# A taxonomy of approaches to preserve location privacy in location-based services

Ali Khoshgozaran* and Cyrus Shahabi

Information Laboratory (InfoLab),
Department of Computer Science,
University of Southern California,
Los Angeles, CA 90089, USA
E-mail: jafkhosh@usc.edu
E-mail: shahabi@usc.edu
*Corresponding author

**Abstract:** The ubiquity of smartphones and other location-aware hand-held devices has resulted in a dramatic increase in popularity of location-based services (LBS) tailored to users' locations. The comfort of LBS comes with a privacy cost. Various distressing privacy violations caused by sharing sensitive location information with potentially malicious services have highlighted the importance of location privacy research aiming to protect users' privacy while interacting with LBS. This paper presents a taxonomy of different approaches proposed to enable location privacy in LBS and elaborates on the strengths and weaknesses of each class of approaches.

**Keywords:** location privacy; location-based services; LBS; spatial databases; spatial query processing; taxonomy.

**Reference** to this paper should be made as follows: Khoshgozaran, A. and Shahabi, C. (2010) 'A taxonomy of approaches to preserve location privacy in location-based services', *Int. J. Computational Science and Engineering*, Vol. 5, No. 2, pp.86–96.

**Biographical notes:** Ali Khoshgozaran recently completed his PhD in Computer Science at the University of Southern California, where he worked in the Information Laboratory (Infolab) on privacy in location-based services and geospatial information management. His research interests include geospatial databases, location-based services and location privacy.

Cyrus Shahabi is a Professor and the Director of the Information Laboratory at the Computer Science Department and also the Director of the NSF's Integrated Media Systems Center (IMSC) at the University of Southern California. His research interests include multimedia and geospatial information management. He received his PhD in Computer Science from USC.

## 1 Introduction

The availability of low cost smartphones equipped with various positioning technologies has dramatically increased the ubiquity of location-based services (LBS). The market for navigation and search devices and subscriptions and services will nearly triple in revenue in 2008, to $1.3 billion from $485 million in 2007, and will reach $8 billion in 2011 (New York Times, 2008). Users now benefit from various services offered by LBS tailored to their current location information. They can use their cell phones to search for nearby points of interest (POI) such as restaurants and hotels and track their friends and family.

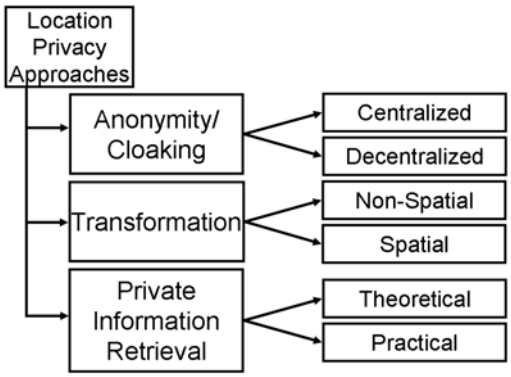
The benefits of LBS come at the cost of sharing private identity and location information of users with potentially untrusted entities offering such services. The explosive growth of such *location servers* has made it impossible for users to verify the authenticity of all location servers they interact with. Sharing such sensitive information with untrusted servers has recently resulted in various distressing violations of users' privacy. Several breaches of subscriber's privacy by misusing their location information have been reported (e.g., Washington Post, 2005; CNN, 2007). In particular, Bettini et al. (2005) illustrate how the location information in the history of user-requests can act as a quasi-identifier to identify users.

To protect against various privacy threats while using LBS, several studies have proposed different approaches to protect the privacy of users while interacting with potentially untrusted location servers, hence coining the term *location privacy*. In this paper, we present a taxonomy of approaches proposed for the location privacy problem. As illustrated in Figure 1, these approaches are based on *anonymity/cloaking*, *transformation* and *private information retrieval* (PIR) techniques. We study each group in more details and briefly show how each approach supports sample spatial queries used in LBS.

We stress that this work by no means is a full treatment of all the location privacy literature. We review the taxonomy of the current state of the art in location privacy

and further discuss some of the dominant studies in each category.

**Figure 1** Location privacy taxonomy




Depending on the services provided, several architectures are currently practised in the context of LBS. We consider a generic case that covers the majority of the location-based applications. With this setting, *users (clients)* are interested in certain information about their location and carry location-aware devices such as cell phones whose location can be found with an embedded GPS device or by triangulating user's location-based on nearby cell phone towers. Users subscribe to a *location server* who usually offers a set of services customised based on user's provided location data. In order to serve a user, the location server, or in short *the server*, possesses a set of various geospatial data sources that might include maps, aerial imagery, POI, etc. for a large region. Moreover, the server also has enough resources to process large loads of queries received from several users. The location server is not trusted even though it may not be malicious. This is because several breaches of private user information occur inadvertently by an accidental security breach, insecure client/server communication or an existing bug in server applications.

Typical user interactions with LBS are often expressed as range or K-nearest neighbour (KNN) queries. With range (KNN) queries, users are interested in objects that fall within a certain region (the K closest objects to a query location) specified by the user. In the above queries, the static objects represent POI and the query points represent user's locations. We first study the case where users query the location server about the static objects and later in Section 5.2, we discuss the challenges in relaxing this assumption and extending the model to allow users query other users as well as POIs.

The remainder of this paper is organised as follows. Section 2 studies the class of cloaking and anonymity approaches proposed for location privacy. Sections 3 and 4 present various techniques aiming to protect users' privacy based on transformation techniques and PIR, respectively. In Section 5, we discuss some of the privacy threats associated with querying static objects in LBS and proceed to discuss the challenges associated with the more general case of querying static POIs as well as dynamic locations of other users. Finally, Section 6 concludes the paper.

## 2 Anonymity/cloaking

The main idea behind the class of anonymity/cloaking approaches is to blur a user's exact location in a larger *cloaked* region and to make her indistinguishable among the set of other (real or dummy) users located in the cloaked region. Depending on where the cloaking is taking place, these approaches can be grouped into two classes of centralised and decentralised cloaking.

