Cache Management Techniques for Privacy Preserving Location-based Services

Yu Chen[1] Jie Bao[2] Wei-Shinn Ku[2] Jiun-Long Huang[3]
[1]*Dept. of Electrical & Computer Engineering, SUNY - Binghamton, Binghamton, NY 13902*
[2]*Dept. of Computer Science & Software Engineering, Auburn University, Auburn, AL 36849*
[3]*Dept. of Computer Science, National Chiao Tung University, Hsinchu, Taiwan 300*
ychen@binghamton.edu, {baojie, weishinn}@auburn.edu, jlhuang@cs.nctu.edu.tw

Abstract

In order to access location-based services, mobile users have to disclose their exact locations to service providers. However, adversaries could collect the location information for purposes against mobile users' privacy. There are existing solutions for privacy protection by utilizing the K-anonymity model. However, the computational and communication costs are high. This research proposes cache management techniques for further improving user privacy protection, saving computational power, and decreasing communication costs.

1. Introduction

As a result of recent advances in wireless technologies, more and more personal mobile devices (e.g., cell phones, PDAs, etc.) possess the ability to access the Internet ubiquitously. In addition, Global Positioning System (GPS) receiver modules gradually become a standard component in new generation handheld devices. Consequently, novel location based services (LBS) allow users to launch location-dependent queries ubiquitously. Sample queries of such location based services include "*find me the nearest ATM*" and "*show me the gas station with the lowest price within one mile*." In order to fulfill these queries, mobile users have to reveal their current locations to service providers. However, service providers may disclose the trajectory of a certain user to attackers which decreases the dependability of LBS.

For protecting mobile user's privacy, recent research in [12] proposed a framework for location based services without compromising location privacy by leveraging the *K*-anonymity concept [15]. In order to implement the *K*-anonymity mechanism, the framework contains a trusted server to collect user location information and perform cloaking procedures. Then, the trusted server will send the location-dependent query along with the cloaked spatial area to service providers to retrieve query results. The returned query solutions will be sent back to individual users by the trusted server as well. Since the trusted server has the knowledge of all query results, we propose to store them in memory and use the cached data to answer future queries. Our solution has two main advantages. First, user privacy protection can be further improved, because the trusted server does not have to forward every query to service providers and it is much more difficult for adversaries to launch correlation attacks [3]. Second, with our cache management techniques, fewer queries have to be answered by service providers. Consequently, computational resources and communication costs can be effectively saved.

The rest of the paper is organized as follows: Section 2 surveys the related work of privacy protected query processing. Our own system design is detailed in Section 3. The experimental results are presented in Section 4. We conclude this paper in section 5, and we also raise some open issues for future research.

2. Related Work

Traditional privacy protection solutions rely on encryption and pseudonym techniques to safeguard users' communication and identities. However, the queries launched by users may contain sensitive information (e.g., physical locations), which could harm users' privacy. Recently, researchers have proposed several novel techniques [8], [6], [12], [13] for location-based spatial queries without compromising privacy based on the well-known *K*-anonymity mechanism [15]. A user can demand the cloaked area to cover the location of $K - 1$ closest peers for anonymizing its exact location. In order to

keep a reasonable size of the cloaked area in high user density regions, the user is able to decide the minimum acceptable cloaked region size. Their system model is similar to the architecture depicted in Figure 1. However, each of them applies different cloaking mechanisms and user location management data structures. In order to avoid the single point of failure (the location cloaker) problem in the aforementioned systems, Ghinita et al. [6] and Chow et al. [4] proposed peer-to-peer architecture based spatial cloaking techniques. On the location-based service provider side, there these system ([8], [6], [12], [13]) also provide solutions for cloaked nearest neighbor queries and range queries. For cloaked range queries, the general solution is to extend the received cloaked region outward by the search distance *d* on all dimensions. For cloaked nearest neighbor queries, most existing solutions are based on the range nearest neighbor technique [5], which retrieves the nearest neighbors for every point within a range.

