Electronic Preprint for Journal of Information Processing Vol.25

Invited Paper

# Full Location Privacy Protection Through Restricted Space Cloaking

Guolei Yang[1,a)] Ying Cai[1,b)]

Received: December 1, 2016, Accepted: June 2, 2017

**Abstract:** Anonymous location data may be correlated with restricted spaces like home and office for subject identification. This creates significant privacy risks to users when they disclose their location to applications like location-based services. In this paper, we address this problem with a novel approach named restricted space cloaking. This scheme cloaks a user's location only when the location is inside a restricted space. When in non-restricted spaces, the location is reported as is. We show that this cloaking strategy is capable of full location privacy protection: given a restricted space, the adversary does not know when its owner was there; given a non-restricted location, the adversary does not know who has ever visited there. Such protection is not available from the existing cloaking techniques. In addition to full location privacy protection, the proposed strategy makes it possible for users to cloak their locations without disclosing accurate locations to either a trusted anonymizer or other users. We discuss the implementation challenges and present corresponding solutions. The performance of the proposed solutions is evaluated through simulation.

**Keywords:** location privacy, location cloaking, restricted space cloaking

## 1. Introduction

When users disclose their location to applications like location-based service (LBS), they are subject to two significant privacy risks: query privacy and location privacy. The former refers to the fact that the service provider knows who uses the services, whereas the latter, a user's whereabouts. Simply using pseudonyms for identity protection in service uses does not solve the problem because anonymous location itself may be correlated with restricted spaces such as home and office to identification the subject. A single location may not reveal a user's real-world identity, but a time-series sequence of location samples that form a trajectory will eventually do.

The above problem, referred to as *Restricted Space Identification* [1], has received great research interests in the last decade. A number of technical solutions have been proposed, among which the most intensively studied one is location cloaking. The idea put in a simple way is to reduce location resolution to achieve a desired level of protection. Instead of reporting their exact position, users disclose a cloaking region as their location. The challenge here is how to compute a cloaking region that can provide a certain level of guarantee that it cannot be linked to some specific individual. Existing techniques can be classified into two categories:

- Neighbor Cloaking [1], [2], [3], [4], [5]: The techniques in this category ensure that each cloaking region has at least $k$ users inside it at the time when the cloaking region is reported.
- Footprint Cloaking [6], [7]: This type of techniques guarantees that each cloaking region contains at least $k$ different users' footprints, each defined to be a location sample collected at some historical time point.

While location cloaking has been shown to be effective in privacy protection, existing techniques fall short in various aspects. Neighbor cloaking supports only anonymous service uses, but not location privacy protection. Given a cloaking region that contains at least $k$ users, an adversary may not know which of these users requested the service, but knows that all of them were there at the time when the location was reported. When compared to a single user's location, revealing the presence of a group of people together in a particular area is even more threatening – it is well said that "where you are and whom you are with are closely correlated with what you are doing" [17]. On the other hand, using footprints for cloaking was aimed at addressing this problem, but it can protect a user's location privacy only at the time dimension. Given a cloaking region with $k$ visitors' footprints, an adversary does not know when they visit the region, but does know that they have all visited this region. This remains a significant privacy concern, especially when a cloaking region is part of sensitive places such as clinic or entertainment centers.

In this paper, we consider the problem of providing *full location privacy protection* to the users of LBS. Given a restricted space, an adversary shall not know when the corresponding owner was there; given a non-restricted space, the adversary shall not know who has ever visited there. We call such protection full protection because for restricted spaces, the presence of their owners is considered public knowledge and the best one can do is

[1] Department of Computer Science Iowa State University, Ames, Iowa 50011, USA
[a)] yanggl@iastate.edu
[b)] yingcai@iastate.edu

to prevent an adversary from knowing the time of their presence. We summarize our main contributions as follows:

1) We introduce the concept of full location privacy protection and show that the existing cloaking techniques fail to provide such protection.

2) We present a novel cloaking technique called *Restricted Space Cloaking* (RSC). This scheme cloaks a user's location only when the location belongs to a restricted space. Otherwise, the location is submitted as is. We prove that this scheme protects can provide users full location privacy protection. Moreover, it incurs less communication and computation overhead than existing techniques that cloak a user's every location.

3) We introduce a new concept called *Cloaking Map*, which allows users to cloak their location by themselves. This feature of self-cloaking is a significant contribution since most existing techniques require users to disclose their exact location to either a trusted anonymizer [1], [2], [4], [6], [7] or their current neighbors [3] to compute their cloaking region. This is problematic because the anonymizer and neighboring nodes may be as untrustworthy as the provider of LBS. One exception that supports non-exposure location cloaking is the technique proposed in Ref. [12]. This scheme, however, relies on secure multiparty computing techniques, which incur significant computation and communication overheads among mobile nodes.

4) We have implemented the proposed techniques with a detailed simulator and evaluated their performance from various aspects using realistic synthetic data.

The rest of this paper is organized as follows. In Section 2, we give a formal definition of the problem and introduce the basic idea of the proposed restricted space cloaking. We describe the implementation challenges and corresponding solutions in Section 3 and then evaluate their performance in Section 4. We discuss more related work in Section 5 and conclude this paper in Section 6.

## 2. Basic Idea: Restricted Space Cloaking

In restricted space identification, the adversary uses the restricted spaces to identify the subjects of anonymous location data. A space $S$ is said to be restricted to (or owned by) a person $u$ if having a service request originating from $S$ reveals the presence of $u$ in $S$ at the time when the request was sent. Notice that $u$ may or may not be the one who requests the service and a person may own more than one restricted space. Examples of restricted spaces include house and office which are considered public domain knowledge (e.g., available from public sources like the Internet or housing data). If a location is inside or contains a restricted space, the subject at this location is very likely to be the owner of the restricted space.

The adversary is interested in location privacy intrusion. Let $k_u$ be the desired level of protection of a user $u$. We say a privacy protection protocol $P$ supports *Full Location Privacy Protection* if for any location $(l, t)$, where $l$ is the user's location reported at time $t$, the following conditions hold:

1) If $l$ contains a restricted space $S$ owned by $u$, the probability that the adversary knows $u$ was in $S$ at time $t$ is no greater than $1/k_u$.

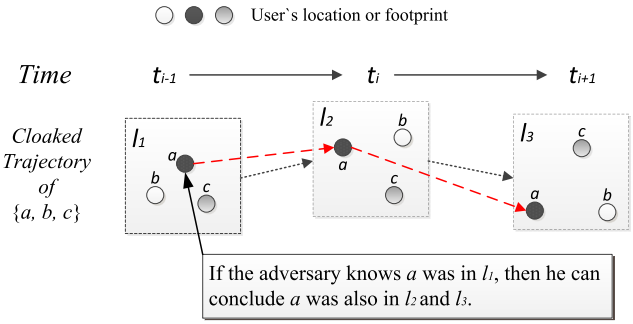



**Fig. 1** Privacy leak in existing cloaking techniques.

2) If $l$ does not contain any of $u$'s restricted spaces, the probability that the adversary knows $u$ has visited $l$ is no greater than $1/k_u$.

