Towards Privacy-Aware Location-Based Database Servers*

Mohamed F. Mokbel

Department of Computer Science and Engineering
University of Minnesota, Twin Cities
mokbel@cs.umn.edu

Abstract

The wide spread of location-based services results in a strong market for location-detection *devices (e.g., GPS-like devices, RFIDs, handheld devices, and cellular phones). Examples of location-based services include location-aware emergency service, location-based advertisement, live traffic reports, and location-based store finder. However,* location-detection *devices pose a major privacy threat on its users where it transmits private information (i.e., the location) to the server who may be untrustworthy. The existing model of location-based applications trades service with privacy where if a user wants to keep her private location information, she has to turn off her location-detection device, i.e., unsubscribe from the service. This paper tackles this model in a way that protects the user privacy while keeping the functionality of location-based services. The main idea is to employ a trusted third party, the* Location Anonymizer*, that expands the user location into a spatial region such that: (1) The exact user location can lie anywhere in the spatial region, and (2) There are k other users within the expanded spatial region so that each user is k-anonymous. The location-based database server is equipped with additional functionalities that support spatio-temporal queries based on the spatial region received from the location anonymizer rather than the exact point location received from the user.*

1. Introduction

The explosive growth of location-detection devices (e.g., cellular phones, GPS-like devices, RFIDs, and handheld devices)[1] along with wireless communications and mobile databases results in realizing location-based services as commercial products (e.g., see [38, 39, 49, 50]) and research prototypes (e.g., see [21, 43, 53]). Location-based services are applications that deliver specific information to their users *based on* their current location. Examples of such applications include finding the nearest restaurant, delivering weather and/or traffic information, and sending coupons to nearest customers.

The flood of information that come out from location-detection devices along with the large number of mobile users that utilize location-based services call for migrating the functionality of location-based services with the recent technologies in database management systems. The basic idea is that user requests to location-based services can be modeled as spatio-temporal queries that can be efficiently executed over large numbers of mobile users through database management modules, e.g., data indexing, query processing, and query optimization [27, 28, 32, 34, 41]. Spatio-temporal queries deal with objects and/or queries that change their locations and/or sizes over time. Efficiency in spatio-temporal queries and location-based services is crucial where any delay in the query response may result in an obsolete answer as the status of data and queries may change. In addition, scalability in terms of large number of continuous queries (i.e., large number of concurrent users to location-based services) can be efficiently realized through database management techniques (e.g., see [42, 54])

Although location-based services and location-detection devices promise safety and convenience, they threaten the privacy and security of their customers [52]. Location-detection devices continuously send the location information of their users to

* This work is supported in part by the Grants-in-Aid of Research, Artistry, and Scholarship, University of Minnesota.

1 According to the Cellular Telecommunication and Internet Association, CTIA, there are about 200 Million wireless customers in the Unites States [11].

the location-based database server. With untrustworthy servers, such model provides several privacy threats if the private location information is being misused by the server or hacked by a third untrusted party. For example, an employer may check on her employee behavior by knowing the places she visit, the personal medical records can be revealed by knowing which clinic each person is visiting, or someone can track the locations of his ex-friends. In fact, in many cases, GPS devices have been used in stalking personal locations (e.g., see [16, 51]).

The traditional approach of *pseudonymity* (i.e., using a fake identity) [45] may not be applicable to location-based applications where a location of a person can directly lead to her true identity. For example, asking about the nearest Pizza restaurant to my home using a fake identity will reveal my true identity (a resident of the home). The main reason of such privacy threats is that location-based applications rely mainly on an implicit assumption that the users agree to trade their location privacy by the service. If a user wants to keep her private location information, she has to turn-off her location-aware device and (temporarily) unsubscribe from the service. As a result of such privacy threats, recent studies about user concerns in location-based services reported that the location privacy is the most important issue [8, 23, 30]. Thus, there is a real concern that many of the users would stop using location-based services in order to protect their privacy [1].

In this paper, we aim to provide research directions towards achieving privacy in location-based database servers in which users of location-based applications can still obtain a high quality service without sacrificing their own privacy. The main idea is to employ a third trusted party, termed the *Location Anonymizer*, that: (1) Receives the exact point location from mobile users, (2) Blurs the location point into a *cloaked* spatial region according to certain constraints provided by the users, and (3) Sends the *cloaked* spatial region to the location-based database server. Then, the location-based database server is equipped with special modules that modify its functionality to work on the *cloaked* spatial region rather than an exact point location. As a result of having a blurred location information, the location-based database server may not be able to provide a high quality service to its users. Users would have the ability to tune a set of parameters to achieve a personal trade-off between the amount of information they would like to reveal about their locations and the quality of service that they obtain from the location-based database server.

Although there is a lot of research in both the areas of database privacy (e.g., see [4, 5, 35]) and location privacy (e.g., see [9, 13, 31, 45]), the migration of both areas is not yet explored. For example, database privacy research mainly focus on the privacy of the existing data through k-anonymity paradigm [29, 37, 40]. Yet, highly updated data and the privacy of the users who issue the query are not explored. On the other hand, location privacy research mainly focus on disturbing the user location data before sending it to the server to avoid tracking the user behavior [17, 18, 19]. Yet, issues like scalability in terms of the number of users, user queries for non-tracking applications, and privacy constraints are not explored by these approaches. In this paper, we aim to combine the recent technologies in both location privacy and database privacy to enable location-based services while keeping the privacy of both the stored data at the database server and the users who issue the queries to the location-based database server.