From the perspective of queries, privacy is protected only for a single snapshot location-based query in these aforementioned systems. Users are not protected from query tracking attacks and *correlation attacks* [1], [3]. For example, if a mobile user launches the same query from different locations (i.e., continuous query), the mobile user's location can be identified by comparing the users in all the related cloaked regions. Our cache management techniques can effectively decrease the number of queries which has to be forwarded to service providers and successfully alleviate correlation attacks.

Location-Based Services can be generally defined as services that integrate the location of mobile devices with other information so as to provide added value to mobile device users [14]. For example, a motorist can find his/her closest gas station through LBS when he/she drives on a highway. LBS have been well developed during the past decade and many of them are very popular in our daily lives (e.g., navigation services, friend finding services [5], etc.). However these prevalent location-based services could be a potential threat to user privacy. Consequently both location-based service providers and mobile users should be careful and sensitive regarding the way location related information is handled. In addition, governments (both the U.S. and EU) have also legislated regulations on the usage of personal location information [14].

3. System Design

In this section, we describe our cache management techniques for supporting privacy preserving spatial queries in mobile environments. The fundamental idea behind our methodology is to leverage the cached results from prior spatial queries for answering future queries at the location cloaker.

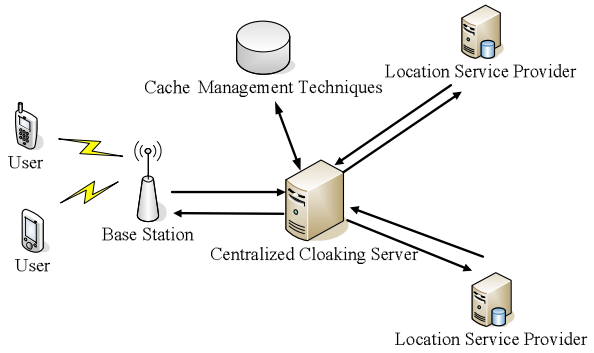

Figure 1. System architecture.

3.1 System Architecture

Our system architecture is consisted of four main entities: *mobile users*, the *location cloaker*, *cache management techniques*, and *location-based service providers* as illustrated in Figure 1. We consider mobile clients such as cell phones, personal digital assistants (PDA), and laptops, that are equipped with global positioning systems for continuous position information. In addition, we assume that there are access points/base stations around the system environment for mobile devices to communicate with the location cloaker. All users are mobile and travel on underlying road networks.

The location cloaker is an intermediate agent which can be trusted by mobile users. The location cloaker receives continuous location updates from mobile users and stores their locations with an index structure. In addition, the location cloaker also anonymizes the location of any query requesting mobile user to a cloaked region before forwarding the query to related location-based service providers. Any user identity related information in the query is also removed by the location cloaker during the cloaking process.

Location-based service providers play the role of spatial data maintainers and spatial query processors in our system. In order to handle privacy protected spatial queries, location-based service providers implement privacy protected query processors in their databases. The privacy protected query processor has the ability to process cloaked spatial queries efficiently and retrieves the inclusive result set (i.e., the minimal set which covers all the possible answers) for query requesters. After receiving the result set, mobile users can distill the exact answers from their locations in

linear time [10]. Basically, strict privacy requirements increase the complexity of processing a location-based query.

3.2 Cache Based Spatial Query Processing

Caching is a key technique to improve data retrieval performance in mobile environments [9]. As we can see in Figure 1, all the spatial queries and returned query results have to pass through the location cloaker. Consequently, if the location cloaker can cache the received query results from service providers, the cached results can be utilized to fulfill new spatial queries from mobile users. By applying this cache based solution, mobile users' privacy protection can be further improved. Since the location cloaker can solve a certain number of queries without forwarding them to service providers, it would be much more difficult for adversaries to launch correlation attacks.

For each received spatial query result, the location cloaker calculates the minimum bounding rectangle (MBR) of all the returned spatial data objects. Then, these retrieved data objects will be inserted into the cache and the boundary of the cached region with be adjusted based on the MBR. Figure 2 demonstrates the relationship between the cached region (the shaded area) and the whole search space. Since k nearest neighbor (kNN) query and window query are two common types of spatial queries, we focus on the two spatial query types in this paper. We introduce our cache based spatial query processing techniques as follows.