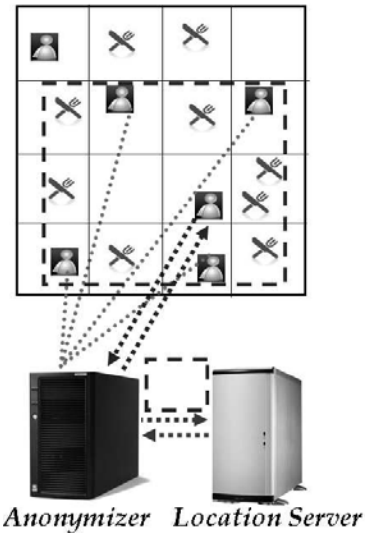
### *2.1 Centralised cloaking*

Many existing approaches in location cloaking rely on the existence of a trusted *location anonymiser* which protects a user's private location and identity information from an untrusted location server (e.g., Mokbel et al., 2006; Gruteser and Grunwald, 2003; Gedik and Liu, 2005a, 2005b; Du et al., 2007). The main idea in centralised cloaking is to put an anonymiser between the users and the location server to prevent the server from learning users' precise location information and identities.

#### *2.1.1 Architecture and query processing*

Figure 2 illustrates the system architecture for centralised cloaking framework. The framework consists of a location anonymiser and an untrusted location server which hosts a privacy-aware query processor. In order to enable location privacy, the anonymiser maintains the current locations of all subscribed users. Instead of sending the location query to the LBS, the user contacts the anonymiser, which generates a cloaked region enclosing the user as well as $k - 1$ other users in her vicinity.

**Figure 2** Centralised cloaking




As processing anonymised nearest neighbour queries is more complex than anonymised range queries, we focus on how nearest neighbour queries are processed with cloaking-based approaches. Among the central cloaking approaches that consider the query processing of the cloaked region, the end-to-end query resolution process can

be divided in the following two phases. First, upon receiving a query, the anonymiser employs a cloaking algorithm to generate a cloaked region. While different algorithms are proposed for cloaking a user's location, the common objective is to blur a user's location in an area of size at least $A_{min}$ and/or among a set of at least $k - 1$ other users. Depending on the approach, these parameters can be specified by each user independently, or are chosen as system parameters. During the second phase, the privacy-aware location server, which is modified to process a cloaked region query, generates a *candidate list* which is guaranteed to include the nearest neighbour of any point inside the cloaked region. This list is then transferred to the client side for further refinement to obtain the final result set.

### 2.1.2 Strengths and weaknesses

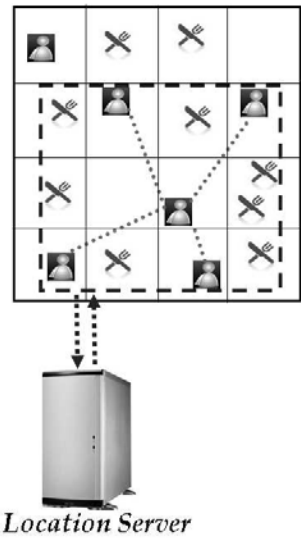
One of the key benefits of centralised cloaking approaches is the fact that the sophisticated anonymiser can perform various complex operations to enable an untrusted server to process complex queries. In other words, range, KNN and other types of spatial queries can be easily supported as long as the privacy-aware server is instructed to perform such queries on a cloaked region. Furthermore, as we elaborate in Section 5.2, allowing an anonymiser to continuously monitor the exact location of all users greatly reduces the challenges associated with supporting queries over dynamic objects (e.g., a nearby friend).

However, as outlined by several recent studies (e.g., Ghinita et al., 2008; Khoshgozaran and Shahabi, 2007; Yiu et al., 2008; Khoshgozaran et al., 2008), centralised cloaking and anonymity approaches have several drawbacks. The first drawback of such approaches originates from the fact that by design they require an anonymiser, as sophisticated as the location server itself, to act as a proxy between users and the server per query. Aside from creating a single point of failure/attack and bottleneck, this approach has another important drawback. In many scenarios *cloaking* users' location information in a larger region or among $k - 1$ other users does not protect user's location information. This is due to the fact that based on user distributions in the space and the value of $k$ (or similarly size of the cloaked region), precise user location can be derived (Kalnis et al., 2006) using several techniques such as monitoring a sequence of queries over time, correlation attacks or reasoning about the possible location of the query point [some studies such as Du et al. (2007) strive to provide more privacy against trace analysis attacks by considering mobility patterns in location cloaking].

## 2.2 Decentralised cloaking

Anonymising a user's query by a trusted anonymiser has several drawbacks. To address the drawbacks of centralised cloaking, several studies propose the non-centralised approach in constructing the cloaked region (Duckham and Kulik, 2005; Kido et al., 2005; Ghinita et al., 2007b, 2007c; Chow et al., 2006).

**Figure 3** Decentralised cloaking




### 2.2.1 Architecture and query processing

In order to avoid a central anonymiser, Kido et al. (2005) propose the use of user-generated dummies to make a user's exact location indistinguishable in an *anonymity set* which contains the locations of the dummy users as well as the user's exact location. Depending on the availability of other users' location information to the user querying the system (via communicating with other users), two variants of generating dummies are proposed.

The approaches proposed by Chow et al. (2006) and Ghinita et al. (2007b, 2007c) assume users communicate with each other to collaboratively form a cloaked region. The cloaked region in Chow et al. (2006) is constructed by having each user communicating with other users around its vicinity until it finds enough users to form a cloaked region which contains $k$ users. If enough users are not found, each request receiver recursively broadcasts the request until $k$ users are found.