As mentioned earlier, neither neighbor cloaking nor footprint cloaking supports full location privacy protection. Their problem roots from the fact that these techniques cloak a user's location whenever the location is reported. Let $(R_1, t_1) \rightarrow (R_2, t_2) \rightarrow \ldots \rightarrow (R_n, t_n)$ be an anonymous trajectory. The adversary knows that this trajectory is computed based on the trajectories from $k$ different users. If restricted space identification allows the adversary to know any one of these users being a subject of $R_i$, then he can further conclude that this user must be a subject of all other cloaking regions. **Figure 1** demonstrates this privacy leak.

In this paper, we present a novel cloaking strategy that circumvents the above problem. We observe that locations reported in non-restricted space area such as highway or parking lot can hardly lead to subject identification because the number of potential visitors to such locations is usually huge. Thus, to prevent identity disclosure, a user's accurate location needs to be cloaked only when he is inside his restricted space. In light of this, we propose to cloak a user $u$'s restricted spaces with the restricted spaces owned by at least $k_u - 1$ restricted space owned by other users. The set of users whose restricted spaces are cloaked with $u$'s restricted spaces is called $u$'s *cloaking set* and the resulted cloaking regions form a *cloaking map*. Let $U$ be a cloaking set that consists of $n$ users $\{u_1, u_2, \ldots, u_n\}$. A cloaking map for $U$ consists of a number of cloaking regions that is required to satisfy the following properties:

1) Every cloaking region contains at least $n$ restricted spaces, each owned by a different user in $U$;
2) For every user in $U$, each of its restricted space is covered by at least one cloaking region.

**Figure 2** shows a cloaking map that contains four cloaking regions generated from for a cloaking set containing three users. Given a cloaking map, a user cloaks its location as follows: when having to report its location for a service, the user simply checks its position against the cloaking regions in the map. If the location is inside a cloaking region, the user reports the cloaking region as its location. Otherwise, it reports its accurate location as is, i.e., without being cloaked.

We refer to the above strategy as *Restricted Space Cloaking* (RSC), alluding to the fact that this scheme cloaks a user's location only when the location is inside a restricted space. This approach has several clear advantages: 1) When a user's loca-

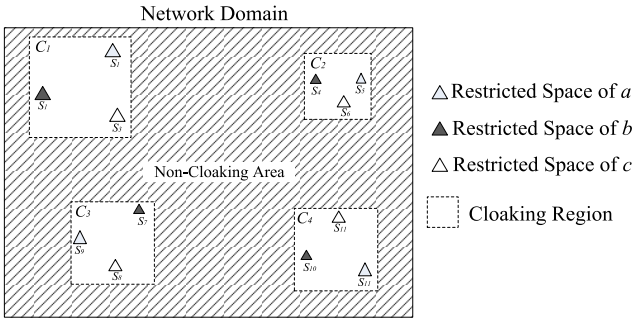



**Fig. 2** A cloaking map for $U = \{a, b, c\}$.

tion is reported as is, best location resolution is achieved, which benefits both the user (improves quality of service) and the service provider (does not need to retrieve/send unnecessary query results); 2) The cloaking map enables users to cloak their location by themselves, without having to disclose accurate location to any third party; 3) The concept of cloaking map simplifies the cloaking process since a user does not need to calculate a cloaking region every time he reports location, but simply look up an appropriate cloaking region in his cloaking map. And the cloaking map does not need to be recalculated unless someone changes his restricted spaces.

The most important advantage, however, is that this strategy has the potential to enable full location privacy protection. Without loss of generality, let $(l_i, t_i)$ and $(l_j, t_j)$ be two continuous locations generated by RSC for a user $u$ with a protection level of $k_u$, where $t_i < t_j$. There are four cases (showed in **Fig. 3**):

1) Both $l_i$ and $l_j$ are non-cloaked locations. Since such locations do not belong to any restricted space, any user could be the subject. As such, the probability of $u$ being at $l_i$ and $l_j$ is no greater than $1/k_u$.
2) $l_i$ is a cloaking region and $l_j$ is a non-cloaked location. In this case, $l_i$ must contain $u$'s restricted space and at least $k_u - 1$ other owners' restricted spaces. The adversary knows all these owners have visited $l_i$, and for any one of them, the chance of being there at time $t_j$ is $1/k_u$. The adversary also knows for sure that one of these users moves from $l_i$ to $l_j$, but does not know who does so. So the probability of having any one of these $k_u$ users visiting $l_j$ is no greater than $1/k_u$.
3) Location $l_i$ is non-cloaked and $l_j$ is a cloaking region. This is similar to case 2. The $k_u$ owners identified from $l_j$ have an equal chance of being at $l_i$.
4) Both $l_i$ and $l_j$ are cloaking regions. The two cloaking regions reveal two sets of users. Here we can make sure that the two sets contain at least $k_u$ common users so that the chance of having any one of them at $l_i$ at $t_i$ and at $l_j$ at time $t_j$ is no greater than $1/k_u$.

So far our discussion follows the same assumption as in existing researches on the adversary's background knowledge: it knows only restricted spaces and their corresponding owners. Under this assumption, the owners of the restricted spaces covered in a trajectory have the equal chance to be the subject. In reality, however, other side information may be available to the adversary. We now consider a common situation that the adversary uses road network information, such as Google Map, to re-