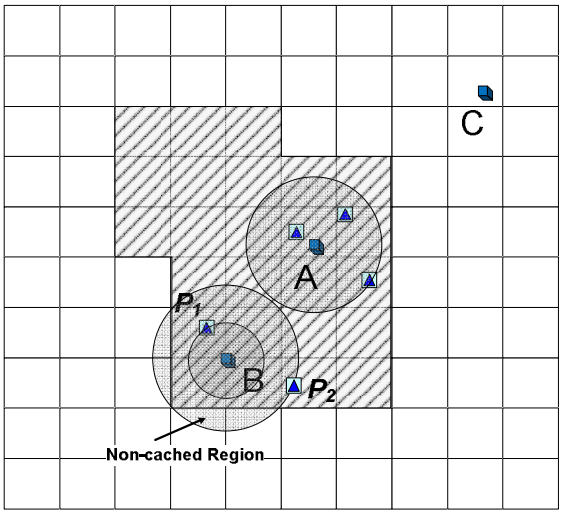

Figure 2. kNN query examples.

3.2.1 k Nearest Neighbor Queries

For kNN queries launched by mobile users, the location cloaker first checks if the query point is covered by the cached area. If the query point is covered by the cached region (e.g., points A and B in Figure 2), the location cloaker will try to retrieve k cached objects to answer the kNN query. Basically, there are two possible conditions – the query can be totally fulfilled or the query can only be partially fulfilled. The mobile user at point A requests for three nearest points of interest (POI) and we can retrieve three nearest objects from the cache based on their spatial relationships. Therefore, the query of mobile user A can be solved without forwarding the query to any service providers. Similarly, the location of mobile user B is covered by the cached region and its k value is equal to two. For this query, we can only retrieve one POI P_1 whose verification circle (with the distance between P_1 and B as the radius and B as the center point) is totally covered by the cached region. POI P_2 demonstrates a counter example. Since we are not sure if there is any POI within the non-cached region outside the cached area, P_2 cannot be count as a nearest neighbor of B [11]. Consequently, the kNN query of B still needs to be forwarded to service providers. However, the partial result (i.e., POI P_1) can be returned to mobile user B for decreasing the response time. If approximate results are acceptable, POI P_2 will also be returned. For mobile user C, since its location is out of the boundary of the cached region, the location cloaker will forward its kNN query to service providers.

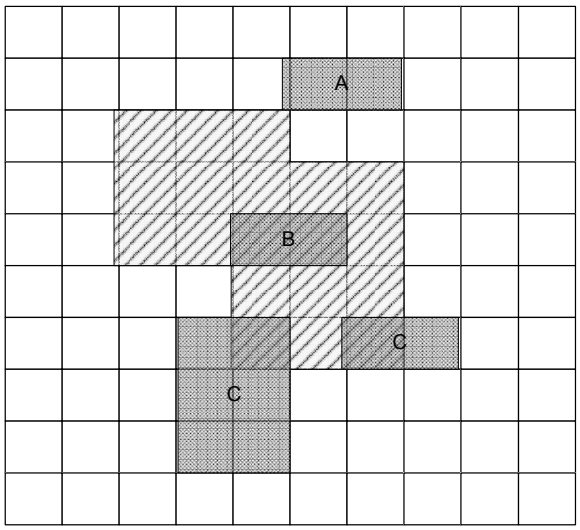

Figure 3. Window query examples.

3.2.2 Window Queries

Window queries find data objects within a specified area – the query window. Generally, there are three possible spatial relationships between the query window and the cached region. First, the query window is totally cover by the cached area (e.g., query window B in Figure 3). The query can be directly answered by the location cloaker without forwarding it to service providers. Second, the query window is partially covered by the cached area (e.g., query windows C and D in Figure 3). For this condition, the location cloaker still needs to forward the reduced query window (the portion not covered by the cached area) to service providers. However, both the query processing time and communication costs can be effectively decreased. Third, there is no overlapping between the query window and the cached area. Consequently, the cached data set cannot be applied to improve query evaluation performance and privacy protection.