The peer-to-peer spatial cloaking algorithm discussed above is shown to have significant privacy leaks for many user distributions since the user initiating the query is usually located close to the centre of the cloaked region. Ghinita et al. (2007b) propose a hierarchical overlay network resembling a distributed B+ tree for constructing the cloaked region that overcomes the above drawback. However, it suffers from very slow response time. Ghinita et al. (2007c) propose methods which provide stronger privacy than Chow et al. (2006) for various distributions and do not suffer from slow response time of Ghinita et al. (2007b). The authors propose a distributed method to find a random set of $k$ adjacent users based on their 1-D Hilbert ordering. Finally, Duckham and Kulik (2005) propose a graph model to represent possible user's locations and denote the cloaked region by a set of vertices in the graph. The client progressively gives more information about her precise location until the query result set reaches her desired accuracy. This study does not consider the query processing.

Once the cloaked region is constructed using any of the above peer-to-peer approaches, the server receives an anonymised region. Therefore, similar to the centralised approach, the server should be modified to be able to respond to anonymised queries. However, the methods proposed in Kido et al. (2005) and Duckham and Kulik (2005) require the server to process a spatial query for every element inside the anonymity set instead of the entire cloaked region.

### *2.2.2 Strength and weaknesses*

The most obvious superiority of decentralised cloaking approaches to their centralised peers is avoiding a central trusted anonymiser. Similar to centralised cloaking, processing complex spatial queries is feasible as long as the privacy aware server can perform them on a cloaked region. Finally, processing a spatial query for each member of the anonymity set as proposed by Kido et al. (2005) and Duckham and Kulik (2005) further simplifies the framework since their proposed methods can be built on top of any of the conventional spatial query processing algorithms currently in use.

However, with the above decentralised approaches, the privacy threats are even more significant as the server knows the exact location of the user is provided in the anonymity set. Therefore, monitoring a sequences of queries can easily reveal valuable information to the server about the real location of the user. More importantly, in cloaking approaches, users should in fact trade-off their privacy with the accuracy of the query result or the efficiency of the query processing because a larger anonymity set (or similarly, the cloaked region) may result in a significantly larger query result set which includes many unnecessary data points that should be filtered. Furthermore, forming a large anonymity set prohibitively increases the communication cost between the users in the peer-to-peer architecture. Alternatively, decreasing $k$ (or the size of the cloaked region) will directly increase the probability of identifying the user's location. Therefore, preserving users' location information might not always be possible regardless of the size of $k$ (or the cloaked region). Finally, decentralised techniques assume all users subscribed to a service are trusted in order to collaboratively create the cloaked region. This assumption might be far from reality in typical LBS frameworks.

The problems stated above inspired designing more robust and privacy preserving schemes for location privacy. Sections 3 and 4 present two other classes of approaches based on transformation and PIR, respectively, to address the privacy concerns associated with cloaking and anonymity approaches.

## 3 Transformation

In this section, we present a class of approaches that do not employ cloaking techniques and anonymisers to achieve anonymity. We denote these techniques as *transformation*-based approaches since they are based on transforming the query to prevent the server from learning information about the users locations. Although all approaches discussed in this section utilise transformation to protect user's private location information, based on the proposed transformation scheme, they can be divided into two different groups: non-spatial and spatial transformation-based techniques.

### *3.1 Non-spatial transformations*

The class of approaches under this category are mainly standing on the shoulders of applied cryptographic protocols to achieve privacy (Indyk and Woodruff, 2006; Zhong et al., 2004, 2007). With these approaches, the query is evaluated in an encrypted space. Therefore, the transformation employed is some form of encryption.

### *3.1.1 Architecture and query processing*

The class of non-spatial transformation techniques blind the untrusted party (i.e., the server or another user) by utilising secure multi-party computation schemes. As we discuss later, the three techniques studied in this subsection do not address conventional spatial queries such as range and KNN queries. Each method instead supports a specific query type of interest (such as privately computing the distance between two parties or learning whether two parties are located in the same region). Therefore, we study each method separately in more details.

The scheme proposed by Indyk and Woodruff (2006) involves a two-party computation protocol between Alice and Bob to privately evaluate the distance between Alice's point and other $n$ points that Bob owns. After executing the protocol, Bob knows nothing about Alice's point and Alice only learns the nearest neighbour from Bob's points. Although the solution proposed is mainly of theoretical interest and does not focus on spatial queries or LBS, it can be considered as a method for protecting users' privacy in LBS. In other words, one can think of a privacy-aware LBS framework by treating Bob as an untrusted server and Alice as a user interacting with the server.

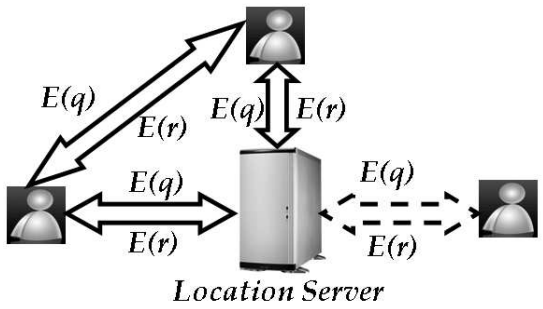
Zhong et al. (2007) propose three solutions to what they define as the *nearby-friend problem*. The problem is defined as allowing users to learn information about their friends' locations if and only if their friends are actually nearby. The three protocols are all efficient in terms of the amount of computation and communication required by each party. Each protocol is an instance of a multi-party computation scheme with certain strengths and restrictions (in terms of number of messages transferred and the resilience to a malicious party).

Finally, Zhong et al. (2004) provide two protocols aiming at protecting users' location information in LBS. While the first protocol allows a user to share her location information with other users via an untrusted server, the second protocol enables a dating service where a user learns whether other users with similar profiles (found by the server) are located in the same region she is located. This

protocol, which is of more interest to location privacy in LBS, assumes the entire user profile is known to the server, and the server first finds any potential matches between a user and all other users. The server then sends all matched profiles to the requester so that she can blindly compare their locations with her own location. Similar to Zhong et al. (2007), a multi-party computation protocol is proposed which involves the requester, the dating service and any other matched user.

