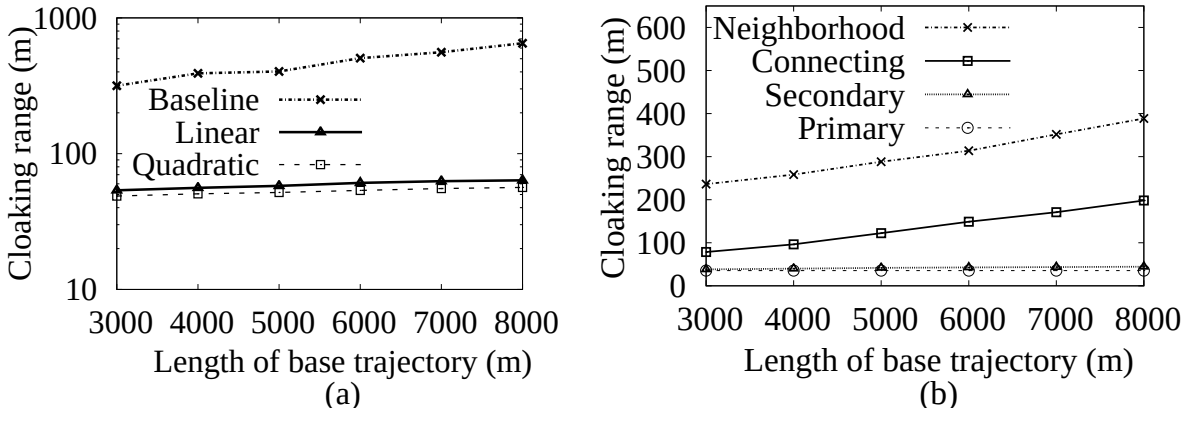

Figure 6.4 Effect of trajectory length

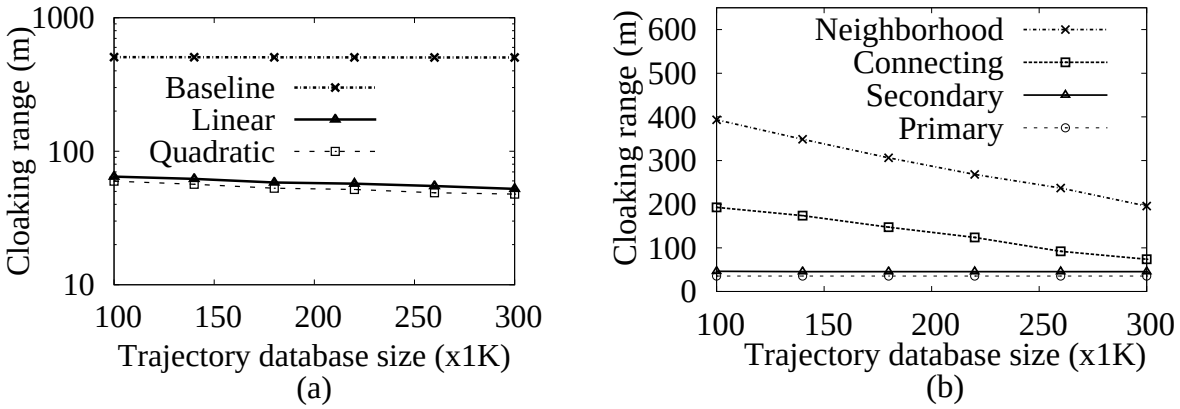

Figure 6.5 Effect of trajectory database size

6.2.3 Effect of the number of historical trajectories

This study investigates the impact of the number of trajectories in the footprint database. We varied the number of trajectories in the database from 100,000 to 300,000. For each simulation, we generated 200 base trajectories, each averaged at 5000 meters with a deviation of 500 meters. We set $K = 15$ for each service request. The performance results are plotted in Figure 6.5. It is shown in Figure 6.5 (a) that the curve for Baseline is flat. This is not a surprise since this scheme uses only the current position of mobile nodes for cloaking. As for Linear and Quadratic, both have better cloaking results when the database contains more trajectories. Clearly, more historical trajectories means more choices in selecting additive trajectory candidates for cloaking. With the same anonymity level, it can then find enough additive trajectories by searching in a smaller range for a base trajectory. Thus, the generated KATs have a smaller cloaking range. Since base trajectories can be added to the database for future cloaking, our proposed techniques can generated better and better cloaking results as more and more footprints are collected. This feature makes them especially attractive for large-scale anonymization services. Figure 6.5 (b) shows that the increase of the number of historical trajectories has a significant impact on the average cloaking range on unpopular roads, but not on popular roads. On the expressway or state roads, there is a sufficient number of footprints for cloaking, even when the database contains as few as 100,000 trajectories. In

contrast, for the unpopular roads, adding some new trajectories could increase their popularity substantially.

