L2P2: Location-aware Location Privacy Protection for Location-based Services

Yu Wang* Dingbang Xu† Xiao He‡ Chao Zhang‡ Fan Li‡ Bin Xu§
*Department of Computer Science, University of North Carolina at Charlotte, Charlotte, North Carolina, USA
†Department of Computer Science, Governors State University, University Park, Illinois, USA
‡School of Computer Science, Beijing Institute of Technology, Beijing, China
§Department of Computer Science and Technology, Tsinghua University, Beijing, China

***Abstract*—Location privacy has been a serious concern for mobile users who use location-based services provided by the third-party provider via mobile networks. Recently, there have been tremendous efforts on developing new anonymity or obfuscation techniques to protect location privacy of mobile users. Though effective in certain scenarios, these existing techniques usually assume that a user has a *constant* privacy requirement along spatial and/or temporal dimensions, which may not be true in real-life scenarios. In this paper, we introduce a new location privacy problem: *Location-aware Location Privacy Protection* (L2P2) problem, where users can define dynamic and diverse privacy requirements for different locations. The goal of the L2P2 problem is to find the smallest cloaking area for each location request so that diverse privacy requirements over spatial and/or temporal dimensions are satisfied for each user. In this paper, we formalize two versions of the L2P2 problem, and propose several efficient heuristics to provide such location-aware location privacy protection for mobile users. Through multiple simulations on a large data set of trajectories for one thousand mobile users, we confirm the effectiveness and efficiency of the proposed L2P2 algorithms.**

***Index Terms*—Location privacy, k-anonymity, cloaking, location based service, mobile networks**

I. Introduction

Recent extensive media reports about Google and Apple's smart phones (i.e., Android phones and iPhones) being able to store and gather users' location data have attracted national attention, and even the lawmakers from the congress expressed their concerns [1]. The privacy concerns from mobile users on location data have significant impact on usage and development of location based service applications and systems.

Location based service (LBS) is a type of service where the information is provided based on a mobile user's geographical location. In recent years, mobile devices with positioning capabilities have been extensively used in our daily lives. Examples of such devices are smart phones and tablets. These devices provide great convenience to many users, however, privacy issues have been a big concern when location data has to leave local devices to a third party for LBS. The broadcast nature of wireless networks usually makes it challenging to protect a user's privacy including identities and locations. Location data is sensitive since it can reveal where you live and work, where you go for movies and dinner and even if you stay at someone else's house. As defined by Beresford and Stajano in [2], *location privacy* is "the ability to prevent other parties from learning one's current or past location." In an example scenario of LBS application, a mobile user may issue an LBS query "where is the closest Chase bank?" From privacy protection perspective, this user may not want to disclose his identity, nor his exact location information, especially when the service is sensitive, but he still wants to get the query resolved by LBS providers. Therefore, location privacy has been a great challenge for location based services in mobile environment.

Over the past a few years, many different approaches have been proposed to protect a user's location privacy, for example, Gruteser and Grunwald's k-anonymity based approach [3], where a cloaking area in which at least k users are present is provided, and Xu and Cai's entropy based approach [4], where a cloaking area is decided based on multiple users' footprints in the area. General speaking, the approaches to protecting location privacy can be roughly divided into four categories: (1) regulatory approaches, (2) privacy policy based approaches, (3) anonymity based approaches, and (4) obfuscation based approaches. Anonymity based approaches separate users' identities from their location information. E.g., a user's identity may be replaced with pseudonyms [2]. Obfuscation based approaches downgrade the quality of users' location information to protect location privacy. E.g., a cloaking area (instead of the user's exact location) may be reported based on multiple users' footprints in the region [4]. See Section II for more detailed review. In this paper, we focus on designing cloaking area based approaches.

Though effective in certain scenarios, these existing techniques usually assume that a user has a *constant* privacy requirement, which may not be true in certain real-life scenarios. In real world, different locations or different types of LBS requests may reveal different private information of the mobile user, thus the user may have diverse privacy requirements over various locations. For example, a user may have higher privacy requirement when he is in a hospital compared with when he is in a shopping mall. Therefore, addressing a user's *diverse*

The work of Y. Wang is supported in part by the US National Science Foundation (NSF) under Grant No. CNS-0915331 and CNS-1050398, and by Tsinghua National Laboratory for Information Science and Technology (TNList). The work of F. Li is supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 60903151 and Beijing Natural Science Foundation under Grant No. 4122070. The work of B. Xu is supported in part by the NSFC under Grant No. 61170212.

and *dynamic* privacy requirements which may depend on his location would be necessary for location privacy protection.

In this paper, we introduce and investigate a new location privacy problem: *Location-aware Location Privacy Protection* (L2P2) problem, which addresses the *dynamic* and *diverse* privacy requirements from mobile users. We assume that a mobile user can have diverse and dynamic location privacy requirements, depending on where or when the user requests a location based service. Each LBS request is associated with a specific privacy requirement, and we generate a cloaking area to fulfill this requirement. Notice that privacy requirements can be expressed through either k-*anonymity* or *entropy* based metrics, and our approach can accommodate both. To be more specific, for k-anonymity based metric, if the privacy requirement is k, the cloaking area should have at least k users including the user makes the request; for entropy based metric, the footprint frequencies from multiple users in the cloaking area can be used to compute a privacy value, and this value should be no less than the requirement. We will give the formal definitions for both metrics in Section III.

Considering that mobile users can have a sequence of LBS requests to make the scenarios more complicated, to tackle this, we further define two versions of L2P2 problems: basic L2P2 and enhanced L2P2. In basic L2P2, each user request can be seen as an independent event. To generate a cloaking area, all users in the area are considered during the calculation of privacy values, and the privacy value provided by the cloaking area should be equal to or larger than the requirement. We provide a simple cloaking area generation algorithm to find the minimum-sized cloaking areas for basic L2P2. In enhanced L2P2, since a sequence of LBS requests will generate a sequence of cloaking areas, we choose a conservative approach in favor of privacy protection. To be more specific, we enforce a much stronger restriction, where only the common users among this sequence of cloaking areas are considered for computing privacy values. This restriction comes from the possibility that an attacker may be able to shorten the list of possible users through discovering the common users in a sequence of cloaking areas. Existing cloaking methods do not work for this problem, since the privacy values of the sequence of cloaking areas are not independent any more. To address this enhanced L2P2 problem, we propose four different heuristics to generate the cloaking areas in polynomial time. All proposed algorithms can provide diverse privacy protection for multiple users over both temporal and spatial domains to fulfill the mobile scenarios. In order to evaluate our approaches, we also conduct extensive simulations over a large set of mobile user location data (generated by a network-based traffic generator [5]). Results show that our methods can fulfill diverse privacy requirements with slight downgrade of the quality of original location data.