The rest of the paper is organized as follows. Section 2 highlights related work in the areas of location privacy, database privacy, and privacy models for data communications. Our proposed architecture of the *privacy-aware location-based database server* is outlined in Section 3. As the proposed architecture have three main components, the following three sections discuss each component in detail where Section 4 presents the requirements of mobile users to keep their privacy. The *Location anonymizer* is described in details in Section 5. Query types and *privacy-aware* query processing in the location-based database server are discussed in Section 6. Finally, Section 7 concludes the paper.

2. Related Work

In this section, we highlight the related work to privacy-aware location-based database servers in three different areas, location privacy, data privacy, and privacy models.

2.1. Location Privacy

Recent attempts for providing location privacy in location-based services (e.g., see [9, 13, 17, 18, 19, 24, 25, 31]) and other location-aware applications (e.g., context-aware computing [46] and sensor networks [20]) focus only on hiding a single user location information. Although such techniques would be valuable in small scale location-based services, the practicality in real location-based database servers is doubtful where these techniques lack two main issues: (1) Scalability. In a typical location-based service, there are large numbers of concurrent users. Trying to protect the loca-

tion privacy of each user individually would not scale up to the large number of users. (2) Query processing. By protecting user location information from being disclosed to the location-based database server, processing traditional queries become challenging where new techniques need to be presented to provide efficient query processing while not being able to know the exact user locations.

In general, four different approaches have been explored: (1) False dummies [31]. For every location update, a user would send n different locations to the server with only one of them is true while the rest are dummies. Thus, the server cannot know which one of these locations is the actual one. (2) Landmark objects [25]. Rather than sending the exact location, the user would refer to the location of a certain landmark or a significant object. (3) Location perturbation [13, 17, 18]. The main idea is to *blur* the exact location information to a spatial region using either spatio-temporal cloaking [17, 18] or location obfuscation [13]. The blurred spatial region can be based either on the k-anonymity concept [47, 48] (i.e., the region should contain k users) or on a graph model that represents a road network [13]. (4) Avoid location tracking [9, 19]. While the previous three approaches focus only on hiding a certain instance of the user location, this approach aims to avoid tracking the user behavior. Thus, the user would have the ability to hide her location as long as she is navigating in a sensitive area.

2.2. Database Privacy

Recent literatures in database privacy are mainly concerned about protecting the privacy of existing stored data (e.g., see [4, 5, 6, 35]). The main objective is to provide access to the stored data without disclosing privacy sensitive information. Towards this goal, several techniques have been proposed to maintain the privacy of each data record as k-anonymous [29, 37, 36, 40], i.e., a data record should not be distinguishable among other k records [47, 48]. However, a direct extension to any of these approaches to the case of location-based database servers is not trivial due to the following three main reasons: (1) Such techniques aim to preserve the privacy of the stored data. In our model, we aim not to store the data at all. Instead, we store perturbed version of the data. Thus, data privacy is managed before storing the data. In this case, the risk of privacy threats can be minimized. (2) These approaches aim to protect the data not the queries. In the *privacy-aware* location-based database server, we aim to protect the person who issues the query. For example, a person who wants to ask about her nearest ATM machine needs to protect her location while the ATM location information does not have to be protected. (3) The presented k-anonymity models are provided and guaranteed only for a certain snapshot of the database. This could be valid for traditional databases where data updates are not so frequent. In location-based environments, data and queries are continuously updated with high rates. Such dynamic behavior calls for new techniques to provide k-anonymity for highly updated data.

2.3. Privacy Models

During the last decade, several architectures have been explored to provide secure data transformation from the client to the server machines. Secure-multiparty communication [12, 22] organizes the communication among m parties such that each party can have the knowledge only of a certain function but not the actual data for other parties. Yet, the computation overhead of such scheme prevents its direct application to database problems. Thus, the minimal information sharing [3] paradigm is proposed where it uses encryption/decryption techniques to perform join and intersection operations. Yet, the computation cost and the inability to serve other queries make such paradigm not suitable for real time applications. The untrusted third party [15] paradigm has been proposed in the context of peer-to-peer systems. The main idea is to employ a third party that executes queries by collecting secure information from multiple data sources (i.e., peers). The most commonly used model is the trusted third party [2, 26] paradigm. The main idea is to employ a third party that is trusted by the users and acts as a middle layer between the user and the database server. Such paradigm is commonly used for location privacy techniques discussed in Section 2.1 (e.g., [9, 17, 18]). Having an intermediate trusted party between the user and the service provider is commercially applied in other fields. For example, the Anonymizer [7] is responsible for private web surfing to internet users while the PayPal [44] system is a trusted third party where a user can buy products without giving her credit card information to the provider. Yet, none of these commercial products deal with the location information. In this paper, we use the trusted third party model as an interface between large number of continuously moving users and the location-based database server.