Figure 4 illustrates the general framework of non-spatial transformation methods. All possible communications are shown as bidirectional arrows. The requester sends an encrypted request $E(q)$ to the other party who returns the encrypted response $E(r)$. The dashed arrows represent the message transfers between the two parties in Indyk and Woodruff's (2006) framework while solid arrows denote the possible client-client and client-server negotiations of Zhong et al.'s (2004, 2007) schemes.

**Figure 4** Non-spatial transformation




### *3.1.2 Strength and weaknesses*

The main advantage of the three methods discussed in Section 3.1.1 is their perfect privacy guarantee. Building their framework on well-known cryptographic primitives and widely used one-way functions, these protocols do not suffer from severe privacy leaks of anonymity-based methods. Furthermore, their problem-specific designs allow very efficient implementations of the protocols mostly involving only a handful of computations and few message transfers [specifically in Zhong et al. (2004, 2007)].

The major drawback of non-spatial transformations is their high computation or communication complexity when being used for spatial query processing. To illustrate, let us discuss two important properties that make any privacy-aware transformation-based spatial query processing a viable approach.

- *One-wayness property:* Virtually all studies related to location privacy in LBS assume the existence of an untrusted location server with which users have to interact to receive their query responses. Therefore, it is important to prevent the server from learning a user's location while responding to her query. This property is achieved by the one-wayness property in transformation-based approaches. In other words, the server processes encrypted queries and therefore, cannot reverse (i.e., decrypt) the transformed query or dataset to gain information about a user's exact location.
- *Locality preserving property:* While the one-wayness property is necessary, it is not sufficient to enable efficient implementation of privacy-aware LBS. This is due to an important fact that if the spatial relationship between the objects is not preserved by a transformation (e.g., while using encryption), the server has to blindly perform a linear scan of all object in the database in order to evaluate a query. Therefore, a transformation, while being one-way, also has to preserve the locality and proximity of objects to avoid a linear scan of the entire database for each query.

The inherent limitation in using non-spatial techniques for blind evaluation of spatial queries is rooted in the second property discussed above. To illustrate, assume a server uses any of the techniques discussed in Section 3.1.1 to compute the encryption of the Euclidean distance between an encrypted point (i.e., the query origin) and each point of interest to find the results of a *KNN* query. These encrypted distances can then be sent back to the client who can decrypt them and find the top $K$ results. Trivially, this protocol satisfies the first property discussed above since the location of neither the query point nor the result set is revealed to the server. However, the main limitation here is that the distance between query point and each and every point of interest must both be computed or transferred to the client, i.e., $O(n)$ computation or communication complexity where $n$ is the size of the database. This is because the POI are treated as vectors with no exploitation of the fact that they are in fact points in space. Therefore, the main limitation of encryption-based techniques discussed above is the loss of spatial information via encryption. This loss either results in a linear scan of the entire database if used to evaluate a spatial query [as in Indyk and Woodruff (2006)], or makes the protocol unusable for spatial query processing [as in Zhong et al. (2007, 2004).

Based on the above discussion, it is now easy to observe that the solution proposed for the 'private near neighbour problem' in Indyk and Woodruff (2006) incurs at least a $\Omega(n)$ communication complexity where $n$ represents the number of data points and $O\left(\sqrt{n}\log^{O(1)}(n)\right)$ computation cost for finding the exact nearest neighbour of the query point.

Similarly, with the protocols proposed by Zhong et al. (2007), Alice will know whether a certain user Bob is nearby. However, verifying whether a certain friend is nearby Alice is a different problem than finding Alice's nearest friends. Therefore, as we discussed in this section, finding the nearby friends using these protocols involves a costly multi-party computation between each user and every other user's friends in the system. This cost is prohibitive in terms of the amount of communication and computation required.

Finally, the work of Zhong et al. (2004) suffers from the same drawbacks since it only describes a matching protocol in an encrypted space between two users located in the same region. However, the real challenge is finding nearby matches which is not possible in an encrypted space. Furthermore, a potential privacy leak in the method proposed by Zhong et al. (2004) is that the service provider learns the exact locations of all users and only needs to learn the identity of each user at a certain location.

In this section, we showed how utilising cryptographic primitives of multi-party computation in evaluating queries suffers from prohibitive communication and/or computation cost. While all approaches offer efficient protocols to calculate a function (e.g., distance or region similarity) privately between two parties, they all come short of proposing techniques to enable *locating* the two parties which should follow the proposed protocols. We also argued that the main reason for the inefficiency of the above protocols lies in their lack of maintaining the privacy-preserving property. The techniques discussed in the next section overcome this limitation by utilising spatial transformations to maintain the relationship between the objects while protecting the user's information while responding to location queries.

### *3.2 Spatial transformations*

We proceed to present the spatial transformation-based methods proposed for location privacy. As we discussed in Section 3.1.2, the main intuition behind the techniques under this category is to somehow blind the server from learning the query location while still preserving the locality of objects with regards to each other and the query point.

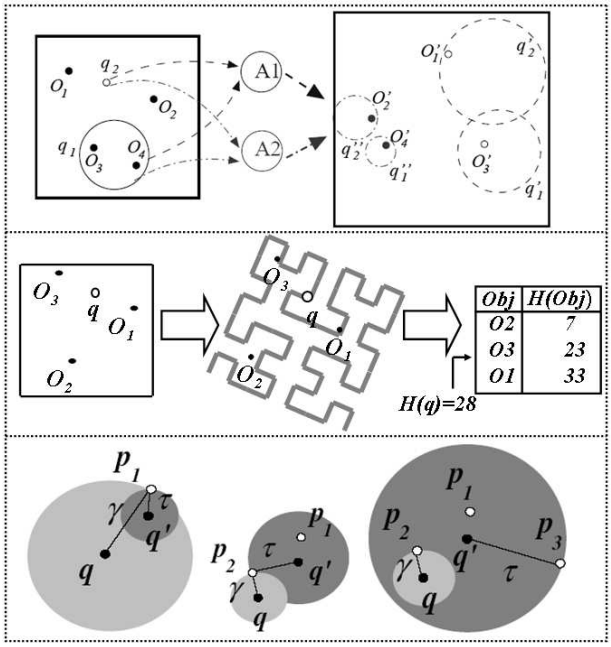
#### *3.2.1 Architecture and query processing*

The first method we study in this class is the work of Lin (2006). The author proposes a transformation-based approach to enable location privacy through the use of several agents. The key idea is to modify users and query location information through the use of various geometric transformations such as rotation, translation and scaling. The framework utilises several agents interposed between users and service providers to perform the transformations (the upper box in Figure 5). The agents serve as intermediaries and do not store user information since their only responsibility is to transform information received from other users or the server. To preserve privacy, users randomly choose the agent to perform the transformation.