Our major contributions are summarized as follows:

- We introduce a new concept of *Location-aware Location Privacy Protection* (L2P2), which allows mobile users to define dynamic and diverse privacy requirements.
- We formalize an optimization problem for cloaking area generation, called L2P2 problem, which aims to find the cloaking areas with minimum sizes such that diverse privacy requirements are still satisfied.
- We propose a set of polynomial-time heuristics to address basic and enhanced L2P2 problems.
- We conduct extensive simulations on a large location data set generated by a traffic generator to evaluate all proposed methods.

The remainder of this paper is organized as follows. Section II discusses related work on location privacy protection. Section III introduces the model and assumptions used in our study, and formally defines the location-aware location privacy protection problem. To address this challenging problem, a set of heuristic algorithms are proposed in Section IV. Section V presents simulation results of our proposed algorithms and Section VI concludes this paper.

II. Related Work

To protect location privacy, many approaches have been proposed [2]–[4], [6]–[15]. For example, Gruteser and Grunwald [3] propose to apply the concept of k-*anonymous*, where for a subject, a cloaking area including the location of this subject and at least $k-1$ other subjects is generated for location based services. According to [16], the strategies in protecting location privacy can be divided into four categories: (1) regulatory approaches, (2) privacy policy based approaches, (3) anonymity based approaches, and (4) obfuscation based approaches, where regulatory approaches are those related to making rules, regulations, and legislations to protect privacy, privacy policies are those mechanisms that can prohibit certain misuse of location data, anonymity based approaches separate users' identities from their location information, and obfuscation based approaches downgrade the quality of users' location information. In this section, we follow this classification and give an overview of the approaches in Categories (3) and (4).

Anonymity based approaches. Several approaches belong to this category. Beresford and Stajano [2] propose a framework of frequently changing a user's identities through pseudonyms. Moreover, the concept of mix zones in anonymous communication has also been applied to provide location privacy. To measure the location privacy, two metrics, where one is based on entropy and the other is based on anonymity sets, are also proposed in [2]. Another approach falling in this category is the aforementioned approach proposed by Gruteser and Grunwald [3]. In this approach, a user's location will be reported as a two-dimensional spatial cloaking area where at least $k-1$ other users are also in the same area. A quadtree based cloaking algorithm has been designed, and the size of the anonymity set k is used to measure the degree of anonymity. In Bettini et al.'s approach [6], location-based quasi-identifiers are defined, and based on the concept of historical k-anonymity, a formal framework has been proposed to see the potential risk of location information leading to the identity disclose. In Kido's approach [7], dummy location data has been generated and mixed with real location data, so that it is difficult for the LBS providers to differentiate them.

Obfuscation based approaches. This category includes several approaches. It is also possible that some approaches may belong to both anonymity and obfuscation categories, for example, the approach [3]. In a feeling based approach proposed by Xu and Cai [4], a user's privacy requirement is defined through specifying a public region (e.g., a restaurant), and the public region's *popularity* is computed through an entropy based approach regarding the region's users (i.e., visitors) and footprints. A user's location may be disclosed in the form of a cloaking box, only if the disclosed cloaking box's popularity is equal to or greater than that of the pre-specified public region. Xu and Cai further propose the concept of *P-Popular Trajectory*, which is related to the temporal and spatial aspects of a mobile (moving) user, and propose quadtree based algorithms to select cloaking sets and compute cloaking boxes. In this approach, an entropy based metric has been proposed to measure the location privacy. In another approach proposed by Duckham and Kulik [8], a formal obfuscation model (with weighted graph) is presented, and a negotiation algorithm (between users and location based service providers) is designed. The size of the obfuscation set is used to measure the location privacy in this approach.

There are also other location privacy protection techniques not discussed here, please refer to [16], [17] for details. In this paper, we focus on designing cloaking area based approaches to address *dynamic* and *diverse* privacy requirements.

III. L2P2: Location-aware Location Privacy Protection

In this section, we will introduce the motivation of location-aware location privacy protection, and formally define the related optimization problems.

A. Motivation

All existing techniques on location privacy protection [2]–[4], [6]–[15] usually assume that a user has a constant privacy requirement (a k-value in k-anonymity models or a given region to define the popularity requirement in entropy-based models) along spatial and/or temporal dimensions. This assumption may not always be true in real world, since a user may have diverse location privacy requirements, depending on where or when a user requests a service. For example, in the spatial dimension, Bob may have higher privacy requirement when he is in a hospital compared with the case when he is in a shopping mall; and in the temporal dimension, David may have higher privacy requirement in a workday morning compared with the case in a weekend afternoon due to the specialty of his job. Moreover, spatial and temporal dimensions usually are mixed together and may result in more complicated privacy requirements. Therefore, in this paper, we will propose a new model that can formally represent these requirements, and explore novel approaches to fulfill these diverse location-aware privacy requirements.

B. Location-Based Service Model

We assume a general model for location-based services (e.g., in [3], [4]), where there are three critical components: *mobile users*, trusted *location anonymization server*, and *location based service providers*. See Fig. 1 for illustration. In this model, a mobile user u_i sends a location based service request to the trusted anonymization server, which includes his location data (x, y), timestamp information t, as well as his privacy requirement r. Hereafter, we use (u_i, x, y, t, r) to represent such request. During this step, user authentication and message encryption can be performed to provide security protection. After the anonymization server gets the request message, it will perform location anonymization (generating a clocking area c which covers the user's location (x, y)) to provide location privacy protection, then the anonymized location information (the cloaking area c) will be sent to the location based service providers for the services. In this model, *location privacy* of a mobile user refers to his exact location. By using the generated cloaking area instead, the anonymization server hides the user's exact location from possible adversary and thus protect his privacy. Our focus of this paper is *how to perform location anonymization to fulfill location-aware privacy requirements at the location anonymization server*. Notice that the location anonymization server (not necessary a centralized server but a group of distributed servers) has the location and timestamp information of all requests from all users, and it uses a footprint database $\mathcal{F}$ (each request will leave a footprint (u_i, x, y, t) in the database) to save all historical data. Here, we also assume that the temporal domain is divided into equal time intervals.