3. Architecture

Our ultimate goal is to protect the users' privacy while maintaining the functionality of location-based

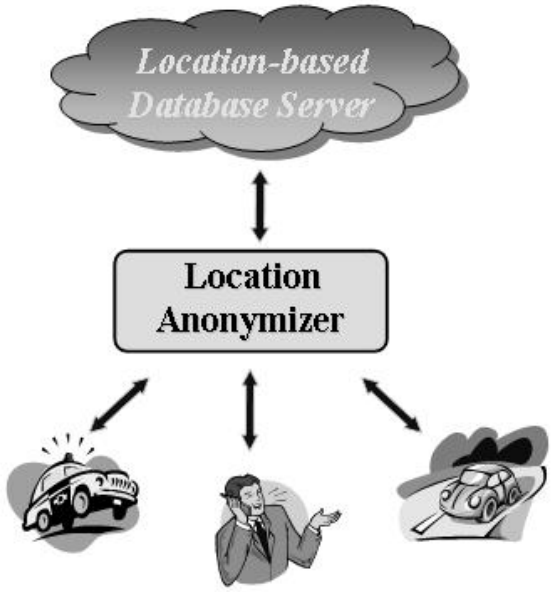


Figure 1. The Location Anonymizer.

database servers. The main idea is to build a *Location Anonymizer* that acts as a trusted third party between the mobile users and the location-based database servers. Figure 1 depicts the proposed architecture. Mobile users register with the *location anonymizer* via a *privacy profile* that outlines the privacy requirements of each user. While continuously moving, mobile users send their exact location updates to the *location anonymizer*. The *location anonymizer* cloaks the exact point locations received from the users to a spatial region that satisfies the mobile user *privacy profile*. Finally, the *location anonymizer* sends the *cloaked* spatial region to the location-based database server. Notice that the *location-anonymizer* does not need to store the exact location information. Instead, the *cloaked* spatial region can be computed through some metadata or statistics maintained through the course of execution.

Spatio-temporal queries processed at the location-based database server may come either from mobile users or from an untrusted third party. Queries that come from mobile users should pass by the *location anonymizer* to hide the query identity and blurs the location of the user who issued the query. In this case, the location-based database server will provide the query answer based on the blurred location received from the *location anonymizer*. Spatio-temporal queries that are issued from an untrusted party do not need to pass through the *location anonymizer*, instead they are directly submitted to the location-based database server. The database serve will answer such queries based on the stored blurred location information of all mobile users.

The following three sections discuss in details the three main entities in Figure 1, namely, the mobile users, the location anonymizer, and the location-based database server.

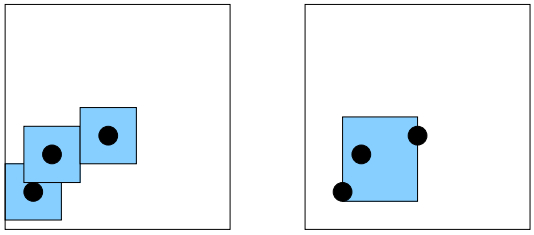
4. Privacy Profiles of Mobile Users

Mobile users that are willing to share their private location information can register directly with the location-based database server. On the other hand, mobile users who want to protect their private information should register with the *location anonymizer*. A mobile user can be in one of three modes, *passive* mode, *active* mode, or *query* mode. A *passive* user does not share her information neither with the location anonymizer nor with the location-based database server. *Active* users continuously send their locations to the location anonymizer. A certain user is considered in the *query* mode whenever she seeks a certain location-based service via a spatio-temporal query (e.g., asking about the nearest fast food restaurant). Our focus in this paper is on users that are either in *active* or *query* modes. Upon registration with the *location anonymizer*, mobile users should indicate their initial *privacy profile*. A user *privacy profile* basically contains the following:

- The level of anonymity (k). A user should specify her convenient level of privacy by introducing the anonymity parameter k. A k-anonymous user is not distinguishable among other k users. Larger k indicates more restrictive privacy, which indicates less quality of service.
- Minimum area (A_{min}). A_{min} represents the minimum area requirement of the *cloaked* spatial region. Determining the minimum area is helpful in dense areas or within large buildings. For example, a user in a stadium with $k = 100$ may have a very small *cloaked area*. Similarly, a user in a shopping mall may want to guarantee that her *cloaked* area is beyond the mall boundary. Larger A_{min} indicates more restrictive privacy.
- Maximum area (A_{max}). A_{max} represents the maximum area requirement for the *cloaked* spatial area. Such parameter is particulary useful in sparse areas. For example, a user in rural way with $k = 100$ may find that her *cloaked* spatial region is so large, which yields a low-quality service. Thus, such user may like to place an upper bound for the area of the *cloaked* spatial region. Lower A_{max} indicates more restrictive privacy.
- Temporal constraints. A mobile user may specify multiple instances of the above parameters for dif-

Time	*K*	*Min. Area*	*Max. Area*
8:00 AM –	**1**	—	—
5:00 PM –	**100**	**1 mile**	**3 miles**
10:00 PM –	**1000**	**5 miles**	—

Figure 2. Example of a privacy profile.



