iPDA: Supporting Privacy-Preserving Location-Based Mobile Services*

Jing Du Jianliang Xu
Hong Kong Baptist Univ.
Kowloon Tong, Hong Kong
{jdu,xujl}@comp.hkbu.edu.hk

Xueyan Tang
Nanyang Technological Univ.
Singapore
asxytang@ntu.edu.sg

Haibo Hu
Hong Kong Baptist Univ.
Kowloon Tong, Hong Kong
haibo@comp.hkbu.edu.hk

Abstract

*This demonstration presents iPDA, a system to support privacy-preserving data access in location-based mobile services. The iPDA system consists of three main components: 1) a mobility-aware location cloaker that cloaks the user's location with a region and transforms a location-based query to a region-based query, 2) a progressive query processor that efficiently evaluates a result superset for the location-based query and, 3) a result refiner that refines the superset to generate the exact query result for the user. We discuss in detail the architecture and functionalities of our iPDA system. In addition, a tourist information system named **iGuide**, as an iPDA application, is prototyped for demonstration.*

1. Introduction

Location based mobile services (LBS) are growing dramatically along with development of GPS-powered cellular phones and PDAs. In LBS, users with location-aware mobile devices can query about their surroundings anywhere and at any time. Spatial range query and k-nearest-neighbor (kNN) query are two most commonly used queries. For example, a user can use a range query to find out all shopping centers within a certain distance, or use a kNN query to find out the k nearest ATM machines. To get the query results, the user has to provide the server with his/her current location. However, the disclosure of location information to the server raises privacy concerns, which have hampered the widespread use of LBS. Thus, provisioning LBS while preserving location privacy has been a hot research topic recently [1, 2, 3].

In this demonstration, we present an iPDA system (see Figure 1 for its architecture) to support privacy-preserving location-based data access based on our recent work [4]. A user can specify for each location-based query the privacy requirement with a minimum spatial area he/she wants to hide his/her location. For example, a user can specify the minimum acceptable spatial area is 1 square mile when in a shopping center or 10 square miles when in the Disneyland. Upon a location-based spatial query, the location cloaker cloaks the user's current location with a *cloak region* according to his/her privacy requirement. The location-based spatial query is thus transformed to a *region-based spatial query* before being submitted to the server. Upon receiving the region-based query, the query processor on the server evaluates and returns a *result superset* which contains the query results for all location points in the cloak region. Finally, the client refines the result superset based on the user's actual location to generate the exact result.

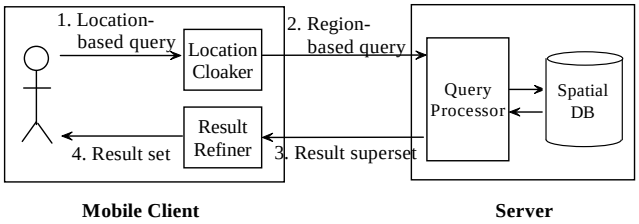
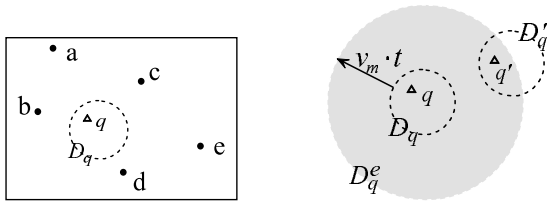


Figure 1. The iPDA Architecture

A number of challenging technical issues present in iPDA, including 1) how to effectively cloak user locations to meet user-specified privacy requirements and, 2) how to efficiently evaluate result supersets for region-based queries. To address these issues, we develop a *mobility-aware* cloaking technique that resists trace analysis attacks by considering mobility patterns in location cloaking. We also develop an efficient *progressive* query processor, which returns the query results incrementally and parallelizes query processing and result transfer. The progressive mode of query processing is particularly suitable for slow mobile networks.

To sum up, this demonstration shows the architecture and functionalities of our iPDA system through a tourist information system, named **iGuide**, for the urban area of Hong Kong. The remainder of this paper proceeds as follows. Section 2 presents the iPDA system in detail. Section 3 describes the demonstration system **iGuide**.

*This work was supported by the Research Grants Council, Hong Kong SAR, China under Project No. HKBU211206.



(a) Dynamic cloaking (b) Isolated cloaking

Figure 2. Dynamic Location Cloaking

2. iPDA System

2.1. Overview

The system is based on a client-server architecture. The clients are mobile and are equipped with GPS. Users are interested in querying public spatial objects (e.g., ATM machines, restaurants, hotels etc.) related to their current locations. These objects are maintained by a spatial database on the server. Following the iPDA architecture in Figure 1, Figure 2(a) shows an example for a 1-NN query using our iPDA system. Instead of providing its exact location q, the client submits a randomly generated cloak region D_q to the server. The server then returns the set of objects that are potentially a nearest neighbor of some point in D_q, i.e., $\{b, c, d\}$. Finally, the client uses the exact location q to find out its nearest neighbor, i.e., b. Throughout this query processing, the server only knows the region D_q in which the user is located, but not his/her exact location.

A circle is used to represent a cloak region. We adopt a simple yet practical privacy measure, i.e., the spatial area of the cloak region. The quality of location cloaking is measured by entropy. Suppose we can infer that the likelihood of the client being at location *(x, y)* in cloak region *D* is *prob(x, y)*, the entropy is defined by:

$$-\iint_D prob(x,y)\ln(prob(x,y))\,\mathrm{d}x\mathrm{d}y. \quad (1)$$

The large is the entropy, the more uncertain is the user location. It is obvious that the entropy is maximized when the probability of the client being at any location in the region is equal.

