PRESERVING QUERY PRIVACY WITH A QUERY-BASED MEMORIZING ALGORITHM

A Thesis by

Jiang Cheng

Bachelor of Science, Xi'an University of Post and Telecommunication, 2009

Submitted to the Department of Electrical Engineering and Computer Science
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

May 2014

PRESERVING QUERY PRIVACY WITH A QUERY-BASED MEMORIZING ALGORITHM

The following faculty members have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Electrical Engineering.

_____
Murtuza Jadliwala, Committee Chair

_____
Vinod Namboodiri, Committee Member

_____
Esra Buyuktahtakin, Committee Member

ACKNOWLEDGEMENTS

I am particularly thankful to my thesis advisor, Dr. Murtuza Jadliwala. He has given me much inspiration when I start to write the thesis. Besides, he is also very patient to give me suggestions and directions. I am grateful to his help during my writing of the thesis. I also want to thank the committee members for their time and effort.

ABSTRACT

Query privacy is a critical concern to users of location-based services. A majority of existing query privacy protection techniques are based on the notion of *k*-anonymity, wherein a user's exact location is obfuscated into a spatial range containing at least *k* users, called the cloaking region. Thus, the user who issues the query cannot be distinguished from *k*-1 other users. However, when mobile users issue continuous queries using such a *k*-anonymity scheme, an adversary can exploit the overlapped areas of the corresponding cloaking regions to determine the query issuer with a significantly higher probability. This thesis proposes a query-based memorizing algorithm to specifically address this issue. The main idea in this thesis is to memorize the identity of the users in an anonymity set or cloaking region. When a user issues sequential location-based queries, the cloaking regions are determined such that they include a maximum number of users that have appeared in the past cloaking regions. The query-based memorizing approach is empirically evaluated by means of simulation experiments and a detailed comparative analysis with three other popular privacy protection algorithms using standard privacy metrics is performed. The results show that the proposed algorithm efficiently protects users' query privacy against the overlapped area attack, especially when users are highly mobile.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS/ NOMENCLATURE

AP	Access Point
AS	Anonymizing Server
CCR	Candidate Cloaking Region
CR	Candidate Result
DLS	Dummy Location Selection
LBS	Location-Based Service
PAD	Privacy-Area Aware Dummy
POI	Point of Interest
QoS	Quality of Service
TTP	Trusted Third Party

CHAPTER 1

INTRODUCTION

1.1 The Problem of Location Privacy

With the rapid proliferation of smart phones and smart devices, location-based services and applications on these devices are gaining popularity. According to the Pew Internet & American Life Project, 18% of smart phone users use online check-in services to share their current location with their friends and 78% use their phones to get real-time location-based information [1]. Location-based services (LBS) compute user queries based on the current location of the user (or the user's device), which is considered to be a part of the user's personal information. There are serious concerns about the loss of user location privacy in LBSs as service providers may use the users' location information for business or other purposes without the knowledge or consent of the user. For example, the LBS provider may spam users with unwanted advertisements. Inferences about the identity or living habits of a user can also be made based on the points of interest (POI) visited by the user [2]. Alternatively, if it is revealed that a user visits a hospital, then one may conclude that the user is sick or suffering from a particular disease. This loss of location information is a grave threat to users' privacy, which may lead users to new dangers such as stalking, or even physical harassment [3].

According to Duckham and Kulik [4], location privacy can be defined as "a special type of information privacy which concerns the claim of individuals to determine for themselves when, how, and to what extent location information about them is communicated to others." When location privacy is compromised, the user may leak other personal information, in addition to location information, to an adversary. The amount of information about a user that an adversary can obtain depends on the rate and granularity of location information disclosed by the user, and the extent of background information available to the adversary [5][6].

Location privacy protection has been a widely researched topic in the literature. How to protect users' private location information from disclosure without compromising the Quality of Service (QoS) in location-based service remains a challenge. According to Rinku Dewri [7], location privacy is an often misused term which should be separated into (i) *location privacy* and (ii) *query privacy*. Location privacy deals with preventing an adversary from knowing the user's exact location, while query privacy deals with preventing an adversary, who already has certain location information of users, from mapping a location-based query to the query issuer. Among all the query privacy protection schemes, the most popular method is *location obfuscation*, which is a technique that alters user's exact location in location-based queries into a coarse grained spatial range. Instead of sending her exact location, a user can provide a spatial range, known as the *cloaking region*, which includes her current location. The cloaking region has to be large enough to contain k users, including the one sending the query, and these k users form an *anonymity set*. This technique is also known as location *k-anonymity*.

1.2 Shortcomings of Existing Obfuscation

Unfortunately, existing techniques [2-4, 8-11] for location k-anonymity have one shortcoming: when the user issues multiple continuous queries, multiple cloaking regions are generated as a result of these queries. The overlapped area of these cloaking regions can be exploited by the adversary to identify, with a high probability, the user who issued the query. The details of the attack are described in Chapter 2.

Various techniques [10, 12-16] have been proposed to address the problem of overlapped regions in LBS. However, these techniques have at least one of the following shortcomings: (i) they do not focus on query privacy, (ii) they have high computational cost, which results in lower QoS, or (iii) they do not work well when users are highly mobile, which is often the case. Specifically, the scheme in [10]

is not suitable to cases in which users are highly mobile the scheme in [14] does not consider query privacy.

1.3 Thesis Contributions

To address the above shortcomings, a query-based memorizing algorithm is proposed as an improvement over the schemes in [10, 14]. In this thesis, the main focus is to protect query privacy of mobile users by considering the adversary's prior knowledge and the privacy preferences of the querying users (details in Chapter 3). The main idea of the proposed privacy preserving algorithm is to anonymize the querying user with other LBS users who have been in the same anonymity set in the past. Unlike other memorizing algorithms in the literature, the algorithm proposed in this thesis takes users' privacy preferences into account and adapts to user mobility. In this thesis, a series of simulations are conducted to evaluate the performance of the proposed query-based algorithm by using location data generated on a real road map. Experimental result shows that the proposed query-based memorizing algorithm is efficient in protecting users' query privacy, especially under conditions when users are highly mobile, while providing a QoS (measured by the size of the resulting cloaking region) similar to traditional location *k*-anonymity [11].

1.4 Thesis Organization

The thesis is organized as follows. Chapter 2 outlines the related work on query privacy protection and the motivation of the thesis. Chapter 3 provides an overview of the system architecture and the adversary model. Chapter 4 introduces the proposed query-based memorizing algorithm in this thesis. Chapter 5 presents the evaluation results for the proposed query-based memorizing algorithm. Finally, Chapter 6 outlines the conclusions of the thesis and provides directions for future research.

CHAPTER 2

RELATED WORK AND MOTIVATION

2.1 Location *k*-anonymity

Location *k*-anonymity is a commonly used technique to preserve users' location privacy and query privacy in LBS. The fundamental idea in location *k*-anonymity is to apply the "need-to-know" principle which states that an LBS user should only provide information to a service provider enough to complete the query with certain level of granularity, and nothing more [4]. Most existing research efforts related to location *k*-anonymity [2-4, 8-11], adopt an anonymizing proxy, provided by a trusted third party, that generates an obfuscation or cloaking region for each query issued by the LBS user. In this architecture, the user sends her query and a value *k* to the trusted third party. The third party then chooses *k*-1 other users nearest to the current query issuer and creates a minimal spatial range or region that covers all these *k* users including the current query issuer. This spatial range is the cloaking region for the query issuing user. The third party forwards the user query, with the cloaking region as the user location, to the service provider. In this way, the trusted third party knows all the exact locations of the *k* users along with the identity of the query issuer and the service provider only knows the cloaking region for the query. The service provider processes the query based on the cloaking region and sends the result back to the third party. Generally, there are two ways to process the query: (i) find the POIs nearest to the centroid of the cloaking region; (ii) find all the POIs nearest to every possible location in the cloaking region. Upon receiving the response back from the LBS, the third party filters the results and sends the filtered results back to the query issuer. The privacy level of the query is determined by the value of the parameter *k*. With an increasingly large value of *k*, a user can get better query anonymity because the size of query anonymity set increases with *k*.

2.2 The Overlapped Area Problem

In practice, users are highly mobile and they usually send multiple continuous location-based queries to the LBS. For each query, the anonymizing proxy will create a new cloaking region based on the privacy level (in terms of the size of anonymity set) the user has set before forwarding the query to the service provider. One shortcoming of existing *k*-anonymity schemes is that when a user issues multiple queries, it will result in overlapped areas formed by the cloaking regions of the respective queries. When a user issues continuous queries, the adversary (service providers or attackers who can access information from the service providers) can observe that multiple similar queries within nearby cloaking regions have been sent in a short period of time, thus deducing that these queries were sent by the same user with high probability [14]. With the multiple distinct cloaking regions of these queries, the adversary can also deduce with high probability that the user must be located in the overlapped area of these cloaking regions. The more queries a user sends, the more accurately an adversary can track her. This attack is depicted in Figure 1. Suppose, Alice is looking for the nearest hospital while she is driving. To simplify the exposition, it is assumed that only Alice is moving and that her privacy level remains unchanged. It is also assumed the adversary knows the exact locations of every user, the cloaking region for each query and whether the user has changed her privacy level or not. At time t_1, she sends her first query requiring for 5-anonymity and she is cloaked in the area R_{t1}. At time t_2 and t_3, she issues another two queries, and she is cloaked in the area R_{t2} and R_{t3} separately with different users. To the adversary, her location is then limited to the overlapped area of R_{t1}, R_{t2} and R_{t3}. It is clear that, given the above knowledge, the adversary can deduce the identity of the query issuer with respect to region R_{t3} with a probability higher than 1/*k*.