(a) Naive cloaking **(b) MBR cloaking**

Figure 3. Data-dependent location anonymizer.

ferent time intervals. For example, a certain user may have a less conservative *privacy profile* during the week days while the same user may have so conservative *privacy profile* at nights and weekends.

Figure 2 gives an example of a typical *privacy profile* for a mobile user. The first entry of the *privacy profile* indicats that the user accepts to reveal her location information ($k = 1$) at the daytime (8:00 AM to 5:00 PM). The second entry indicates that the same user would like to have some reasonable privacy-service trade-offs between 5:00 PM and 10:00 PM. During this time, the user may be in some places where she does not want to reveal her exact location, yet she wants to be able to have reasonable quality when using location-based services. Finally, after 10:00 PM, the same user has very restrictive privacy constraints ($k = 1000$, $A_{min} = 5$) that indicate the unwilling of the user to reveal any information about her location. It is important to note that mobile users have the ability to change their *privacy profiles* at any time.

5. The Location Anonymizer

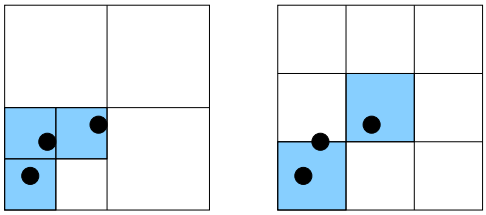
Initially, mobile users register with the *location anonymizer* through their *privacy profiles*. Similar to the proposed model in [14], the *location anonymizer* may charge the mobile users based on their required protection level. Upon receiving the exact location information from the mobile user m, the *location anonymizer* checks for the privacy profile of m and *cloaks* the point location into a *cloaked* spatial region R that mostly satisfy the user requirements. A *privacy profile* may contain some contradicting requirements. For example, a user may specify a very small minimum and maximum area with large k. Thus, the job of the *location anonymizer* is a best effort where it tries to satisfy the user requirements as much as possible. In general, we identify the following three main requirements that need to be satisfied in the *cloaked* spatial region:

1. The spatial *cloaked* region should contain at least k users so that each user becomes k-anonymous, i.e., not distinguishable among k users, where k is a user-specified parameter. Having k achieved, the *location anonymizer* tries to locate a spatial region with an area A, such that $A_{min} < A < A_{max}$.
2. The spatial *cloaked* region should not reveal any information about the exact user location. In other words, an adversary should not be able to do reverse engineering to know the exact user location from the spatial *cloaked* area.
3. The *cloaking* algorithm should be computationally efficient to cope with the continuous movement of mobile users and real time requirements of spatio-temporal queries.

The first requirement is the minimum requirement that any *location anonymizer* should provide, however the second and third requirements are mainly concerned with the efficiency and quality of the *location anonymizer*. Existing techniques for location *cloaking* mainly focus only on the first requirement. In general, constructing the spatial cloaked area from the exact location information can be either *data-dependent* or *space-dependent*. Such classification is similar to multi-dimensional indexing structures that are classified to data-partitioning and space-partitioning data structures. Figures 3 and 4 give examples of *data-dependent* and *space-dependent location anonymizer*, respectively. The exact point locations are plotted as black circles while the spatial *cloaked* areas are plotted as gray rectangles.

5.1. Data-dependent Location Anonymizer

Figure 3 gives two examples of *data-dependent* spatial cloaking. The main idea is to construct the *cloaked* area based on the knowledge of the mobile users' locations. In general, data-dependent techniques are sub-



(a) Quad–tree (b) Regular grid

Figure 4. Space-dependent location anonymizer.

ject to reverse engineering where some information can be revealed about the exact point location. For example, consider the naive case in Figure 3a where the *location anonymizer* expands the point location *equally* in all areas till the user *privacy profile* is satisfied. Although such data-dependent *location anonymizer* may satisfy the user requirements in terms of k, A_{min}, and A_{max}, an adversary can easily deduce the exact location as being the middle point of the *cloaked* spatial region. A more smart *data-dependent* cloaking technique is given in Figure 3b and has been utilized in [17]. The main idea is to construct the spatial *cloaked* area of several point locations as their minimum bounding rectangle (MBR). Although, there is no direct reverse engineering that can reveal the exact point location form the MBR, yet the MBR encounters some information leakage. Having the MBR indicates that there is at least one data point on each edge. If k is small, then an adversary would guess that the exact point location is on the MBR boundary. Such information leakage degrades the performance of the *location anonymizer*.

