Achieving k-anonymity in Privacy-Aware Location-Based Services

Ben Niu*, Qinghua Li†, Xiaoyan Zhu*, Guohong Cao‡ and Hui Li*
*National Key Laboratory of Integrated Networks Services, Xidian University, China
†Department of Computer Science and Computer Engineering, University of Arkansas, AR, USA
‡Department of Computer Science and Engineering, The Pennsylvania State University, PA, USA
*xd.niuben@gmail.com, †qinghual@uark.edu, *{xyzhu, lihui}@mail.xidian.edu.cn, ‡gcao@cse.psu.edu

***Abstract*—Location-Based Service (LBS) has become a vital part of our daily life. While enjoying the convenience provided by LBS, users may lose privacy since the untrusted LBS server has all the information about users in LBS and it may track them in various ways or release their personal data to third parties. To address the privacy issue, we propose a Dummy-Location Selection (*DLS*) algorithm to achieve k-anonymity for users in LBS. Different from existing approaches, the *DLS* algorithm carefully selects dummy locations considering that side information may be exploited by adversaries. We first choose these dummy locations based on the entropy metric, and then propose an *enhanced-DLS* algorithm, to make sure that the selected dummy locations are spread as far as possible. Evaluation results show that the proposed *DLS* algorithm can significantly improve the privacy level in terms of entropy. The *enhanced-DLS* algorithm can enlarge the cloaking region while keeping similar privacy level as the *DLS* algorithm.**

I. Introduction

With the rapid development of mobile devices and social networks, Location-Based Service (LBS) has become a vital part in our daily activities in recent years. With smartphones or tablets, users can download location-based applications from Apple Store or Google Play Store. With the help of these applications, users can easily send queries to LBS servers and obtain LBSs related to some point of interests. For example, users can check the bus schedule, the price information of nearby restaurants or gas stations, etc.

By submitting LBS queries, users can enjoy the convenience provided by LBS. However, since the untrusted LBS server has all the information about users such as where they are at which time, what kind of queries they submit, what they are doing, etc., he may track users in various ways or release their personal data to third parties. Thus, we need to pay more attention to user's privacy.

To address the privacy issue, many approaches [1], [2] have been proposed in the literature over the past few years. Most of them are based on location perturbation and obfuscation, which employ well-known privacy metrics such as *k-anonymity* [3] and rely on a trusted third-party server. To achieve *k-anonymity*, a LBS related query is submitted to the LBS server via a centralized *location anonymizer* [4], [5], which enlarges the queried location into a bigger *Cloaking Region (CR)* covering many other users (e.g., $k-1$) geographically. As a result, it is hard for the untrusted LBS server to distinguish the user's real location from the other $k-1$ dummy locations. However, these approaches of using *k-anonymity* have some limitations. First, it heavily relies on the *location anonymizer*, which suffers from a single point of failure. If the adversary gains control of it, the privacy of all users will be compromised. There also exists a performance bottleneck, since all the submitted queries have to go through the *location anonymizer*. Second, although dummy locations can be used to achieve *k-anonymity*, how to select these locations is a challenge. Most of the existing approaches [6], [7], [8], [4], [9] assume that the adversary has no *side information* [10], [11], such as user's query probability related to location and time, and information related to the semantics of the query such as the gender and social status of the user, and then dummy locations are generated based on a random walk model [7], [4], or virtual circle/grid model [9]. Since some adversary (e.g., the LBS server) may have such *side information*, these dummy generation algorithms may not work well. For example, some improperly selected dummy locations may fall at some unlikely locations such as lakes, swamps, and rugged mountains, and can be easily filtered out by the adversary. Thus, it is hard to effectively guarantee the desired *k-anonymity*.

In this paper, we design *Dummy-Location Selection (DLS)* algorithms to achieve *k-anonymity* for users in LBS. Different from existing approaches, *DLS* carefully selects dummy locations considering that *side information* may be exploited by adversaries. We first choose these dummy locations based on the entropy metric [12], and then enhance the algorithm, called *enhanced-DLS*, by making sure that the selected dummy locations are spread far away. The major technical contributions of this paper are as follows.

- To protect user's location privacy against adversary with *side information*, we design an entropy-based *DLS* algorithm to achieve *k-anonymity* by carefully choosing dummy locations.
- We propose an *enhanced-DLS* algorithm, which considers both entropy and *CR* to maintain the entropy while ensuring that the selected dummy locations are spread as far as possible.
- We present a WiFi access point based solution to implement our idea. Analytical and simulation results show that our algorithms can achieve our objectives efficiently.

The rest of the paper is organized as follows. We discuss the related work in Section II. Section III presents some

preliminaries of this paper. We present the *DLS* and the *enhanced-DLS* algorithms in Section IV, together with some security analysis and a practical solution to implement our work. Section V shows the evaluation results. We conclude the paper in Section VI.

II. Related Work

Privacy issues in mobile social networks have been well studied in the literature (e.g., privacy in LBSs [13], [14], privacy in mobile sensing [15], [16], etc.). In this section, we review some existing research on user's privacy issues for LBSs.

A. Metrics for Location Privacy

Since a location can always be specified as a single coordinate, to quantify the location privacy, we should find out how accurately an adversary might infer about this coordinate. Based on this principle, several location privacy metrics have been proposed. Most of the existing metrics are uncertainty-based. Gruteser *et al.* [6] proposed to measure the ability of the adversary to differentiate the real user from others within the anonymity set. The ability of the adversary to link two pseudonyms of a particular user or distinguish the paths along which a user may travel has been investigated in [2] and [17], respectively. Therefore, a straightforward parameter to determine the privacy degree is the size of the anonymity set, for example, *k-anonymity* [3] (or variations like *l-diversity* [18] and *t-closeness* [19]), which tries to hide the real information of a user into other k-1 users. Based on this model, a number of follow-up works appear, such as [8], [18]. Recently, as a good measurement for the uncertainty of location privacy, entropy-based metrics have been adopted [2], [17], [20], [21], [22], [23]. Some other metrics [24], [25] are based on the estimation error of the adversary to quantify the location privacy. However, in our work, we do not introduce any location error on either the real location or the dummy locations. Thus, we choose entropy-based metric to quantify location privacy.