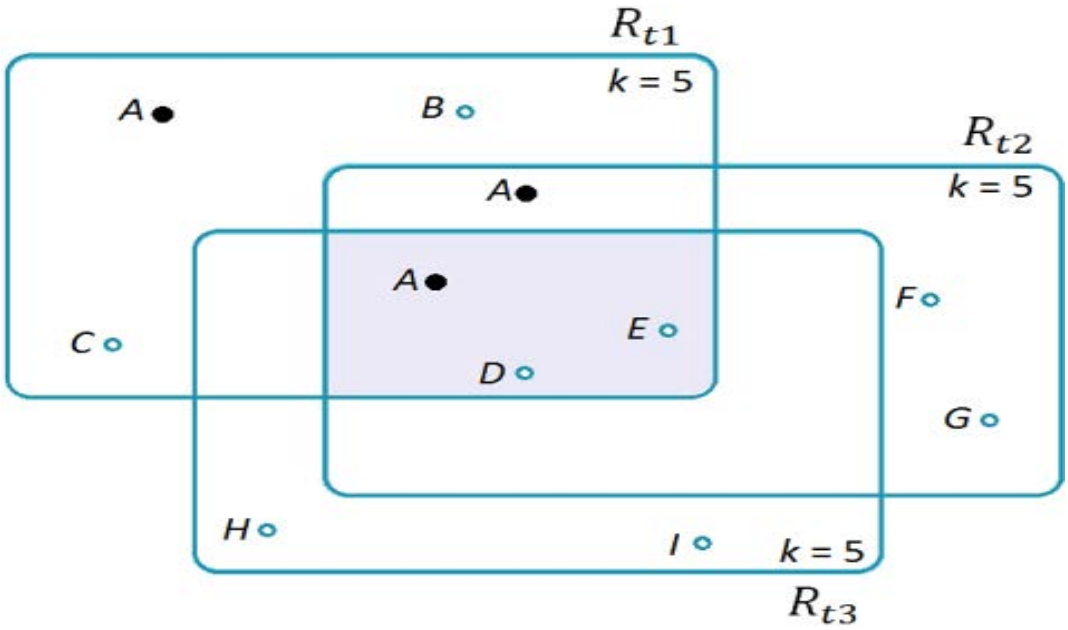


Figure 1: The Overlapped Area Problem

2.3 Techniques to Address the Overlapped Area Problem

One technique that is proposed to overcome the overlapped area problem is to use fake or dummy queries in addition to real queries. Wang and Wu [12] propose that each user sends fake queries constantly to the service provider in order to achieve better obfuscation. In this scheme, no third party is required for location obfuscation as the user himself/herself can generate fake queries. Lu et al. [16] have done some work in a similar direction. They propose a Privacy-area Aware Dummy (PAD) approach where a privacy region is generated with dummy locations. Niu et al. [17] improves the dummy location technique by proposing a Dummy-Location Selection (DLS) algorithm, which achieves better location *k*-anonymity by selecting distributed dummy locations.

Besides these, Amini et al. [13] propose a cache-based solution to address the overlapped area problem. The querying user uses a cache to store location data about her past queries and results. Then, instead of sending multiple queries to the service provider, the user only needs to query from her cache. This method is effective when the user often makes repeated queries within a certain region such as a street or a city. Truong et al. [14] propose a solution based on memorization to address the overlapped

area problem. The query region is divided into grid cells. For each query, the anonymizer will memorize the cloaking region used. If the user is still inside the cloaking region when issuing another query and her privacy level (in terms of the number of grid cells) is unchanged, the same cloaking region will be used for anonymization in that query. In addition, the authors also propose other algorithms for situations in which the user changes her privacy level. However, in this scheme, location *k*-anonymity is not guaranteed as a user's privacy level is determined by how many cells she uses. In low-density regions, even if the user requires a high privacy level, which leads to a large cloaking region, there may still not be enough users in the anonymity set as the current user desires, and the adversary can easily deduce the identity of the user who issued the query.

Similar research on memorization of past queries has been done in [10] and a new memorization algorithm is proposed. In this scheme, users with the same *k* value for *k*-anonymity will form a group and use the same cloaking region. As long as the users are still inside the cloaking region, the group remains. When there are not enough users in the cloaking region, the group expands and then a new cloaking region is used. One obvious problem in this scheme is that when users are highly mobile, it is difficult to maintain the same group for a long time. With members in the group changing constantly, there will be multiple different cloaking regions and thus the overlapped area problem still exists.

As existing location privacy preserving techniques have shortcomings when protecting the users from attacks on the overlapped area problem, an algorithm that is highly resilient to this kind of attack is urgently needed. In order to solve the overlapped area problem, while still guaranteeing location *k*-anonymity for users, this thesis proposes a query-based memorizing algorithm. This algorithm is an improvement to the memorizing algorithms outlined in [10, 14] and focuses on protecting users' query privacy. By means of grid-based maps, the proposed scheme allows the user to choose her privacy level based on how many users she wants to mix with. For a certain user in the past, instead of memorizing

the cloaking region, the proposed scheme allows the anonymizing proxy to memorize the identity of the other users in the anonymity set. For each query from a certain user, the third party will find a group of *k* users (including the current user) that contains as many users as possible from past anonymity sets. The reason for determining cloaking regions in such a way is because an adversary may have additional knowledge about the locations of the user. Thus, to simply memorize the cloaking region is not sufficient to protect users' query privacy. Later, by means of extensive simulations, this thesis will prove that the proposed memorizing algorithm with *k*-anonymity scheme is more secure in preserving query privacy, and does not have a significant impact on the QoS.

CHAPTER 3

SYSTEM ARCHITECTURE AND ADVERSARY MODEL

3.1 System Architecture

Similar to most existing research efforts in the literature [2-4, 8-11, 15], the system in this thesis employs a centralized architecture, as shown in Figure 2. The system architecture considered in this thesis consists of three entities; (i) mobile users, (ii) trusted third party (TTP) and (iii) LBS provider. First, it is assumed that users can determine their exact locations at all times by means of either GPS, cell tower locations or nearest Wi-Fi access points (AP). The TTP runs an anonymizing server (AS), which acts as a gateway or proxy between the user and service provider. Thus, all queries from the user to the LBS provider must pass through the AS. The communication between the user and the AS and the communication between the AS and the service provider is secured using suitable cryptographic techniques. The query region is divided into a grid (consisting of grid cells of certain granularity) in order to accurately retrieve users' location data. In this thesis, the shape of the grid cells is assumed to be exactly square for ease of explanation. The grid and the size of the cells are the same for each user of the system.

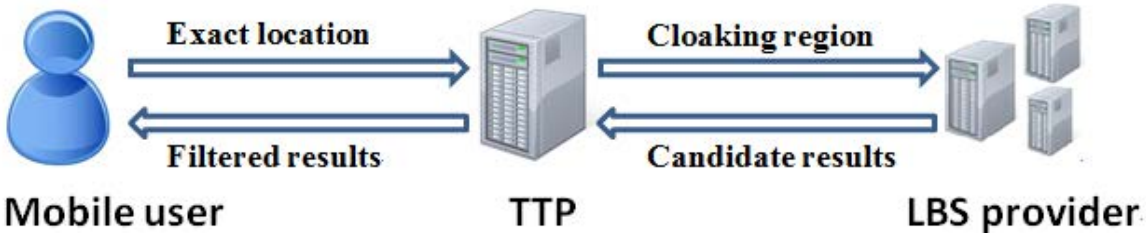


Figure 2: System Architecture

In the architecture, it is assumed that users need to update their location periodically to the AS while they are moving. When a user issues a location-based query (e.g., "where is the nearest gas station"), the client application on the user device sends to the LBS a request in the form of a tuple $Q = \{id, k, l, r, c\}$, where id is the identity of the user, k is the anonymity requirement in terms of the

minimum number of neighboring users (or users in the anonymity set) required, l is the user's exact location, $r \times r$ is the minimum size of the cloaking region required in terms of the number of grid cells, and c is the content of the query. Thus, the user's privacy level is determined by the parameter k and r, which are set by the user herself. Upon receiving a query Q, the AS replaces the user id with a new pseudonym id', and generates a cloaking region $R = \{k, m, x, y\}$, where $m \times m$ ($m \geq r$) is the number of cells in the cloaking region, based on the user's privacy requirements and x, y are the coordinates to represent the exact location of R on the map. Then, the AS will forward this new query $Q' = f(Q) = \{id', R, c\}$ to the LBS server. Here, f is the proposed query privacy-preserving algorithm (details discussed in Chapter 4), which takes a user's actual query Q as an input, generates an obfuscated query Q', and stores the information of both the actual and obfuscated queries in a database maintained at the AS. Records in this database, called the *query records*, stores the history of all the users' past queries and is of the form of $Q'' = \{id, id', ID, t, R\}$, where ID represents the set of the identities of the other users in the anonymity set and t represents the maximum time period after which the record will be removed from the AS database. The set of all the query records Q'' for a user id is denoted as Rec_{id}. Upon receiving Q', the LBS server processes the query and returns the *Candidate Results* (*CR*) of Q'. As the query is processed based on R, the CR for Q' is a superset of the results that could have been obtained based on Q. The CR will then be refined by the AS based on the user's exact location l, and sent to the user through a secure channel.

3.2 Adversary Model

All entities that can get access to the query information sent from the AS to LBS provider, including the LBS provider, are considered as potential adversaries in this thesis. In this thesis it is assumed that the service provider (and other adversaries) is honest but curious. This means that the service provider honestly executes the LBS protocol, but is curious in mapping each query to its

corresponding issuer (or user) based on the knowledge it gets during the execution of the protocol. This includes information it receives before executing the protocol, intermediate information generated due to the protocol execution and the protocol output. The adversary's main objectives are the following: (i) reveal the user's current location or even the actual trace of the user, and (ii) associate a query to a certain user, in accordance with her location.