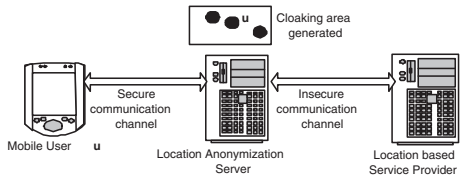


Fig. 1. **LBS model:** Location anonymization server performs location anonymization to provide privacy protection for mobile users.

Similar to approaches in [3], [4], we use a quadtree T [18] to partition the spatial domain recursively into cells. A cell at level l is partitioned into four smaller cells in level $l+1$. The partitioning stops when the size of cells becomes less than a threshold. Assume that T has L levels. Fig. 2(a) illustrates an example of such a quadtree. Let jth cell at level l be $c(j, l)$ and its area be $a(c(j, l))$ or $a(j, l)$. We assume that all generated cloaking areas by location anonymization server are cells in the quadtree T. In other words, for an LBS request at position (x, y) which is contained at $c(j, L)$, all possible cloaking areas of this request are $c(j, L)$ and its ancestors in T, as shown in Fig. 2(b). Obviously, smaller cells (at higher level) provide better quality of location data but with potentially smaller privacy values because less users may be involved.

C. Location-aware Location Privacy

The user privacy requirement r included in the LBS request is given by the user u_i, and it could be dynamic and diverse over both spatial and temporal dimensions. In other words,

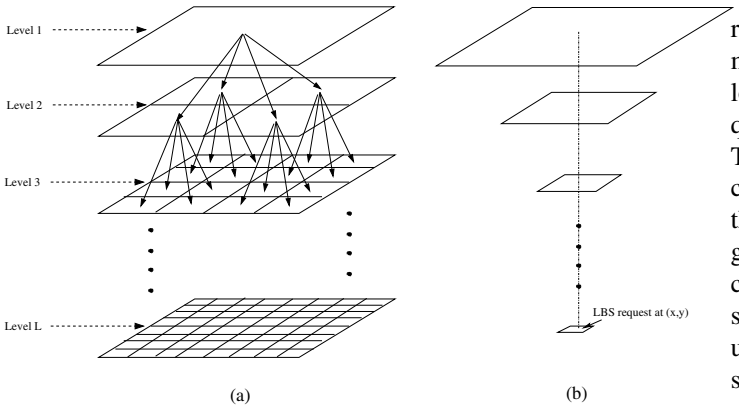


Fig. 2. **Quadtree:** (a) the network is recursively partitioned into a quadtree T; (b) all possible cloaking areas of an LBS request happened in cell $c(j, L)$ are $c(j, L)$ and its ancestors.

it could be treated as a function of (u_i, x, y, t). Thus, we call our model location-aware location privacy. As we discussed in Section II, mainly there are two models to measure location privacy protection: *k-anonymity model* [3] and *entropy-based model* [4].

Definition 1: **k-anonymity privacy ([3])**: Let c represent a cloaking area and $U(c) = \{u_1, u_2, \cdots, u_m\}$ represent the set of users whose footprints are in c at time t. The k-anonymity privacy value $p_k(c)$ of c is the size of $U(c)$, i.e., $p_k(c) = m$.

Definition 2: **entropy-based privacy ([4])**: Let c denote a cloaking area and $U(c) = \{u_1, u_2, \cdots, u_m\}$ denote the set of users whose footprints are in c based on a footprint database $\mathcal{F}$. Let n_i represent the number of u_i's footprints in c, and $N = \sum_{i=1}^{m} n_i$ represent the total number of footprints from user set $U(c)$. The entropy of c is $E(c) = -\sum_{i=1}^{m} \frac{n_i}{N} \log \frac{n_i}{N}$, and the popularity privacy value of c is $p_e(c) = 2^{E(c)}$.

In both models, for each LBS request (u_i, x, y, t, r), the goal of location privacy protection is to find a cloaking area c such that its privacy value ($p_k(c)$ or $p_e(c)$) is no less than r. Hereafter, we use $p(c)$ to represent the privacy value of either $p_k(c)$ or $p_e(c)$ for cloaking area c. If we only consider a user subset $U'(c) \subseteq U(c)$ instead of all users in $U(c)$, we can also define privacy value of c with respect to the subset U' accordingly.

Though k-anonymity and entropy-based metrics are used in our study, our proposed cloaking algorithms can adopt any privacy measurement to quantify the location privacy value. Shokri *et al.* [19] recently show that k-anonymity and entropy-based metrics are not correlated with the attacker's success rate, thus may not be perfect metrics for location privacy. They also provide a new measurement tool to quantify location privacy, which could be used by our proposed algorithms instead of k-anonymity and entropy-based measurement.

D. Optimization Problem for L2P2

While the single privacy request is easy to satisfy, the issue becomes more complicated when a user makes a sequence of requests in different locations with different privacy requirements. Now we are ready to formally define the location-aware location privacy protection (L2P2) problem. When a user requests a continuous LBS, it sends a sequence of LBS requests. The location anonymization server generates a sequence of cloaking areas to provide location privacy protection and sends them to the LBS provider. See Fig. 3 for illustration. The generated cloaking area is required to satisfy the following conditions: (1) it contains the user's current location; (2) it should provide enough privacy protection as specified by the user; and (3) it would be as small as possible. We can define such a problem as the following optimization problem.