5.2. Space-dependent Location Anonymizer

Figure 4 gives two examples of *space-dependent* spatial cloaking. The main idea is to construct the *cloaked* area based on partitioning the space. Since *space-dependent* cloaking does not rely on the exact point location, it is almost impossible to reveal any information about the exact location information. *Space-dependent* cloaking can partition the space based on the existing data [18] as in Figure 4a or based on a fixed space-partitioning as in Figure 4b. The main idea in Figure 4a is that the *location anonymizer* starts from the whole space and checks if it satisfies the mobile user requirements in terms of k, A_{min}, and A_{max}. If this is the case, the *location anonymizer* will keep partitioning the space into four quadrants till it encounters a quadrant that does not satisfy the user requirements. In this case, the latest quadrant that has satisfied the user requirements is returned as the spatial *cloaked* area. Figure 4b gives another example of *space-dependent* cloaking where the whole space is partitioned into fixed grid cells. For each mobile user m, the *location anonymizer* locates the gird cell g in which m lies in. Then, the *location anonymizer* checks if g satisfies the user *privacy profile*. If this is the case, g is returned as the spatial *cloaked* area. Otherwise, g is merged with other adjacent grid cells till the *location anonymizer* satisfies the user *privacy profile*. Also, it may be the case that g is already satisfying the user requirements yet with a very relaxed area. In other words, the *cloaked* area can be reduced more while still satisfying the user requirements. Thus, g can be partitioned again into other fixed grids. Keeping fixed multi-level grids would be an optimization for Figure 4b.

5.3. Efficiency of the Location Anonymizer

In general, *space-dependent* cloaking is preferred over the *data-dependent* cloaking in terms of preserving the privacy of mobile users and preventing any information leakage from the spatial *cloaked* area. However, both the presented *data-dependent* and *space-dependent* approaches address the privacy requirements for each individual mobile user. To exploit scalable techniques for both the *location anonymizer* and the *location-based database server*, two main approaches need to be investigated: (1) Incremental evaluation. The main idea is to avoid continuous computation of the *cloaked* region as users continuously update their locations. Instead, computing a *cloaked* region at time t should benefit from the computation of the *cloaked* region of the same user at time $t-1$. Similarly, processing the continuous queries at the location-based server should be done incrementally. (2) Shared execution. Since both the server and the anonymizer do similar functionalities for different users, many of the required procedures can be *shared* among different users. Our plan is to identify such *shared* procedures and execute them only once for all users.

6. Location-based Database Server

In our architecture, we extend traditional location-based database servers that deal with accurate location information and queries to deal with *cloaked* spatial regions and inaccurate queries. An example of such accurate queries is *"Find all the gas stations that can be nearest to any point in the given region R"* where R is the *cloaked* spatial region computed from the *location anonymizer*.

6.1. Data and Query Types

The *privacy-aware* location-based database server keeps track of two types of data; *public* data and *private* data.

- *Public data.* This type of data includes stationary objects such as hospitals, restaurants, gas stations, and coffee shops or moving objects such as police cars and on-site workers. Such persons and facilities do not want to hide their location information.
- *Private data.* This type of data mainly contains personal information of mobile users with a *privacy profile* of non-zero k or A_{min}. Such persons adjust their *privacy profile* to satisfy their privacy requirements.

Based on the data stored in the *privacy-aware* location-based database server, two novel types of spatio-temporal queries need to be supported:

- *Private queries over public data.* An example of such query is that a person is asking (i.e., private query) about her nearest gas station (i.e., public data). In this case, the location-based database server does not have the exact location information of the person who issued the query while the exact location information of the target objects (i.e., gas stations) are known.
- *Public query over private data.* An example of such query is that an administrator wants to query (i.e., public query) about the number of mobile users (i.e., private data) in a certain area. In this case, the *privacy-aware* location-based database server knows the exact query information, yet it does not know the exact locations of mobile users.

Notice that traditional location-based database servers (e.g., [21, 43, 53]) can support only *public queries over public data* where the complete knowledge of location information of both data and queries are available. At the other end of the spectrum, *private queries over private data* can be reduced to any of the above two query types.

6.2. Privacy-aware Query Processing

The *privacy-aware* location-based database server is equipped with a non-traditional *privacy-aware* query processor that deals with the two novel query types.

6.2.1. Private query over public data A naive way to deal with private queries that request information about public objects is to ask the location-based database server to send *all* target objects to the mobile user. Then, the mobile user evaluates her query locally. Although such naive way will completely preserve the user privacy, it is not practical due to the cost of transmitting large sizes of target information and to the limited computation and storage capability of mobile users. As a trade-off between the data transmission and the user privacy, the mobile user should only request a subset of *candidate* target objects that, with a high probability, contain the actual answer. Then, internally, the mobile user will go through the *candidate list* to find the actual answer.

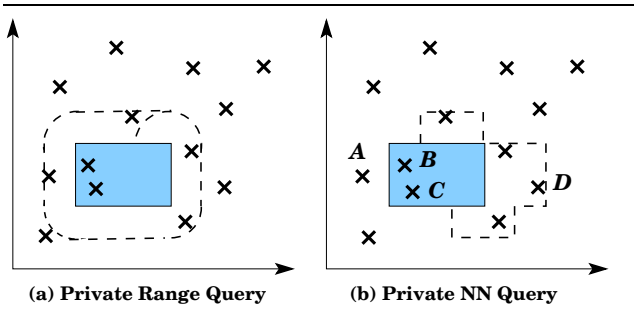
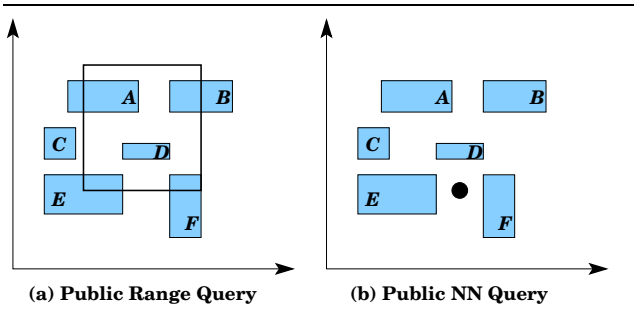


Figure 5. Private queries over public data.