2.2. Mobility-Aware Location Cloaking

Several techniques have recently been proposed for cloaking user locations with different privacy metrics [1, 2, 3]. However, all prior techniques perform location cloaking in an isolated manner for each query. They did not take into consideration the spatial locality of user movement and, hence, are vulnerable to trace analysis attacks. Consider our previous example. Suppose the user issues a second query at location q' with a randomly generated cloak region D'_q (see Figure 2(b)). The server can then link the query trace and mobility pattern to infer the user location. For example, knowing the user's maximum moving speed, v_m, the server can draw an ever-expanding region D^e_q (the shaded area in Figure 2(b)) from the last cloak region D_q based on the product of v_m and the elapsed time t since the last query. Thus, it can be inferred that the user must be in the grey area of D'_q, which lowers the cloaking quality and may fail to meet the expected privacy requirement. The cloaking quality would further deteriorate with the analysis of more successive queries.

To address this issue, in iPDA, we have developed an optimal mobility-aware cloaking technique that resists trace analysis attacks by considering mobility patterns in location cloaking. Instead of generating cloak regions in an isolated manner for each query, we would like to control the generation of cloak regions such that the user is equally likely to be at any point in the newly generated region given the previous cloak region, which maximizes the cloaking quality. Denote by O the center of the old cloak region produced for the last query. We define $q(y|z)$ as the probability density function of the new object location being distance y away from O given that the center of the new region is distance z away from O. The expression of $q(y|z)$ under optimal cloaking can be obtained by mathematical analysis.[1]

To model controlled random generation of new cloak region, we define $p(z|y)$ as the probability density function of the center of the new cloak region being distance z away from O given that the new object location is distance y away from O. Thus, our problem is to find $p(z|y)$ to satisfy the constraints of $q(y|z)$. Note that the relation between $p(z|y)$ and $q(y|z)$ is given by the Bayes' rule, i.e.,

$$q(y|z) = \frac{p(z|y)\cdot u(y)}{\int_{\max\{0,x-r\}}^{\min\{R-r,x+r\}} p(z|x)\cdot u(x)\mathrm{d}x}, \quad (2)$$

where $u(x)$ is the probability density function of the new object location being distance x away from O at the time of the new query.

After obtaining $p(z|y)$'s, the distance between O and the center of the new region can be randomly generated. Given this distance, the center of the new region can be randomly generated on the corresponding circle perimeter. The detailed cloaking algorithm can be found in [4].

2.3. Region-based Query Processing

The evaluation of a region-based range query is straightforward since it is still a range query (with an extended range), which simply retrieves all objects within the spatial range. Thus, we focus on the evaluation of region-based

[1] Due to space limitations, the detailed expression of $q(y|z)$ is not presented here. Interested readers are referred to [4] for details.

kNN queries. Following Theorem 1 in [4], the results of a region-based kNN query include all objects in the region and the kNNs of any point on the region's perimeter (denoted by Ω).

Let $\{p_1, p_2, \cdots, p_M\}$ be the set of spatial objects. The basic idea of kNN query processing is to scan the objects one by one, and during each scan we maintain the set of arcs on Ω for each object to which this object is the 1st, 2nd, $\cdots$, and k-th NN. Each time a new object p_i is scanned, we compare it against each existing kNN candidate object, and check whether the new object will become the new i-th NN of an existing arc (or a part of an existing arc) by taking the place of its original i-th NN. After scanning the entire dataset, those objects which have at least one l-th-NN arc ($l \leq k$) constitute the kNN result set for Ω.

To support progressive query processing, after scanning of each object, we randomly pick an end point of an unchecked arc on Ω as the check point and compute its full kNN results. If any of the kNN results has not been returned to the client, it is output for immediate transfer. In this way, the query processing and result transfer are parallelized.

Furthermore, in order to speed up the convergence of the kNN candidate set, we sort the objects and scan the closest objects to Ω first because they are most likely to appear in the final kNN results. The detailed algorithm is described in [4]. As shown in [4], this progressive query processing algorithm has a worst-case time complexity of O(kM^3), where M is the cardinality of the spatial dataset.

3. Demo Description

The client side of our prototype system is implemented using MS Visual C# .NET 2003 and runs on a PocketPC operated by Windows Mobile 5.0. The PocketPC is equipped with GPS and a WiFi network interface card. On the server side, we develop our spatial database server based on UC Riverside's Spatial Index Library, running on Red Hat Linux 7.3 Server platform.

The functionalities of our system are demonstrated through a tourist guide prototype called **iGuide** (see Figure 3(a)). This application aims at providing location-based information (such as ATM, cafe, and restaurant) to a tourist while preserving his/her location privacy. The client side of **iGuide** functions as an interface to accept user queries and to display results. With GPS, the position of a tourist is tracked and provided for queries.[2] As shown in Figure 3(b), the client interface allows a tourist in the urban area of Hong Kong to specify the business of interest, e.g., a cafe, the privacy requirement in terms of the radius of cloak region, the type of search (either a kNN or a range query), and the associated query parameter (the k value for a kNN query or the

[2] At the time of demonstration, we emulate the user movement on the ground by the stylus movement on the screen.

Figure 3. Interacting with iGuide

radius for a range query). The user-specified query is taken and processed by the underlying cloaking module to form a region-based query. After receiving the result superset from the server, the client side displays the candidate results (see Figure 3(c)) and then refines them to show the final result based on the user's position (see Figure 3(d)).

References

[1] B. Gedik and L. Liu. A customizable k-anonymity model for protecting location privacy. *IEEE ICDCS*, 2005.

[2] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. *ACM MobiSys*, 2003.

[3] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The New Casper: Query procesing for location services without compromising privacy. *VLDB*, 2006.

[4] J. Xu, J. Du, X. Tang, and H. Hu. Privacy-Preserving Location-based Queries in Mobile Environments. Technical Report, Hong Kong Baptist University, 2006.