Definition 3: **Basic Location-aware Location Privacy Protection (Basic L2P2):** Given a quadtree T, the footprint database $\mathcal{F}$, and a sequence of LBS requests from user u in the format of (u, x_i, y_i, t_i, r_i) for $i = 1$ to m, L2P2 aims to generate a sequence of cloaking areas $c_i = c(j_i, l_i)$ (which are cells in T) for $i = 1$ to m such that

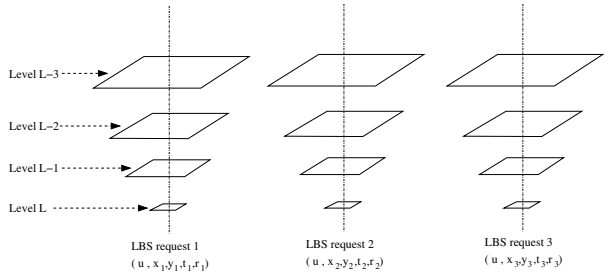
1. each cloaking area $c_i = c(j_i, l_i)$ includes the user's location (x_i, y_i) at t_i.
2. for any cloaking area c_i, its privacy value satisfies the corresponding privacy requirement, i.e., $p(c_i) \geq r_i$.
3. the total area of all cloaking areas $\sum_{i=1}^{m} a(j_i, l_i)$ is minimized.

Notice that in basic L2P2, we assume that each user request r_i $(1 \leq i \leq m)$ is an independent event among the request sequence, so this basic L2P2 problem is easy to address. However, in some cases, simply satisfying basic L2P2 requirements is not enough for privacy protection. This is because that an attacker may be able to shorten the list of possible users through discovering the common users in a sequence of cloaking areas. Therefore, similar to [4], to prevent such attacks, we may want to consider an enhanced version of L2P2, where only the common users (in a sequence of cloaking areas) are considered for privacy value computation inside each cloaking area.

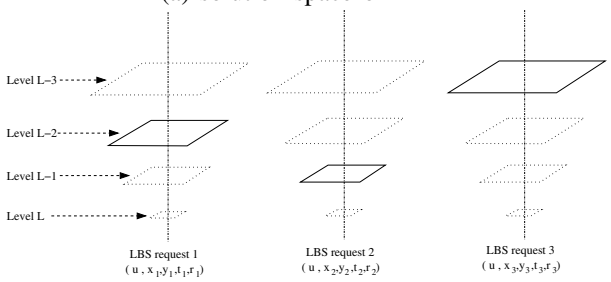
Definition 4: **Enhanced Location-aware Location Privacy Protection (Enhanced L2P2):** Given a quadtree T, the footprint database $\mathcal{F}$, and a sequence of LBS requests from user u in the format of (u, x_i, y_i, t_i, r_i) for $i = 1$ to m, enhanced L2P2 generates a sequence of cloaking areas $c_i = c(j_i, l_i)$ (which are cells in T) for $i = 1$ to m such that

1. each cloaking area $c(j_i, l_i)$ includes the user's location (x_i, y_i) at t_i.
2. for any cloaking area $c(j_i, l_i)$, its privacy value with respect to common user set U' satisfies user's requirement. Here U' is the set of common users among a sequence of cloaking areas, i.e., $U' = \cap_{1 \leq i \leq m} U(c(j_i, l_i))$.
3. the total area of all cloaking areas $\sum_{i=1}^{m} a(j_i, l_i)$ is minimized.

Based on the above definition, we can see that in enhanced L2P2, we must ensure that the privacy value of each cloaking area with respect to the common users is no less than the privacy requirement. This enhances the location privacy protection for mobile users, but makes the problem much more



(a) solution space of L2P2



(b) a possible solution of L2P2

Fig. 3. Example of L2P2 problem for a sequence of three requests. (a) all possible cloaking areas for each request; (b) a possible solution of L2P2 problem in which the privacy values satisfy the privacy requirements.

challenging. *All existing cloaking algorithms do not work for the enhanced L2P2 problem, since the privacy values of the sequence of cloaking areas are not independent any more and moving one cloaking area will affect all others.*

Notice that in enhanced L2P2, when k-anonymous model is used, the privacy level of each generated cloaking area must be at least $\max_{i=1}^{m} r_i$, since the privacy level (i.e., the number of users) is defined over the intersected set of users in the whole sequence. However, when the entropy-based model is used, the privacy level of each generated cloaking area could be less than $\max_{i=1}^{m} r_i$, since the entropy value is based on the distribution of common users' footprints not just the number of users. Therefore, the enhanced L2P2 do fulfill various location privacy requirements.

IV. Our Solutions: L2P2 Algorithms

In this section, we present five different cloaking algorithms to provide location-aware location privacy protection for a mobile user with a sequence of LBS requests. For simplicity, we present our algorithms in offline fashion (with the footprint repository and m LBS requests as their inputs), but all of them can be converted into online algorithms by using the techniques proposed in [4]. We start with a simple algorithm to address basic L2P2 problem, then four more heuristics are proposed for enhanced L2P2 problem.

A. Algorithm for Basic L2P2

For the basic L2P2 problem, it is easy to find the optimal solution within polynomial time. Since each LBS request (u, x_i, y_i, t_i, r_i) is an independent event in the request sequence, we can simply find the best cloaking area for each request separately. For request (u, x_i, y_i, t_i, r_i), we start at the leaf node in quadtree T where the cells are smallest cloaking areas. First, we find the leaf node which contains location of (x_i, y_i) and use it as the initial cloaking area, then traverse the tree heading towards the root node (as shown in Fig. 3(b)) until the privacy value of the current cloaking area fulfills the requirement from the user. This can guarantee that the generated cloaking area is the smallest one satisfying the privacy requirement. Algorithm 1 shows the detail. The time complexity of this algorithm is $O(mL)$.

Algorithm 1 Cloaking Algorithm 1 for Basic L2P2

Input: A quadtree T, a footprint repository $\mathcal{F}$, and a sequence of m LBS requests (u, x_i, y_i, t_i, r_i) (for $i = 1$ to m).
Output: A sequence of m cloaking areas c_i ($i = 1$ to m).

1: **for** $i = 1$ to m **do**
2: Find the leaf node $c(j_i, L)$ in quadtree T which contains position of (x_i, y_i).
3: Let $c_i = c(j_i, L)$.
4: **while** $p(c_i) < r_i$, i.e., the privacy value of c_i does not fulfill the privacy requirement r_i **do**
5: Let c_i be the parent node of c_i in T, i.e., move the cloaking area one level up in tree T towards the root.
6: **end while**
7: **end for**
8: Return $c_1, c_2, \cdots, c_m$.