Figure 5 gives two examples of private range queries and private nearest-neighbor queries. Target objects are plotted all over the space. The shaded area in Figure 5 represents the spatial *cloaked* region produced from the *location anonymizer*. In Figure 5a, a mobile user in the shaded area is asking about all target objects within three miles of her location. Since the *privacy-aware* location-based database server has no idea about the exact location of the mobile user within the shaded area, it should return all target objects that can be within three miles from *any* point in the shaded area. As a result, the five objects within the rounded dashed rectangle are returned to the mobile user. Then, the mobile user evaluates her query internally on the five returned objects. Notice that any object that lies in the rounded dashed rectangle is candidate to be an answer. We have the rectangle rounded to be exact, however, on a real implementation, the rounded rectangle will be approximated by its minimum bounding rectangle.

Figure 5b gives an example where a mobile user in the shaded area is asking about her nearest target object. The *privacy-aware* query processor should manage to compute the set of target objects that can be nearest to any point in the shaded area. As in Figure 5b, six objects are returned to the user. Two of them are already in the shaded area. The other four are within the dashed polygon. Notice that the returned objects



(a) Public Range Query (b) Public NN Query

Figure 6. Public queries over private data.

may not be the nearest to the rectangle. For example, target A is nearest to the shaded area than target D, yet the *privacy-aware* query processor would return only D as a candidate answer. The main reason is that it is guaranteed that targets B and C would be nearest to any point in the shaded area than target A. Thus, target A is eliminated from the candidate set. On the other side, there could be a possibility that the exact mobile user location is on the right boundary of the shaded area. In this case, target D could be nearest to some point in the shaded area than targets C and D. Similar cases apply for the rest of returned objects.

6.2.2. Public query over private data A naive way to deal with public queries that request information about private objects is to consider that the private data are non-zero size objects, i.e., rectangular objects. Then, all the queries are converted to ask about rectangular objects that are returned by the *location anonymizer* rather than exact point locations. Although such technique would keep the data privacy, it may end up with very inaccurate query results. For example, consider the case of a user with restrictive privacy profile and hence a large spatial *cloaked* area. Such user may end up *mistakenly* contributing in many query results, thus degrading the accuracy of such queries.

Figure 6 gives two examples of public queries over private data. There are six private data objects, A to D, that are represented by their spatial *cloaked* regions (the shaded area). Figure 6a seeks the count of mobile users inside a certain rectangular area. Dealing with each object as a non-zero size object would return five as the query answer, which is totally inaccurate answer. Thus, it is better to deal with each object individually. For example, we are sure 100% that object D will contribute to the query answer while object C does not contribute to the answer. However, for the other objects, we are confident only by 75%, 50%, 20%, and 25% that objects A, B, E, and F, respectively, will contribute to the query answer. These ratios are computed based on the ratio of the overlapped area of each object with the query area to the area of the spatial *cloaked* region. The underlying assumption behind these computations is that the *location anonymizer* generates the *cloaked* area so that the exact location information could be anywhere within this area. As a result, the answer of the query can be represented in various formats: (1) As an absolute value by adding the probabilities of each object, i.e., $1+0.75+0.5+0.2+0.25 = 2.7$, (2) As an interval, i.e., the number of objects is in the interval [1,5], or (3) As a probability density function, i.e., in the form of (i, p_i) where i is an integer number in the interval [1,5] and p_i is the probability that the query answer is i. Similar answer representation has been employed in the context of probabilistic queries over imprecise queries (e.g., see [10, 33]).

Figure 6b gives an example of a public object (e.g., a gas station) that asks about its nearest mobile user to send her a personalized e-coupon. The gas station is represented as a black dot. The *privacy-aware* query processor would eliminate the mobile users A, B, and C from the answer where any location of object D within its *cloaked* region would be more near to the gas station than any location of these objects. On the other hand, there is a possibility that objects E and F are more near to the query point than D. Thus, the answer of the query can be represented in any of the following formats: (1) As a set of potential nearest users, i.e., $\{E, D, F\}$, (2) As only one object with the highest probability to be a nearest user, i.e., D, or (2) As a probability density function, i.e., $\{(E, p_E), (D, p_D), (F, p_F)\}$ where p_E, p_D, and p_F are the probabilities that objects E, D, and F are the nearest users to the query point, respectively.