3.2.1 Adversarial Strength

According to [7], based on the knowledge an adversary has access to, it can be classified into two broad types, namely, *locator* and *holder*. A locator has information about the user's exact location *l*. A perfect locator knows all the exact locations $L = \{l_1, l_2 \ldots l_k\}$ of the users upon receiving a query in a *k*-anonymous cloaking region. However the adversary doesn't know which one of the *k* users issues the query. In this case, query privacy is preserved with no location privacy. An approximate locator has partial knowledge (say, an area instead of the exact location) about the users' locations. A holder has access to the query from LBS provider. A holder must at least have approximate location knowledge of the users. A perfect holder can accurately map the query with the query issuer.

3.2.2 Specific Adversarial Assumptions

This thesis makes the following assumptions about the adversary:

Assumption 1: As the adversary can access query information (query content and cloaking region), the adversary is a holder. In this thesis, it is assumed that the adversary is not a perfect holder.

Assumption 2: The adversary knows the privacy preservation algorithm. In other words, the adversary knows the algorithm *f*, but not the privacy parameter values $\{id, k, r\}$ for a certain query *Q*.

Assumption 3: The adversary has prior knowledge of the mobility pattern of users. For example, if the cloaking region is in a city, the adversary can deduce with high probability that the users are following roads and highways. The maximum and minimum speed of the moving users is also known by the

adversary. Therefore, the adversary can draw a spatial range based on the observed mobility pattern by observing a user query. The spatial range indicates the possible approximate user location in next query. By observing a second query, the adversary can limit the cloaking region into a smaller region by selecting the overlapped area of the spatial range she draws and the cloaking region for the second query.

Assumption 4: The adversary may know the technology used by the user for localization (e.g., if the LBS provider is also the cellular service provider for a user, then the provider can get an approximate location for the user based on the location of the cell tower the user is connected to). The accuracy of the estimation of the user's current location depends on the adversary's capability. In other words, the adversary can be a locator. In this thesis, an adversary who is a perfect locator is referred to as a strong adversary. For a strong adversary, her only objective is to associate a query to its issuing user.

CHAPTER 4

QUERY-BASED MEMORIZING ALGORITHM

4.1 Basic Idea

In this chapter, the query-based memorizing algorithm (*f*) to achieve location obfuscation in continuous queries is outlined. As mentioned in Chapter 3, the query region is divided into predefined square-shaped grid cells. As discussed in Chapter 2, when a user issues multiple continuous queries, traditional *k*-anonymity schemes will results in multiple overlapped areas due to which user queries can be easily deanonymized. The proposed query privacy-preserving algorithm focuses on solving this problem. The main idea of the proposed algorithm is to select the users that have been in the same anonymity set with the current user in the past queries.

4.2 Algorithm Details

The details of the algorithm is described as follows: the input to the algorithm is a user query $Q = \{id,k,l,r,c\}$. Upon receiving Q, the AS searches all the $r \times r$ regions containing the query issuer for those regions that also contain at least k-1 other users. If all these searched $r \times r$ regions contain less than k-1 other users, the AS repeats the search by increasing the size of the searched region by one grid cell, i.e., it searches all $(r+1) \times (r+1)$ regions. The AS repeats this process for regions of size $(r+i) \times (r+i)$ grid cells, where i=2, 3… until regions containing at least k-1 other users are found. The cloaking regions obtained in such a fashion are called c*andidate cloaking regions* (CCR). After determining the various CCRs for the query Q of user *id*, the AS verifies if there is at least one past query record in Rec_{id} which matches (using some similarity measure) the users cloaking requirements.

If none of the memorized queries in Rec_{id} matches the user's requirement, then the AS randomly selects a cloaking region R from the set of CCRs. The AS creates a new identifier *id'* for the query and generates a new (obfuscated) query $Q' = \{id', R, c\}$. The AS will collect the identity of the users inside

the cloaking region and generates a $Q'' = \{id, id', ID, t, R\}$. The AS then stores Q'' in its database Rec_{id} and forwards Q' to the LBS provider.

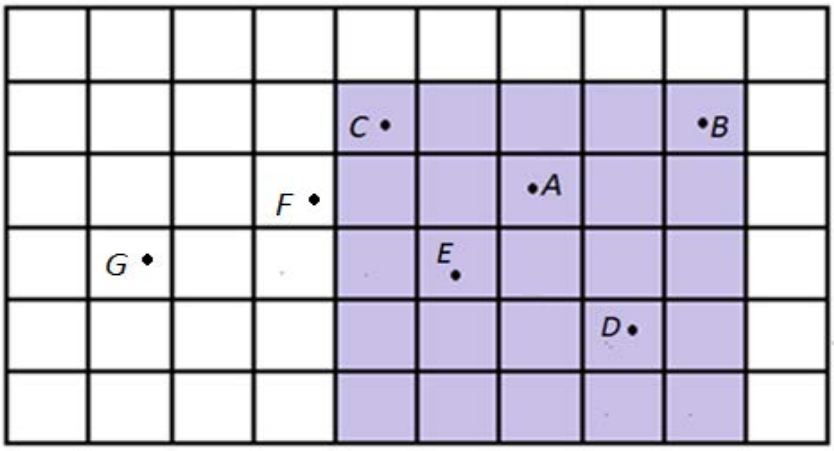


Figure 3: Choosing Square Cloaking Region (1)

Let's consider an example as shown in Figure 3. At time t_1, Alice sends a query Q_1 asking for 5-anonymity with minimum cloaking region size requirement of 3 × 3 cells to the AS. The privacy parameters for Alice are $r = 3$ and $k = 5$. As there are no cloaking region with 3 × 3 cells containing at least 5 users, the AS expands its search to larger regions repeatedly until it finds a region that can contain at least 5 users. A cloaking region R of size 5 × 5 cells will be randomly selected among the regions that cover the user. In this case, the selected cloaking region R is the square in shadow and the corresponding anonymity set is $ID_1 = \{A,B,C,D,E\}$. The AS will store $Q_1'' = \{id_A, id_A', ID_1, t, R\}$ into its database.

Now on the contrary, if there are matching past cloaking regions in Rec_{id}, the AS compares the CCRs with the matched cloaking regions in the query record set Rec_{id} and returns the CCR that is most similar to any matched cloaking region in the query record set Rec_{id}. The similarity value between two regions is computed using the following metrics:

$$sim(R_1, R_2) = \left\{ \begin{array}{l} \dfrac{ID_{R_1} \cap ID_{R_2}}{k}, if \left(\left(ID_{R_1} \cap ID_{R_2} \right) < k \right) \\ 1, if \left(\left(ID_{R_1} \cap ID_{R_2} \right) \geq k \right) \end{array} \right\} \quad (1)$$

In Equation (1), ID_{R_1} and ID_{R_2} denote the set of users, including the query issuer, in the anonymity set for cloaking regions R_1 and R_2, respectively. The maximum value of $sim(R_1, R_2)$ is 1, when the users in the two anonymity sets corresponding to the two regions have at least k users in common. Otherwise, the similarity value will be decided by the number of the intersected users in the anonymity sets corresponding to the regions. For each CCR, the maximum similarity value with the matched cloaking regions in Rec_{id} is set as the similarity value of the CCR. The CCR that has the largest similarity value is then selected as the actual cloaking region for the query Q. If there are multiple CCRs that have the largest similarity value, the AS will randomly choose one and forward it to the LBS providers in Q'.

Figure 4: Choosing Square Cloaking Region (2)

Let's review this with the help of an example as shown in Figure 4. At time $t_2((t_2 - t_1) < t)$, Alice sends a second query Q_2 which has the same privacy parameter (r=3, k=5) as Q_1. The locations of the LBS users are shown in Figure 4. The AS checks its database and finds that at t_1, the anonymity set

for Alice was ID_1 = {*A*,*B*,*C*,*D*,*E*}. There are four candidate CCRs, which are shown as the shaded region, the regions enclosed within the dotted lines and the region within the solid line in Figure 4. The corresponding anonymity sets of the CCRs are {*A*,*B*,*C*,*E*,*F*}, {*A*,*B*,*D*,*H*,*I*} and {*A*,*B*,*E*,*F*,*H*}. The region that has its anonymity set containing a largest number of users in common with the users in the anonymity set ID_1, is the shaded region. The anonymity set of the shaded region and the anonymity set of ID_1 have four users in common. The AS will choose the anonymity set of the shaded region, i.e., ID_2 = {*A*,*B*,*C*,*E*,*F*}, as the anonymity set for the current query and send the corresponding 5 × 5 cell region (the shaded region) as the corresponding cloaking region to the LBS provider.