B. Algorithms for Enhanced L2P2

While the basic L2P2 is easy to solve, the enhanced L2P2 becomes more complicated. The major reason is that only the set of common users in a sequence of cloaking areas would be considered for computing privacy metrics. This can provide better location privacy protection for mobile users, however, it also makes the problem of L2P2 at location anonymization server much more challenging. In enhanced L2P2, whether a cloaking area for request r_i can be satisfied is not independent on other requests in this sequence, because the privacy value of such a cloaking area is calculated with respect to the common users inside all cloaking areas generated from a request sequence. In such situation, moving the cloaking area along one branch of a quadtree for one request will affect the privacy values of cloaking areas at other branches for other requests. In other words, to increase the privacy value of a cloaking area c_i of request r_i, we can either expand the cloaking area of such request by moving it up toward the root node in the quadtree, or expand the cloaking areas of other requests, which may enlarge the common user set. Therefore, how to dynamically and efficiently generate the cloaking areas for a sequence of LBS requests is very challenging. The issues such as what leaf nodes we should start and what stopping criteria we should have require more thorough investigation. We certainly can try a brute force method, which examines all combinations and chooses the best one, but it will lead to the complexity of $O(L^m)$. Thus, it is necessary to have some polynomial heuristics to reduce the complexity.

Next, we propose four different heuristics, denoted by Algorithm 2 to 5, to generate cloaking areas for enhanced L2P2. These heuristics share one basic idea: they all start from initial cloaking areas at bottom of the quadtree T, and iteratively move cloaking areas up along T to increase the privacy values, until all cloaking areas fulfill the user requirements. One key difference among these four heuristics is the order of moving cloaking areas along the quadtree. The first two heuristics move cloaking areas in order while the latter two move cloaking areas greedily based on certain criteria. All algorithms have polynomial complexity of $O(mL)$.

The first algorithm (Algorithm 2) starts with the output of Algorithm 1, since cloaking areas satisfying the privacy requirements of enhanced L2P2 (with respect to common users) must first satisfy the privacy requirements for the corresponding basic L2P2. Then the algorithm first expands the cloaking area c_1 for the 1st request and checks whether the privacy values of all requests are fulfilled. If not, it continues moving the first cloaking area up until it reaches the root node. At this point, if the requirements are not met yet, it begins to move the cloaking area c_2 for the 2nd request. This process goes on until all requests are fulfilled. Note that all requests can always be fulfilled when all cloaking areas become the root node in the quadtree. Fig. 4(a) illustrates the idea and the detailed algorithm is given in Algorithm 2.

Algorithm 2 Cloaking Algorithm 2 for Enhanced L2P2

Input: A quadtree T, a footprint repository $\mathcal{F}$, and a sequence of m LBS requests (u, x_i, y_i, t_i, r_i) (for $i = 1$ to m).
Output: A sequence of m cloaking areas c_i ($i = 1$ to m).

```
1: Run Algorithm 1 for basic L2P2 to initialize c_1, ..., c_m.
2: Let i = 1.
3: while i ≤ m do
4:    Let l = the level of c_i in T.
5:    while l > 1 do
6:       Compute p(c_1), ..., p(c_m) with respect to U', where
         U' is the set of common users in c_1, ..., c_m.
7:       if p(c_1), ..., p(c_m) do not satisfy the privacy re-
         quirements r_1, ..., r_m then
8:          Move c_i to be its direct parent in T.
9:          l = l − 1.
10:      else
11:         Return c_1, c_2, ..., c_m.
12:      end if
13:   end while
14:   i = i + 1.
15: end while
```

The second algorithm (Algorithm 3) starts with cloaking areas at the leaf node level. It first expands the cloaking area c_1 for the 1st request and checks whether the privacy values of all requests are fulfilled. If not, it moves the cloaking area c_2 for the 2nd request and checks whether all requests are fulfilled. This process goes on until all requests are fulfilled. Fig. 4(b) illustrates this procedure, and the detailed algorithm is given as Algorithm 3. This algorithm can guarantee that the level difference among all generated cloaking areas is within one in the quadtree T.

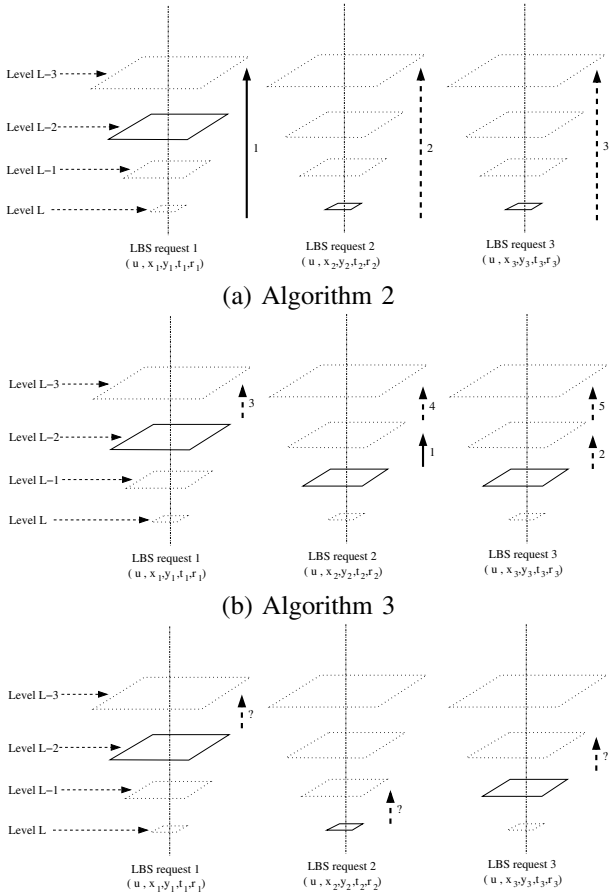


Fig. 4. **Illustrations of Algorithms 2-5:** Algorithms 2 and 3 raise the cloaking area in order while Algorithms 4 and 5 raise the cloaking area greedily based on certain privacy gain.