B. Protecting Location Privacy

Protecting user's location privacy in LBSs has received considerable attention over recent years. Among these Location Privacy Protection Mechanisms, location perturbation and obfuscation have been widely used. It protects location privacy through pseudomyzation, perturbation, adding dummies and reducing precision. In some early work [6], Gruteser *et al.* introduced *k-anonymity* into location privacy, which protects privacy by hiding user's location from the LBS server. More specifically, they design an adaptive interval cloaking algorithm which generates spatio-temporal cloaking boxes containing at least k_{min} users and use the boxes as the location sent to the LBS server. Later, *CliqueCloak* [26] was proposed as a personalized *k-anonymity* model in which users can adjust their level of anonymity. Unfortunately, most of these works still rely on a *location anonymizer* to enlarge the queried location into a bigger cloaking region, and hence the anonymizer becomes the central point of failure and the performance bottleneck.

To address this problem, Kido *et al.* [7] proposed to use dummy locations to achieve anonymity without employing anonymizer. However, they only concentrate on reducing the communication costs. Moreover, they employ a random walk model to generate dummy locations and it cannot ensure privacy when the server has some *side information*. Lu *et al.* [9] designed two dummy location generating algorithms called *CirDummy* and *GridDummy*, which achieve *k-anonymity* for mobile users considering the *privacy-area*. *CirDummy* generates dummy locations based on a virtual circle that contains user's location, while *GridDummy* is based on a virtual grid covering user's location. In these algorithms, the dummy generation is configurable and controllable, and the location privacy of a user can be controlled. Aside from the aforementioned location privacy protection mechanisms, policy-based approaches [27] and Cryptography primitive-based approaches [28] have also been investigated.

Different from existing work, the proposed scheme achieves *k-anonymity* by carefully generating dummy locations. It provides desired privacy level of *k-anonymity* for mobile users without relying on any trusted entity and can deal with adversaries with some *side information*.

III. Preliminaries

In this section, we first introduce some basic concepts used in this paper, and then introduce the adversary model. Finally, we present the motivation and the basic idea of our solution.

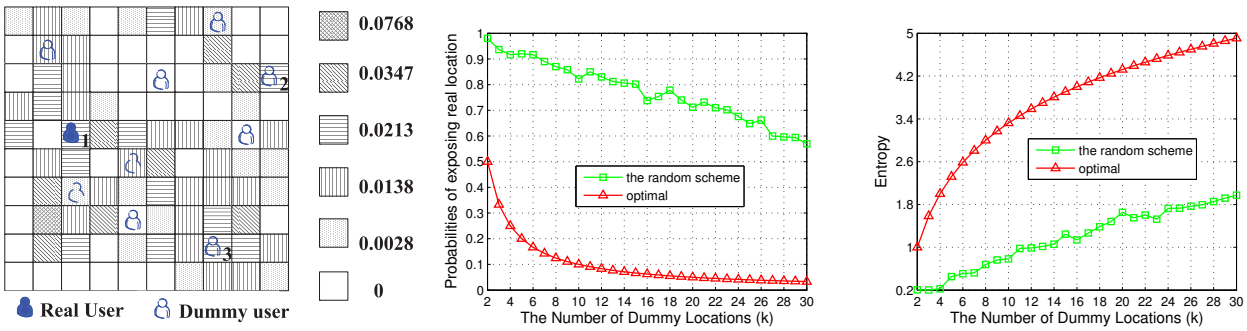
A. Basic Concepts

In this paper, the *side information* is limited to user's query probability at a particular location. Users can obtain two kinds of *side information* from our system: *partial information* and *global information*. The *partial information* represents the information collected by a user, and the *global information* represents all the query information in the system, i.e., all users' query probabilities at all locations. For a particular user, the ideal case is that he knows *global information*, and can take an optimal strategy to select dummy locations. A more realistic way is to retrieve the *side information* from his collection, even though the retrieved *side information* may be partial.

In this paper, we use entropy to measure the degree of anonymity. It can be seen as the uncertainty in determining the current location of an individual [12] from all the candidates. To compute the entropy, each possible location has a probability of being queried in the past, denoted by p_i, and the sum of all probabilities p_i is 1. Then, the entropy H of identifying an individual in the candidate set is defined as

$$H = -\sum_{i=1}^{k} p_i \cdot \log_2 p_i. \tag{1}$$

Our aim is to achieve the maximum entropy, i.e., the highest uncertainty to identify an individual from the candidate set. The maximum entropy is achieved when all the k possible



(a) Experiment setup (b) Experimental result - I (c) Experimental result - II

Fig. 1. The problem of random-based dummy selection

locations have the same probability $\frac{1}{k}$, where the maximum entropy will be $H_{max} = \log_2 k$.

B. Adversary Model

The goal of the adversary is to obtain sensitive information about a particular user. We consider two types of adversaries: *passive adversary* and *active adversary*. Any entity can be a passive adversary if he can monitor and eavesdrop on the wireless channels between entities or compromise users to obtain other users' sensitive information. A passive adversary can perform *eavesdropping attack* to learn extra information about a particular user. An active adversary can compromise the LBS server and obtain all the information that the server knows. In this work, we directly consider the LBS server as the active adversary. Then, he is able to obtain *global information* and monitor the current queries sending from the users. He can also obtain the historic data of a particular user as well as the current situation. Additionally, he knows the location privacy protection mechanism used in the system. Based on these information, he tries to infer and learn other sensitive information about the user.