7. Conclusion

In this paper, we have identified the privacy threats imposed by the use of recent technologies in location-detection devices in obtaining location-based services. As these privacy threats may hinder the use of location-based services, in this paper we have addressed challenges and research directions towards achieving a *privacy-aware location-based database server*. The main objective of the *privacy-aware location-based database server* is to protect the private sensitive location information of mobile users while being able to deliver the functionality of traditional location-based database servers, yet with less quality. *Privacy-aware location-based database servers* tend to balance between the amount of information released from the mobile user to the quality of service delivered to the same user. Towards this goal, we have proposed to employ a third

trusted party, namely, the *Location Anonymizer* that receives the exact location information from the mobile user, turns the exact information into a *cloaked* spatial region according to a certain *privacy profile* that the mobile user provides, and finally sends the *cloaked* region to the location-based database server. The functionality of the location-based database server is modified to deal with *cloaked* spatial regions rather than exact point locations. Two general approaches have been discussed for the location anonymizer, namely *data-dependent* and *space-dependent* location anonymizer. In addition, two novel query types are presented and discussed, namely, *private queries over public data* and *public queries over private data*.

References

[1] L. Ackerman, J. Kempf, and T. Miki. Wireless location privacy: A report on law and policy in the united states, the europrean union, and japan. Technical Report DCL-TR2003-001, DoCoMo Commuinication Laboratories, USA, 2003.

[2] G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, N. Mishra, R. Motwani, U. Srivastava, D. Thomas, J. Widom, and Y. Xu. Vision Paper: Enabling Privacy for the Paranoids. In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 708–719, 2004.

[3] R. Agrawal, A. V. Evfimievski, and R. Srikant. Information Sharing Across Private Databases. In *Proceedings of the ACM International Conference on Management of Data, SIGMOD*, pages 86–97, 2003.

[4] R. Agrawal, R. J. B. Jr., C. Faloutsos, J. Kiernan, R. Rantzau, and R. Srikant. Auditing Compliance with a Hippocratic Database. In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 516–527, 2004.

[5] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic Databases. In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 143–154, 2002.

[6] R. Agrawal and R. Srikant. Privacy-Preserving Data Mining. In *Proceedings of the ACM International Conference on Management of Data, SIGMOD*, pages 439–450, 2000.

[7] Anonymous surfing. http://www.anonymizer.com.

[8] L. Barkhuus and A. K. Dey. Location-Based Services for Mobile Telephony: a Study of Users' Privacy Concerns. In *Proceeding of the IFIP Conference on Human-Computer Interaction, INTERACT*, pages 709–712, 2003.

[9] A. R. Beresford and F. Stajano. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[10] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Evaluating Probabilistic Queries over Imprecise Data. In *Proceedings of the ACM International Conference on Management of Data, SIGMOD*, pages 551–562, Madison, WI, 2003.

[11] The cellular telecommunication and internet association, ctia. http://www.wow-com.com/.

[12] W. Du and M. J. Atallah. Secure Multi-Party Computation Problems and their Applications: A Review and Open Problems. In *Proceeding of the New Security Paradigms Workshop*, 2001.

[13] M. Duckham and L. Kulik. A Formal Model of Obfuscation and Negotiation for Location Privacy. In *Pervasive*, pages 152–170, 2005.

[14] S. Duri, J. Elliott, M. Gruteser, X. Liu, P. Moskowitz, R. Perez, M. Singh, and J.-M. Tang. Data Protection and Data Sharing in Telematics. *Mobile Networks and Applications*, 9(6):693–701, 2004.

[15] F. Emekci, D. Agrawal, A. E. Abbadi, and A. Gulbeden. Privacy Preserving Query Processing using Third Parties. In *Proceedings of the International Conference on Data Engineering, ICDE*, 2006.

[16] Foxs News. Man Accused of Stalking Ex-Girlfriend With GPS. http://www.foxnews.com/story/0,2933,131487,00.html. September, 04, 2004.

[17] B. Gedik and L. Liu. A Customizable k-Anonymity Model for Protecting Location Privacy. In *Proceeding of the International Conference on Distributed Computing Systems, ICDCS*, 2005.

[18] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services, MobiSys*, pages 163–168, 2003.

[19] M. Gruteser and X. Liu. Protecting Privacy in Continuous Location-Tracking Applications. *IEEE Security and Privacy*, 2(2):28–34, Mar. 2004.

[20] M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald. Privacy-Aware Location Sensor Networks. In *Proceedings of the Workshop on Hot Topics in Operating Systems, HotOS*, pages 163–168, 2003.

[21] R. H. Güting, V. T. de Almeida, D. Ansorge, T. B. Z. Ding, T. Höse, F. Hoffmann, M. Spiekermann, and U. Telle. SECONDO: An Extensible DBMS Platform for Research Prototyping and Teaching. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 1115–1116, 2005.

[22] L. M. Haas, R. J. Miller, B. Niswonger, M. T. Roth, P. M. Schwarz, and E. L. Wimmers. Transforming Heterogeneous Data with Database Middleware: Beyond Integration. *IEEE Data Engineering Bulletin*, 22(1):31–36, Jan. 1999.

[23] U. Hengartner and P. Steenkiste. Access Control to Information in Pervasive Computing Environments. In *Proceeding of the Workshop on Hot Topics in Operating Systems*, pages 157–162, 2003.

[24] U. Hengartner and P. Steenkiste. Protecting Access to People Location Information. In *Proceeding of the International Conference on Security in Pervasive Computing, SPC*, pages 25–38, 2003.

