Research Article

A Privacy-Preserving Location-Based System for Continuous Spatial Queries

Doohee Song and Kwangjin Park

Information Communication Engineering, Wonkwang University, Iksan-shi, Republic of Korea

Correspondence should be addressed to Kwangjin Park; kjpark@wku.ac.kr

Received 21 June 2016; Accepted 25 September 2016

Academic Editor: Chang Xu

Copyright © 2016 D. Song and K. Park. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

K-anonymization generated a cloaked region (CR) that was K-anonymous; that is, the query issuer was indistinguishable from K − 1 other users (nearest neighbors) within the CR. This reduced the probability of the query issuer’s location being exposed to untrusted parties (1/K). However, location cloaking is vulnerable to query tracking attacks, wherein the adversary can infer the query issuer by comparing the two regions in continuous LBS queries. This paper proposes a novel location cloaking method to resist this attack. The target systems of the proposed method are road networks where the mobile clients’ trajectories are fixed (the road network is preknown and fixed, instead of the trajectories), such as subways, railways, and highways. The proposed method, called adaptive-fixed K-anonymization (A-K_F), takes this issue into account and generates smaller CRs without compromising the privacy of the query issuer’s location. Our results show that the proposed A-K_F method outperforms previous location cloaking methods.

1. Introduction

With the growth of location-based services (LBS) in mobile computing, many businesses are interested in analyzing user location data to better understand patterns and relationships. For example, social marketing relying on social network services takes the form of coupons or advertising directed at customers based on their current locations. In general, mobile clients must expose their exact location information to an LBS provider before receiving their desired services. The location of a mobile client can be obtained via a variety of outdoor and indoor positioning technologies (e.g., Global Positioning System and Wi-Fi). LBS include services to identify the location of a person or object, such as the nearest point of interest (POI) or the whereabouts of a friend or employee. Typical LBS applications include road navigation and vehicle tracking services [1–4]. As LBS have become more numerous and diverse, user privacy violations have become more commonplace. Unfortunately, laws and regulations regarding LBS and location privacy have tended to become less rigorous. This paper proposes a technical approach for location data protection in LBS. For example, when a user sends a continuous K-anonymity query, the number of K − 1 clients may change the expected user. Thus, a user’s ID will be exposed by the service provider. In other words, the service provider can store the information from user’s query contents and cloaked regions (CRs) from client. Therefore, we propose a novel algorithm to protect users’ query contents and CRs (trajectory). In this scheme, if random K_F were to travel in opposite directions from the user, the CRs would increase, at which point a service provider may search many POIs.

Much research has been done on protecting a user’s location. Yi et al. [5] introduced a method for protecting various categories of data. The first categories were information access control, mix-zone, and K-anonymity. To function, this method required an anonymization server; a trusted server, such as middleware, that functioned as an intermediary between the client and the LBS server; and every client to be stored in the anonymization server. Alternatively, the PIR technique required more than one server (K ≥ 2).
We propose protecting users’ locations using a K-anonymity technique. The K-anonymity functions as follows.

Location cloaking blurs a user’s location into a CR that satisfies the privacy parameter (the K-anonymity metric) specified by the user at query time. Location cloaking has attracted a tremendous amount of research as a solution to protect user privacy in LBS. Previous location cloaking methods perform K-anonymization (i.e., identification of K-anonymous users) at the moment that a user issues a query with K-anonymity [6–11].

Figure 1 illustrates an example of 4-anonymization. The anonymization server, a trusted third party that functions as an intermediary between the client and the LBS server, identifies a CR that satisfies the 4-anonymity requirement. This enables the query issuer to have the query result without disclosing his or her exact location to the LBS server. K-anonymization generates a CR that is K-anonymous; that is, the query issuer is indistinguishable from K – 1 other users (nearest neighbors) within the CR. This reduces the probability of the query issuer’s location being exposed to untrusted parties to 1/K. However, location cloaking is vulnerable to query tracking attacks, and a query issuer is not safe when launching continuous LBS queries. For example, if a client issues two queries at times t – 1 and t with corresponding CRs, it is easy for an adversary to compare these two regions to find the query issuer [12–15].