A second study based on evaluating a query in a transformed space is proposed by Khoshgozaran and Shahabi (2007). The authors utilise space filling curves as one-way transformations to *encode* the locations of both user(s) and POI into an encrypted space and to evaluate a query in this transformed space (the middle box in Figure 5). The transformed space maintains the distance properties of the original space which enables efficient evaluation of location queries. At the same time, the one-way transformation proposed can be viewed as a *space encryption scheme* that allows fast computation of its inverse given some extra knowledge, termed *trapdoor* (Schroeder, 1990) or transformation *key*. Subsequently, the client can encrypt the query using its *key* and the server performs the query in the encrypted space and returns back to client the encrypted answers for client's fast decryption.

Most recently, Yiu et al. (2008) propose a framework termed SpaceTwist to blind an untrusted location server by incrementally retrieving POI based on their ascending distance from a fake location near the query point termed the *anchor* point. The query processing starts by the requester picking the anchor point. The entire area is divided into a supply and a demand space. The supply space centred at the anchor is the part of space already explored. The demand space denotes the space to be covered before the client is guaranteed to be able to produce an accurate result. The privacy is achieved by ensuring that only the client knows both the demand space and the supply space (whereas the server knows only the supply space). The query processing continues until the supply space is expanded enough that eventually covers the shrinking demand space (see the lower box in Figure 5). It is formally shown that this property ensures the query result set completeness. Note that with this approach, the query is still evaluated in the original space but the query point is transformed to an anchor point.

**Figure 5** Spatial transformation




*Source:* Top and bottom images taken from Lin (2006) and Yiu et al. (2008), respectively

### 3.2.2 Strength and weaknesses

Based on the two properties discussed in Section 3.1.2, it is clear that the dominant advantage of transformation-based methods is their locality preserving property. For the rest of this section, we elaborate more on the strengths and weaknesses unique to each method.

A strong feature of Lin (2006) is the provision of answering a wide range of conventional location-based spatial queries such as *KNN* and range queries on both static and dynamic datasets. Secondly, the proposed algorithms provide exact answers for both types of queries. However, this approach suffers from the following weakness which is common across all solid geometric transformations. A careful comparison of the original dataset with the transformed version can reveal significant amount of information to the server to reverse the transformation. For instance, using any combination of the proposed geometric functions will always map the central points of the space, to a central region of the transformed space and similarly moves the points near the edges into points that stay close to another edge in the transformed dataset. Secondly, this work requires users to trust the agents and share their location information with them. This brings up several issues of trust similar to what discussed in Section 2.2.

The choice of *space filling curves* as candidate space encoders for the framework proposed by Khoshgozaran and Shahabi (2007) and showing how it can be treated as a space encoder is the key idea to enable blind evaluation of *KNN* queries. This property makes query processing very efficient in terms of the computation and communication complexity. However, the proposed framework only addresses *KNN* queries since the space filling curves, by design, are very efficient for evaluating proximity-related queries as opposed to range queries. Furthermore, the *KNN* algorithm proposed is approximate although it is shown that the amount of approximation is quite satisfactory. The existence of a single key protected in tamper-proof devices and shared by all users is another weakness of the scheme proposed by Khoshgozaran and Shahabi (2007).

Finally, one of the main advantages of Yiu et al.'s (2008) framework is its lack of need for a transformation of the entire data set as required by the previous two approaches. Furthermore, utilising the existing query processing index structures present in a non-privacy aware servers makes it readily applied to existing location servers. However, similar to cloaking approaches, Spacetwist suffers from several privacy leaks and costly computation/communication if exact results and strict privacy are required. In other words, choosing an anchor point too close to the query point makes the framework's *privacy region* very small while choosing an anchor point too far away significantly increases the computation and communication costs of Spacetwist. More importantly, the supply space (and hence user's privacy region) can become very small for dense distributions and small values of *K*. That is, there is no lower bound for the size of the privacy region. Note that this region can become even smaller than a cloaked region.

## 4 Private information retrieval

The methods studied in Sections 2 and 3 each attempted to improve the efficiency or the privacy aspects of evaluating spatial queries privately in LBS. However, they mostly suffer from a privacy/quality of service trade-off. While on one extreme end the non-spatial transformation techniques provide perfect privacy, they result in very costly spatial query processing schemes. Similarly, on the other side of the spectrum, efficient cloaking or spatial transformation approaches might result is severe privacy leaks under certain user, object or query distributions.

The approaches studied in this section are based on the solutions proposed to the well-known problem of PIR. There is a wide spectrum of scenarios in which a user needs to gain access to a specific record of a database but does not want to reveal the record in which she is interested. More formally, a PIR protocol allows a user to retrieve the *i*th record from a database of size *n* stored at an untrusted server, without revealing *i* to the server. The main intuition behind using a PIR protocol for location privacy is to disguise the selection of records hosted at the untrusted server which are required to process a spatial query. Therefore, the remaining challenge becomes efficient retrieval of a subset of the server's database using PIR.