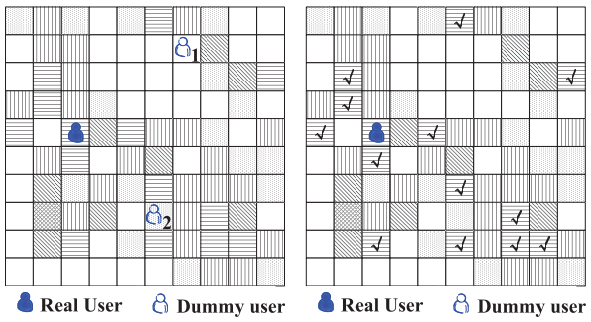
C. Motivation and Basic Idea

In existing LBS, a user submits a query to the LBS server, including the identifier, exact location, the interest and the query range, and then receives the corresponding reply from the server. To protect user privacy, one method is to use cloaking [4], [5], which has several weaknesses. The most important weakness is the *location anonymizer*, which is the bottleneck from both privacy and system performance points of view. Also, even though the k-1 users can be found nearby, it still reveals user's location privacy with a high probability since the chosen dummy users may be very close to the real user. Generating dummy locations can address these problems, but may lead to other problems, e.g., *how to choose the dummy locations*? Most existing works [7], [4] rely on random-based method or the random walk model. However, it is not a good way to protect user privacy against adversaries with *side information*, as shown in the following example.

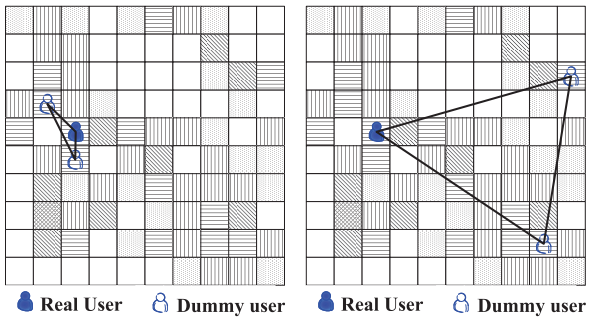
Fig. 1(a) shows the setting of the experiment. The area (location map) is divided into a grid of 10×10 cells. Different shades of the cell represent different query probabilities, which are generated based on the Borlange Data Set. This data set was collected over two years (1999-2001) as part of an experiment on traffic congestion that took place in Borlange (see [29] for more details). We consider two dummy generation schemes: the optimal scheme and the random scheme. The optimal scheme represents the ideal case in theory. In the random scheme, the dummy locations are chosen randomly. To send a query, a user generates k-1 dummy locations randomly and sends them together with its own location to the LBS server. Then, the user believes the probability of exposing the real location is $\frac{1}{k}$, which is the theoretical result of *k-anonymity*. However, since the server has *side information* about the query probabilities of locations in the map, the achieved privacy level is much less. The server can guess that the user is in a cell which has the highest query probability. With such *side information*, the server can infer the real location with a higher probability as $\frac{1}{k-k_d}$, where k_d represents the number of dummy locations that the server will filter out based on their low query probabilities. In the case shown in Fig. 1(a), since the query probabilities in locations 1, 2 and 3 are bigger than others, the server believes that the real location is between them, which means $k - k_d = 3$. As a result, the entropy drops significantly from $\log_2 k$ to $\log_2(k - k_d)$.

Fig. 1(b) indicates the probability of finding the user's real location by the server. We can easily see the difference between the random scheme and the optimal scheme. Fig. 1(c) illustrates the evaluation results by showing the entropy of the random scheme and the optimal scheme. The performance of the optimal scheme is always better since all the candidates have the same probability to be targeted as the real user's location. While in the random scheme, the entropy drops significantly. For example, when $k = 20$, the entropy drops from 4.32 to 1.65, that means, about $2^{4.32} - 2^{1.65} \approx 17$ dummy locations may be filtered out from the submitted 20 locations.

The general idea of our solution is to optimize the selection of dummy locations considering that the adversary may exploit some *side information*. We enhance user privacy from two aspects. First, we try to choose dummy locations with similar query probabilities. Second, if there are several candidates, the dummy locations should be spread as far as possible. This is



(a) Existing k-anonymity based approach (b) Our DLS algorithm where candidate cells are marked with $\surd$



(c) Dummies with smaller CR (d) Dummies with bigger CR

Fig. 2. Our main idea

based on the intuition that people always want to hide their real locations into a large area.

Fig. 2 further explains our basic idea. It is based on Fig. 1(a), where different shades of the cell represent different query probabilities. In this example, the user generates two dummy locations to achieve *k-anonymity* where $k = 3$. In the random scheme, as shown in Fig. 2(a), dummy locations are randomly assigned, and it is possible that dummy location 1 is assigned to a cell without any query in the past. As a result, the server may filter out this dummy location since it does not look like a real location. Then, *3-anonymity* cannot be ensured.

In our Dummy-Location Selection (*DLS*) algorithm, dummy locations are generated in cells which have similar query probabilities with the real location of the user. Fig. 2(b) shows all the candidate cells which are marked with $\surd$, and the two candidates are chosen among them.

This approach can achieve as high entropy as the optimal *k-anonymity*, but it has a problem. As shown in Fig. 2(c) and Fig. 2(d), both cases choose dummy locations with similar query probabilities, and hence both cases have similar entropy. However, in the solution in Fig. 2(c) dummy locations are too close to the real location. Certainly, we prefer the solution in Fig. 2(d) where the dummy locations are spread out. Thus, we need to design a scheme to carefully choose dummy locations among the cells which have similar query probabilities.