The third and fourth algorithms (Algorithm 4 and Algorithm 5) are greedy-based algorithms, where they choose one cloaking area (among all m cloaking areas) to expand in each step. The choice of cloaking areas is based on certain criteria/metrics. See Fig. 4(c) for illustration. To be more specific, for the third algorithm, we pick the cloaking area whose expansion can lead to maximum increase of privacy values. While in the fourth algorithm, we consider both privacy and area increment, that is, we pick the cloaking area whose expansion can lead to the maximum increase of the ratio between total privacy values and total areas.

Discussion. In summary, the enhanced L2P2 problem is a very challenging problem. A simple brute force method can find the optimal solution, but have exponential time complexity, which makes it very time-consuming in practice especially when the number of LBS requests is large. Therefore, we propose four different heuristics, which can find a sequence of cloaking areas in polynomial time to fulfill the user requirements. Each of these heuristics has a unique criterion

Algorithm 3 Cloaking Algorithm 3 for Enhanced L2P2

Input: A quadtree T, a footprint repository $\mathcal{F}$, and a sequence of m LBS requests (u, x_i, y_i, t_i, r_i) (for $i = 1$ to m).
Output: A sequence of m cloaking areas c_i ($i = 1$ to m).

1: **for** $i = 1$ to m **do**
2: Find the leaf node $c(j_i, L)$ in quadtree T which contains position of (x_i, y_i). Let $c_i = c(j_i, L)$.
3: **end for**
4: Let $l = L$ the height of quadtree T.
5: **while** $l \geq 1$ **do**
6: Let $i = 1$.
7: **while** $i \leq m$ **do**
8: Compute $p(c_1), \cdots, p(c_m)$ with respect to U', where U' is the set of common users in $c_1, \cdots, c_m$.
9: **if** $p(c_1), \cdots, p(c_m)$ do not satisfy the privacy requirements $r_1, \cdots, r_m$ **then**
10: Move c_i to be its direct parent in T.
11: **else**
12: Return $c_1, c_2, \cdots, c_m$.
13: **end if**
14: $i = i + 1$.
15: **end while**
16: $l = l - 1$.
17: **end while**

Algorithm 4 Cloaking Algorithm 4 for Enhanced L2P2

Input: A quadtree T, a footprint repository $\mathcal{F}$, and a sequence of m LBS requests (u, x_i, y_i, t_i, r_i) (for $i = 1$ to m).
Output: A sequence of m cloaking areas c_i ($i = 1$ to m).

1: Run Algorithm 1 for basic L2P2 to initialize $c_1, \cdots, c_m$.
2: Compute $p(c_1), \cdots, p(c_m)$ with respect to U', where U' is the set of common users in $c_1, \cdots, c_m$.
3: **while** $p(c_1), \cdots, p(c_m)$ do not satisfy the privacy requirements $r_1, \cdots, r_m$ **do**
4: **for** $i = 1$ to m **do**
5: Let c_i' be c_i's direct parent in T.
6: Compute the privacy values $p(c_1), \cdots, p(c_i'), \cdots, p(c_m)$ with respect to U'', where U'' is the set of common users in $c_1, \cdots, c_i', \cdots, c_m$. Note that if $p(c_j) > r_j$, let $p(c_j) = r_j$.
7: $\mathcal{P}[i] = \sum_{j=1}^{i-1} p(c_j) + p(c_i') + \sum_{j=i+1}^{m} p(c_j)$.
8: **end for**
9: Pick the index i which maximizes $\mathcal{P}[i]$.
10: Move c_i to be its direct parent in T.
11: **end while**
12: Return $c_1, c_2, \cdots, c_m$.

Algorithm 5 Cloaking Algorithm 5 for Enhanced L2P2

Input: A quadtree T, a footprint repository $\mathcal{F}$, and a sequence of m LBS requests (u, x_i, y_i, t_i, r_i) (for $i = 1$ to m).
Output: A sequence of m cloaking areas c_i ($i = 1$ to m).

1-6: same as Algorithm 4.
7: $\mathcal{P}[i] = \frac{\sum_{j=1}^{i-1} p(c_j) + p(c_i') + \sum_{j=i+1}^{m} p(c_j)}{\sum_{j=1}^{i-1} a(c_j) + a(c_i') + \sum_{j=i+1}^{m} a(c_j)}$.
8-12: same as Algorithm 4.

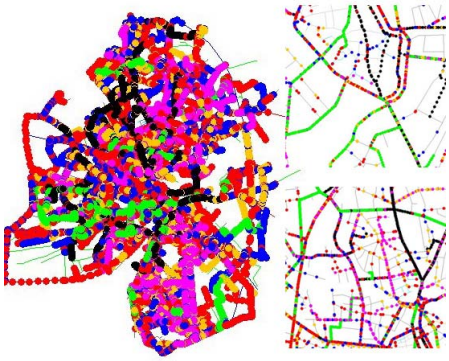


Fig. 5. Footprints of $1,000$ mobile users in the real road map of Oldenberg, Germany, generated via the genertator [5].

to expand cloaking areas: either following certain order as in Algorithm 2 and Algorithm 3, or based on privacy gain as in Algorithm 4 and Algorithm 5. We understand that these algorithms may not lead to optimal solutions for enhanced L2P2 problems. So in the next section, we evaluate these methods using a large mobile user location data set generated by a network-based traffic generator. Simulation confirms that all these methods can fulfill diverse privacy requirements, and two of them can achieve this with just slight downgrade of the quality of original location data (using small cloaking areas).

V. Simulation Results

To evaluate the effectiveness of our approach, we tested our algorithms extensively through a series of simulations. To generate the coordinates of mobile users and their LBS requests, we use a *network based generator of moving objects* proposed and implemented by Brinkhoff [5]. We randomly generate $1,000$ mobile users and simulate their movement on the real road map of Oldenberg, a city in Germany. For the moving speeds, we use the default setting in the generator, which changes users' speeds at each intersection based on the road type. We run the simulated $1,000$ mobile users for $1,000$ unit time, which generates about $35K$ footprints with timestamp and location information. Fig. 5 shows the global and partial view of the map of Oldenberg with footprints of mobile users. All these footprints $\mathcal{F}$ are saved in MySQL (http://www.mysql.com/) as a footprint database. We implement all five proposed algorithms (Algorithms 1 to 5) using Java as the programming language.