The class of PIR approaches can be roughly divided into cryptographic and hardware-based (also known as practical) approaches. While the former class makes use of homomorphic encryption, quadratic residues and other cryptographic properties to achieve PIR, the latter techniques utilise a secure coprocessor (SC) which acts as a securely protected computing space residing at the untrusted *host* machine that enables private querying of the data. Note that while PIR is not restricted to the schemes we discuss in this section, these are the only currently proposed techniques used to enable location privacy.

The recent techniques proposed by Ghinita et al. (2008) and Khoshgozaran et al. (2008) are both based on using PIR to achieve location privacy in the framework discussed above. However, while Ghinita et al. (2008) utilises theoretical PIR protocols to blind the server, Khoshgozaran et al. (2008) uses practical PIR techniques to enable location privacy.

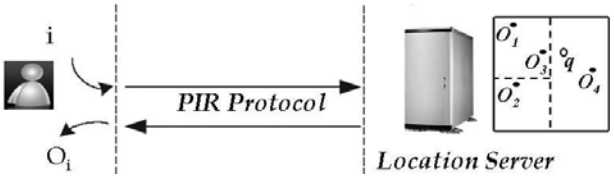
### 4.1 Theoretical PIR

The PIR problem was first proposed by Chor et al. (1998) in an information theoretic setting, which also proves that any theoretical PIR scheme has a lower communication bound equal to the database size. However, relaxing the problem to computationally bounded adversaries, a PIR framework is proposed by Kushilevitz and Ostrovsky (1997) based on the *quadratic residuosity assumption*. The client and server follow a secure two-party computation which allows the client to privately retrieve the *i*th bit from a bit string of size *n* owned by the server. Ghinita et al. (2008) build a framework on top of this PIR protocol to enable location privacy.

### *4.1.1 Architecture and query processing*

In order to privately evaluate nearest neighbour queries, during an offline process the untrusted server indexes the POIs with a kd-tree partitioning the space into several regions. While responding to an NN query, the client first learns the region *R* she is located at and privately requests the server for all objects within that region. Note that this method is approximate as there might be objects in other regions that are closer to the client than her nearest neighbour in *R*. Ghinita et al. (2008) further extend their approach to an exact solution. Utilising Voronoi diagrams superimposed by a regular grid, the client privately queries the server for the generators of the Voronoi cells intersecting with the grid-cell containing her. The returning result set is then decrypted and tested to find the actual NN to the client's location. Figure 6 illustrates the theoretical PIR framework proposed for blind evaluation of approximate NN queries.

**Figure 6** Theoretical PIR




### *4.1.2 Strengths and weaknesses*

The key strength of the technique proposed by Ghinita et al. (2008) is the perfect privacy guarantee against the most powerful adversaries and correlation attacks. As we discussed in Sections 2.1.2, 2.2.2, 3.1.2 and 3.2.2, the major concern with the two other classes of proposed approaches for location privacy is the existence of various attack models that can help adversaries to pinpoint (or approximate) the clients' locations. This information leak is theoretically avoided by PIR. Therefore, regardless of the users, objects and query distributions, the server is entirely blinded from learning a user's location. However, this stringent privacy guarantee comes at the cost of executing expensive protocols that result in significant communication and computation overhead. Furthermore, the proposed framework is not yet extended to the general case of *KNN* or other types of queries such as range queries.

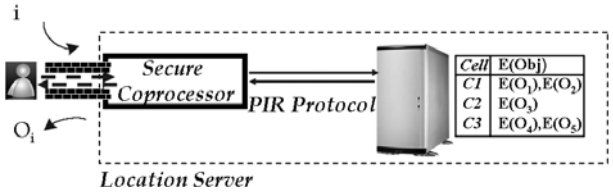
## *4.2 Practical PIR*

An SC is a general purpose computer designed to meet rigorous security requirements that assure unobservable and unmolested running of the code residing on it even in the physical presence of an adversary (Smith and Safford, 2001). The idea behind using a SC is to place a trusted entity as close as possible to the untrusted host to disguise the selection of desired records within a black box. In order to avoid the linear cost of going through each record in the host or sending the entire dataset to the user [i.e., $O(n)$ computation and communication cost, respectively) the technique proposed by Asonov and Freytag (2002) achieve optimal (i.e., constant) query computation and communication complexity at the cost of performing as much offline precomputation as possible. By generating a random permutation vector, *SC* privately generates an encrypted shuffled version of the original database and writes it back to the untrusted server while storing the permutation vector in its own memory. Building a PIR protocol based on the above database, a user can privately retrieve any record from the untrusted server via *SC*.

### *4.2.1 Architecture and query processing*

In order to avoid a linear retrieval of the encrypted shuffled database, Khoshgozaran et al. (2008) use regular grids to spatially index the object information for each grid cell. This spatial index is then converted by the *SC* to a permuted shuffled database stored at the server. Therefore, this *private spatial index* allows efficient and private retrieval of a small subset of all objects in the server's database to evaluate spatial queries. For instance, while evaluating range queries, knowing the granularity of the grid, one can efficiently find a set of cells whose objects partially or fully overlap with the query window. The tuples of such cells can then be privately retrieved from the server through *SC* and the PIR protocol discussed above. Figure 7 illustrates the practical PIR framework for evaluating range queries.

**Figure 7** Practical PIR




Using practical PIR to enable location privacy is also proposed by Hengartner (2007). The architecture uses PIR to privately retrieve location-specific object information from the server and trusted computing to implement a PIR algorithm. Two LBS scenarios such as proximity and tracking service have been discussed. However, the proposed framework is not yet implemented.