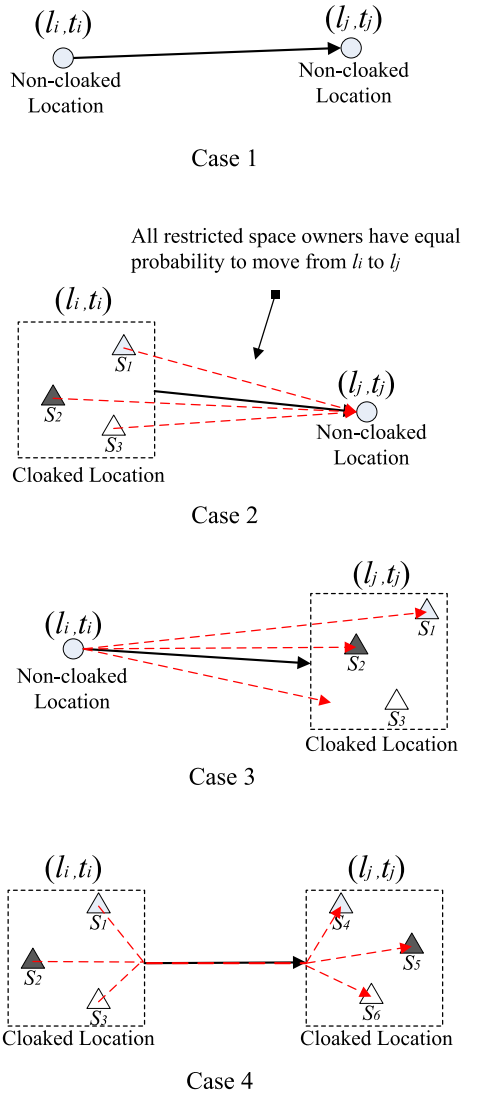



**Fig. 3** Cloaking results of restricted space cloaking.

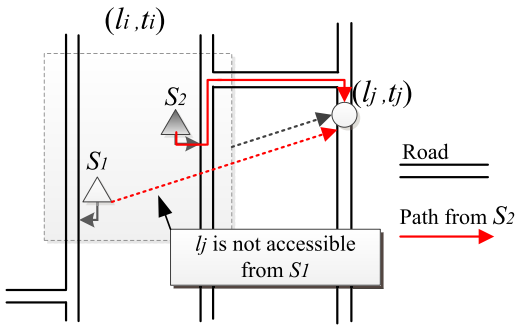



**Fig. 4** Accessibility attack.

fine the cloaking set of a user. Consider two adjacent locations $(l_i, t_i)$ and $(l_j, t_j)$ in a trajectory generated for a same user (**Fig. 4**). $l_i$ is a cloaking region with two restricted spaces $S_1$ and $S_2$, and $l_j$ is a non-cloaked location. If there is no road from $S_1$ to $l_j$ or it is impossible for one to move from $S_1$ to $l_j$ within a time period of $t_j - t_i$, then the adversary can conclude the owner of $S_2$ is more likely to be the subject of these two locations.

This attack (which we refer to as *Accessibility Attack*) arises when a cloaking region's exit or entrance point is not accessible from a restricted space inside the region. To prevent this attack,

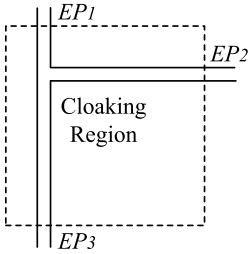



**Fig. 5** Exit/entrance points of a cloaking region.

we first introduce the notion of Exit/Enter Point (EP). An EP is a point where a public road intersects with the border of a cloaking region. In other words, an EP is a point where one can move into or out of a cloaking region. **Figure 5** shows a cloaking region with three EPs. With the notion of EP in place, we can prevent the accessibility attack by ensuring that each cloaking region $c$ has the following *Accessibility Requirements*:

1) For each restricted space $S$ inside $l$ and each EP of $l$, there exists at least one road $P$ from $S$ to EP.
2) All such paths are completely within $c$.

The two requirements are there to guarantee that every EP of a cloaking region $c$ is accessible from all restricted spaces inside it without having to move out of $c$. As such, given a trajectory that contains $c$, any owner of the restricted spaces inside $c$ has an equal chance of moving into or out of $c$ from or to other locations on the trajectory.

## 3. Implementation Details

### 3.1 System Overview

We now consider how to implement the proposed cloaking strategy. In general, cellular phone users access the Internet through their wireless service providers (WSP) such as AT&T and Verizon. We assume such providers are interested in assisting their clients to preserve their location privacy and therefore are willing to provide the cloaking map for them as an add-on value. To compute the cloaking map, the WSP needs to have the following information: 1) each user's required level of protection; 2) each user's restricted spaces; and 3) road networks. Note that in computing cloaking maps, the WSP does not require any location data of users except for the location of their restricted spaces (which is considered public knowledge), so the WSP does not have to be trusted. An overview of the system structure is demonstrated in **Fig. 6**.

Users can report their desired level of protection and restricted spaces to the WSP. Here a restricted space can be any location where a user wants it to be cloaked should a service request is sent within it. Another way to gather restricted space information is from public sources like housing data available from city assessor. It is also possible for the WSP to find out restricted spaces by analyzing the location data they collect from their clients through signal triangulation. Here we simply assume that the WSP has the information of restricted spaces of its users without further discussion on how to acquire this information. As for road networks, the WSP can acquire from public sources like GIS databases.

We represent road network by an undirected graph $G_{road}$, where a road is represented by an edge while road intersection

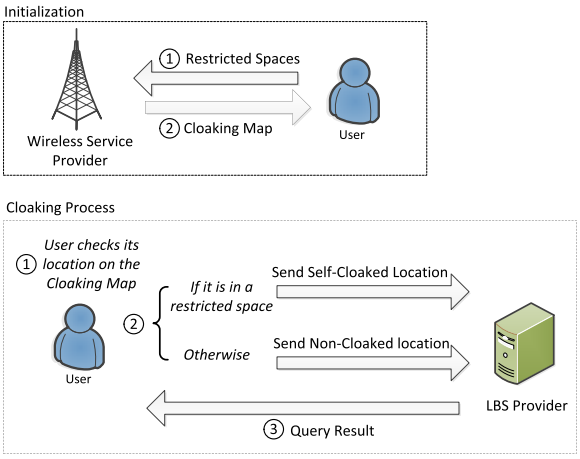



**Fig. 6** System structure.

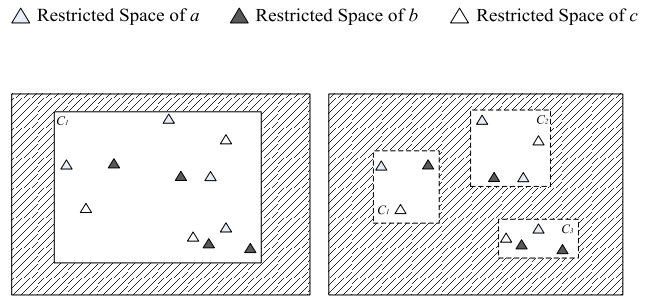
points are represented by vertices. Coordinates of an end point or an intersection point of the road network are stored in the corresponding vertex. As such we can use BFS to find a path between any two points. To accelerate the process of generating cloaking maps, we also index the roads using a segment tree such that we can easily compute which roads are inside and/or intersect a cloaking region.

Let $U = \{u_1, u_2 \dots u_n\}$ be the set of users. A user $u_i = <uid, k_{ui}, RS(u_i)>$ contains a user's $id$, his desired protection level, and the set of restricted spaces owned by him. Let $S = \{s_1, s_2 \dots s_m\}$ be the set of all restricted spaces, where $s_i = <sid, MBR(s_i), r_i>$ contains the id of a restricted space, its minimal bounding rectangle, and the road to which the restricted space is directly connected, such that a subject leaves or enters the restricted space exclusively through this road.

The challenge here is how to compute a cloaking map that satisfies a user's desired protection level and meanwhile, is of good quality. The quality of a cloaking map can be measured by the total area of cloaking regions in the map. A smaller area will result in a higher cloaking resolution, which in turn allows a user to receive a better quality of service. Moreover, it is likely to incur less computation and communication costs to both the server and users. This is because when a user is not inside a cloaking region, he can report his accurate location to the LBS provider. We propose a two-step approach:

- Step 1. *Cloaking Set Selection*: Partition the users into a number of cloaking sets such that each user belongs to one and only one cloaking set. To satisfy the required protection level of all users, the cardinality of each cloaking set $U$ must be no less than $k_u$ for any $u \in U$.
- Step 2. *Cloaking Map Generation*: For each cloaking set, generate a cloaking map. Because of the criteria for users to be in a cloaking set, this map can be distributed to and shared by all users in the cloaking set to cloak their location.

Note that the above two steps only need to be performed once for a given set of users and their restricted spaces, which usually do not change frequently. In the next subsections, we discuss these two steps in detail. To make it easy to follow, we will discuss how to perform Step 2 first.