This paper proposes a novel location cloaking method called A-KF that resists query tracking attacks. This proposed method can generate minimized CRs while protecting the location and trajectory privacy of the query issuer.

The contributions of this paper are as follows.

(i) A systematic model prevents both the query contents and CRs (trajectory) from being exposed to continuous spatial queries, because the query issuer is indistinguishable from KF.

(ii) The proposal of an effective anonymization method, A-KF, can reduce CRs within KF and resist query tracking attacks (refer to Section 4). Previous location cloaking methods [15, 16] perform K-anonymization at each moment, whereas the proposed method prevents a query’s trajectory from KF.

(iii) The demonstration of the performance of the proposed method is presented in a variety of settings.

The rest of the paper is organized as follows. Section 2 reviews existing works on location anonymity. Section 3 introduces the problem statement, and Section 4 presents the system model and algorithms for the proposed method. In Section 5, the results of the experiment are presented. Finally, Section 6 concludes the paper.

The terms frequently used in this paper are defined in Definition of Terms Section.

2. Related Work

2.1. Issues Related to Location Privacy Protection. Today’s mobile devices, typically smartphones, enable users to gain access to various LBS that provide dynamic content based on the user’s location. In LBS, the transmission and sharing of user location data are necessary, and such data can be analyzed by third parties for various purposes. For example, one can infer sensitive private information about a person’s health conditions or lifestyle by analyzing his or her whereabouts, length of stay, and movement patterns. Analyzing user locations along with other personal information such as credit card details allows for the creation of more sophisticated and precise user information, which also gives rise to privacy and safety concerns. Hence, businesses and government organizations have made numerous efforts to protect location privacy. However, mandatory controls and regulatory standards that determine the priority between protection of location privacy and development of LBS and other location-based technologies are still lacking; therefore, there are currently no clear and objective criteria regarding this issue [17–20].

2.2. Research Trends. Among various techniques that aim to protect the privacy of LBS users, a dummy is created when the mobile user queries the LBS, during which he or she sends many random locations to the LBS provider to obfuscate his or her location. However, the dummy is not derived from real clients. Thus, we cannot compare our method with the dummy method [21].

Private information retrieval (PIR) allows a user to retrieve a record from servers. To do so, PIR needs more than one server (K ≥ 2). Therefore, this technique cannot be compared with our technique [22–24].

Location cloaking based on K-anonymity predominates and a great deal of research has been conducted on this technique [6–15, 25–30].

Figure 1 presents an example of 4-anonymization. In this example, the minimum CR that satisfies the 4-anonymity requirement is outlined by a red rectangle (the CR contains 4-anonymous clients Cq, C, C, , and C2). One problem this presents is that the size of the CR can increase when all K clients are kept in the CR after they are selected. To address this problem, a method that forms a CR with K – 1 clients...
that are nearest to the query issuer at a given time has been suggested. However, this method is vulnerable to query tracking attacks. It is very likely that the initial CR members other than the query issuer are updated in continuous queries. The adversary can easily guess the authentic query issuer by monitoring the CRs at different time points, and the one that constantly remains in the CRs is the query issuer (e.g., \( C_q, C_1, C_2, C_3 \rightarrow C_q, C_3, C_10, C_12 \)).

Solutions have been proposed to resolve this problem. In [7], \( K-1 \) clients are found in proximity to the query issuer and a temporary CR is set that is twice as large as the initially calculated CR. In this method, the anonymization server must calculate the movement paths of all the clients, which increases the computational cost. Additionally, the accuracy of query results might be low due to the use of a movement probability matrix.

### 3. Problem Statement

This section presents the definitions for the proposed A-\( K_F \) method. Previous location anonymization methods have experienced location privacy threats related to continuous queries, as depicted in Figure 2. A-\( K_F \) method proposed in this paper is designed to solve this problem. The terms and variables frequently used in the proposed method are summarized in Definition of Terms Section.

**Definition 1.** A given set of clients, \( C \), includes \( C_N \) (\( C_N \geq K \)). That is, \( C \ni C_N \ni A-K_F \) (\( C_N \): candidate set of \( K \) close to a querier).