IV. Dummy-Location Selection Algorithms

In this section, we present our Dummy-Location Selection (*DLS*) algorithm and the *enhanced-DLS algorithm*, as well as their security analysis. Finally, we propose an AP-based method to collect the *side information* and address some implementation issues.

A. The DLS Algorithm

The main purpose of this algorithm is to generate a set of realistic dummy locations. Suppose the location map is divided into $n \times n$ cells with equal size. Each cell has a query probability based on the previous query history, which is denoted as

$$q_i = \frac{\#\ of\ queries\ in\ cell\ i}{\#\ of\ queries\ in\ whole\ map}, i = 1, 2, \cdots, n^2, \quad (2)$$

where

$$\sum_{i=1}^{n^2} q_i = 1.$$

Generally speaking, our *DLS* algorithm needs to search a big database to find an optimal set of dummy locations. Given a degree of anonymity k, besides the real location, we need to determine the other *k-1* cells to assign the dummy locations. The following shows how the *DLS* algorithm addresses this problem.

(i) As the first step, a particular user needs to determine a proper degree of anonymity k, which is closely related to the user's location privacy and the system overhead. Specifically, a bigger k leads to higher degree of anonymity but also higher overhead due to the cost incurred by the selected dummy locations.

(ii) The maximum entropy is achieved when the submitted k locations have the same probabilities to be treated as the real location on the server side. At the beginning of our *DLS* algorithm, the user needs to read all the obtained query probabilities and then sorts all cells by the order of the query probabilities. In the sorted list, if there are multiple cells which have the same query probability as the real location, we put half of them before and the other half after the real location. In the sorted list, the user chooses the k cells right before and the k cells right after the real location as $2k$ candidates. Then, the user derives m sets of cells, each with k cells. For each set, one cell is the real location, and the other $k-1$ cells are randomly chosen from the $2k$ candidates. The j^{th} ($j \in [1, m]$) set can be denoted as $\mathcal{C}_j = [c_{j1}, c_{j2}, \cdots, c_{ji}, \cdots, c_{jk}]$. Based on the original query probabilities of the chosen cells, the normalized query probabilities of the included cells can be denoted as $p_{j1}, p_{j2}, \cdots, p_{ji}, \cdots, p_{jk}$ and computed by

$$p_{ji} = \frac{q_{ji}}{\sum_{l=1}^{k} q_{jl}}, i = 1, 2, \cdots, k, \quad (3)$$

such that their sum is 1. The reason for choosing $2k$ locations as candidates of dummies is to increase the anonymity degree, and the size of this set can be changed according to user's requirement.

(iii) Now, we need to determine an optimal set to effectively achieve *k-anonymity* for the user. The privacy degree of

our solution is guaranteed by employing the entropy-based metric, which is widely used in measuring user's privacy. Specifically, for a particular chosen set $\mathcal{C}_j$, we compute the entropy by

$$H_j = -\sum_{i=1}^{k} p_{ji} \cdot \log_2 p_{ji}. \quad (4)$$

At last, the *DLS* algorithm outputs the set with the highest entropy:

$$\mathcal{C} = \arg\max H_j. \quad (5)$$

Algorithm 1: Dummy-Location Selection Algorithm

Input : query probabilities in history q_i, real location l_{real}, number of sets m, k
Output: an optimal set of dummy locations

1 sort cells based on their query probability;
2 choose $2k$ dummy candidates among which k candidates are right before l_{real} and k candidates are right after l_{real} in the sorted list;
3 **for** $(j = 1; j \leq m; j++)$ **do**
4 construct set $\mathcal{C}_j$ which contains l_{real} and $k-1$ other cells randomly selected from the $2k$ candidates;
5 compute the normalized probability p_{ji} for each cell c_{ji} in the set;
6 $H_j \leftarrow -\sum_{i=1}^{k} p_{ji} \cdot \log_2 p_{ji}$;
7 **end**
8 output $\arg\max H_j$;

Algorithm 1 shows the formal description of the *DLS* algorithm. It can provide *k-anonymity* efficiently and effectively. Although the *DLS* algorithm can achieve better degree of privacy in terms of entropy, as explained in Section III-C, when choosing dummy locations, it is better to spread these dummy locations far away. As a result, *DLS* can be enhanced so that dummy locations are spread into a larger area (i.e., bigger cloaking region (*CR*)).

B. The Enhanced-DLS Algorithm

To improve the privacy level, the *DLS* algorithm can be enhanced by considering both entropy and *CR*. Since two factors are considered, the dummy selection problem can be formulated as a Multi-Objective Optimization Problem (MOP). On the one hand, we want to maximize the required privacy level based on the metric of entropy. On the other hand, we want to maximize *CR* to spread the dummy locations as far as possible.

One fundamental question is how to measure the *CR*. Intuitively, the sum of the distances between pairs of dummy locations can be used to measure the *CR*, which is $\sum_{i \neq j} d(c_i, c_j)$ where $d(c_i, c_j)$ denotes the distance between cell c_i and c_j. However, it may not be as good as the product of the distances between pairs of dummy locations, which is $\prod_{i \neq j} d(c_i, c_j)$. We consider an example in Fig. 3. In this example, *A* is the real location of the user. *B* is chosen as a dummy location since it is the furthest location from *A*. Suppose we have two choices to assign the third dummy location, *C* and *D*. If we choose it based on the sum of the distances between pairs of dummy locations, we can choose either of them, because $CA + CB = DA + DB$. However, from the privacy point of view, we prefer *C* rather than *D* since it spreads the dummy locations further. As a result, instead of using the sum of the distances between pairs of dummy locations, we use their multiplications. In this case, $CA \cdot CB > DA \cdot DB$, and hence we choose C as the dummy location.