3.3 Cache Space Management and Replacement Policies

In order to improve the performance of our cache based solution (i.e., cache hit rate), we have to develop efficient cache space management mechanisms. As illustrated in Figure 2, the cached data in the location cloaker is indexed by a grid structure. Based on statistics, the query frequency of each POI type is not uniformly distributed during a day. For example, there are significantly more queries for restaurants during dinning time (i.e., noon and evening). The frequency of queries for gas stations increases during rush hours and there are more queries for hotels in the evening. According to the variation of query frequency for different POI types, we design a temporal dynamic cache space allocation mechanism as illustrated in Figure 4. We verified the feasibility of our design with extensive simulations and the results are presented in Section 4.

For cache replacement policies, we apply three methods to decide which grid cell should be replaced based on time, retrieval frequency, and mobile user density.

- *Time Based Policy*

  The weight of each grid cell is according to a timer, which records the time interval from the last visit to present. Similar to the Least Recently Used (LRU) algorithm, the cell, which has the largest time interval, will be discarded first.

- *Retrieval Frequency Based Policy*

  Since retrieval frequency reflects the popularity of a certain data object/spatial region, this method decides the weight of each grid cell based on the number of times which it has been searched. The cell with the lowest visit frequency will be replaced first.

- *Mobile User Density Based Policy*

  Mobile users usually interest in POIs close to their current locations. Accordingly, it is an ideal strategy to keep grid cells which have high mobile user density and discard low user density cells.

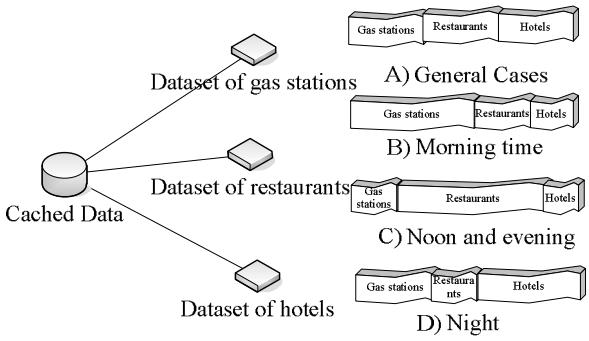

Figure 4. Dynamic allocation of cache space based on spatial query frequency.

4. Experimental Results

To evaluate the performance of our approach, we have implemented our cached based query solutions and cache management mechanisms within a simulator. The objective of our design is to decrease the number of queries which have to be forwarded to service providers to preserve mobile users' privacy, save computational power, and decrease communication costs. Based on our novel cache replacement policies, the cache hit rate can be effectively increased.

4.1 Simulator Implementation

Our simulator consists of four main components, the *mobile environment*, the *location cloaker*, the *cache management module*, and the *location-based service provider*. For the mobile environment, we applied the network-based moving objects generation framework [1] to generate a set of mobile users and the underlying road network inside the city boundary of Oldenburg in Germany. Each mobile user is an independent object which encapsulates all its related parameters (e.g., its current speed and destination). We implemented our query processing and cache management techniques as new modules for interacting with mobile users to improve query performance and privacy protection.

Every simulation has numerous intervals (whose lengths are Poisson distributed), and during each interval, the simulator selects a random subset of

mobile users to launch spatial queries (the query intervals are also based on the Poisson distribution). The subset size is controlled by the user defined mean number of queries per minute (e.g., 1000 queries per minute).

To obtain results that closely correspond to real-world conditions, we obtained our simulation parameters from public data sets, for example, mobile user and gas station densities in Oldenburg.

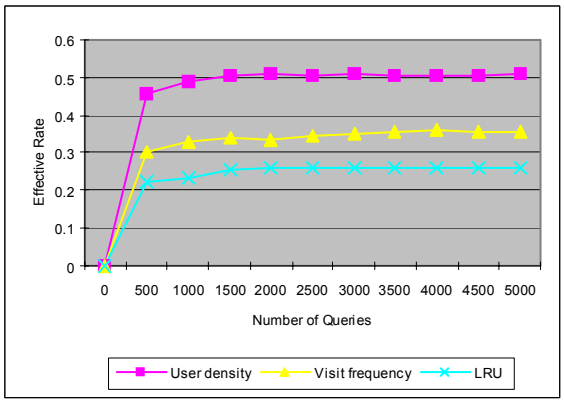

Figure 5. The cache hit ratio of the three cache replacement policies with increasing *k*NN query number.