**Definition 2.** A-\( K_F \) = \( K \) and A-\( K_F \) = \( K_F + K_{NF} \) (refer to Definition of Terms Section).

The criteria for selecting A-\( K_F \) are as follows: (1) \( C_N \) denotes the number of clients that are searched by \( C_q \) (\( C_N = C_N + C_q \)) and, (2) among \( C_N \) members, those with the smallest distance between the origin and the destination are chosen as \( K_F \) members (\( K_F \ni C_q \)).

**Definition 3.** Under fixed-\( K \), \( K_F = K \). \( K_F \) is greater than or equal to 1 and less than or equal to \( K \) (\( K \geq K_F \geq 1 \)) (refer to Definition of Terms Section).

**Definition 4.** A set of clients, \( C \), includes \( C_q \) and \( C_N = \{C_1, C_2, \ldots, C_{n-1}, C_n, C_q\} \) (\( C_N \) must be greater than \( K \) and can include all the clients except \( C_q \)).

In Figure 1, \( C_N = \{C_1, C_2, \ldots, C_{11}, C_{12}, C_q\} \). In Figure 2, when \( C_N = 4 \) and \( K_F = 2 \), Algorithm 1 computes that \( K_F \) member other than \( C_q \) is \( C_2 \); that is, \( K_F = \{C_q, C_2\} \).

Figure 2 depicts the problem in which the location of the querying client can be exposed in continuous queries with \( K \)-anonymity. Figure 1 shows 4-anonymization at time \( t = 0 \), and Figures 2(a) and 2(b) present 4-anonymization.
Input: query issuer's current location and destination, CR chosen by the query issuer, K and K_F (number of clients for location anonymity), content of the query

Output: K-anonymous clients in a minimum CR, K_F member clients

1. Check the CR chosen by the query issuer using the anonymizer.
2. Calculate the minimum distance (K, K_F, K_{NF}, K[i] = null).
3. If |K| < K_F + K_{NF} - 1 then return 0
4. e ← 0
5. while e ≠ |K|
6. e ← |K|
7. K[i] = e;
8. i + 1;
9. if (i < K - 2);
10. continuous;
11. else end if;
12. end while;
13. if (periodically measure the CRs of dist(q, K[i]) and sort the CRs in ascending order (i.e., from the smallest to the largest))
14. return K[i]; (ith is from 1 to K)

Algorithm 2: Anonymizer’s query processing.

at t = 1 and t = 2, respectively (CRs are represented by a rectangle). This example indicates that the location of C_q as well as its trajectory can be disclosed over time. When the CR is increased to reduce the probability of revealing a query issuer’s location, it may be necessary for the LBS server to send more objects corresponding to the increased CR to the query issuer, which increases the communication and computational costs.

Alternatively, lowering K of the K-anonymity requirement decreases the size of the CR but increases the probability of exposing a query issuer’s location to third parties. The proposed method assumes a road network environment where the client movement trajectories are fixed (e.g., subway, railway, and highway networks). Suppose that the clients nearest to the query issuer are selected as fixed CR members in such an environment. If clients that move in directions opposite to a query issuer are bounded in a CR along with the query issuer, the size of the CR increases dramatically over time.

4. Protection of User Location and Trajectory Privacy

4.1. System Model. In Figure 1, C_q issues a query with 4-anonymity (i.e., K = 4). The anonymizer (Algorithm 2), a location anonymization server in a LBS system that knows the locations of clients and generates blurred locations for them, checks the locations of all clients C_i and generates a minimum CR that contains 4 clients including C_q (see the solid rectangle in Figure 1).

The anonymization server then sends queries with CRs to the LBS server that stores information about the queried objects.

In the proposed method, the query issuer first determines the destination, K (nonfixed K (K_{NF}) + fixed K (K_F)), and C_N (the number of clients for selecting K_F members). The query issuer then issues a nearest-neighbor query (C_N ⊃ K ⊃ K_F ⊃ C_q). The anonymizer checks the current locations and the destinations of the clients. As C_N increases, the computational cost increases, but the size of the CRs decreases.

