ISSN:0975-9646



# Proficient and Privacy-Attentive Framework for Continuously Moving Objects

Palukuri.Jhansy, T.Sunitha
*QIS College of Engineering and Technology, Ongole (dt)*

***Abstract*— Performance decline of periodic monitoring schemes of mobile clients for handling continuous spatial queries in terms of parameters like monitoring accuracy efficiency, and privacy has forced the development of a new scheme that holistically addresses the issues of location updating with respect to the above parameters. Location-based services are the scalable processing of location monitoring requests on a large collection of mobile objects. Previously, a mobile client encapsulates its exact position in a bounding box, and the timing and mechanism with which the box is updated to the server are decided by a client-side location update. Advances in sensing and tracking technologies create new opportunities for location-based applications but they also create significant privacy risks. An important privacy issue in Location Based Services (LBS) is to hide a user's identity while still provide quality location based services. In this paper, we propose a technique for preserving location privacy in moving-object environments. Our approach is based on the idea of sending to the service provider suitably modified location information. Our technique not only prevents the service provider from knowing the exact locations of users, but also protects information about user movements and locations from being disclosed to other users who are not authorized to access this information. This significantly improves the monitoring efficiency, privacy and accuracy compared to the periodic or deviation update methods.**

## I INTRODUCTION

In mobile and spatiotemporal databases, monitoring continuous spatial queries over moving objects is needed in numerous applications such as public transportation, logistics, and location-based services. Today people are increasingly aware of privacy issues and do not want to expose their personal information to unauthorized subjects or organizations. An important problem is represented by the possibility that a piece of personal information released by an individual to a party be combined by this party, or other parties, with other information, leading to the disclosure of sensitive personal information.

Therefore, there is an important concern for *location privacy* in location-based services. For example, GPS users who do not want to disclose their locations to the system may still require service such as "is there any of my friends close to me now?" There are two privacy requirements for this query. First, service providers are not allowed to know the real locations of users. Second, users can only query an authorized dataset.

In this paper, we address such a problem by developing a technique to preserve location privacy in moving-object environments. The basic idea of our approach is to send transformed user location data to the service provider. Transformations are performed by agents interposed between users and service providers. Agents are only responsible for transforming information either received from the users or the server. The service providers receive the transformed data and compute answers to queries on these transformed data.

The server stores for each agent a sub-dataset specific to the agent. A query is thus executed by the server separately on the sub-dataset of each agent. Our technique not only prevents service providers from inferring the exact locations of users, but also keeps information about the location of an individual private from other individuals not authorized to access such information. A key characteristic of our approach is that privacy is achieved without degrading service quality.

## II RELATED WORK

Recent works focus on the development of anonymization techniques specific to location-based service environments. A common technique is based on the notion of spatial temporal cloaking. The idea is firstly introduced by Gruteser et al. They propose the application of the k-anonymity technique to cloak location information in order to support anonymous applications. Specifically, a user's location is represented by a region in which other $k - 1$ users are also present. This model has later been improved by Gedik et al. Their approach supports the assignment of different values for different users to the k parameter in a system. Also as part of their work, they investigate the tradeoff between anonymity and accuracy requirements.

Beresford et al. use the k-anonymity metric in pseudonymous applications. The idea is to rename user's identity when there are at least k users in the same zone. When there are less than k users in the same zone, a user may refuse to disclose his location. Recently, Cheng et al. investigated the trade-off of location cloaking, privacy and quality of service. They developed queries that evaluate cloaked data and provide probabilistic answers. They also presented quality metrics in order to quantify the effect of cloaking on service quality. Based on the similar idea, Mokbel et al. propose a framework to protect mobile users in location-based services, which adopts the cloaking idea and supports various k parameters.

Some other approaches are based on cryptographic techniques. Hore et al. suggest encrypting location data and using a privacy-preserving index for executing range queries over encrypted data. However, this technique only works for specific query operators and is unable to provide accurate

query answers. Similarly, Khoshgozaran et al. also propose a one-way transformation to encode all static and dynamic objects and resolve the query blindly in the encoded space. Again, they are not able to generate the exact query answers. To rectify the shortcomings of previous work, Yiu et al. have proposed a client-side query processing technique that retrieves points of interest from the server incrementally until accurate query answers are obtained. The main problem of this approach is the expensive communication cost since users need to receive much more data than just query answers. Ghinita et al. propose a framework to support private nearest neighbor queries based on Private Information Retrieval (PIR). Their approach does not require users to trust any third-party anonymizer and can return exact answers.