(a) Naïve approach (b) Advanced approach

**Fig. 7** Cloaking map for $U = \{a, b, c\}$.

### 3.2 Cloaking Region Generation

Given a cloaking set, a naïve way to create corresponding cloaking map is to generate a cloaking region that contains all restricted spaces of these users. **Figure 7** (a) illustrates this approach. This approach is easy to implement, but can result in poor cloaking resolution since the cloaking region can be very large.

Here we present a more advanced approach. Let $U = \{u_1, u_2 \dots u_k\}$ be a cloaking set and $S = \{s_1, s_2 \dots s_n\}$ the set of all restricted spaces owned by the users in U. To generate qualified cloaking regions, our idea is first divide the set into smaller subsets. We partition $S$ into $m$ subsets $\{S_1, S_2 \dots S_m\}$ such that $\bigcup_{i=1}^{m} S_i = S$ and $S_i \cap S_j = \emptyset \ \forall i, j$. Given a subset $S_i$, we can then use the minimal bounding rectangle of the restricted spaces in $S_i$ as its cloaking region. The problem here is how to make sure that the generated cloaking regions have the minimum area. A partition is said to be optimal if the area of the corresponding cloaking regions is smallest. Note that when partitioning $S$ into $m$ subsets $\{S_1, S_2 \dots S_m\}$, we need to guarantee that every subset $S_i$ contain at least 1 restricted space of each user. This approach could result in better cloaking resolution because we are able to cloak nearby restricted spaces together with $m$ smaller regions as demonstrated in Fig. 7 (b).

We formulate the above optimization problem as follows. Given the cloaking set $U = \{u_1, u_2 \dots u_k\}$, we first define the following variables:

$$O_{ij} = \begin{cases} 1 & \text{if } s_i \text{ belongs to } u_j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The above variable $O_{ij}$ indicates the ownership of each restricted space. Note that here we assume each restricted space can only belong to one user, thus, the following condition holds at any time:

$$\sum_{j=1}^{k} O_{ij} = 1 \quad i \in \{1, 2 \dots n\} \tag{2}$$

$$P_{ij} = \begin{cases} 1 & \text{if } s_i \text{ is in } S_j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

This variable reflects which subset a restricted space is assigned to. In reality a restricted space can belong to multiple owners. To deal with this situation, we can represent such a restricted space as multiple restricted spaces with the same location, each belongs to one owner. Let $x_i$ and $y_i \in \mathbb{N}$ denote the $x$ and $y$ coordinates of restricted space $s_i$. The area of a cloaking region (i.e., the minimal bounding rectangle) generated for subset $S_j$ can be calculated by:

$$MBR(S_j) = \left[\max_{i=1}^{n}(P_{ij}x_i) - \min_{i=1}^{n}(x_i/P_{ij})\right] \times \left[\max_{i=1}^{n}(P_{ij}y_i) - \min_{i=1}^{n}(y_i/P_{ij})\right] \tag{4}$$

For convenience of computation, we simple define $1/P_{ij}$ to be a very large positive number when $P_{ij} = 0$. To find the optimal partition of a given cloaking set, the following optimization problem need to be solved for $P_{ij}$:

$$\textit{Minimize}: \quad \sum_{j=1}^{m} MBR(S_j) \tag{5}$$

*Subject to* :

$$\sum_{j=1}^{m} P_{ij} = 1 \quad i \in \{1, 2 \dots n\} \tag{6}$$

$$\sum_{i=1}^{n} O_{ij}P_{il} \geq 1 \quad j \in \{1, 2 \dots k\},\ l \in \{1, 2 \dots m\} \tag{7}$$

$$O_{ij}, P_{ij} \in \{0, 1\} \quad \forall \text{i and } j \tag{8}$$

Equation (6) requires that a restricted space must be assigned to exactly one cloaking region. While Eq. (7) requires that each cloaking region must contain at least 1 restricted space of each user. Note that these two restrictions guarantee that property 1 and 2 are satisfied for each cloaking region generated in this step. Here, $O_{ij}$, $x_i$, and $y_i$ are given as input.

Solving the above linear programming is known to be NP-hard. For a small number of restricted spaces, one could iterate through all possible partitions of $S$ to find the optimal partition that will result in minimal area of restricted spaces. This approach we will refer to as *Optimal Cloaking Region Generation* algorithm. Exhaustive search like this, however, is computational-infeasible as the total number of partitions grows exponentially to $|S|$. Alternatively, we propose an *Heuristic Cloaking Region Generation* algorithm with running time linear to the size of the cloaking set. Pseudo code of this algorithm is given below:

**Algorithm 1.** *Generate_CMap*$(U, S)$

```
1   CMap ← ∅
2   u* ← the user with least number of restricted spaces.
3   m ← the number of restricted spaces of u*
4   For each restriced space s_i of u*, do
5       S_i ← s_i
6   For each u ∈ U and u ≠ u*, do:
7       S_u = the set of restricted spaces of u.
9       Allocate S_u into {S_1, S_2 ... S_m} such that each subset
        contains at least 1 restricted space in S_u.
10      For each subset S_i ⊂ S, do
11          Compute the area of the Minimal Bounding
            Rectangle of all restricted spaces in S_i.
12      While there exists other allocation scheme of S_u
13      Allocate restricted spaces in S_u into {S_1, S_2 ... S_m}
        with the optimal allocation scheme
14  For each subset S_i, do
15      Add the Minimal Bounding Rectangle of restricted
        spaces in S_i into CMap.
16  Return CMap
```

Our proposed algorithm, as showed in Algorithm 1, starts with a pivot user $u^*$ who has the least number of restricted spaces. Suppose $u^*$ has $m$ restricted spaces. We first generate $m$ subsets $\{S_1, S_2 \dots S_m\}$ such that each subset contains exactly one restricted space of $u^*$. These subsets are used as the initial partition. Then, for each user $u$, we try to allocate his restricted spaces into $\{S_1, S_2 \dots S_m\}$ such that 1) each subset $S_i$ is assigned at least 1 restricted space of $u$; and 2) each subset's cloaking region is minimally expanded. To this end, we compare all the possible ways to assign $u$'s restricted spaces to $m$ subsets, which is substantially smaller than the number of all possible partitions of $S$. We perform this greedy allocation for users one by one, until all users' restricted spaces are allocated into exactly one of the subsets.

**Algorithm 2.** *Expand_CRegion*$(c, G_{\text{road}})$

1. **For each** restricted space $s$ inside $c$, **do**
2. **For** each exit point $EP$ of $c$, **do**
3. $Path \leftarrow$ the shortest path from $s$ to $EP$ in $G_{road}$.
4. **If** $Path = \emptyset$ **or** is not totally inside $c$, **then**:
5. Expand $c$ to fully contain $Path$.
6. Go back to step *1* and repeat.
7. **Return** $c$

Recall that cloaking regions must also satisfy the accessibility property to prevent accessibility attack. In order to do so, we rectify all raw cloaking regions generated by Algorithm 1 through an expanding process as showed in Algorithm 2. In this algorithm, we take an intuitive approach by keep expending the cloaking region to include more roads and road intersections until the requirement is satisfied. Since the whole road network is assumed to be connected, this process will guarantee to create cloaking regions that satisfy the property. Note that the cloaking regions generated by the above algorithms may overlap with each other. To avoid cloaking regions being refined by the adversary, we require the user to randomly choose a cloaking region to report if his current location falls into multiple overlapped cloaking regions.