When C_q moves from the origin to the destination, the clients are sorted so that those nearest to C_q are listed first (\sum_{i=0}^{t=2} dist(C_0, C_i)). Subsequently, the top K_F clients in the sorted list are selected as K_F members (K_F is given by the query issuer). This procedure is represented in Algorithm 1.

4.2. Adaptive-Fixed K-Anonymization (A-K_F). In Figure 3(a), C_q’s nearest neighbors are C_1, C_2, and C_3 when C_N = 4. At t = 0, the client nearest to C_q is C_4, the second nearest is C_1, and the third nearest is C_2. Thus, the sorted order of C_N member clients is \{C_q, C_4, C_1, C_3\}. Figures 3(b) and 3(c) show that the movement of clients will cause changes in the distance between C_N members and C_q. Suppose that the time period between the moment that a client issues a query and the moment that the client arrives at the destination is T. T is divided by n (T/n = t), and the distances of C_q (C_N ⊃ C_q) to C_q are calculated at every t second. In Figures 3(a)–3(c), the movement of C_q is depicted, and the sorted order of C_N members is changed to \{C_2, C_4, C_1\} according to the updated distance, \sum_{i=0}^{t=n} dist(C_q, C_i). K_F members are C_4 and C_3 when K_F = 2. And Figure 4 shows that K_F members are C_q, C_2, C_4, and C_1 when K_F = 4.

In Figures 4(a)–4(c), the movement of C_q is depicted, and the sorted order of C_N members is changed to \{C_2, C_4, C_1\} according to the updated area, \sum_{i=0}^{t=n} area(C_q, C_i). The proposed method selects C_N and K_F based on the query issuer’s request. To decrease the amount of information to be transmitted while preserving location privacy, the anonymizer generates a minimum bounding rectangle (MBR) that includes the CRs.
5. Performance Evaluation

Figure 5 shows the possible directions of movement of a client. Initially, a query issuer $C_q$ can move in one of eight different directions ($w = 0$ through $w = 7$). After $t + 1$, $C_q$ moves to $w = 2$. It is assumed that $C_q$ can move in the directions at a ±45° angle from the current direction of movement. Here, the client moved in a set direction. This assumption was made to obscure a client’s movement pattern because, if a client was moving back and forth repeatedly, there might be a discernible location.

5.1. Experimental Settings. This section evaluates the performance of the proposed method in comparison with that of the existing AMV method. The experiments were carried out using a computer with a 2.9 GHz processor, 4 GB memory, and Microsoft visual C++ 6.0. It was assumed that LBS clients are moving and that they are evenly distributed throughout the grid cells. The dataset comprised simulated uniform data. The length of a single grid cell was assumed to be 1 meter (m), and time ($t$) was in seconds. Our proposed method was compared with all fixed $K$ [16] and nonfixed $K$ [15] methods. We assumed the service provider and anonymization server in our experimental environment [15]. Table 1 describes the parameter settings for the experiments.

5.2. Experiment Results. Figure 6 shows how the sizes of the CRs associated with fixed anonymous clients ($K_F$) change over time. When $K = 5$, $K_F$ is 2. The fixed-$K$ method determines which five clients are nearest to the query issuer ($K_F = 5$). A-$K_F$ method generates a CR for the query issuer when $C_N = 10$ and $K_F = 2$ (in which case a reconfiguration is needed). Compared to the AMV method, the sizes of the CRs created by A-$K_F$ method are 12% lower. The proposed A-$K_F$ method can generate smaller CRs than the AMV method because A-$K_F$ selects optimal $K_F$ clients by monitoring $C_N$ members’ movements and updating the distances of moving clients to the query issuer $C_q$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of grid cells</td>
<td>200 x 200</td>
</tr>
<tr>
<td>Number of clients</td>
<td>4,000</td>
</tr>
<tr>
<td>Client’s movement radius (number of grids)</td>
<td>1–3</td>
</tr>
<tr>
<td>$K$-anonymity ($K$)</td>
<td>10</td>
</tr>
<tr>
<td>Update time (sec; $t$)</td>
<td>0, 2, 4, 6</td>
</tr>
<tr>
<td>The average number of experiments</td>
<td>10,000</td>
</tr>
</tbody>
</table>
Figure 5: Directions of movement of a mobile client.