[25] J. I. Hong and J. A. Landay. An Architecture for Privacy-Sensitive Ubiquitous Computing. In *In Proceedings of The International Conference on Mobile Systems, Applications, and Services, MobiSys*, pages 177–189, 2004.

[26] N. Jefferies, C. J. Mitchell, and M. Walker. A Proposed Architecture for Trusted Third Party Services. In *Proceedings of the International Conference on Cryptography: Policy and Algorithms*, pages 98–104, London, UK, 1995.

[27] C. S. Jensen. Database Aspects of Location-Based Services. In *Location-Based Services*, pages 115–148. Morgan Kaufmann, 2004.

[28] C. S. Jensen, A. Friis-Christensen, T. B. Pedersen, D. Pfoser, S. Saltenis, and N. Tryfona. Location-based Services: A Database Perspective. In *Proceedings of the 8th Scandinavian Research Conference on Geographical Information Science, ScanGIS*, pages 59–68, 2001.

[29] R. J. B. Jr. and R. Agrawal. Data Privacy through Optimal k-Anonymization. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 217–228, 2005.

[30] E. Kaasinen. User needs for location-aware mobile services. *Personal and Ubiquitous Computing*, 7(1):70–79, 2003.

[31] H. Kido, Y. Yanagisawa, and T. Satoh. An Anonymous Communication Technique using Dummies for Location-based Services. In *Proceedings of IEEE International Conference on Pervasive Services, ICPS*, pages 88–97, 2005.

[32] J. Krumm and S. Shafer. Data Store Issues for Location-based Service. *IEEE Data Engineering Bulletin*, 28(3):35–42, Sept. 2005.

[33] I. Lazaridis and S. Mehrotra. Approximate Selection Queries over Imprecise Data. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 140–152, Boston, MA, 2004.

[34] D. L. Lee, M. Zhu, and H. Hu. When Location-Based Services Meet Databases. *Mobile Information Systems*, 1(2):81–90, 2005.

[35] K. LeFevre, R. Agrawal, V. Ercegovac, R. Ramakrishnan, Y. Xu, and D. J. DeWitt. Limiting Disclosure in Hippocratic Databases. In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 108–119, 2004.

[36] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian Multidimensional K-Anonymity. In *Proceedings of the International Conference on Data Engineering, ICDE*, 2006.

[37] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient Full-Domain K-Anonymity. In *Proceedings of the ACM International Conference on Management of Data, SIGMOD*, pages 49–60, 2005.

[38] Locatrix Communications. http://www.locatrix.com/.

[39] MapInfo. http://www.mapinfo.com/.

[40] A. Meyerson and R. Williams. On the Complexity of Optimal K-Anonymity. In *Proceedings of the ACM Symposium on Principles of Database Systems, PODS*, pages 223–228, 2004.

[41] M. F. Mokbel, W. G. Aref, S. E. Hambrusch, and S. Prabhakar. Towards Scalable Location-aware Services: Requirements and Research Issues. In *Proceedings of the ACM Symposium on Advances in Geographic Information Systems, ACM GIS*, pages 110–117, New Orleans, LA, Nov. 2003.

[42] M. F. Mokbel, X. Xiong, and W. G. Aref. SINA: Scalable Incremental Processing of Continuous Queries in Spatio-temporal Databases. In *Proceedings of the ACM International Conference on Management of Data, SIGMOD*, pages 443–454, Paris, France, June 2004.

[43] M. F. Mokbel, X. Xiong, W. G. Aref, S. Hambrusch, S. Prabhakar, and M. Hammad. PLACE: A Query Processor for Handling Real-time Spatio-temporal Data Streams (Demo). In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 1377–1380, Toronto, Canada, Aug. 2004.

[44] Paypal. http://www.paypal.com/.

[45] A. Pfitzmann and M. Kohntopp. Anonymity, Unobservability, and Pseudonymity - A Proposal for Terminology. In *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability*, pages 1–9, 2000.

[46] A. Smailagic and D. Kogan. Location Sensing and Privacy in a Context-aware Computing Environment. *IEEE Wireless Communication*, 9(5):10–17, 2002.

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

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

[49] TargusInfo. http://www.targusinfo.com/.

[50] Telostar. http://www.telostar.com/.

[51] USAToday. Authorities: GPS system used to stalk woman. http://www.usatoday.com/tech/news/2002-12-30-gps-stalker_x.htm. December, 30, 2002.

[52] J. Warrior, E. McHenry, and K. McGee. They Know Where You Are . *IEEE Spectrum*, 40(7):20–25, 2003.

[53] O. Wolfson, H. Cao, H. Lin, G. Trajcevski, F. Zhang, and N. Rishe. Management of Dynamic Location Information in DOMINO. In *Proceedings of the International Conference on Extending Database Technology, EDBT*, pages 769–771, 2002.

[54] X. Xiong, M. F. Mokbel, W. G. Aref, S. Hambrusch, and S. Prabhakar. Scalable Spatio-temporal Continuous Query Processing for Location-aware Services. In *Proceedings of the International Conference on Scientific and Statistical Database Management, SSDBM*, pages 317–328, Santorini Island, Greece, June 2004.