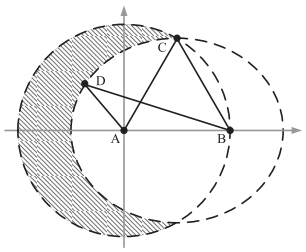


Fig. 3. The enhanced-DLS scenario

Let $\mathcal{C} = [c_1, c_2, \cdots, c_k]$ denote the set of real and dummy locations. The MOP can be described as

$$Max\{-\sum_{i=1}^{k} p_i \cdot \log_2 p_i, \prod_{i \neq j} d(c_i, c_j)\}, \quad (6)$$

where $c_i, c_j \in \mathcal{C}$, p_i and p_j denote the query probabilities of c_i and c_j, respectively.

It is hard to satisfy all objectives at the same time in MOP. For us, the primary goal is to confuse the adversary to target a particular location to a user. This objective is represented as

$$\mathcal{C} = \arg\max(-\sum_{i=1}^{k} p_i \cdot \log_2 p_i), \quad (7)$$

which is the basic condition used to choose a set of dummy locations to achieve a higher entropy. Then, we try to find the optimal combination of the k candidates, which are far away from each other. It can be denoted as

$$\mathcal{C} = \arg\max \prod_{i \neq j} d(c_i, c_j). \quad (8)$$

Based on the aforementioned analysis, we propose a heuristic solution for the MOP which first selects a redundant set of dummy locations to maximize entropy and then selects the final $k-1$ dummy locations out of the redundant set to maximize the *CR*.

Algorithm 2 gives the formal description of the *enhanced-DLS* algorithm. Following the approach in *lines 1-2* of Algorithm 1, we first choose $4k$ candidate dummy locations. Then following the approach in *lines 3-8* of Algorithm 1, we construct a smaller set of $2k$ candidate dummy locations out of the $4k$ candidates. This process aims to maximize the entropy. Among these $2k$ candidates, our next goal is to select $k-1$

dummy locations (c_1, c_2, $\cdots$, c_{k-1}) which can maximize the *CR*. Let c_{real} denote the cell where the user is currently in. Our heuristic solution chooses c_1, c_2, $\cdots$, c_{k-1} in order through $k-1$ rounds. c_1 is chosen in the first round, c_2 is chosen in the second round, and so on. In each round, each remaining candidate is assigned a certain *weight*, and the dummy of this round is chosen in such a way that each remaining candidate is chosen with a probability proportional to its weight. More formally, let x denote the number of remaining candidates in a round, and w_i denote the weight of candidate i. Then candidate c_i $(i = 1, ..., x)$ is chosen as a dummy in this round with probability $\frac{w_i}{\sum_{j=1,...,x} w_j}$. In the first round, $x = 2k$ and the weight of each candidate is its distance with c_{real}. We can see that cells far away from the real location have a higher probability of being chosen. Here, we do not directly choose the cell that is farthest from the real location because that approach may make it easier to guess the real location in some cases (e.g., the real location is in the central part of the area but all dummy locations are around boundaries). In the second round, $x = 2k-1$ and the weight of each candidate is the product of its distances with c_{real} and the already-selected dummy (which is c_1 here). Other rounds proceed similarly.

Algorithm 2: The enhanced-DLS algorithm

Input : query probabilities in history q_i, current cell c_{real}, number of sets m, and k
Output: an optimal set of dummy locations

follow *lines 1-2* of Algorithm 1 to choose $4k$ candidate dummy locations;
follow *lines 3-8* of Algorithm 1 to choose $2k$ candidate dummy locations $\hat{\mathcal{C}} = \{c_1, c_2, \cdots, c_{2k}\}$;
$\mathcal{C} \leftarrow \{c_{real}\}$;
for $(i = 1; i \leq k-1; i++)$ **do**
    choose c' as one candidate $c_j \in \hat{\mathcal{C}}$ in such a way that c_j is chosen with probability $\frac{\prod_{c_l \in \mathcal{C}} d(c_j, c_l)}{\sum_{c_j \in \hat{\mathcal{C}}} \prod_{c_l \in \mathcal{C}} d(c_j, c_l)}$;
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{c'\}$;
    remove c' from $\hat{\mathcal{C}}$;
end
output $\mathcal{C}$;

C. Security Analysis

In our algorithm, cryptography techniques such as the public key infrastructure (PKI) can be used to deal with *eavesdropping attacks* on the wireless channel between users and other entities. Our schemes can also resist from some other attacks, such as *colluding attacks* and *inference attacks*.

1) Resistance to Colluding Attacks: Passive adversary may collude with some users to learn extra information of other users, or collude with LBS server to predict sensitive information of legitimate users.

Definition 1. *A scheme is colluding attack resistant if the probability of successfully guessing the real location of a user among the k submitted locations does not increase with the size of colluding group.*

Theorem 1. *Our scheme is colluding attack resistant.*

Proof: We consider the case that colluding happens between a group of users. They want to guess the real location of user U out of the submitted k locations. In our schemes, each user only knows *partial information* in the system. When the colluding group contains only one user U_i, the obtained information includes the query probabilities she has collected, her current queries and the query history. Then she intercepts the k locations that user U sends to the server. Since each location in the intercepted set has the same query probability, she has no clue about the real location, which means the probability of successful guessing is $\frac{1}{k}$. Since she might have sent queries in locations with similar query probabilities before, she can get the intersection between her query history and the intercepted queries. The best case for her is that her query history can fully cover the intercepted set. However, since Equation 4 and 5 guarantee high uncertainty for each location, she cannot locate the real user even if she knows how the *DLS* and *enhanced-DLS* algorithms work. Therefore, she can only randomly guess the real location within the intercepted k locations. Similarly, when the colluding group has more members, they can still only randomly guess, which means the probability of successful guessing is still $\frac{1}{k}$. ■

One extreme case for the *passive adversary* is that he can get the *global information* by compromising the LBS server as well as all users. In this case, he actually becomes an *active adversary* and can perform *inference attack* as discussed in the following.