### 3.3 Cloaking Set Partition

We now consider how to perform cloaking set selection. A cloaking set basically is a set of users whose restricted spaces are cloaked together. Similarly, given the original set of users, we can use exhaustive search to find the optimal partition of cloaking sets such that the average cloaking resolution of their corresponding cloaking maps is minimized. Let $\{U_1, U_2 \dots U_m\}$ denote a partition of $U$. The area of cloaking regions generated for a user set $U_i$ is represented by:

$$AREA(U_i) = MBR_{Opt}(S), \tag{9}$$

where $MBR_{Opt}(S)$ is the minimal sum of MBRs as in the target function (5) achieved by solving the 0-1 integer linear programming in previous section. Here, $S$ is the set of restricted spaces owned by users in $U_i$. Define the following variable:

$$X_{ij} = \begin{cases} 1 & \text{if } u_i \text{ belongs to } U_j \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The variable indicates which cloaking set a user has been allocated to. The optimal partition can be found by solving:

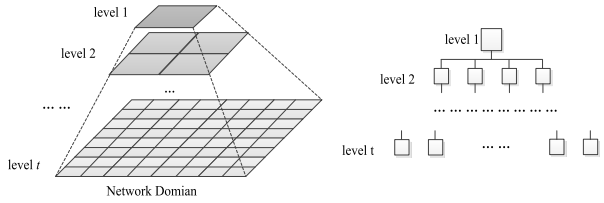



**Fig. 8** Quadtree partition of the network domain.

$$\textit{Minimize:} \quad \sum_{j=1}^{m} AREA(U_j) \tag{11}$$

*Subject to:*

$$\sum_{j=1}^{m} X_{ij} = 1 \quad i \in \{1, 2 \dots n\} \tag{12}$$

$$|U_j| \geq \max_{i=1}^{n}(X_{ij} k_{ui}) \quad j \in \{1, 2 \dots m\} \tag{13}$$

$$X_{ij} \in \{0, 1\}, \quad \forall \text{i and } j \tag{14}$$

Exhaustive search for the optimal solution is infeasible in this case since the running time is exponential to the total number of restricted spaces owned by all users, which can be very large. Again, to circumvent the computational infeasibility, we propose a *Heuristic Cloaking Set Partition* algorithm with quadratic worst case running time to the number of restricted spaces.

To minimize cloaking resolution, the users in a same cloaking set should have a similar number and distribution of their cloaking regions. In light of this, we introduce the *Restricted Space Feature Vector* (RSFV) as a coarse indicator of both the number and distribution of restricted spaces of a user. Specifically, we first partition the network domain into a grid of $n \times m$ homogeneous cells, and then designate each cell a unique index from $(1, 1)$ to $(n, m)$. A user u's RSFV is defined as:

$$\boldsymbol{V_u} = <V_{(1,1)}, V_{(1,2)} \dots V_{(2,1)} \dots V_{(n,\text{m})}> \tag{15}$$

where

$$V_{(i,j)} = \begin{cases} 1 & \text{if } u \text{ has restricted spaces overlaped } cell(i, j) \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

This vector can be considered a spatial down sampling of a user's restricted spaces, so it reflects the spatial feature of these restricted spaces in a rough way. Users with similar feature vectors should be selected into the same cloaking set. We recursively partition the network domain using a quadtree of $t$-levels, as demonstrated in **Fig. 8**. At every level of the quad tree, we compute the RSFV for each user, representing his restricted space distribution in this level of resolution. We now present a Feature Vector-aided cloaking set partition algorithm, showed in Algorithm 3 and 4.

**Algorithm 3.** *Find_CloakingSet*$(Pivot, S, U)$

1 $CSet \leftarrow \{Pivot\}$,
2 Remove $Pivot$ from $U$.
3 **For** $level = t$ to *1,* **do**
4 **For each** $u \in U$, **do**
5 **If** $\boldsymbol{V_u} = \boldsymbol{V_{Pivot}}$ at $level$ **and** $k_u \leq k_{pivot}$ **then**
6 Add $u$ into $CSet$. Remove $u$ from $U$
7 **If** $|CSet| \geq k_{pivot}$, **then go to** step 12

8 **Else, for each** $u \in U$, **do**
9 **If** $Close(\boldsymbol{V}_u, \boldsymbol{V}_{Pivot}, level)$ **and** $k_u \leq k_{pivot}$ **then**
10 Add $u$ into $ASet$. Remove $u$ from $U$
11 **If** $|CSet| \geq k_{pivot}$, **then go to** step 12
12 Return $ASet$

**Algorithm 4.** $Is\_Close(\boldsymbol{V_u}, \boldsymbol{V_v}, level)$

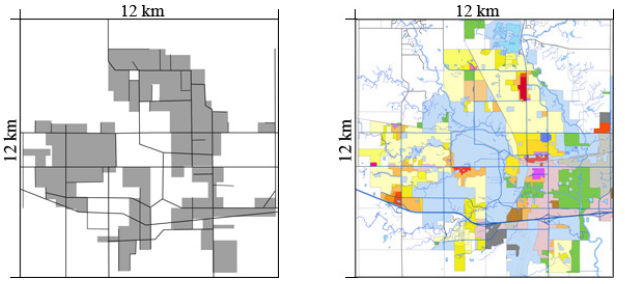
1 $\boldsymbol{V_{mask}} \leftarrow <0, 0, \ldots 0>$ at $level$
2 **For each** $V^u_{(i,j)}$ in $\boldsymbol{V_u}$ at $level$, **do**
3 **If** $V^u_{(i,j)} = \mathbf{1}$, **then**
4 $V^{mask}_{(i,j)}, V^{mask}_{(i-1,j)}, V^{mask}_{(i+1,j)}, V^{mask}_{(i,j-1)}, V^{mask}_{(i,j+1)} \leftarrow 1$
5 **For each** $V^v_{(i,j)}$ in $\boldsymbol{V_v}$ at $level$, **do**
6 **If** $V^v_{(i,j)} = \mathbf{1}$ **and** $V^{mask}_{(i,j)} = 0$, **then**
7 **Return** $FALSE$.
8 **Return** $TRUE$.

The algorithm starts from the $t$-th level the quadtree. First, we randomly choose a user as a pivot form the set of users who do not have a cloaking set yet. Then, we try to find at least $k_{pivot} - 1$ other users to formulate a cloaking set with him. These users should satisfy two conditions: 1) their desired protection level must be no higher than $k_{pivot}$, and 2) their RSFV equals $\boldsymbol{V}_{pivot}$ at this level. If there is no enough user satisfying condition 2), we try to add users whose RSFV is "close" to $\boldsymbol{V}_{pivot}$. Two users' RSFV are said to be close to each other if all of their restricted spaces are located in the same cell or in adjacent cells. If still not enough users are found, we move up the quadtree to the $t-1$ level until reach level 1. This process is repeated until all users are grouped into their cloaking sets. If at certain point, the number of remaining users is less than $k_{pivot}-1$, we then try to allocate the remaining users into existing cloaking sets. For a remaining user $u$, we can allocate him into any cloaking set as long as the number of users in that set is no less than $k_u$. And adding $u$ into the cloaking set will not threaten existing users in that set since it will only increase the protection level of this cloaking set. We assume the highest protection level required by a user is less than the total number of users; otherwise it is impossible to achieve such a level of protection.