### *4.2.2 Strengths and weaknesses*

The practical PIR scheme proposed by Asonov and Freytag (2002) is 'almost optimal', meaning it achieves $O(1)$ online computation and communication complexity. The only parameter left to be improved is the time it takes to preprocess the data (i.e., generate the encrypted shuffled database) and prepare it for the PIR protocol. Therefore, compared to the work of Ghinita et al. (2008), the method proposed by Khoshgozaran et al. (2008) incurs significantly lower computation and communication cost. This is because removing the hardware dependency of a PIR protocol makes it very costly to guarantee perfect privacy. Furthermore, the proposed scheme has very

moderate space requirements compared to the theoretical PIR approach discussed in Section 4.1. However, relying on a SC to perform the PIR protocol has its own drawbacks. In particular, secure coprocessors are significantly slower than the conventional processors and have much less memory available. Therefore, the protocols designed to utilise them should be relatively light-weight. Finally, if enough space is not available to store an unused permuted and reshuffled database, shuffling has to be repeated periodically and its amortised cost adds to the query processing time.

## 5 Discussion

In this section, we first discuss some of the privacy threats associated with querying static POIs (such as restaurants and hotels) privately. While Section 5.1 is by no means a full treatment of possible attacks, it offers a brief overview of dominant privacy vulnerabilities in location-based services. We then proceed to briefly discuss querying other moving users rather than static POIs.

### *5.1 Vulnerabilities in privately querying static data*

The majority of the location privacy literature is focused on a general framework which enables *private queries over public data* (Mokbel et al., 2006). With this setting, users freely move in an area and they query the untrusted location server for information about POI such as restaurants and hospitals without disclosing their private location information.

As these POIs are static, their information is usually publicly available and hence it is imperative that any proposed scheme be resilient against the attacks performed by the server using this *prior knowledge*. Furthermore, the server might utilise the information in a query response to infer the query source. Khoshgozaran and Shahabi (2007) propose *result set anonymity* as a privacy metric to ensure protection against attacks based on learning the query result sets. Several other attacks have also been studied each utilising the object, query or user distributions in an area [e.g., see Kalnis et al. (2006)]. Finally, the proposed frameworks for location privacy in querying static objects should provide security against *active attacks*. With these attacks, the server colludes with malicious users to learn the query source in a cloaked region or the transformed dataset. While protecting users from all of the above attacks is very challenging with cloaking or transformation-based approaches, the studies using PIR can offer such privacy guarantees. Note that active attack is a major concern with peer-to-peer cloaking approaches such as those discussed in Section 2.2 as they assume no malicious user exists in the system.

### *5.2 Querying dynamic data*

A more general setting for LBS extends the above framework by allowing dynamic users to query each other, in addition to the publicly available static points. While many of the privacy vulnerabilities of previously discussed approaches still exist in this *private queries over private data* scenario (Mokbel et al., 2006), more problems arise regarding the issue of trust while querying dynamic objects. The main challenge associated with querying dynamic objects is the fact that there is a need for a central *global* view of all objects locations at any moment (note that we denote both clients and POIs as objects in this section). This dynamism, by itself, makes indexing and querying dynamic objects more challenging in a non-privacy setting. More importantly, with privacy-aware LBS, the centralised entity maintaining object locations cannot be the location server since it is not trusted.

While centralised cloaking techniques can delegate the task of monitoring users' locations to the trusted location anonymiser, many issues arise for enabling dynamic data querying with decentralised cloaking, transformation and PIR approaches. With decentralised cloaking approaches, no client owns the global view of all object locations at any time. Similarly, as we elaborated in Section 3.1.2, the lack of spatial properties of objects in non-spatial transformation approaches makes query processing very costly even for the simpler case of private queries over public data. The challenge with using spatial transformation techniques for querying dynamic objects is the complication of updating a client's location hosted on the server's side while preventing the server from tracking users. This problem is avoided in Lin (2006) by having agents update users' locations. However, users now have to trust the agents which act as local anonymisers. Furthermore, this approach is vulnerable against the server with a prior knowledge about object distributions. Finally, with PIR approaches, the main challenge is to allow users to privately manipulate a server's index structure and modify their location in the encrypted index hosted at the untrusted server.

Finally, we stress that a fundamental difference between the static and dynamic cases in LBS is the fact that users have to trust all other subscribed users (or at list those in their friends list) in the dynamic case. Note that while this assumption is not necessary for the static case, it is inevitable for querying dynamic objects. This is due to the fact that in the static case, users do not query about each other's location, and thus, do not have to trust other users for their querying needs. However, this is not the case for querying other users in the dynamic case.

**Table 1** Comparison of different classes of proposed approaches for location privacy

| *Technique* | *Reference* | *Query type* | *Major strengths* | *Major weaknesses* |
|---|---|---|---|---|
| Centralised cloaking | Mokbel et al. (2006), Gruteser and Grunwald (2003), Gedik and Liu (2005a, 2005b) and Du et al. (2007) | Range/KNN | Spatial query support, support for querying dynamic data | Major privacy leaks, trusting a third party, privacy/quality of service trade-off |
| Decentralised cloaking | Duckham and Kulik (2005), Kido et al. (2005), Ghinita et al. (2007b, 2007c) and Chow et al. (2006) | Range/KNN | No need for a centralised anonymiser, stronger privacy support compared to centralised cloaking | Costly communication complexity, assuming all users are trusted, privacy leaks, privacy/quality of services trade-off |
| Non-spatial transformation | Indyk and Woodruff (2006) and Zhong et al. (2004, 2007) | Customized two-party computation queries (private distance approximate, private co-location comparison, etc.) | Perfect privacy guarantee, very efficient customised Queries | Prohibitive linear computation or communication complexity for classic spatial queries |
| Spatial transformation | Lin (2006), Khoshgozaran and Shahabi (2007) and Yiu et al. (2008) | Range/KNN | Efficient spatial query processing, support for querying dynamic objects | Privacy leaks under certain object distribution, privacy/quality of service trade-off |
| Theoretical PIR | Ghinita et al. (2008) | Nearest neighbour | Perfect privacy guarantee, support for spatial queries | High computation and communication complexity |
| Practical PIR | Khoshgozaran et al. (2008) | Range | Perfect privacy guarantee, support for spatial queries | Hardware dependence, limited secure-coprocessor space and computation power |

## 6 Conclusions

This paper presented three distinct classes of approaches proposed for protecting users' location information in LBS. The first class of approaches, based on cloaking and anonymity techniques, offer flexible schemes to support privacy-aware location servers responding to various spatial queries. However, they suffer from multiple privacy leaks under certain user or query distributions. The second classes of approaches are based on transforming the queries to blind the server from knowing a user's location while evaluating location queries. With these approaches, users have to trade-off their privacy with the quality of service they receive from location-based services. Finally, the third class of PIR approaches addresses all privacy concerns of the previous approaches. However, they incur expensive computations or rely on a trusted platform to execute the queries. Table 1 summarises the properties of each category of approaches. Each table column represents the dominant properties shared among the proposed approaches under each category.