Figure 6: The sizes of the CRs over time.

Figure 7: The sizes of the CRs with regard to changes in $C_N$.

Figure 8 shows the sizes of the CRs with regard to changes in $K_F$ (the number of fixed anonymous clients). When $K = 5$, $K_F$ can be 2, 3, or 4. The fixed-$K$ method selects the five clients nearest to the query issuer as $K_F$ members. A-$K_F$ method generates the CRs for the query issuer with $C_N = 10$ and $K_F = 2$.

Figure 9 shows the sizes of the CRs, which vary according to the velocity ($V$) of the clients’ movement. It is assumed that the clients’ speeds are 1 m, 3 m, and 5 m per second. The CRs created by the proposed A-$K_F$ method when $V = 1$ are smaller than those created when $V = 3$ by 29.9%, and they are smaller than those created when $V = 5$ by 42.7%. That is, the size of the CR increases as $V$ increases.

Figure 10 presents the number of queried objects, which changes according to changes in $K_F$. As shown in Figure 8, the sizes of the CRs increase as $K_F$ increases. This implies that the search area for the query issuer increases as $K_F$ increases, which, in turn, increases the number of objects to be searched.

Figure 11 presents the sizes of the CRs in connection with changes in the number of LBS clients. The sizes of the CRs decrease as the number of clients increases. This is because LBS clients become more densely populated in a grid map.
Figure 8: The sizes of the CRs with regard to changes in $K_F$.

Figure 9: The sizes of the CRs with regard to changes in velocity ($V$).

Figure 10: The numbers of queried objects with regard to changes in $K_F$.

Figure 11: The sizes of the CRs with regard to the number of clients.

Figure 12 shows how $K$ (the anonymity degree) changes under the three different anonymization methods over time. At $t = 0$, both the AMV and A-$K_F$ methods have the same $K$ ($K = 10$). $C_N$ is fixed at 20, and $K_F$ is fixed at 5. As time passes, the anonymity level $K$ gradually decreases, except for under the fixed-$K$ method. $K$ drops to nearly 1 under the AMV method, and $K$ decreases to 4 as $t$ increases under A-$K_F$ method.

Figure 13 presents the probability of protecting a query issuer’s location at different time points $t$. As described in Figure 12, $K$ decreases over time in the AMV and A-$K_F$ methods and is unable to meet the requested anonymity metric of 10. This increases the probability of revealing a query issuer's location to third parties.

6. Conclusion

This paper stated a drawback of existing $K$-anonymous location cloaking methods that can occur in continuous LBS queries and proposed A-$K_F$ method, which is effective in preventing this problem. The proposed A-$K_F$ method determines $K$ based on the query issuer’s request, increasing the query issuer’s satisfaction and decreasing the workload in the anonymization server. The proposed method can achieve smaller CRs than existing location anonymization methods while preserving $K$-anonymity.

In the future, the movement information of mobile LBS clients will be analyzed, and the proposed A-$K_F$ method will be further refined to query requests for time and conditions.
Additionally, algorithms to reduce errors that occur in the process of a movement information analysis will be studied.

**Definition of Terms**

- **CR:** A cloaked region
- **C_q:** i_q, client
- **C_q:** The query issuer (the querying client)
- **K:** The anonymity metric specified by the client; number of anonymous clients satisfying the K-anonymity metric
  \( K = K_{NF} + K_F \)
- **K_F:** The number of anonymous clients that are fixed in the initial K-anonymization process
- **K_{NF}:** The number of nonfixed anonymous clients
  \( (K_{NF} = K - K_F) \)
- **T:** Total query processing time \( (t_0 + t_1 + \cdots + t_{n-1} + t_n) \)
- **Fixed-\( K \):** The method in which all the K-anonymous clients are fixed since the initial K-anonymization process
- **A-\( K_F \):** The method in which only \( K_F \) clients are fixed since the initial K-anonymization process
- **AMV:** The method in which K-anonymous clients are not fixed in K-anonymization.

**Competing Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.

**Acknowledgments**

This paper was supported by Wonkwang University in 2016.

**References**