## 4. Performance Evaluation

We have implemented a detailed simulator that allows us to evaluate the performance of the proposed strategy from various aspects. The performance of cloaking based techniques can be measured by the average resolution of user-reported locations (the smaller the better), which reflects not only the quality of service, but also the potential communication cost. In restricted spaces cloaking, a user cloaks its location based on its cloaking map, so we are also interested in the quality of the cloaking map produced by our approaches. We measure the quality of cloaking map by the proportion of non-cloaking area (area in which the user's location is reported as is) to the area of the whole network domain. In general, the larger this proportion is, the more accurate locations are likely to be reported.

To evaluate the proposed techniques in a real world scenario, we simulate an area of 12 km×12 km (144 km$^2$), which is the size of the city of Ames, IA, USA, based on the GIS data [14] provided by the city authority. Residential areas are added according to the 2010 census and household data published by the US Census Bureau [15] and the housing density data acquired from [14], [16]. **Figure 9** shows the simulated area comparing to Ames urban area as seen on the GIS Map. The area is partitioned into 50 m × 50 m cells using quadtree. Since restricted spaces are usually house and office, we randomly allocate them in residential and commercial zones.




(a) Simulated area. Residential and commercial zones are marked by shadow area

(b) The original GIS map. Residential and commercial zones are marked by yellow and red area.

**Fig. 9** Synthetic data for performance evaluation.

**Table 1** Compared algorithms.

| Name | Algorithms |
|---|---|
| *Footprint* | footprint cloaking[6] |
| *Proposed1* | Proposed strategy with *Heuristic Cloaking Set Partition* + *Heuristic Cloaking Region Generation* |
| *Proposed2* | Proposed strategy with *Heuristic Cloaking Set Partition* + *Optimal Cloaking Region Generation* |

We implement three algorithms, including *Heuristic Cloaking Set Partition*, *Heuristic Cloaking Region Generation*, and *Optimal Cloaking Region Generation*. The optimal cloaking set partition was not implemented since its running time is exponential to the total number of restricted spaces of all users, which is beyond the computational power of our experimental platform. In order to compare performance of the proposed strategy against state-of-the-art cloaking techniques, we also implemented the *footprint cloaking* algorithm proposed in Ref. [6]. When a user is not inside any cloaking region and needs to report location as is, we assume the accurate location has a resolution of 5 meter, which can be achieved by consumer-grade GPS [25] as equipped on most smartphones. We group these algorithms into three categories for comparison (**Table 1**).

Note that restricted space cloaking supports full location privacy protection and self-cloaking, the two features not available from existing cloaking techniques. As such, we only compare the quality of service that they can provide. We simulate a 12 hours duration of service usage of a set of users. The number of queries is set to 30 per user per hour to simulate continuous service uses like navigation. We are interested in how the performance of these techniques is impacted by factors including the number of restricted spaces, the distribution of restricted spaces

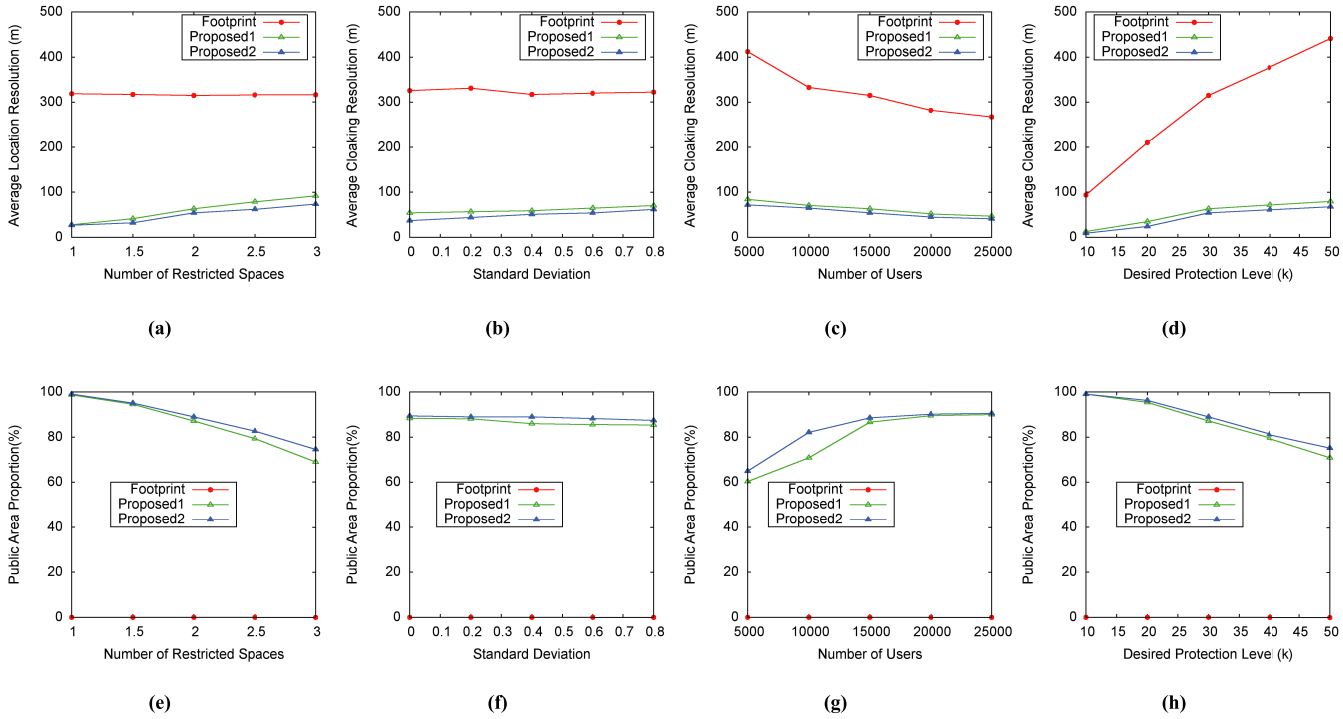



**Fig. 10** Experimental results.

**Table 2** Default settings.

| Parameter | Range | Default |
|---|---|---|
| Number of users | 5000 – 25000 | 15000 |
| Mean number of restricted spaces | 1 – 3/user | 2/user |
| Restricted spaces distribution deviation | $\delta = 0 \sim 0.8$ | $\delta = 0.5$ |
| Desired protection level | 10 – 50 | 30 |
| Number of queries | 30/user*hour | |

among users, the number of users, and users' desired protection level. We use two performance metrics which reflect the Quality of Service (QoS) of LBS. The *Average Cloaking Resolution* is the average diameter of all reported cloaking regions (recall that we use round cloaking regions). A smaller diameter indicates a higher resolution and thus likely to result in higher QoS. The *Public Area Proportion* is the percentage of non-cloaking areas on the whole cloaking map. A larger public area proportion could mean less cloaking is needed. Note that this proportion is always 0 for the footprint cloaking strategy, due to the fact that this strategy requires the user's location to be cloaked at any time. The default experiment parameters are showed in **Table 2**.