Location privacy research is still in its infancy. While creative solutions have been proposed to solve the location privacy problem, there are still many challenges to be addressed. Devising a framework that while ensuring perfect privacy, can very efficiently respond to various spatial queries dealing with both static and dynamic objects is still an open problem and far from what the existing approaches offer.

## Acknowledgements

This research has been funded in part by NSF grants IIS-0238560 (PECASE), IIS-0324955 (ITR), IIS-0534761 and IIS-0742811 (SGER), and in part under JPL SURP program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## References

Asonov, D. and Freytag, J. (2002) 'Almost optimal private information retrieval', *PET*, pp.209–223.

Bettini, C., Wang, X. and Jajodia, S. (2005) 'Protecting privacy against location-based personal identification', *Secure Data Management*, pp.185–199.

Chor, B., Kushilevitz, E., Goldreich, O. and Sudan, M. (1998) 'Private information retrieval', *J. ACM*, Vol. 45, No. 6, pp.965–981.

Chow, C., Mokbel, M. and Liu, X. (2006) 'A peer-to-peer spatial cloaking algorithm for anonymous location-based service', *GIS*, pp.171–178.

Du, J., Xu, J., Tang X. and Hu, H. (2007) 'iPDA: supporting privacy-preserving location-based mobile services', *MDM*, pp.212–214.

Duckham, M. and Kulik, L. (2005) 'A formal model of obfuscation and negotiation for location privacy', *Pervasive*, pp.152–170.

Gedik, B. and Liu L. (2005b) 'Location privacy in mobile systems: a personalized anonymization model', *ICDCS*, pp.620–629.

Gedik, B. and Liu, L. (2005) 'A customizable k-anonymity model for protecting location privacy', *ICDS*.

Ghinita, G., Kalnis, P. and Skiadopoulos S. (2007b) 'PRIVE: anonymous location-based queries in distributed mobile systems', *WWW*, pp.371–380.

Ghinita, G., Kalnis, P. and Skiadopoulos, S. (2007c) 'MobiHide: a mobilea peer-to-peer system for anonymous location-based queries', *SSTD*, pp.221–238.

Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C. and Tan, K. (2008) 'Private queries in location based services: anonymizers are not necessary', *SIGMOD*, pp.121–132.

Gruteser, M. and Grunwald, D. (2003) 'Anonymous usage of location-based services through spatial and temporal cloaking', *MobiSys*, pp.31–42.

Hengartner, U. (2007) 'Hiding location information from location-based services', *First International Workshop on Privacy-Aware Location-based Mobile Services (PALMS)*, in conjuction with *MDM*, pp.268–272.

Indyk, P. and Woodruff, D. (2006) 'Polylogarithmic private approximations and efficient matching', *TCC*, pp.245–264.

Kalnis, P. and Ghinita, G., Mouratidis, K. and Papadias, D. (2006) 'Preserving anonymity in location based services', Technical Report TRB6/06, National University of Singapore.

Khoshgozaran A. and Shahabi, C. (2007) 'Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy', *SSTD*, pp.239–257.

Khoshgozaran, A., Shirani-Mehr, H. and Shahabi, C. (2008) 'SPIRAL: a scalable private information retrieval approach to location privacy', *The 2nd International Workshop on Privacy-Aware Location-based Mobile Services (PALMS)*, In conjunction with *MDM'08*.

Kido, H., Yanagisawa, Y. and Satoh, T. (2005) 'An anonymous communication technique using dummies for location-based services', *IEEE International Conference on Pervasive Services*, p.1248.

Kushilevitz, E. and Ostrovsky, R. (1997) 'Replication is NOT needed: SINGLE database, computationally-private information retrieval', *FOCS*, pp.364–373.

Lin, D. (2006) 'Indexing and querying moving objects databases', PhD thesis, National University of Singapore, pp.122–163.

Mokbel, M., Chow, C. and Aref, W. (2006) 'The new Casper: query processing for location services without compromising privacy', *VLDB*, pp.763–774.

Schroeder, M. (1990) *Number Theory in Science and Communication*, 2nd ed., Springer, New York.

Smith, S. and Safford, D. (2001) 'Practical server privacy with secure coprocessors', *IBM Systems Journal*, Vol. 40, No. 3.

Yiu, M., Jensen, C., Huang, X. and Lu, H. (2008) 'SpaceTwist: managing the trade-offs among location privacy, query performance, and query accuracy in mobile services', *ICDE*, pp.366–375.

Zhong, G., Goldberg, I. and Hengartner, U. (2007) 'Louis, Lester and Pierre: three protocols for location privacy', *PET*, pp.62–76.

Zhong, S., Li, L., Liu, Y. and Yang, Y. (2004) 'Privacy-preserving location-based services for mobile users in wireless networks', Technical Report YALEU/DCS/TR-1297, Yale University.

## Websites

CNN (2007) 'Cabbies threaten strike over GPS systems', available at http://www.cnn.com/2007/TECH/08/01/gps.taxi.strike.ap/index.html.

New York Times (2008) 'Predicting where you'll go and what you'll like', available at http://www.nytimes.com/2008/06/22/technology/22proto.html.

Washington Post (2005) 'Online data gets personal: cell phone records for sale', available at http://www.washingtonpost.com/wp-dyn/content/article/2005/07/07/AR2005070701862 pf.html.