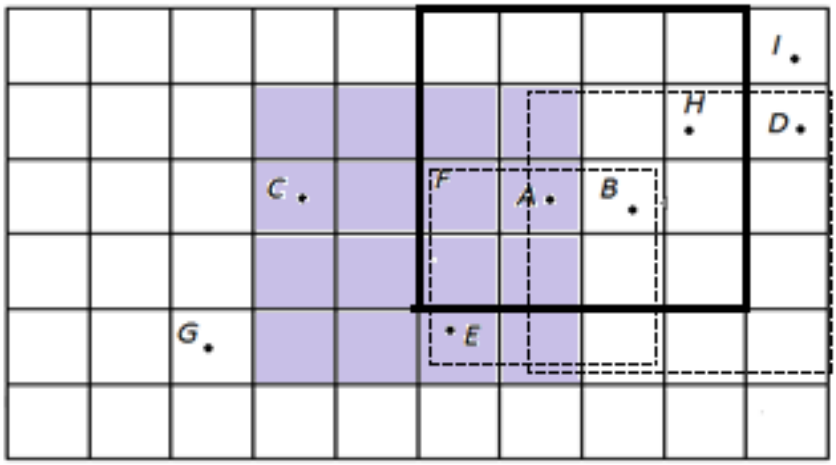


Figure 5: Choosing Square Cloaking Region (3)

This solution also suits the cases in which the user changes her privacy level. Suppose, at time $t_3((t_3 - t_1) < t)$, Alice sends a third query Q_3 asking for 4-anonymity (*k*=4) with minimum area 3 × 3 cells (*r*=3) and the locations of the LBS users is shown in Figure 5. The AS searches the CCR, and gets the corresponding anonymity sets {*A*,*B*,*C*,*G*}, {*A*,*B*,*D*,*H*,*I*}, and {*A*,*B*,*C*,*F*} (see Figure 5). As the anonymity set in her past queries are ID_1= {*A*,*B*,*C*,*D*,*E*} and ID_2 = {*A*,*B*,*C*,*E*,*F*}, the AS will choose the anonymity set {*A*,*B*,*C*,*F*}, and use the respective region as the cloaking region R for that query. The pseudo code of the proposed memorizing algorithm is outlined in Algorithm 1.

Algorithm 1 query-based memorizing algorithm

Require: $Q = \{id, k, l, r, c, x, y\}$
Ensure: cloaking region
  if Rec_id = NULL **then**
    while there is no CCR **do**
      for all *region* with $r \times r$ cells containing the user **do**
        if the region contains no less than k users **then**
          this region is a CCR
        end if
      end for
      if there is no less than 1 CCR **then**
        randomly select one CCR as cloaking region
      else $r = r+1$
      end if
    end while
  else selects anonymity set from every record in Rec_id
    computer similarity value $CCR.sim$ for each CCR
    if CCR_i has maximum similarity value **then**
      CCR_i is the cloaking region
    end if
  end if
  create a record $Q'' = \{id, id', k, l, r, R, c, ID, t\}$

CHAPTER 5

EXPERIMENTAL EVALUATION

In this chapter, the performance of the proposed query-based memorizing algorithm is evaluated by conducting a series of simulation experiments. Let us first outline the methodology that is followed for the simulation experiments and for measuring performance.

5.1 Experimental Methodology and Metrics

The performance of the proposed algorithm is evaluated by comparing it with three other related query privacy preservation algorithms, specifically, (i) the location-based algorithm [14], (ii) the group-based algorithm [10], and (iii) the traditional *k*-anonymity [11]. The comparative evaluation in this thesis is based on the query privacy provided by each algorithm, the QoS implications due to the obfuscation done by each algorithm and the size of the overlapped regions in the obfuscated queries in each of the above algorithms. Below, the metrics used to quantify each of these evaluation criteria are discussed.

As *k*-anonymity is the most widely-used technique for location obfuscation, it is an important baseline for comparison for any location obfuscation scheme. Location-based algorithm [14] and group-based algorithm [10] also have memorizing functions, which are similar to the proposed query-based algorithm. However, location-based algorithm [14] only focuses on location privacy, whereas the group-based algorithm [10] focuses on maintaining the same group of users. As can be seen, the focus of each of these algorithms is different from the proposed approach, but they directly or indirectly affect the query privacy of users, and thus it would be useful to compare them with each other. Moreover, each of the three memorizing algorithm, i.e., the proposed query-based algorithm, the location-based algorithm and the group-based algorithm, implement the memorization function in a unique fashion. Therefore, the comparison of the algorithms can explore the advantages and disadvantages of each algorithm.

5.1.1 Quality of Service

The QoS of the proposed algorithm is analyzed by comparing the size of the cloaking regions generated by each of the four algorithms for a given query. The larger the generated cloaking region is, the lower is the QoS provided by the LBS. In other words, for a given user privacy requirement, an algorithm with good QoS should result in the smallest possible cloaking region satisfying that requirement. The intuition here is that a larger cloaking region will result in coarse-grained results for the querying user and/or more work for the AS (which could result in delays for the querying user), both of which are direct indicators of the QoS provided by the LBS.

5.1.2 Degree of Query Privacy

In the simulation experiments, it is assumed that the attacker is a strong adversary, i.e., the adversary is a perfect locator who has exact location information of every user in the query region. An entropy metric is used to quantify the degree of query privacy. In information theory [18], the concept of entropy is a measure of the uncertainty associated with a random variable. The notion of entropy was first used to evaluate anonymity by Deng et al. [19]. Niu et al [17] uses entropy to evaluate the degree of anonymity upon generating dummy locations. In this thesis, the entropy metric is used to capture the uncertainty of an adversary in correctly linking a query to its issuer, given the corresponding cloaked or obfuscated query and the adversary's prior knowledge about the users. For a particular query, the adversary's uncertainty in correctly linking the query to its issuer is represented by a random variable X, i.e., X takes values representing the issuer of the query with some probability. In other words, the probability distribution over this random variable X is indicative of the adversary's uncertainty. According to [20], given a probability distribution over some random variable X, the entropy $H(X)$ of the attacker can be calculated as shown below in Equation (2).

$$H(X) = -\sum_{i=1}^{N} p_i \times \log_2(p_i) \qquad (2)$$

Here, N is the number of the users in the system, i represents the identity of the user, and $p_i = Pr(X = i)$ represents the probability that the adversary links the query to user i, i.e., i is the query issuer with probability p_i. Thus, a high value of entropy would indicate that the adversary is highly uncertain in deciding which user issues the query, whereas a low value indicates the adversary can deduce with high probability which user issues the query. If $H(X)$ is 0, it means that the adversary is certain which user issued the query. The max value of $H(X)$ is $\log_2(N)$ and it indicates that all the N users have equal probability of issuing the query from the adversary's perspective.

When the adversary collects multiple queries during a short period of time and the content of the query is the same, she can deduce with high probability that these queries are continuous queries from the same user. For a strong adversary, even if the content of the queries cannot be correlated, she can deduce that only the users that exist in every anonymity set of all the queries are more likely to be the query issuer than the other users in the anonymity set. For example, let's assume that at time t_1, the adversary obtains query Q_1', and she notices there are five users *A*, *B*, *C*, *D* and *E* in the cloaking region. At time t_2, the adversary obtains a new query Q_2', and the users in the cloaking regions of Q_2' are *B*, *C*, *D*, *E* and *F*. Finally, at time t_3 the adversary obtains query Q_3' with its anonymity set $\{B,C,D,F,G\}$. Based on the above three queries, the adversary can deduce with high probability that the query issuer for all the three queries must be one user among *B*, *C* or *D*, as only these three users are in the anonymity set of all the queries. The entropy of the third query is much lower and is close to $\log_2 3$, as compared to the optimal value for perfect 5-anonymity which is $\log_2 5$. If the adversary is certain that the third query was issued by one of *B*, *C* or *D*, she can assign a probability of 1/3 to each of these three users to be the query issuer, and a probability of 0 to each of *E* and *F*. As can be seen, the value of the entropy for a query would depend on two factors: first, the largest number of common or intersecting users that exists between the current anonymity set and every (valid) past anonymity set that the

adversary has collected, and second, the number of such past anonymity sets containing the same and largest number of common users. For simplicity, in the evaluations it is assumed that for a certain query, if the number of the users in the current anonymity set is n and if there are m common users in at least one of the valid past anonymity sets (of past queries) obtained by the adversary, then the adversary assigns a probability of $1/m$ to each of these m users (in the current anonymity set) to be the issuer of the current query and a probability of 0 to each of the remaining of n-m users. This current probability assignment is solely based on the number of common users, and can act as a lower-bound on the accuracy of the adversary's guess. With additional information, for example, correlation between query times and query content, this probability can be further improved. As the goal in this thesis is not to improve these attack probabilities, but to comparatively evaluate the various query privacy-preserving mechanisms, the basic probability assignment mechanism (and the related privacy measurement) that is used is sufficient.

5.1.3 Size of the Overlapped Area

The size of the overlapped area (between the cloaking regions of the current query with those of past queries) is also an indicative measure of location privacy leakage. Intuitively, to a weak adversary (adversary who is not a perfect locator), a small overlapped area can help the adversary limit a user's location within a smaller spatial range. In other words, the adversary can obtain more accurate location information from a small overlapped area. Contrary to the above, a larger overlapped area leaks comparatively lesser location information. If the current cloaking region is completely overlapped with the cloaking regions in past queries, then a weak adversary can get no additional information from the queries.

5.2 Experimental Setup

The simulation experiments in this thesis use the tool by T. Brinkoff [21], called Network-based Generator of Moving Objects, to simulate the mobility of users following certain fixed paths such as a road network within a city. The input network is a road map of Oldenburg, Germany, which is roughly 102.96 km^2 in size [22].The datasets for the simulation experiments are generated with varying numbers of users and varying distributions (by using varying user speeds). The four query privacy preserving algorithms discussed above are implemented using Netbeans. Each of these algorithms take as input the user traces generated in the previous step.

The simulations are executed on an Intel 2.30GHz CPU work station with 6GB memory. The traces of the users are generated for duration of 20 seconds. It is assumed that each user inside the query region issues one continuous query every second (for 20 seconds). Besides, the adversary obtains one query (from each user) every two seconds, i.e., the adversary obtains ten queries for each user. It is assumed that the user doesn't change her privacy level when issuing continuous queries. To clearly show the results of the evaluation, a fixed r=3 is used in all the queries. In order to comprehensively evaluate the performance of each algorithm, the average size of the cloaking region, the average entropy and the average size of the overlapped area at each observation time point is calculated. The simulations are repeated for varying numbers of users, privacy levels, and user distributions (or speeds).