### 4.1 Effect of Number of Restricted Spaces

In order to simulate the regular moving pattern of human, users are simulated to move randomly between their restricted spaces and detour to random public locations like supermarket and shopping mall. The number of public locations visited by each user is normally distributed with a mean of 3 and standard deviation of 1.5. The footprint data used in footprint cloaking is also simulated this way by accumulating users' continuous movement for 36 hours following the same pattern. The number of restricted spaces owned by each user is normally distributed with a standard deviation ($\delta$) of 0.5 and then rounded to the nearest integer. We adjust the mean number of restricted space from 1/user to 3/user while using default value for other parameters. The result is plotted in **Fig. 10** (a) and (e). The proposed methods significantly outperform the footprint cloaking strategy, which is generally not affected by the number of restricted spaces. Note that the performance of the cloaking strategy with heuristic algorithm is very close to the one with optimal algorithm, indicating that the proposed heuristic algorithm is very effective in generating high quality cloaking regions.

When each user owns 1 restricted space the performance of our strategy is the best, since it very easy to find users whose restricted spaces are close to each other and generate small cloaking regions. But as the number of restricted spaces increases, the performance drops. This is due to the fact that it becomes harder and harder to find a set of users with similar distribution of restricted spaces. However, the proposed strategy outperforms the footprint cloaking strategy regardless of the number of restricted spaces.

### 4.2 Effect of Distribution of Restricted Spaces

In general, the number of restricted spaces owned by each user is vastly different. A very small number of people may own more properties than the majority. This experiment is to evaluate how this unevenness of ownership impacts on the cloaking performance. Here we fix the mean number of restricted space to 3/user while adjust the standard deviation ($\delta$) from 0 to 0.8. Default value is used for other parameters. In other words, the standard deviation of the number of restricted spaces in a particular area represents the degree of user diversity in terms of restricted spaces ownership. Figure 10 (b) and (f) illustrate the

performance of the compared techniques. The experiment result shows that the proposed strategies achieved the highest map quality when the deviation is small. In other words, the proposed strategy prefers regions where users basically have similar number of restricted spaces. Similarly, in this experiment the heuristic algorithm demonstrates good performance even comparing with the optimal algorithm.

It is interesting that the quality of map reduces as the ownerships become more skew, but the decrease is not as dramatic as affected by factors studied in the other sections. This indicates that the proposed strategy is not highly sensitive to diversified users under the settings of this experiment, at least not as sensitive as to the other three factors. Again, the proposed strategy outperforms the footprint cloaking significantly.

### 4.3 Effect of Number of Users

In this study, we increase the number of users from 5,000 to 25,000 to simulate the impact of the number of users on the proposed strategy. The default settings are used for other parameters. Figure 10 (c) and (g) show that when the number of users increases, the performance of all these cloaking-based strategies improves including the proposed algorithm and the footprint cloaking. This can be explained by the fact that having more users means a better chance to find users with similarly distributed trajectories as well as restricted spaces. Again, the proposed strategy outperforms footprint cloaking due to the fact that even using footprint may generate smaller cloaking regions, but using restricted spaces cloaking, a large portion of locations can be reported as is with highest accuracy.

### 4.4 Effect of Desired Protection Level

We adjust user's desired protection level ($k$-value) from 10 to 50 while use default value for other parameters. As argued in Section 2 and 3, user's protection level determines the size of their cloaking sets. A larger set guarantees a higher degree of protection, but it tends to result in larger cloaking regions. And the large the cloaking set is, the hard it is to generate smaller cloaking regions for the set. This dilemma is confirmed in Fig. 10 (d) and (h). The performance of all algorithms deteriorates as the level of protection increases. However, the decrease of average location resolution using our strategy is far below that of the footprint cloaking, which drops almost linearly to the protection level.

The above experiments show that the proposed strategy outperforms footprint cloaking in terms of average location resolution under our experiment settings. Based on this result, we conclude that the proposed restricted space cloaking can not only achieves full location privacy protection and support user self-cloaking without relying on any trusted third party, but also provides user with higher quality of service than existing techniques.

## 5. Related Works

Privacy-aware uses of Location-based Services have been intensively studied in the past decade. There are two different types of privacy concern, *anonymous service uses* (also called *query privacy* in some literature) and *location privacy*. The former is to prevent the user of a service from being identified, whereas the latter, the subject inside a disclosed location from being identified. These proposed techniques can be classified into a few categories:

1) Location Cloaking [1], [2], [3], [4], [5], [6], [7]. Most techniques in this category are design to support anonymous uses of services, which we have discussed in Section 1. Location cloaking achieves privacy protection at the cost of location resolution, but it is still regarded as a highly practical solution due to its simplicity and effectiveness. The proposed RSC technique complements the existing location cloaking research in enabling full location privacy protection and self-cloaking without requiring users to disclose their location to a central server or nearby users for cloaking.

2) Trajectory Perturbation. A user's time-series locate updates create a trajectory. One can associate each update with a different pseudonym, but successive location samples are highly correlative and could be re-linked using trajectory tracking methods (e.g., Multi-Target Tracking [18]). The work [19] first considered this problem and proposed a concept called *mix zone*.

A mix zone is a spatial region in which a mobile node does not report its location in order to confuse potential adversaries. Nodes in a mix zone exchange their pseudonyms to make it hard for an adversary to link incoming and outgoing paths of these nodes. While this approach relies on pre-defined spatial regions for pseudonym exchange, the path confusion technique [20] allows nodes to switch their pseudonyms when their paths are within some threshold. These approaches reduce, but cannot prevent, location privacy risks. A partial trace, or just a single location sample, can be sufficient for an adversary to identify a user, thus knowing his/her whereabouts.

Another track of work aimed at trace hiding is using fake locations or trajectories, i.e., dummies (e.g., Refs. [8], [9], [10], [21], [22]). For each location submitted to a service provider, it is accompanied by certain false dummies, which are generated to simulate the movement of mobile nodes. By making certain faked traces, the trace of a service user is under $K$-anonymity protection. This approach does not guarantee that a service user cannot be identified. For example, the adversary can identify the dummy trace as a fake if a sample false dummy is located inside a non-habitant region such as a lake, or the trace passes through multiple spatial regions that exclusively belong to different users. Under these circumstances, a faked trace is compromised and the real-world identity of a user might be revealed.

3) Non-location exposure approaches [12], [13], [23]. This type of techniques lets users download location-based information from a server without having to report their location. Such techniques usually apply the theory of Private Information Retrieval (PIR) [24] to prevent an adversary from deriving the user's location based on the downloaded data. This strategy protects a user's location privacy to its maximum extent, but in general users need to download a large amount of data, the amount of which may be prohibitively expensive to mobile users. For example, the technique [23] requires a user to download the square root of the total number of data items stored at the server.