2) Resistance to Inference Attack: In this part of analysis, the LBS server is considered as an *active adversary*. He knows the query probabilities of the whole map, history queries and current queries which include the user's identifier, the mix of real and dummy locations, interest, query range, etc. Based on such information, the adversary can perform *inference attack* to gain the sensitive information about the user. More formally, the information in the *active adversary*'s hand includes: query probability q_i of each individual cell, the interest I, all the submitted k locations $l_1, l_2, \cdots, l_c, \cdots, l_k$. Let $p_G(event)$ denote the probability that the adversary can successfully guess if $event$ is true.

Definition 2. *A scheme is inference attack resistant if*

$$p_G\{l_i \in U | U \in C\} = p_G\{l_j \in U | U \in C\}\}, \quad \forall (0 < i \neq j \leq k). \tag{9}$$

Theorem 2. *Our schemes are inference attack resistant.*

Proof: For each submitted location l_i in the proposed *DLS* and *enhanced-DLS*, the successful guessing probability can be computed as

$$\begin{aligned} p_G\{l_i \in U | U \in C\} &= \frac{p_G\{l_i \in U \cap U \in C\}}{p_G\{U \in C\}} \\ &= \frac{q_i}{p_G\{U \in C\}}. \end{aligned} \tag{10}$$

Similarly, for the submitted location l_j, the successful guessing probability is

$$p_G\{l_j \in U | U \in C\} \quad = \quad \frac{q_j}{p_G\{U \in C\}} \tag{11}$$

Then for the pair of submitted locations l_i and l_j, Eq. 9 holds if

$$q_i = q_j, \ \forall i \neq j. \tag{12}$$

In our schemes, based on Eq. 4 and 5, the maximum entropy appears when all the potential dummy locations have the same probability, which guarantees Eq. 12. ■

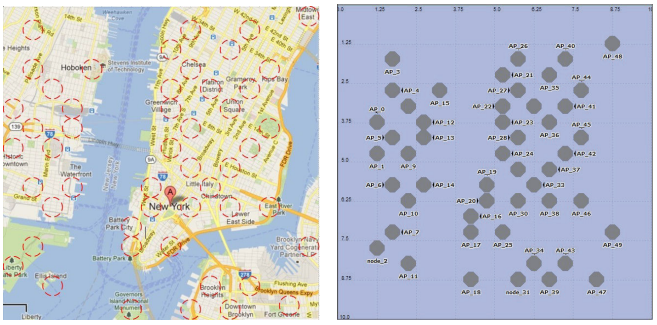
For the *active adversary*, he knows the proposed algorithms (*DLS* and *enhanced-DLS*) as well as all the history data of a particular user. He may try to reverse the algorithms, but this will fail as shown below. Let us recall the *step (ii)* of our *DLS* mentioned in Section IV-A, in which we choose $2k$ candidates to hide the query probability of the real location. Some of the candidates may have a little higher query probabilities while others may have a little lower probabilities. The chosen dummy locations of the *DLS* are randomly selected from the $2k$ candidates, which guarantees the uncertainty of the selection result. In our *enhanced-DLS*, we also employ this technique to guarantee the uncertainty. That is to say, different sizes lead to different dummy selection results. As the result, the LBS server cannot infer the real location by running our algorithms several times with different submitted locations.

D. Implementation Issues

In the proposed *DLS* and the *enhanced-DLS* algorithms, *side information* such as the user's query probability should be known so that the dummy locations can be carefully chosen to achieve *k-anonymity*. In this subsection, we address some implementation issues on how to obtain such *side information*.

One simple solution is to let the LBS server disseminate the users' query probabilities, so that users can get this information from a well known place. Since the users' query probabilities do not change too much, the dissemination interval can be very long, and thus the dissemination overhead is not high.

Another solution is to use WiFi Access Points (APs) to collect the query probabilities. The AP-based approaches [30], [31] have been widely used for LBSs in mobile environments. In our approach, a user can send queries anytime and anywhere, and the query is generated in the format of $\langle (x, y), I, r, others \rangle$, where (x, y) represents the exact location of the user, I represents the queried interest, r is the queried range and *others* include the user's identity, etc. We implement a query probability sharing scheme, this scheme starts when the user comes into a communication range of an AP, then the user can obtain the query probabilities from the AP and merge these information with his own. Additionally, he also needs to share the query probabilities in his hand, or part of them to the AP, based on the user's willingness. Through this way, the query probabilities at each user can be enlarged. The sharing phase happens when a user meets an AP, and the time interval between sharing can also be changed.



(a) Map of New York city (b) APs in OPNET simulator

Fig. 4. Simulation scenario

For each interest I, the original query probabilities in area within each AP are generated under normal distribution. The AP collects the queries within its coverage, and records them into different cells based on the location (x, y) in the query. When a new user joins, the sharing scheme also runs at the APs. After several times sharing with different users, the AP will have query probabilities of other APs. This is good for users since they can obtain more useful information from a single AP.

V. Performance Evaluations

In this section, we evaluate the performance of the proposed *DLS* algorithm and the *enhanced-DLS* algorithm.

A. Simulation Setup

In the simulation, 10000 mobile users are deployed in a $8km \times 8km$ area map of New York city, which is shown in Fig. 4(a). The Levy walk model [32], which has been shown to better describe the mobility patterns of human being [33], is used to generate synthetic user contact events. All the users move in the land area of the map. In this $64km^2$ map, about $23.5km^2$ is covered by sea. Red dashed circles in Fig. 4(a) represent WiFi access points with their communication ranges. Without loss of generality, we use the exact locations of several popular places, such as downtowns of NYC, Brooklyn and New Jersey, some shopping centers and bars since these areas are always covered by APs and with more users. Fig. 4(b) shows the distribution of the APs in our simulator.