5.3 Experimental Result

The results from the simulation experiments are depicted from Figure 6 through 14. Figure 6 depicts the average size of cloaking region for each algorithm under different number of users when the privacy level is fixed. The Y-axis represents a relative value for the average size of the cloaking region. If the value on Y-coordinate is y, it implies that the average size of the region is $y \times y$ cells.

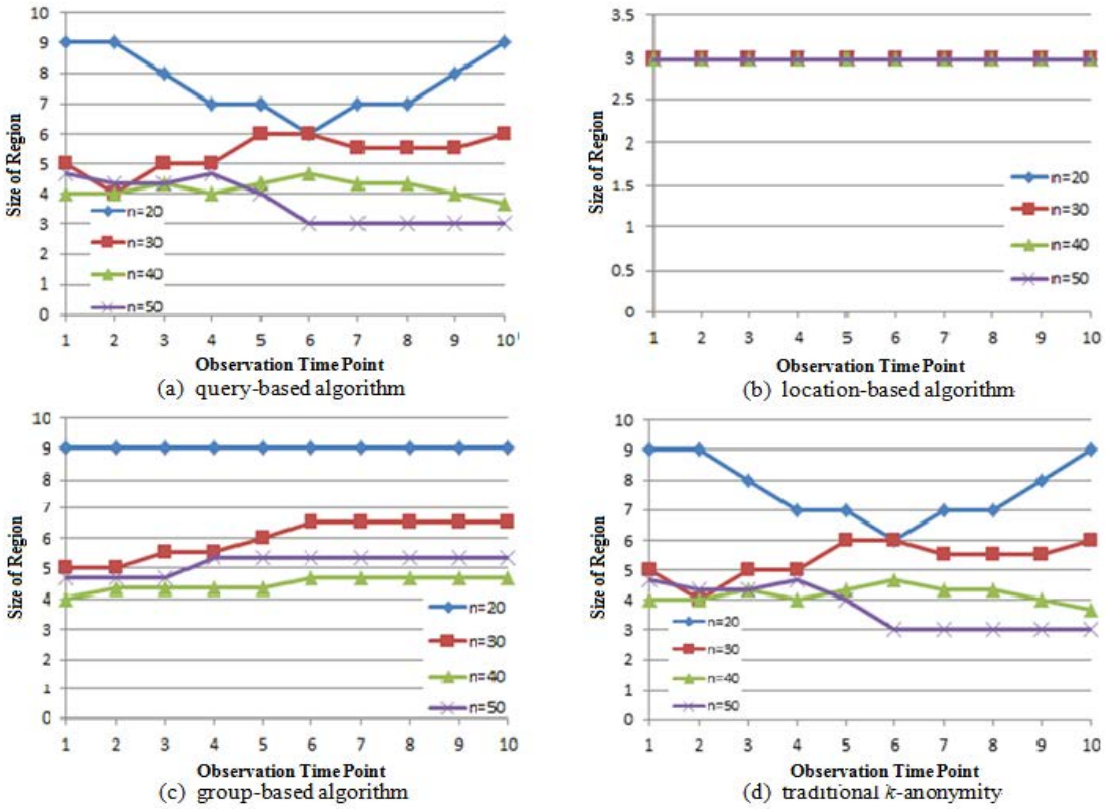


Figure 6: Size of Cloaking Region, k=10

When the number of users increases, the average size of the cloaking region decreases for the query-based algorithm, the group-based algorithm and the traditional k-anonymity approach. However, the size of the cloaking region remains the same for location-based algorithm. This is easy to see because only location-based algorithm does not consider the number of users in the anonymity set. For the other three algorithms, when the number of users increases, the density of users becomes higher and thus, queries require smaller cloaking region to contain enough users. Figure 7 depicts the average size of cloaking region for each algorithm under different privacy level (k) when the number of users is fixed. With increasing privacy level, the average size of the cloaking region increases for query-based algorithm, group-based algorithm and traditional k-anonymity. When the users require larger privacy level, the AS has to generate a larger cloaking region to contain more users in the anonymity set.

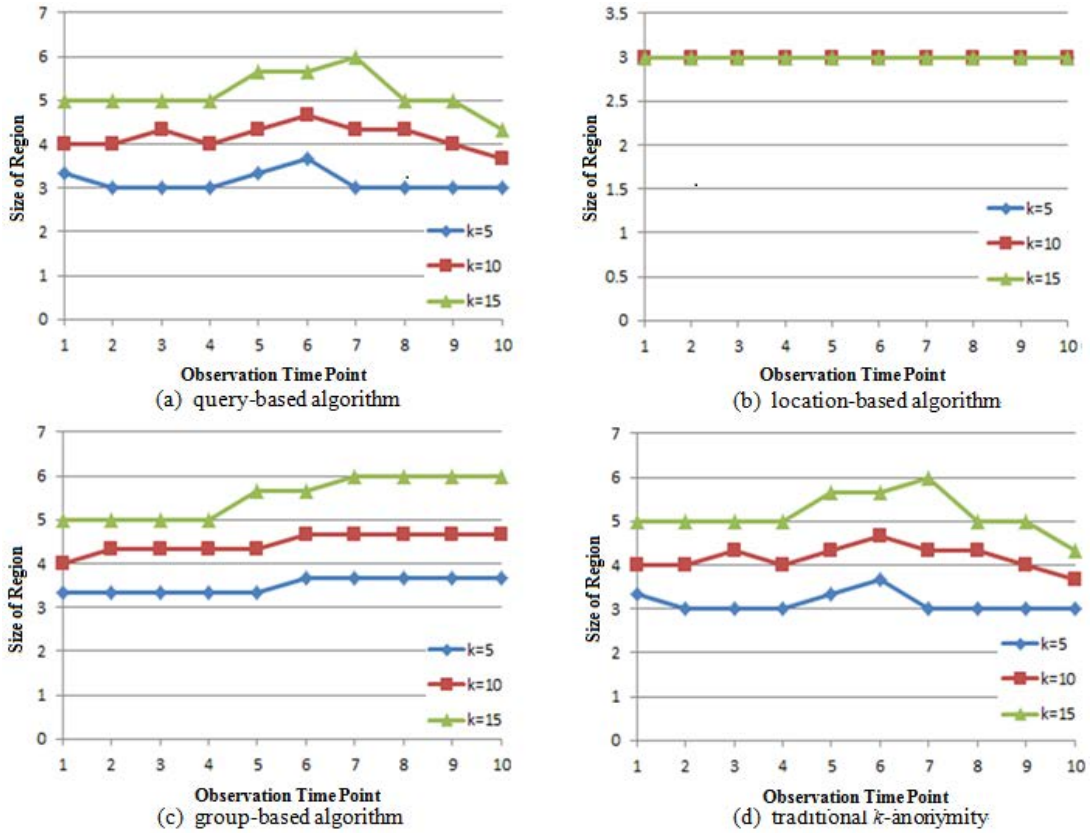


Figure 7: Size of Cloaking Region, n=40

In Figure 8, the QoS of each algorithm is compared under fixed number of users, fixed privacy level and fixed distribution. The experiment results are shown for two distributions in Figure 8 to better evaluate the performance of the algorithms. As shown in Figure 8, group-based algorithm has largest value of the size of cloaking region, and thus has the worst QoS among the four privacy-preserving algorithms. Location-based algorithm has lowest value of size of cloaking region. Query-based algorithm and traditional *k*-anonymity have a moderate value for the average size of region and the values for the two algorithms are almost the same. As the location-based algorithm only uses the size of the cloaking region as the privacy parameter, the size of the region remains the same as long as the user doesn't change her privacy level. For the other three algorithms, the AS has to search larger areas to satisfy the privacy level parameter *k*. As the group-based algorithm retains the same cloaking region as long as there are enough users inside, when user density in the cloaking region becomes higher, group-

based algorithm will not lower the size of the cloaking region. The query-based algorithm and traditional *k*-anonymity approach search for regions that have smallest size of region that satisfy the privacy level parameter *k*. The main difference between the query-based algorithm and traditional *k*-anonymity approach is that the query-based algorithm is constrained by the user's cloaking size requirement whereas the *k*-anonymity approach does not have any such restriction. But in order to have a fair comparison between traditional *k*-anonymity and the other schemes, the same minimum cloaking size requirement for traditional *k*-anonymity is introduced, as other schemes, in the experiments.

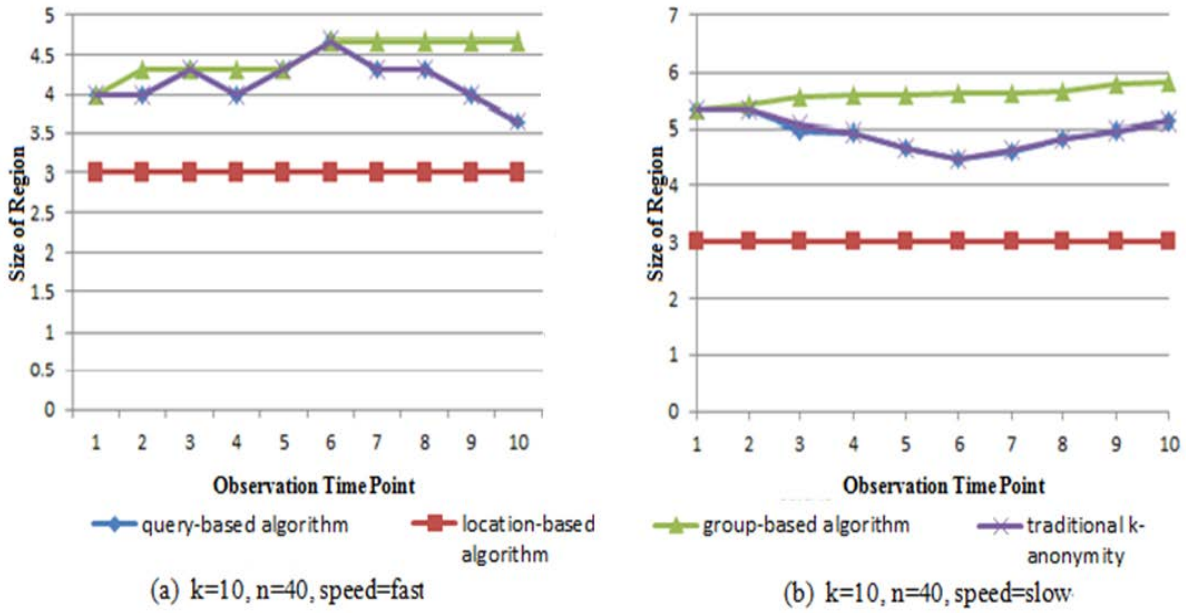


Figure 8: Size of Cloaking Region, *k*=10, *n*=40

Figure 9 depicts the average entropy or adversary uncertainty for each algorithm for an increasing number of users when the privacy level is fixed. The Y-coordinate represents the relative value for entropy. If the value of Y-coordinate is *y*, it means the entropy value is $\log_2 y$. The entropy or adversary uncertainty decreases when the number of observations increases. This is because the adversary can get more information about which user issues the query when she obtains more queries from the users.