## 6. Conclusion

Restricted space identification is arguably the most realistic way for an adversary to discover the real-world identities of the users of location-based services. As a practical solution to this privacy threat, location cloaking has been studied intensively in the past decade. In this paper, we show that existing location cloaking techniques do not give users full location privacy protection and this problem arises from the fact that these schemes cloak every location users report. Protecting every location makes it possible for an adversary to link a set of users to a sequence of locations or trajectories, which is an even more serious privacy leakage.

In contrast to existing techniques, we propose to cloak a user's location only when the location belongs or is close to a restricted space. If a user is inside a public region such as high way and shopping mall which cannot be directly linked to a small set of individual, the user's location can be safely reported as is. We prove that this approach, by not cloaking a user's every location, achieves full location privacy that is not available from existing techniques. With this basic idea in place, we consider the problem of allowing users to cloak their location by themselves, without relying on any trusted third party such an anonymizer, and propose the concept of cloaking map to support such self-cloaking. We have also discussed in details the challenges and solutions of producing cloaking map with high quality. The performance of these solutions is evaluated through simulation with realistic synthetic data, and compared with state-of-the-art cloaking technique.

### References


[1] Gruteser, M. and Grunwald, D.: Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking, *Proc. MobSys'03*, pp.31–42 (2003).

[2] Gedik, B. and Liu, L.: A Customizable K-Anonymity Model for Protecting Location Privacy, *Proc. 25th International Conference on Distributed Computing Systems*, Columbus, Ohio, USA (2008).

[3] Chow, C.-Y., Mokbel, M.F. and Liu, X.: A peer-to-peer spatial cloaking algorithm for anonymous location-based service, *Proc. 14th Annual ACM International Symposium on Advances in Geographic Information Systems*, Arlington, Virginia, USA (2006).

[4] Vu, K., Zheng, R. and Gao, J.: Efficient algorithms for k-anonymous location privacy in participatory sensing, *Proc. IEEE INFOCOM* (2012).

[5] Mokbel, M., Chow, C. and Aref, W.: The new casper: Query processing for location services without compromising privacy, *Proc. 32nd International Conference on Very Large Data Bases*, *VLDB Endowment*, pp.763–774 (2006).

[6] Xu, T. and Cai, Y.: Exploring Historical Location Data for Anonymity Preservation in Location-based Services, *IEEE INFOCOM*, pp.547–555, Phoenix, AZ (2008).

[7] Xu, T. and Cai, Y.: Feeling-based Location Privacy Protection for Anonymity Preservation in Location Based Service, *Proc. 16th ACM Conference on Computer and Communication Security*, pp.348–357 (2009).

[8] Kido, H., Yanagisawa, Y. and Satoh, T.: Protection of Location Privacy using Dummies for Location-based Services, *Proc. 21st International Conference of Data Engineering Workshops*, p.1248 (2005).

[9] Lu, H., Jensen, C.S. and Yiu, M.L.: Pad: Privacy-area Aware Dummy-based Location Privacy in Mobile Services, *Proc. ARES* (2010).

[10] Liu, X., Liu, K., Guo, L., Li, X. and Fang, Y.: A Game-Theoretic Approach for Achieving k-Anonymity in Location Based Service, *Proc. IEEE INFOCOM* (2013).

[11] Hong, J.I.: An Architecture for Privacy-Sensitive Ubiquitous Computing, *Proc. MobSys'04*, pp.177–189 (2004).

[12] Hu, H. and Xu, J.: Non-exposure Location Anonymity, *Proc. 25th International Conference of Data Engineering*, pp.1120–1131 (2009).

[13] Li, X.Y. and Jung, T.: Search Me If You Can: Privacy-preserving Location Query Service, *Proc. IEEE INFOCOM* (2013).

[14] GIS Maps and other geographic data of the City of Ames, available from ⟨http://www.cityofames.org/index.aspx?page=1107⟩.

[15] 2010 Census and Household Data by the United States Census Bureau, available from ⟨http://factfinder2.census.gov/faces/nav/jsf/pages/index.xhtml⟩.

[16] Hammer, R.B., Stewart, S.I., Winkler, R., Radeloff, V.C. and Voss, P.R.: Characterizing spatial and temporal residential density patterns across the U.S. Midwest, 1940-1990, *Landscape and Urban Planning*, Vol.69, pp.183–199 (2004).

[17] Leonhardt, U. and Magee, J.: Security Considerations for a Distributed Location Services, *Journal of Networks and Systems Management*, Vol.6, No.1, pp.51–70 (1998).

[18] Reid, D.: An Algorithm for Tracking Multiple Targets, *IEEE Trans. Automatic Control*, Vol.24, No.6, pp.843–854 (1979).

[19] Beresford, A.R. and Stajano, F.: Location Privacy in Pervasive Computing, *IEEE Security and Privacy*, Vol.2, No.1, pp.46–55 (2003).

[20] Hoh, B. and Gruteser, M.: Protecting Location Privacy Through Path Confusion, *Proc. IEEE/CreateNet Intl. Conference on Security and Privacy for Emerging Areas in Communication Networks*, pp.194–205 (2005).

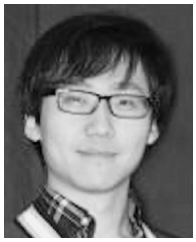
[21] Kido, H., Yanagisawa, Y. and Satoh, T.: An Anonymous Communication Technique using Dummies for Location-based Services, *Proc. An Anonymous Communication Technique using Dummies for Location-based Services*, pp.88–97 (2005).

[22] Meyerowitz, J. and Choudhury, R.R.: Hiding stars with fireworks: Location privacy through camouflage, *Proc. ACM MobiCom'09*, pp.345–356 (2009).

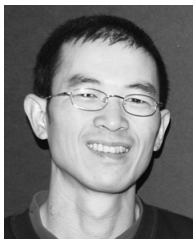
[23] Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C. and Tan, K.-L.: Private Queries in Location-Based Services: Anonymizers are Not Necessary, *Proc. ACM SIGMOD'08*, pp.121–132 (2008).

[24] Chor, B., Goldreich, O., Kushilevitz, E. and Sudan, M.: Private Information Retrieval, *Proc. IEEE Symposium on Foundations of Computer Science*, pp.41–50 (1995).

[25] Wing, M.G., Eklund, A. and Kellogg, L.D.: Consumer-Grade Global Positioning System (GPS) accuracy and reliability, *Journal of Forestry*, Vol.103, No.4, pp.169–173 (2005).

**Guolei Yang** is a Ph.D. student in Computer Science at Iowa State University since 2012. He received his Bachelor degree from Communication University of China in 2008, and Master degree from Peking University in 2011. His research interests include database, data mining, and computer security.



**Ying Cai** received his bachelor and master degree from Xi'an Jiaotong University, China. He graduated from University of Central Florida with a Ph.D. in computer science in 2002. Dr. Cai joined the Department of Computer Science at Iowa State University in 2003 and is now an associate professor there. His current research interests include cloud computing, privacy and security, mobile computing, and multimedia systems.