## III BASIC ARCHITECTURE

The basic architecture of our system is as shown in fig 1. The basic strategy underlying our approach is to reduce the leaking of private information by using data transformation and employing m agents in-between users and servers.

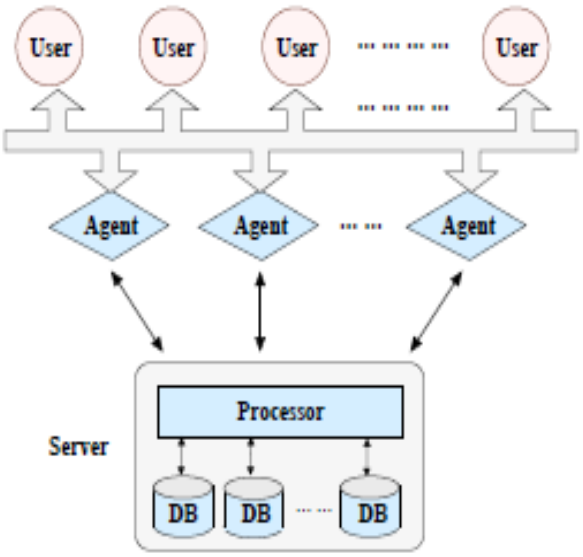


Fig 1: System overview

Each time a user needs to update his position, he does not directly contact the server; instead, he *randomly* selects an agent to which he sends his data. When querying, the user has to send the query to all agents. Then the agents will execute a transformation on the user data or queries and pass the transformed data to the server. The server handles the data processing and returns the query results to the agents. After receiving the results from the server, the agents perform a reverse transformation before returning the results to the user.

**User :** Users are position providers or query issuers. Users' positions are assumed to be unchanged until next update, that is, the *location database* at the service provider keeps the latest position of each user. Users may have a list of qualified agents, and they are assumed to have the ability to randomly choose agents and perform some post processing. A user can be a member of multiple groups and hence he may have a list of group IDs.

**Agent:** Agents are a critical component in our approach. An agent transforms the data received from the users and sends them to the server. It also executes reverse transformations on the data obtained from the server and then forwards them to the users. There are three important features about our agents. First, for the security purpose, agents are independent of the main server, which means they are not under control of the server. Second, agents do not store any user data and hence they are lightweight computers. Third, transformation functions for different types of data do not need to be changed at the same time.

**Server:** The server is responsible for data storage, maintenance and query processing. It also maintains datasets transferred by various agents separately. Any index for moving objects that supports efficient updates and queries can be adopted to manage the datasets in the server.

## IV PROPOSED SYSTEM

In this section, we presented data transformation, queries and updates in our system.

### *A) Data transformation*

Data transformation includes transformation of user IDs, group IDs, user locations, and queries.

First, we will discuss about ID transformation. The main purpose of ID transformation is two-fold. First, we need to prevent the server from identifying the same users through different agents. This can be easily achieved by choosing different transformation functions for different agents. Also, we need to prevent the server from tracking the positions of the same user from one agent. We thus propose to periodically change the transformation functions for each agent which can assign different pseudo-IDs to the same user who sends data at different time instants.

Just transforming IDs is not enough to provide location privacy for users. So, we use location transformation because some locations are strongly associated with user IDs and may thus cause information leak. Possible transformation functions include scaling, rotating, translation, and their combinations. The main challenge in the development of suitable functions for location transformation is to keep the relative distance in each sub-dataset unaltered by the transformation in order to support location based services. The first transformation function can be an arbitrary one, while the following transformation functions need to fulfill some constraints. The differences among the transformed positions obtained by various transformation functions should be kept within a small range.

Due to the multiple transformations on the users' positions, a query has to handle data from different transformations. One solution is to transform the query using all transformation functions, and then execute multiple queries. However, this is not efficient and may disclose the relationship among transformation functions. Therefore, we introduce the concept of *super query*, which covers all queries after multiple

transformations. To characterize the super query, we define its *false negative rate* as the number of missing query answers divided by the number of correct query answers, and define its *false positive rate* as the number of false query answers divided by the number of correct query answers.

Generally, an update is interpreted as a deletion followed by an insertion. To insert, three steps are performed. First, the user randomly selects an agent and sends his information to the agent. Second, the agent transforms the user ID, the group ID list and the location, and then sends the transformed data to the server. During the transformation, the agents will adjust the counters of the transformation functions, and remove the ones with counters equal to 0 which will not be used in the future.