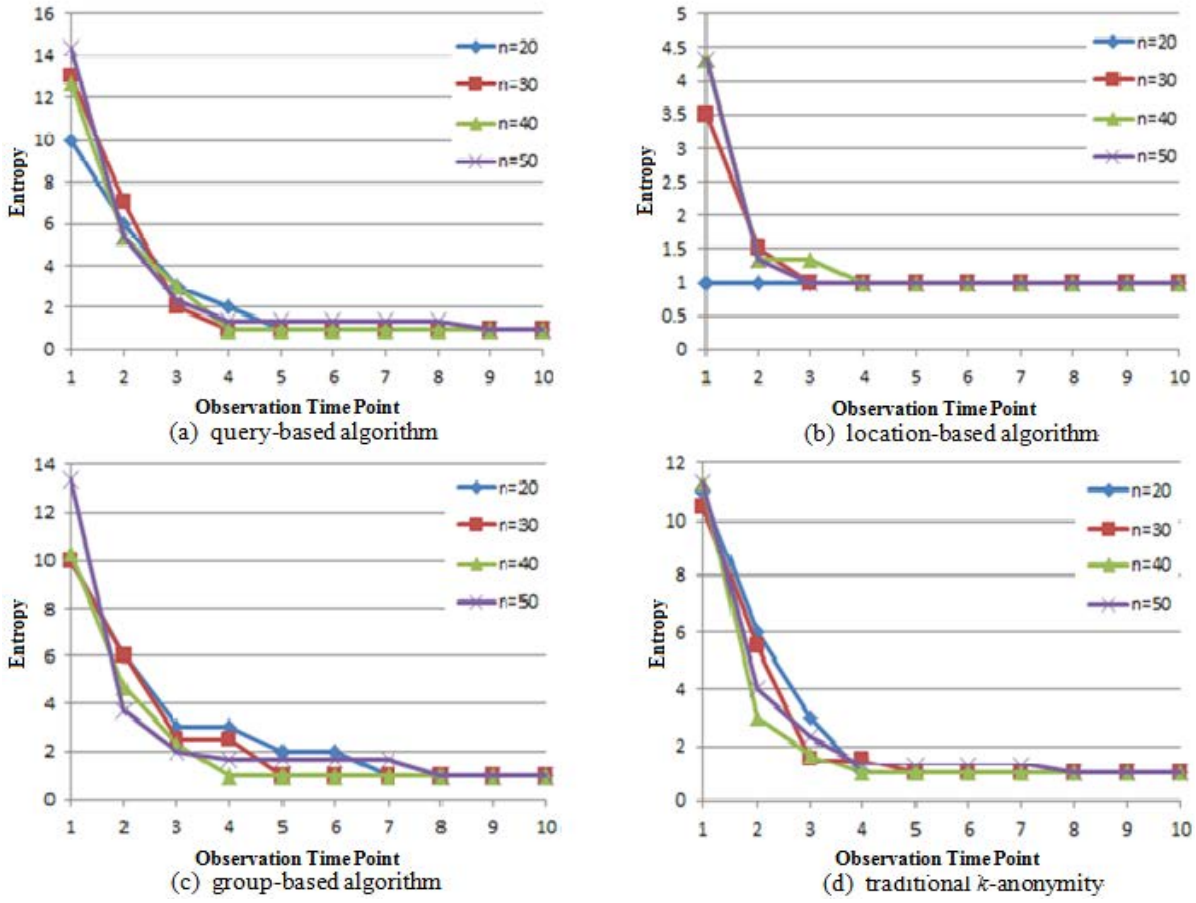


Figure 9: Entropy of the algorithms, k=10

Figure 10 depicts the average entropy for each algorithm with increasing privacy level when the number of users in fixed. The entropy is getting higher with increasing privacy level. This is because more users will be contained inside the cloaking region with increasing privacy level and thus creates more confusion to the adversary.

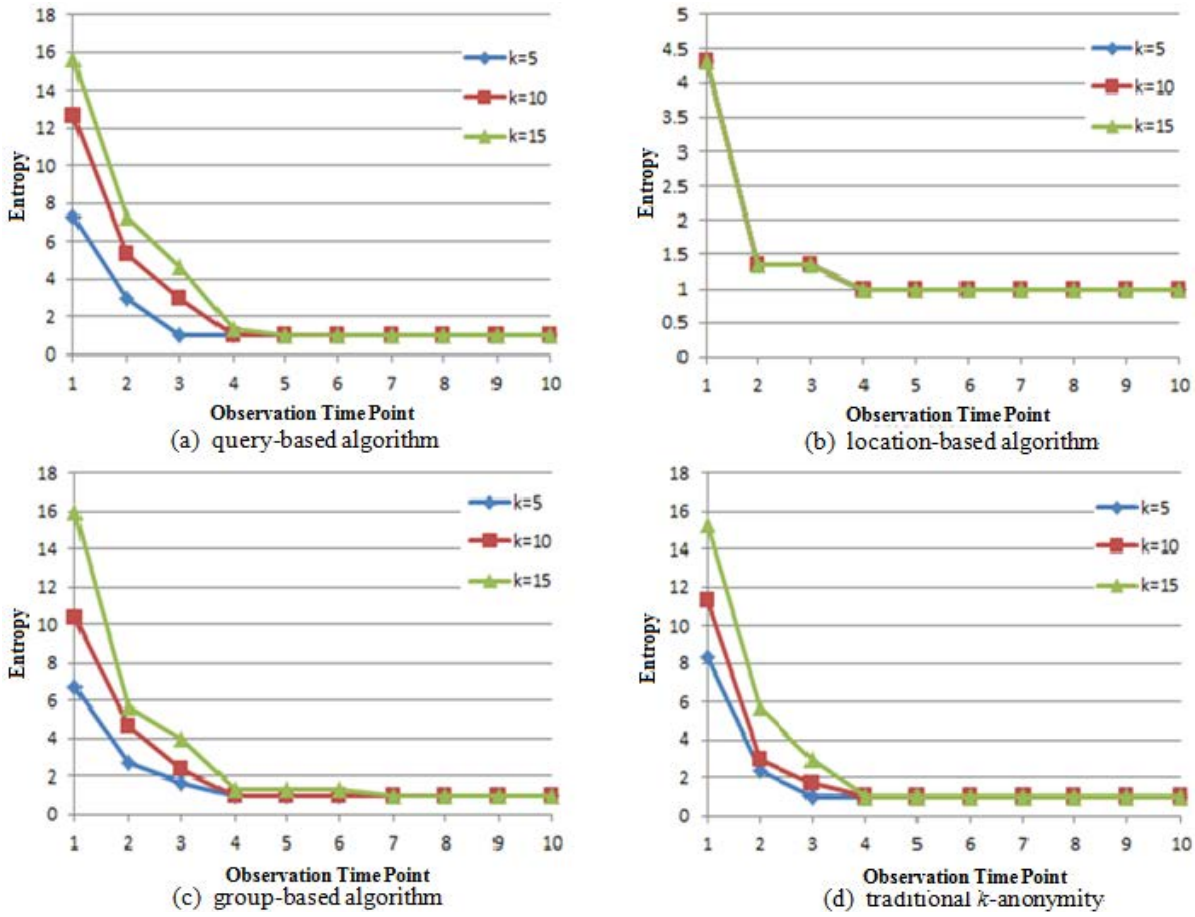
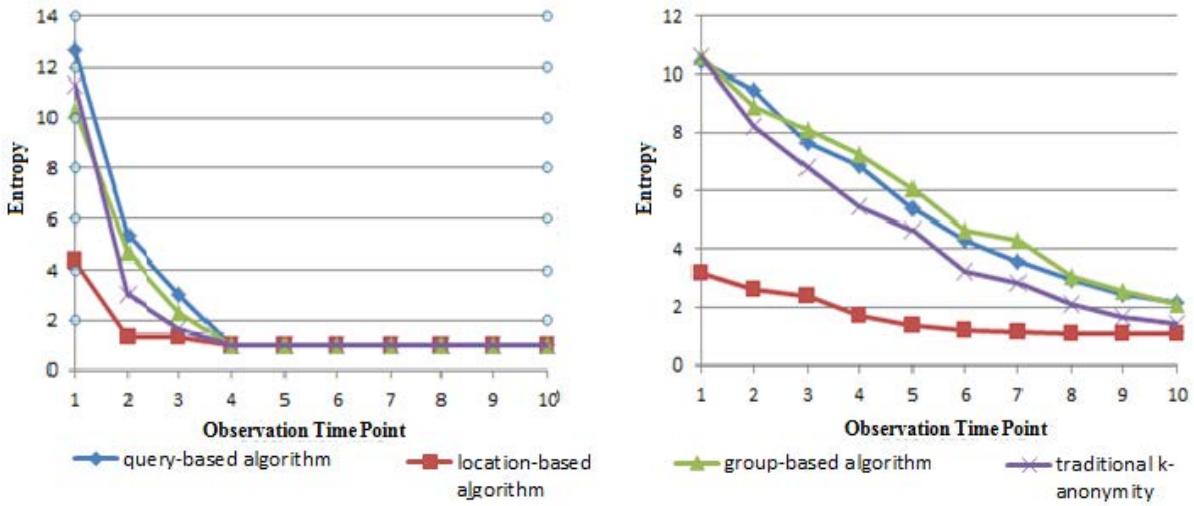