There are several parameters used in our evaluation. k is related to *k-anonymity*, and is commonly set from 2 to 30. r is the radius of the queried area, which is set by the user.

We compare the proposed *DLS* and *enhanced-DLS* algorithms with four other schemes. The baseline scheme represents the dummy selection algorithm in [7], which randomly chooses dummy locations to protect privacy. The *CirDummy* and *GridDummy* schemes are the dummy selection schemes designed in [9], which achieve *k-anonymity* for mobile users. The optimal scheme shows the optimal results of *k-anonymity* in theory.

B. Evaluation Results

In our simulation, when a user issues a LBS query via an AP, the user and the AP exchange their collected *partial*

information. The simulation shows that, when the Levy walk mobility model is used, it needs about 4 hours for all the APs in the $8km \times 8km$ map to collect 99% of all the *partial information* for a single interest (i.e., to obtain roughly *global information*).

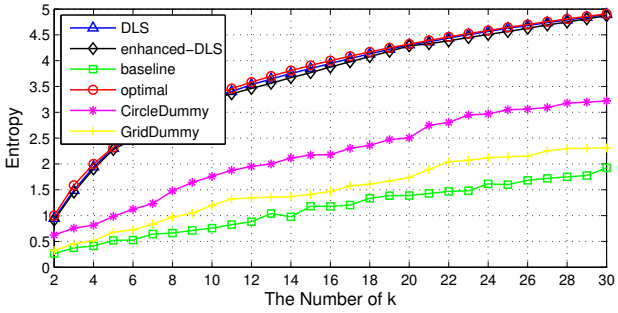
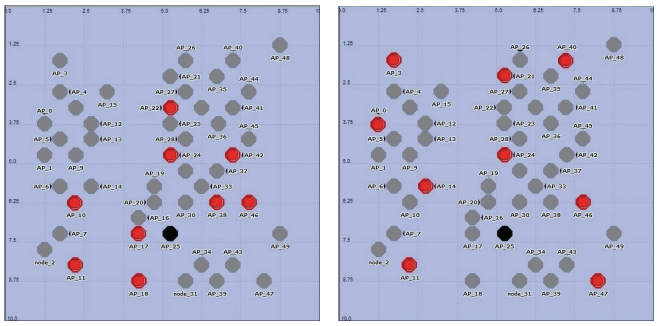


Fig. 5. Entropy vs. k

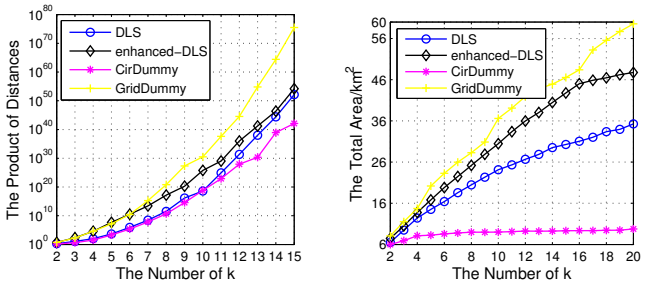
1) Privacy vs. k: We evaluate the relationship between k and the privacy level. Fig. 5 shows the privacy level in terms of entropy of different schemes. Generally, the entropy increases with k. Among these schemes, the optimal scheme has the highest entropy ($\log_2 k$) since all the submitted k locations have the same probability to be treated as the real user. The baseline scheme is the worst since it ignores that the adversary may exploit some *side information* (e.g., query probabilities). As a result, the chosen dummy locations may fall into some cells with very low query probabilities, and are filtered out by the adversary. The performance of *GridDummy* is close to the baseline scheme, since *GridDummy* chooses dummy locations as the vertices of a grid ($\sqrt{k} \times \sqrt{k}$) which are fixed once the map is chosen, and thus its entropy depends on the current query probabilities in the map. The *CirDummy* scheme performs a little better than the baseline scheme. The reason is that all the chosen dummy locations are always within a virtual circle, and the variations of the query probabilities within a small region do not change too much. Compared with the baseline scheme, *GridDummy*, and *CirDummy*, *DLS* and *enhanced-DLS* can achieve much higher privacy levels which are similar to the optimal scheme, because dummy locations in our algorithms are selected from cells with similar query probabilities to guarantee high entropy. Comparing our *DLS* with *enhanced-DLS*, we can see that the entropy of *DLS* is a little bit better than *enhanced-DLS*. This is because *enhanced-DLS* sacrifices some entropy to maximize *CR*.

2) Product of Distances vs. k: We first show two snapshots of two different dummy location selections. Fig. 6(a) shows the result of the baseline scheme. Comparing with the result of *enhanced-DLS* algorithm in Fig. 6(b), it is obvious that locations in Fig. 6(a) are close to each other, and the *CR* they covered is also small. Then, we evaluate the effect of k on the distance product of each pair of users, as well as the queried areas. The results are shown in Fig. 7. This validates the effectiveness of our *enhanced-DLS* algorithm, since all the submitted locations (including both real location and dummy locations) can affect the distance product of each pair of users



(a) Result of the baseline scheme (b) Result of our scheme

Fig. 6. Snapshots of Dummy-Location Selection. Region in black represents the location of real user, and red regions represent dummy locations.