We first build a 5-level quadtree T to divide the entire $16Km \times 16Km$ region of Oldenberg into different size of cloaking areas. The smallest cloaking area at the bottom of the tree T has a size of $1Km \times 1Km$. We then randomly choose a mobile user and generate its privacy requirements. By applying the proposed algorithm, we generate a sequence of cloaking areas for this mobile user, and we verify (1) whether these cloaking areas can satisfy the user's privacy requirements and (2) how efficient these cloaking areas are in term of their average sizes. In other words, in this study, we are mainly interested in the following two performance metrics.

One is *cloaking area*, i.e., the average area of generated cloaking areas, and the other is *privacy protection level*, i.e., the average privacy values achieved by generated cloaking areas. It is obviously that we prefer smaller cloaking area with larger privacy protection level. For all simulations, we perform multiple rounds over multiple users and report the average performance metrics. In addition, we test all methods under both k-anonymity model [3] and entropy-based models [4]. Notice that due to the difference between these two models, we choose different mean values of privacy requirements in our simulations.

Performance on Basic L2P2. In the first set of simulations, we consider the basic L2P2 problem and evaluate the performance of our basic algorithm (Algorithm 1) with different privacy requirements from the user. We fixed the number of requests of each user at 20 and the privacy requirements are randomly chosen from a mean value from 5 to 10 for k-anonymity model and from 5 to 20 for entropy-based model. Fig. 6 shows the detailed results. Here, we also run Algorithm 1 with all privacy requirements set to be the maximum value in the request sequence (denoted as *Alg 1-Max*), which represents the previous work without considering diverse privacy requirements. From Fig. 6(a) and (c), we find that Algorithm 1 uses much smaller cloaking areas than that of *Alg 1-Max*. This confirms our conjecture that considering the diverse privacy requirements can lead to better quality of LBS services. In addition, we also observe that the actual privacy protection levels from *Alg 1* are also smaller than that of *Alg 1-Max*, as shown in Fig. 6 (b) and (d). (Obviously both methods satisfy the user's privacy requirements because they are above the line of required privacy values). Based on these observations, it is desirable to have L2P2 solutions to efficiently protect mobile user's location privacy in LBS systems.

Performance on Enhanced L2P2: Effect of Privacy Requirements. In the second set of simulations, we focus on the enhanced L2P2 problem (where only common users are counted for privacy calculation) and evaluate the other four proposed methods (Algorithms 2, 3, 4, and 5), with the similar setting to the first set of simulations. From Fig. 7(a) and (c), we find that all these algorithms need larger cloaking areas to protect the user privacy, compared with that of Algorithm 1. This is because only common users are counted for privacy calculation in Algorithms 2 to 5, while Algorithm 1 considers all users. In addition, Algorithms 3 and 5 use much smaller cloaking areas, compared with Algorithms 2 and 4. This tells us that Algorithms 3 and 5 are more efficient in term of the quality of cloaking areas generated. All methods need larger areas when users have higher privacy requirements. From Fig. 7(b) and (d), it is clear that all methods can satisfy the privacy requirements over common user set, and the privacy protection level increases along with the increment of privacy requirements from users.

Performance on Enhanced L2P2: Effect of Request Sequence Length. In the last set of simulations, we would like to see how different sequence length of LBS requests (i.e., different numbers of LBS requests) affect our results. We fix the privacy requirement with mean of 7 for k-anonymity model and 12 for entropy-based model, then increase the number of LBS requests from 10 to 50 for both models. Fig. 8 shows the results. From Fig. 8 (a) and (c), we observe that with the increase of LBS requests, Algorithms 3 and 5 consistently outperform Algorithms 2 and 4, in terms of average cloaking areas. However, in Fig. 8 (b) and (d), in term of average achieved privacy values, we do not see a clear trend here. We believe that this is because when more LBS requests are involved, the computation of common user sets are affected by more cloaking areas. Any user change in any cloaking area would impact the privacy value computation, which makes this issue more complicated. We also notice that this simulation result may be data set dependent. In our future work, we will get more data sets to perform more simulations, which may provide us with better observation.

VI. Conclusion

With the increasing importance of user location privacy issues, many approaches have been proposed to protect mobile users' location information. However, we observe that these existing approaches usually assume that users' privacy requirements are *constant*, which may not always be true in real-life scenarios. In this paper, observing that a mobile user's privacy requirements can be dynamic and diverse, we formalize this as the L2P2 problem. We further classify L2P2 problems into *Basic L2P2* and *Enhanced L2P2* problems. The difference between basic and enhanced L2P2 lies in whether the common users or all users in a sequence of cloaking areas would be used for privacy computation. For basic L2P2, we design a simple algorithm (Algorithm 1) to address this problem. While for enhanced L2P2, we propose four heuristics (Algorithms 2 to 5) to generate cloaking areas to satisfy users' privacy requirements, where each heuristic has a different, unique criterion to expand cloaking areas. In addition, to evaluate the effectiveness of our proposed algorithms, we conducted a large number of simulations, and several interesting observations have been reported in Section V.

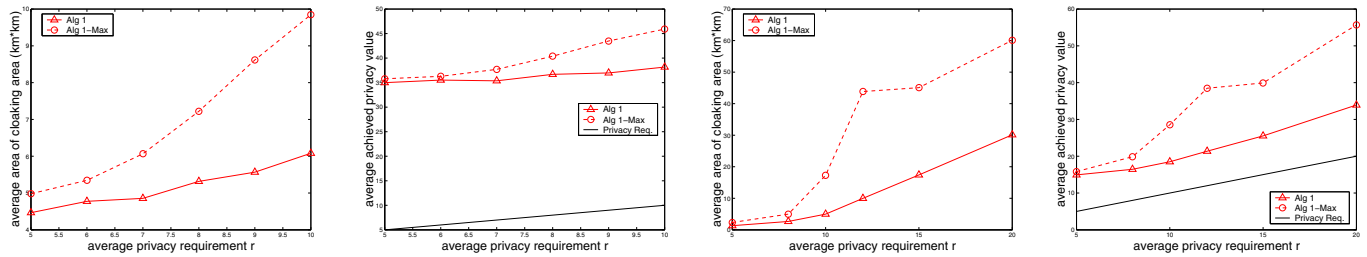
There are a few directions for our future work. First, we will investigate other efficient heuristics for enhanced L2P2 problem. Second, we will test our proposed methods over other location data sets (such as real-life tracking data if possible) and try different location privacy measurements (such as the one in [19]). Last, we are also interested in study of other types of location-based privacy, such as considering the content of actual query which is beyond just the location.