Figure 10: Entropy of the algorithms, n=40

Figure 11 compares the average entropy among the four algorithms when the number of users, privacy level of users and the user distribution are fixed. Two distributions are used to better evaluate the performance of the algorithms. As shown is Figure 11(a), query-based algorithm has largest entropy value. However, in Figure 11(b), group-based algorithm has largest entropy value. This is because group-based algorithm is more suited to conditions when users have low mobility. In the extreme case where users who are inside the cloaking region will not move or disconnect, the entropy value will not decrease for group-based algorithm. When users are highly mobile, group-based algorithm will either frequently regroup or cannot contain the users who are outside the cloaking region but still near the cloaking region. Query-based algorithm can find the nearby users who were in the past anonymity set, no matter whether the user is in the old cloaking region or not.



(a) k=10, n=40, speed=fast (b) k=10, n=40, speed=slow

Figure11: Entropy of the algorithms, *k*=10, *n*=40

Figure 12 depicts the average size of the overlapped area for different number of users for each algorithm when the privacy level (*k*) is fixed. The Y-coordinate represents the number of overlapped cells with the previous query. The size of overlapped area depends on the size of the cloaking region. The largest value for the size of the overlapped area is the size of the cloaking region. If the cloaking region is small, the size of the overlapped area is also small. For query-based algorithm and traditional *k*-anonymity, when the number of users is 20, the size of the overlapped area is largest. This is because when the number of users is small, the size of the cloaking region is relatively large for the two algorithms. As can be observed, the size of the overlapped area in the location-based algorithm and the group-based algorithm has significant fluctuations. This is because the schemes for choosing the cloaking region are similar between these two algorithms and are different from query-based algorithm and traditional *k*-anonymity. Both the location-based algorithm and group-based algorithm maintains the same cloaking region as the previous query as long as the cloaking region in previous query still satisfy the user privacy requirement. In other words, when the size of the overlapped area changes, it implies the current query is generating a new cloaking region. In location-based algorithm, this happens when the user moves out of the previous cloaking region. In group-based algorithm, this happens either

because the user moves out of the previous cloaking region or the number of users inside the previous cloaking region is less than *k*. Therefore, the size of the overlapped area for these two algorithms is more unpredictable.

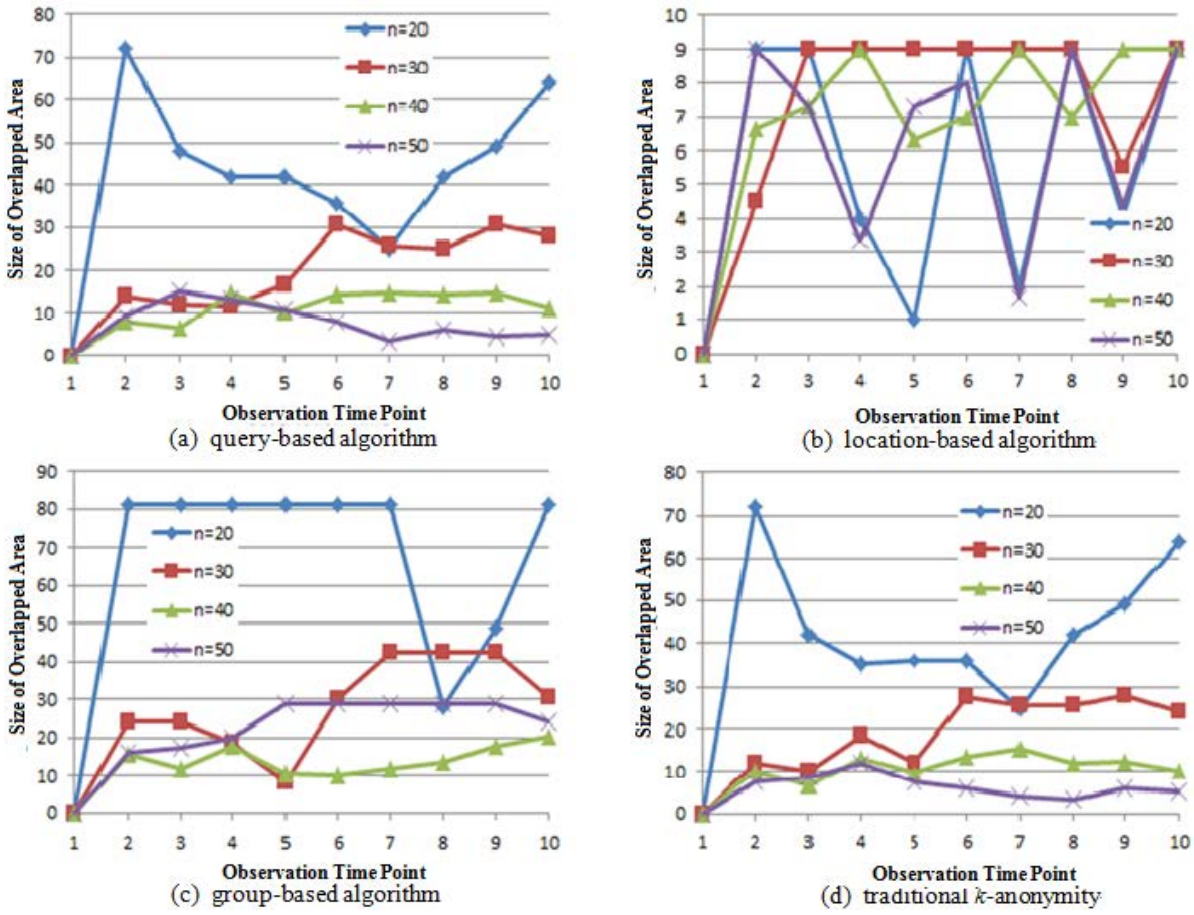


Figure12: Size of the overlapped area, *k*=10

Figure 13 depicts the average size of the overlapped area under varying privacy levels when the number of users is fixed. With higher privacy level, for query-based algorithm, group-based algorithm and traditional *k*-anonymity approach, the size of the region is larger and thus, the size of the overlapped area is larger. Figure 14 compares the average size of the overlapped area for the four algorithms when the number of the users, the privacy level and the user distribution are fixed. Two distributions are used to better evaluate the performance of the algorithms. Overall, the group-based algorithm has the largest value of the size of the overlapped area when the speed of the user is slow, which implies that it best

preserves the location privacy against a weak adversary. However, when the speed of the users is fast, the performance of the group-based algorithm is unsteady. This is because of the nature of its memorization scheme, as explained earlier. The query-based algorithm has a smaller value than the group-based algorithm, but has larger value than traditional *k*-anonymity and the location-based algorithm. This implies that the query-based algorithm performs relatively well in preserving users' location privacy against a weak adversary (or an adversary who is not a perfect locator).