CHAPTER 7. Concluding Remarks

Personal location data can be correlated with restricted spaces such as home and office addresses for subject re-identification. This is probably the most practical and economic way for an adversary to identify the anonymous users of LBSs, which can cover a global wide area with a large number of clients. To address this problem, we depersonalize users' location based on historical location data. For the users of sporadic LBSs, we ensure each location reported to their service providers has been visited by at least K different people. For the users of continuous LBSs, we ensure each trajectory reported has been traversed through by at least K different people. The idea makes it much less costly to support anonymous uses of LBSs as evidenced in our performance study.

Nevertheless, several interesting research problems arise from exploring historical location data for anonymity protection. For instance, to cloak a trajectory, we should also consider selecting the additive trajectories of mobile nodes with similar moving speeds during similar time spans to further prevent an adversary from concluding what historical trajectories may have been used. We also plan to investigate on-the-fly cloaking to provide anonymity protection when a user needs to take a significantly different route after submitting the initial trajectory for cloaking. Simply cloaking the new route may jeopardize the user's anonymity if the two trajectories contain footprints from different sets of users.

BIBLIOGRAPHY

[1] (2003). Computer Scence and Telecommunications Board. *IT Roadmap to Geospatial Future*. The National Academics Press.

[2] C. S. Jensen, A. Friis-Christensen, T. B. Pedersen, D. Pfoser, S. Saltenis, and N. Tryfona. (2001). Location-based services - a database perspective. *Proc. of the Eighth Scandinavian Research Conference on GeoGraphical Information Science*. 59–68.

[3] (2007). NextBus. *http://www.nextbus.com*

[4] (2007). Onstar Technology, Global Positioning System, GPS Maps. *http://www.onstar.com*.

[5] (1990). TIGER/LINE CENSUS FILES. *http://www.land.state.az.us/alris/doc/apendh.txt*.

[6] A. R. Beresford, and F. Stajano. (2003). Location Privacy in Pervasive Computing. *IEEE Security and Privacy*, *2*, 46–55.

[7] T. Brinkhoff. (2002). A Framework for Generating Network-Based Moving Objects. *GeoInformatica*, *6*(2), 153–180.

[8] C. Bettini, X. S. Wang, and S. Jajodia. (2005). Protecting Privacy Against Location-Based Personal Indentification. *Proceedings of the 2nd VLDB Workshop on Secure Data Management*, 185–199.

[9] C. Y. Chow, M. F. Mokbel, and X. Liu. (2006). A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-based Services. *ACM GIS'06*, 171–178.

[10] B. Gedik, and L. Liu. (2005). A Customizable k-Anonymity Model for Protecting Location Privacy, *ICDCS'05*, 620–629.

[11] M. Gruteser, and D. Grunwald. (2003). Anonymous Usage of Location-based Services through Spatial and Temporal Cloaking, *ACM MobiSys'03*, 31–42.

[12] M. Gruteser, and X. Liu. (2004). Protecting Privacy in Continuous Location-Tracking Applications. *IEEE Security and Privacy. 02*, 28–34.

[13] Q. He, D. Wu, P. Khosla. (2004). Personal Control over Mobile Location Privacy. *IEEE Communications Magazine*, *42*(5), 130–136.

[14] B. Hoh, and M. Gruteser. (2005). Location Privacy Through Path Confusion. *IEEE/CreateNet Intl. Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm).*

[15] A. Inan, and Y. Saygin. (2006). Location Anonymity in Horizontally Partitioned Spatial-Temporal Data. *Master Thesis, Sabanci University, Turkey.*

[16] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. (2006). Preserving Anonymity in Location Based Services. *Technical Report TRB6/06, Department of Computer Science, National University of Singapore.*

[17] H. Kido, Y. Yanagisawa, and T. Satoh". (2005). An Anonymous Communication Technique using Dummies for Location-based Services. *IEEE ICPS'05*, 88–97.

[18] K. LeFevre, D. DeWitt, and R. Ramakrishnan. (2005). Incognito: Efficient Full-Domain K-Anonymity. *SIGMOD'05*, 49–60.

[19] K. LeFevre, D. DeWitt and R. Ramakrishnan". (2006). Mondrian Multidimensional K-Anonymity. *ICDE'06*, 25.

[20] A. Meyerson, and R. Williams. (2004). On the Complexity of Optimal K-Anonymity. *PODS'04*, 223–228.