References

[1] K. M. Heussner. Google, Apple track users' location information, but why? ABC News, accessed on April 29, 2011 at http://abcnews.go.com/Technology/google-apple-track-users-location-information/story?id=13436330, 2011.

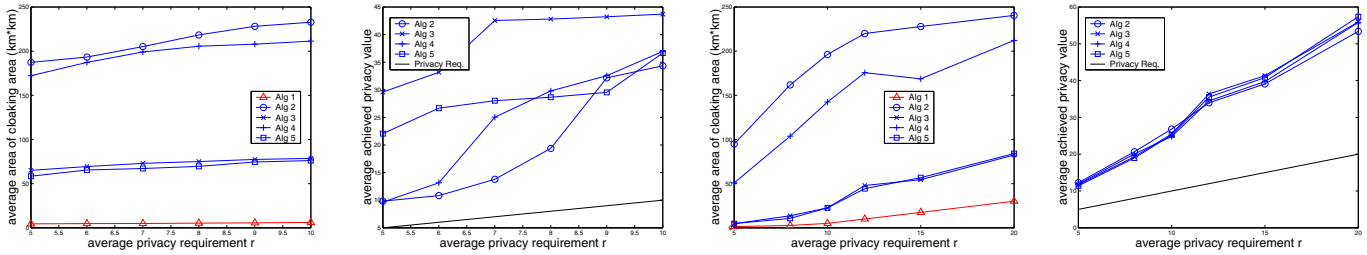
[2] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2:46–55, January 2003.

[3] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. of ACM MobiSys '03*, 2003.



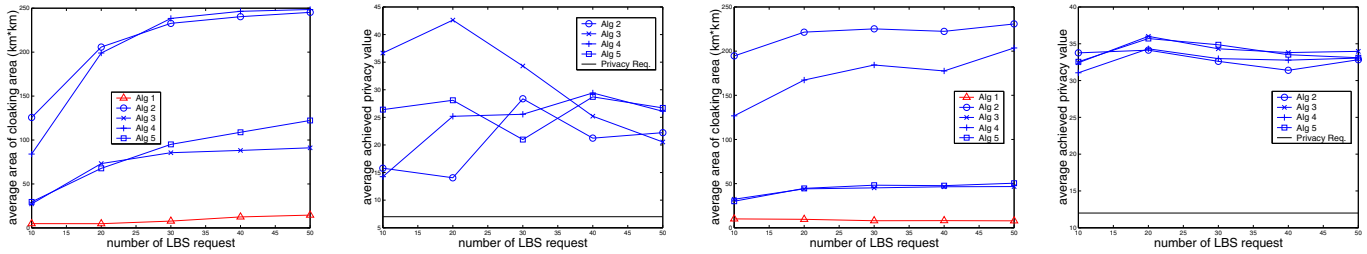
(a) cloaking area (k-anonym.) (b) privacy level (k-anonym.) (c) cloaking area (entropy) (d) privacy level (entropy)

Fig. 6. Results of algorithms for basic L2P2 with different privacy requirements.



(a) cloaking area (k-anonym.) (b) privacy level (k-anonym.) (c) cloaking area (entropy) (d) privacy level (entropy)

Fig. 7. Results of algorithms for enhanced L2P2 with different privacy requirements.



(a) cloaking area (k-anonym.) (b) privacy level (k-anonym.) (c) cloaking area (entropy) (d) privacy level (entropy)

Fig. 8. Results of algorithms for enhanced L2P2 with different number of LBS requests.

[4] T. Xu and Y. Cai. Feeling-based location privacy protection for location-based services. In *Proc. of ACM CCS '09*, 2009.

[5] T. Brinkhoff. A framework for generating network-based moving objects. *GeoInformatica*, 6:153–180, June 2002.

[6] C. Bettini, X. S. Wang, and S. Jajodia. Protecting privacy against location-based personal identification. In *Proc. of 2nd VLDB Workshop on Secure Data Management*, 2005.

[7] H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. In *Proc. of IEEE Int'l Conf. on Pervasive Services (ICPS2005)*, 2005.

[8] M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *Proc. of the 3rd Int'l Conf. on Pervasive Computing (Pervasive 2005)*, 2005.

[9] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Preserving user location privacy in mobile data management infrastructures. In *Proc. of the 6th Workshop on Privacy Enhancing Technologies*, 2006.

[10] C.-Y. Chow, M. F. Mokbel, and X. Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *Proc. of ACM int'l symp. on Advances in geographic information systems (GIS)*, 2006.

[11] B. Gedik and L. Liu. A customizable k-anonymity model for protecting location privacy. In *Proc. of IEEE ICDCS'05*, pages 620–629, 2004.

[12] M. Gruteser and B. Hoh. On the anonymity of periodic location samples. In *Proc. of the 2nd Int'l Conf. on Security in Pervasive Computing*, 2005.

[13] B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *Proc. of the 1st Int'l Conf. on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM)*, 2005.

[14] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in GPS traces via uncertainty-aware path cloaking. In *Proc. of ACM CCS*, 2007.

[15] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new casper: query processing for location services without compromising privacy. In *Proc. of VLDB '06*, 2006.

[16] M. Duckham and K. Lars. Location privacy and location-aware computing. *Dynamic & Mobile GIS: Investigating Change in Space and Time*, pages 34–51, 2006.

[17] J. Krumm. A survey of computational location privacy. *Personal Ubiquitous Computing*, 13:391–399, August 2009.

[18] H. Samet. *The design and analysis of spatial data structures*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.

[19] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux. Quantifying location privacy. In *Proc. of IEEE Symp. on Security and Privacy (S&P)*, 2011.