- *Mobile Users*: The population in Oldenburg is 159,282 based on Wikipedia. According to the mobile device penetration rate in Germany, we estimate that there are around 5,000 mobile users served by one location cloaker.
- *Points of Interest*: We obtained the information concerning the density of the of-interest objects (e.g., gas stations, restaurants, etc.) in Oldenburg from Google Maps. Because gas stations are commonly the target of spatial queries, we use them as the sample POI types for our simulations. According to Google Maps, there are 1,399 gas stations inside the city boundary of Oldenburg.

4.2 Performance of the *k*NN Query

We first tested the performance of our three cache replacement policies with *k* nearest neighbor query. We increased the number of queries per time interval from 1 to 5000. As we can see in Figure 5, the cache replacement policy based on mobile user density prevails over two other strategies. The cache hit ratios of our two novel replacement policies are remarkably higher than the traditional LRU solution.

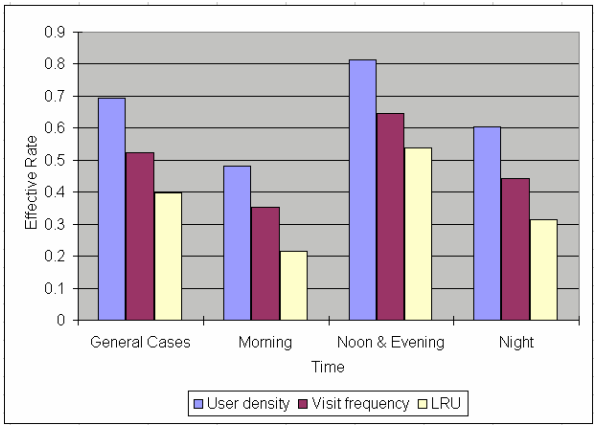

Figure 6. The cache hit ratio of different time intervals during a day with our dynamic cache space allocation mechanism.

Figure 6 shows the effect of our temporal dynamic cache space allocation mechanism. Our technique improved the cache hit rate for one time interval, Noon & Evening. However, there was no improvement in other two time intervals. Since mobile users' behavior varies at different locations, users may decide when to apply our mechanism based on statistics and experimental results.

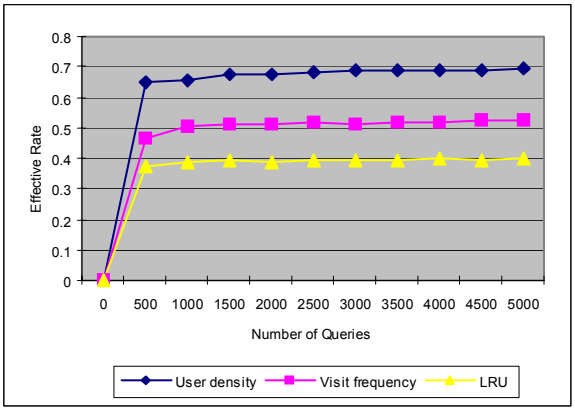

Figure 7. The cache hit ratio of the three cache replacement policies with increasing window query number.

4.3 Performance of Window Query

To see the effect of our cache replacement policies on window queries, we increased the query number from 1 to 5000 and the result is demonstrated in Figure 7. Similar to *k*NN query, the cache replacement policy based on mobile user density outperforms two other strategies and the performance of our two solutions are better than LRU.

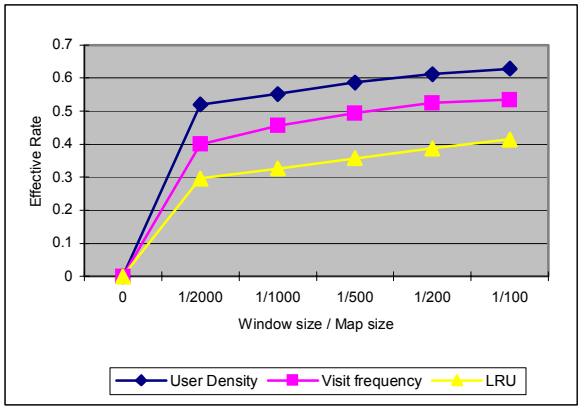

Figure 8. The cache hit ratio of the three cache replacement policies with increasing query window size.