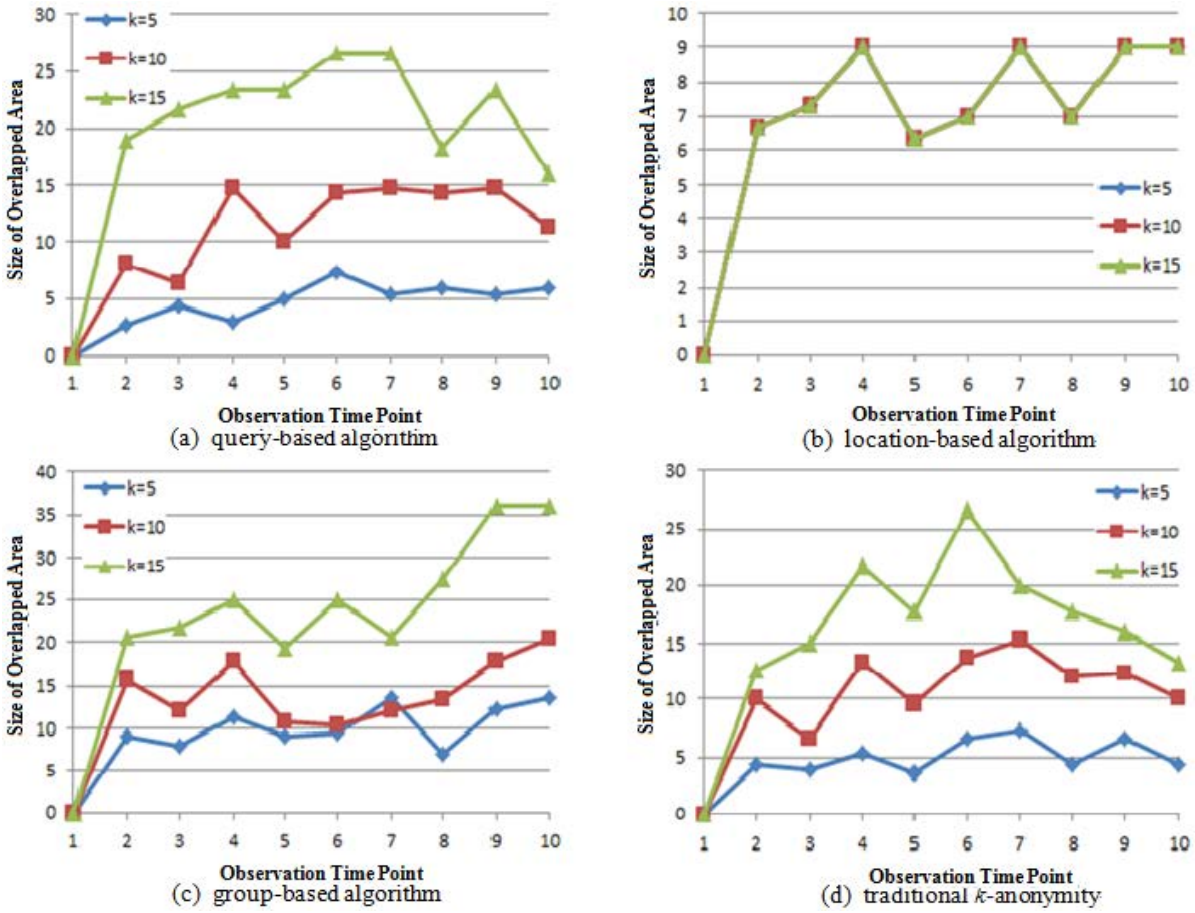
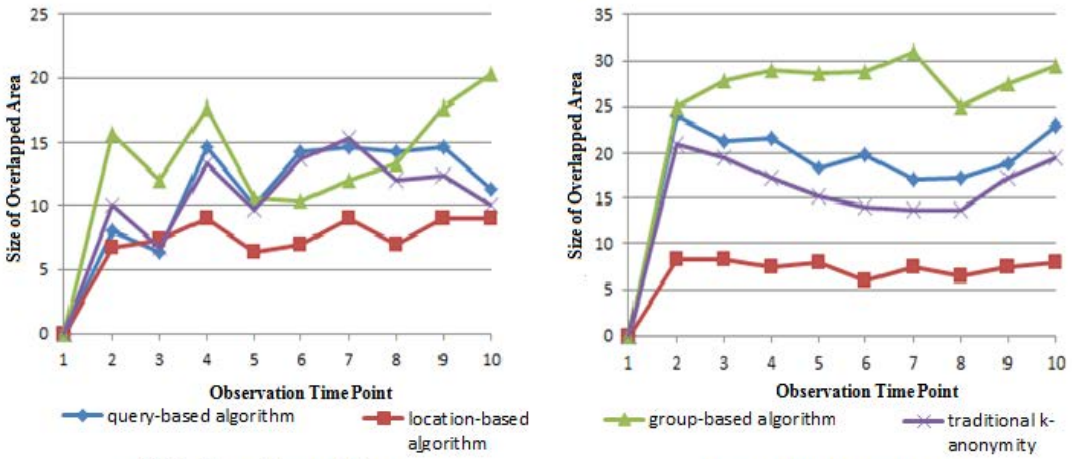


Figure13: Size of the overlapped area, *n*=40



(a) k=10, n=40, speed=fast (b) k=10, n=40, speed=slow

Figure14: Size of the overlapped area, *k*=10, *n*=40

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In this thesis, the problem of query privacy preservation in location-based services is studied. It is argued that when the user issues continuous queries, existing *k*-anonymity schemes have an overlapped area problem which can help the adversary deduce which user in the cloaking region issues the query. It is shown that existing techniques to solve the overlapped area problem does not efficiently protect user's query privacy. In this thesis, a query-based memorizing algorithm that runs on a TTP is proposed to address the overlapped area problem. The algorithm focuses on protecting user's query privacy by memorizing the other users in the past anonymity set. An extensive comparative analysis (using well-known privacy metrics) of the proposed query-based memorizing algorithm with three other popular query privacy preservation techniques is performed. Experimental results showed that location-based algorithm has best scalability while group-based algorithm has worst scalability. Under conditions when users are highly mobile, query-based algorithm provided the best query privacy. When users have low mobility, group-based algorithm has best query privacy. For a weak adversary, group-based algorithm has the best location privacy. Overall, it is shown that the proposed algorithm is scalable and efficient against a strong adversary when the LBS users are highly mobile.

6.2 Future Work

In future research, it would be more convincing to evaluate the algorithms using real location-based service data and develop the algorithm into an application. Besides, the evaluation should consider the cases when the cloaking region is not a square (e.g., a rectangle or round). Another interesting aspect is to study the privacy degree when the adversary is not a strong adversary. As the

price to be a perfect locator is high, it is highly possible that there are more weak adversaries than strong adversaries.

REFERENCES

REFERENCES

[1] K. Zickuhr, “Three-quarters of smartphone owners use location-based service”, http://pewinternet.org/Reports/2012/Location-based-services/Summary-of-findings/Overview.aspx [cited May 11, 2012].

[2] M. Gruteser and D. Grunwald, “Enhancing Location Privacy in Wireless LAN Through Disposable Interface Identifiers: A Quantitative Analysis,” *J. Mobile Networks and Applications - Special issue: Wireless mobile wireless applications and services on WLAN hotspots*, vol. 10, no. 3, pp. 315-325, Jun. 2005.

[3] C.A. Ardagna, M. Cremonini, E. Damiani, S. Vimercati, and P. Samarati, “Location privacy protection through obfuscation-based techniques,” *Proc. 21st annual IFIP WG 11.3 working conf. Data and applications security*, 2007.

[4] M. Duckham and L. Kulik, “Location privacy and location-aware computing,” *Dynamic and Mobile GIS* : *Investigating Change in Space and Time*, J. Drummond, R. Billen, D. Forrest, D. & E. Joao, eds., CRC Press, pp. 34-51, 2006.

[5] R. Shokri, G. Theodorakopoulos, J.-Y.L. Boudec and J.-P. Hubaux, “Quantifying location privacy,” *Proc. 2011 IEEE Symp. Security and Privacy* (*SP*), May.2011.

[6] L. Bindschaedler, M. Jadliwala, I. Bilogrevic, I. Aad, P. Ginzboorg, V. Niemi and J.-P. Hubaux, “Track Me If You Can: On the Effectiveness of Context-based Identifier Changes,” *Proc. 19th Symp. Annual Network & Distributed System Security* (*NDSS ‘12*), Feb. 2012.

[7] R. Dewri, "Local Differential Perturbations: Location Privacy Under Approximate Knowledge Attackers,” *IEEE Trans. Mobile Computing*, vol. 12, no. 12, pp. 2360-2372, Dec. 2012.

[8] B. Gedik and L. Liu, “Location Privacy in Mobile Systems: A Personalized Anonymization Model,” *Proc. IEEE 25th int’l conf. Distributed Computing Systems*, 2005.

[9] C. Zhang and Y. Huang, “Cloaking Locations for Anonymous Location Based Services,” *J. GeoInformatica*, vol. 13, no. 2, pp. 159-182, Jun. 2009.

[10] C. Chow and M. Mokbel, “Enabling Private Continuous Queries For Revealed User Locations,” *Proc. 10th Int’l Conf. Advances in Spatial and Temporal Databases* (*SSTD ‘07*), Jul, 2007.

[11] M. Gruteser and D. Grunwald, “Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking,” *Proc. 1st Int’l Conf. Mobile Systems, Applications and Services* (*MobiSys ‘03*), May, 2003.

[12] L. Wang and S. Wu, "Protecting location privacy through Identity Diffusion," Int'l Conf. Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT '09), 2009.

[13] S. Amini, J. Lindqvist, J. Hong, J. Lin, E. Toch and N. Sadeh, "Cache: caching location-enhanced content to improve user privacy," *Proc. 9th Int'l conf. Mobile systems, applications, and services* (*MobiSys '11*), 2011.

[14] Q. Truong, A. Truong and T.K. Dang, "Privacy preserving through a memorizing algorithm in location-based Services," *Proc. 7th Int'l Conf. Advances in Mobile Computing and Multimedia* (*MOMM '09*), 2009.

[15] M. Yiu, C. Jensen, X. Huang, and H. Lu, "SpaceTwist: Managing the Trade-Offs Among Location Privacy, Query Performance, and Query Accuracy in Mobile Services," *Proc. 24th int'l conf. Data Engineering* (*ICDE*), Apr. 2008.

[16] M. Yiu, C. Jensen, X. Huang, and H. Lu, "PAD: Privacy-Area Aware, Dummy-Based Location Privacy in Mobile Services," *Proc. 7th ACM int'l Workshop. Data Engineering for Wireless and Mobile Access* (*MobiDE '08*), June. 2008.

[17] B. Niu, Q. Li, X. Zhu, G. Cao and H. Li, "Achieving k-anonymity in privacy-aware location-based services," *Proc. 33rd Annual IEEE int'l conf. Computer Communications*(*INFOCOM '14*), Apr. 2014.

[18] S. Ihara, "Information theory for continuous systems," World Scientific Pub Co Inc, 1993.

[19] Y. Deng, J. Pang and P. Wu, " Measuring anonymity with relative entropy," *Proc. 4th int'l conf. Formal Aspects in Security and Trust* (*FAST '06*), August. 2007.

[20] C. Díaz, S. Seys, J. Claessens and B. Preneel, "Towards measuring anonymity," *Proc. 2nd int'l Conf. Privacy Enhancing Technologies* (*PET '02*), Apr. 2003.

[21] T. Brinkoff, "A framework for generating network-based moving objects," *J. Geoinformatica*, vol. 6, no. 2, pp. 153-180, Jun. 2002.

[22] Wikipedia, "Oldenburg," http://en.wikipedia.org/wiki/Oldenburg [cited Nov 28, 2013].