Finally, the server tags the data with the agent ID and stores them. For the deletion, the user needs to submit his old information to the same agent which handled the insertion of this information. Agent will check the transformation and apply corresponding function at the update time. Then, the agent will use this function to transform user information, and decrease the counter of this function by one. If the counter is 0, the function (except for the first one) will be removed from the transformation table. The remaining process for deletion is similar to the insertion.

### *B) Queries*

Here, we will discuss about k- nearest neighbor queries which is based on range queries.

A range query retrieves all objects whose location falls within the circular range at a given query timestamp. As object positions are transformed in different ways through different agents, we have to send a query to all agents. Each agent will generate and send a super query to the server. After receiving the query answers from the server, the agent needs to transform them back and to check whether they are the correct answers to the original query. Finally, users will aggregate the partial results obtained from the agents. If user ranks or group IDs are to be taken into by the query, one more filtering step will be carried out by the server in order to prune unqualified answers.

The k nearest neighbor query (kNN query) retrieves k objects for which no other objects are nearer to the query object at a given query timestamp. For simplicity, we propose to compute the kNN query by iteratively performing range queries with an incrementally expanded search region until k answers are obtained. Like the range query, a kNN query also needs to be sent to all agents.

The main difference is that each agent needs to convert the kNN query to a range query first. Then the agent transforms the range query and the expansion parameter, and sends them to the server. The server will keep processing the range query with the radius extended by each time, and return the query result to the agent once it obtains k qualified answers. Finally, each agent computes the correct distance, and sends the distance along with the user IDs to the user that issued the query. The user then combines these to find his true k nearest neighbors.

## V PERFORMANCE

The trade-off issues between privacy and communication costs have been widely studied in context of network-level privacy protection. In particular, techniques have been devised to enhance network privacy by increasing the communication costs. In our system, there are two types of operations: update and query operations. An update needs one round of communication between a user (agent) and an agent (server). Its communication cost is independent of the number of agents.

Our technique not only prevents service providers from inferring the exact locations of users, but also keeps information about the location of an individual private from other individuals not authorized to access such information. Specifically, users have a list of group IDs that indicate which groups they belong to. Based on these group IDs, the server can remove the query answers that are not in the qualified groups, so that users can avoid their privacy leaked to other users not belonging to the same group. Our approach can be easily applied to multiple-server environments as the sub-databases in the server are relatively independent of one another. If so, the software contention may be reduced to more extent than traditional approaches. In fact, our approach provides increased opportunities for parallel execution.

## VI CONCLUSION

In this paper, we propose an approach to address the problems of location privacy in moving-object environments. Our approach uses a number of agents in-between users and servers. our system can prevent servers from knowing exact locations of users, and even map topology. The basic idea of our approach is to send transformed user location data to the service provider. Transformations are performed by agents interposed between users and service providers. Agents are only responsible for transforming information either received from the users or the server. The service providers receive the transformed data and compute answers to queries on these transformed data.

## REFERENCES

[1] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Preserving user location privacy in mobile data management infrastructures. In *Proc. Workshop on Privacy Enhancing Technologies*, 2006.

[2] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L.Tan. Private queries in location based services: Anonymizers are not necessary. In *Proc. SIGMOD*, 2008.

[3] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. MobiSys*, pages 31–42, 2003.

[4] S. Babu and J. Widom, "Continuous Queries over Data Streams," Proc. ACM SIGMOD, 2001.

[5] R. Benetis, C.S. Jensen, G. Karciauskas, and S. Saltenis, "Nearest Neighbor and Reverse Nearest Neighbor Queries for Moving Objects," Proc. Int'l Database Eng. and Applications Symp. (IDEAS), 2002.

[6] B. Hore, S.Mehrotra, and G. Tsudik. A privacy-preserving index for range queries. In *Proc. VLDB*, pages 720–731, 2004.

[7] C.-Y. Chow, M.F. Mokbel, and X. Liu, "A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-Based Services," Proc. ACM Int'l Symp. Geographic Information Systems (GIS), pp. 171-178, 2006.

[8] G. Ghinita, P. Kalnis, and S. Skiadopoulos, "Mobihide: A Mobile Peer-to-Peer System for Anonymous Location-Based Queries," Proc. Int'l Symp. Spatial and Temporal Databases (SSTD), 2007.