We also studied the effect of various query window sizes by enlarging the query window size from 0 to 1/100 of the whole search space and the results are shown in Figure 8. Basically, the result trend is very similar to the previous experiment.

5. Conclusion and Future Work

This paper presented novel cache management techniques which can be applied in location cloakers. Significantly, our solution can further improve user privacy protection, save computational resources, and decrease communication costs. The experiment results indicate that our method can increase cache hit ratio remarkably. We plan to extend the proposed solutions to support more spatial query types and also experiment the corresponding performance of our mechanisms in the future.

6. References

[1] Claudio Bettini, Xiaoyang Sean Wang, Sushil Jajodia. Protecting Privacy Against Location-Based Personal Identification. In *Security Data Management*, pages 185-199, 2005.

[2] Thomas Brinkhoff. A Framework for Generating Network-Based Moving Objects. *GeoInformatica*, 6(2):153–180, 2002.

[3] Chi-Yin Chow, Mohamed F. Mokbel. Enabling Private Continuous Queries for Revealed User Locations. In *International Symposium on Spatial and Temporal Databases* (SSTD), pages 258-275, 2007.

[4] Chi-Yin Chow, Mohamed F. Mokbel, and Xuan Liu. A Peer-to-peer Spatial Cloaking Algorithm for Anonymous Location-based Service. In *Proceedings of the 14th ACM International Symposium on Geographic Information Systems*, pages 171–178, 2006.

[5] Alon Efrat and Arnon Amir. Buddy Tracking - Efficient Proximity Detection Among Mobile Friends. In *IEEE International Conference on Computer Communications* (INFOCOM), 2004.

[6] Bugra Gedik and Ling Liu. Location Privacy in Mobile Systems: A Personalized Anonymization Model. In *Proceedings of the 25th International Conference on Distributed Computing Systems*, pages 620–629, 2005.

[7] Gabriel Ghinita, Panos Kalnis, and Spiros Skiadopoulos. PRIVE: Anonymous Location-based Queries in Distributed Mobile Systems. In *Proceedings of the 16th International Conference on World Wide Web* (WWW), pages 371–380, 2007.

[8] Marco Gruteser and Dirk Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of the First International Conference on Mobile Systems, Applications, and Services* (MobiSys), 2003.

[9] Haibo Hu, Jianliang Xu, Wing Sing Wong, Baihua Zheng, Dik Lun Lee, Wang-Chien Lee. Proactive Caching for Spatial Queries in Mobile Environments. In *Proceedings of the 21st International Conference on Data Engineering* (ICDE), pages 403-414, 2005.

[10] Wei-Shinn Ku, Roger Zimmermann, Wen-Chih Peng, and Sushama Shroff. Privacy Protected Query Processing on Spatial Networks. In *Proceedings of the 23rd International Conference on Data Engineering Workshops*, pages 215–220, 2007.

[11] Wei-Shinn Ku, Roger Zimmermann, Haixun Wang. Location-based Spatial Queries with Data Sharing in Wireless Broadcast Environments. In *Proceedings of the 23rd International Conference on Data Engineering* (ICDE), pages 1355-1359, 2007.

[12] Mohamed F. Mokbel, Chi-Yin Chow, and Walid G. Aref. The New Casper: Query Processing for Location Services without Compromising Privacy. In *Proceedings of the 32nd International Conference on Very Large Data Bases* (VLDB), pages 763–774, 2006.

[13] Kyriakos Mouratidis, Panos Kalnis, Gabriel Ghinita and Dimitris Papadias. Preventing Location-Based Identity Inference in Anonymous Spatial Queries. *IEEE Trans. Knowl. Data Eng.*, 2007.

[14] Sarah Spiekermann. General Aspects of Location Based Services. *Location-Based Services*, pages 9–26. 2004.

[15] Latanya Sweene. k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.