[21] M. F. Mokbel, C.-Y. Chow and W. G. Aref. (2006). The New Casper: Query Processing for Location Services without Compromising Privacy. *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB'06)*, 763–774.

[22] D. Reid. (1979). An Algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control*, *24*(6), 843–854.

[23] K. Ren, W. Lou, K. Kim and R. Deng. (2006). A Novel Privacy Preserving Authentication and Access Control Scheme in Pervasive Computing Environments. *IEEE Transactions on Vehicular Technology*, *55*(4), 1373–1384.

[24] N. Roussopoulos, S. Kelley, and F. Vincent. (1995). Nearest Neighbor Queries. *Proceedings of ACM SIGMOD'95*, 71–79.

[25] K. Sha, Y. Xi, W. Shi, L. Schwiebert and T. Zhang. (2006). Adaptive Privacy-Preserving Authentication in Vehicular Networks (Invited Paper). *Proceedings of IEEE International Workshop on Vehicle Communication and Applications.*

[26] L. Sweeney. (2002). A Model for Protecting Privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, *10*(5), 557–570.

[27] L. Sweeney. (2002). Achieving k-anonymity Privacy Protection Using Generalization and Suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, *10*(5), 571–588.

[28] (2002). DIRECTIVE 2002/58/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL. http://europa.eu.int/eur-lex/pri/en/oj/dat/2002/l_201/l_20120020731en00370047.pdf.

[29] J. R. Cuellar, J. B. Morris, and D. K. Mulligan. (2002). Geopriv Requirements. *Internet draft*". http://www.ietf.org/internet-drafts/draft-ietf-geopriv-reqs-01.txt.

[30] E. Snekkenes. (2001). Concepts for Personal Location Privacy Policies. *Proceedings of the 3rd ACM Conference on Electronic Commerce.* 48–57.

[31] S. Duri, M. Gruteser, X. Liu, P. Moskowitz, R. Perez, M. Singh, and J. Tang. (2002). Framework for Security and Privacy in Automotive Telematics. *Proceedings of the second International Workshop on Mobile Commerce*. ACM Press. 25–32.

[32] G. Myles, A. Friday and N.Davies. (2003). Preserving Privacy in Environments with Location-based Applications. *IEEE Pervasive Computing.(02)* 56–64.

[33] M. Langheinrich. (2003). A Privacy Awareness System for Ubiquitous Computing Environments. *4th International Conference on Ubiquitous Computing 2498*, 237–245.

[34] P. A. Karger, and Y. Frankel. (1995). Security and Privacy Threats to ITS. *Proceedings of the Second World Congress on Intelligent Transport Systems. 5*, 2452–2458.

[35] R. Want, A. Hopper, V. Falco, and J. Gibbons". (1992). The Active Badge Location System. *ACM Transactions on Information Systems (TOIS). 10*(1), 91–102.

[36] P. E. Agre. (1995). Transport Informatics and the New Landscape of Privacy Issues. *Computer Professionals for Social Responsibility (CPSR) Newsletter. 13*(3)

[37] L. Barkhuus, and A. Dey. (2003). Location-based Services for Mobile Telephony: A Study of Users' Privacy Concerns. *9th International Conference on Human-Computer Interaction.*

[38] J. Warrior, E. McHenry, and K. McGee. (2003). They Know Where You Are. *IEEE Spectrum.*

[39] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Preserving User Location Privacy in Mobile Data Management Infrastructure. *6th Workshop on Privacy Enhancing Technologies.* 393–412.

[40] Z. Song, and N. Roussopoulos. (2001). K-nearest Neighbor Search for Moving Query Point. *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases (SSTD'01).* 79-96.

[41] Y. Tao, D. Papadias, and Q. Shen. Continuous Nearest Neighbor Search. (2002). *Proc. of International Conference on Very Large Data Bases (VLDB'02)*. 287-298. Hong Kong, China.

[42] B. Zheng, W.-C. Lee, and D.L. Lee. (2004). Search Continuous Nearest Neighbors on the Air. *the First International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous'04)*. 236-245. Boston, MA, U.S.A..

[43] M. F. Mokbel, X. Xiong, and W. G. Aref. (2004). SINA: Scalable Incrementable Processing of Continuous Queries in Spatio-temporal Databases. *SIGMOD '04*. 623-634. Paris, France.

[44] Y. Cai, K. A. Hua, G. Cao, and T. Xu. (2006). Real-Time Processing of Range-Monitoring Queries in Heterogeneous Mobile Databases. *IEEE Transactions on Mobile Computing*. *5*(7), 931-942.