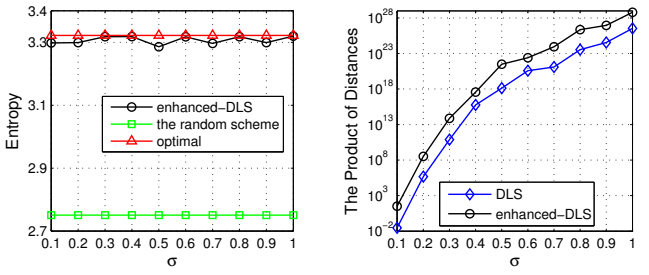
(a) The product of distances vs. k (b) Queried area vs. k

Fig. 7. Effect of our enhanced-DLS algorithm

and the size of the covered area. For simplicity, a user in a particular cell is considered as the user in the center of the cell. We note that the *GridDummy* algorithm has the largest product of distances and queried area, since in this algorithm dummy locations are generated to cover as much portion of the map as possible. However, its overall privacy level is still low as shown in Fig. 5. The *CirDummy* algorithm performs much worse, since it depends on a randomly chosen radius of the virtual circle. Generally, bigger radius leads to bigger product of distances as well as the queried area, and vice versa. We ignore the queried area of the baseline scheme in our comparisons, since it just randomly chooses dummy locations in the whole map, and does not consider the *side information*.

In Fig. 7(a), we measure the distance product of each pair of the chosen dummy locations. Compared with *DLS*, the *enhanced-DLS* algorithm performs much better. Fig. 7(b) indicates the queried areas of different schemes. In the simulations, we use the queried range $r = 1km$ as an example. With the increase of k, both of the *DLS* and *enhanced-DLS* algorithms cover bigger areas. However, the *enhanced-DLS* algorithm has better performance, since it uses a greedy method to spread the dummy locations as far as possible. As a result, it can enlarge the queried area by almost 20% or more when $k > 12$.

3) Privacy vs. σ: Since users can get query probabilities from a single AP or several APs, we introduce a parameter σ to describe the obtained *partial information* over the *global information*. In our evaluation, we use 50 APs, and $\sigma = 0.5$ represents that the user knows query probabilities from 25 APs. We show the effect of σ on entropy and the product of



(a) Entropy vs. σ (b) The product of distances vs. σ

Fig. 8. Effect of partial information

distances in Fig. 8. In the following simulations, for simplicity, we let $k = 10$, $r = 1km$, and change σ from 0.1 to 1.0. Fig. 8(a) shows the effect of σ on entropy. As can be seen, the *enhanced-DLS* algorithm has similar entropy with the optimal scheme, and both are better than the baseline scheme. That is because we always choose cells with similar query probabilities. Since the *DLS* algorithm has similar entropy with the *enhanced-DLS* algorithm, it is not shown in Fig. 8(a). In Fig. 8(b), different σ affects the distance product. When there are only less number of cells, we have to choose candidates from them even though they are close to each other. The evaluation results indicate that the *enhanced-DLS* algorithm outperforms the *DLS* algorithm.

VI. Conclusions

In this paper, we proposed a Dummy-Location Selection (*DLS*) algorithm to protect user's location privacy against adversaries with *side information*. Based on the obtained *side information* and the entropy metric, *DLS* carefully selects the dummy locations to achieve the optimal level of *k-anonymity*. We also proposed an *enhanced-DLS* algorithm which considers both entropy and cloaking region (*CR*) to maintain entropy and try to ensure that the selected dummy locations are spread as far as possible. Finally, we presented an AP-based solution to implement our idea. Evaluation results show that the proposed *DLS* algorithm can significantly improve the privacy level in terms of entropy. The *enhanced-DLS* algorithm can enlarge the cloaking region while keeping similar privacy level as the *DLS* algorithm.

Acknowledgment

This work was supported by National Natural Science Foundation of China under Grant 61003300, Fundamental Research Funds for the Central Universities under Grant K5051201041, and China 111 Project under Grant B08038. The work of Dr. Hui Li was supported by the National Project 2012ZX03002003-002, 863 Project 2012AA013102, IRT1078 and NSFC 61170251.

References

[1] A. Beresford and F. Stajano, "Location privacy in pervasive computing," *Pervasive Computing, IEEE*, vol. 2, no. 1, pp. 46 – 55, jan-mar 2003.
[2] T. Jiang, H. J. Wang, and Y.-C. Hu, "Preserving location privacy in wireless lans," in *ACM MobiSys 2007*.
[3] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002.
[4] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: query processing for location services without compromising privacy," in *ACM VLDB 2006*.
[5] C.-Y. Chow, M. F. Mokbel, and W. G. Aref, "Casper*: Query processing for location services without compromising privacy," *ACM Trans. Database Syst.*, vol. 34, no. 4, 2009.
[6] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *ACM MobiSys 2003*.
[7] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proceedings of International Conference on Pervasive Services*, 2005, pp. 88 – 97.
[8] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *IEEE ICDCS 2005*.
[9] H. Lu, C. S. Jensen, and M. L. Yiu, "Pad: privacy-area aware, dummy-based location privacy in mobile services," in *ACM MobiDE 2008*.
[10] C. Y. Ma, D. K. Yau, N. K. Yip, and N. S. Rao, "Privacy vulnerability of published anonymous mobility traces," in *ACM MobiCom 2010*.
[11] X. Liu, K. Liu, L. Guo, X. Li, and Y. Fang, "A game-theoretic approach for achieving k-anonymity in location based services," in *IEEE INFOCOM 2013*.
[12] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Proceedings of the 2nd international conference on Privacy enhancing technologies*, 2003, pp. 41–53.
[13] Z. Zhu and G. Cao, "Applaus: A privacy-preserving location proof updating system for location-based services," in *IEEE INFOCOM 2011*.
[14] K. Shin, X. Ju, Z. Chen, and X. Hu, "Privacy protection for users of location-based services," *Wireless Communications, IEEE*, vol. 19, no. 1, pp. 30–39, 2012.
[15] Q. Li and G. Cao, "Providing privacy-aware incentives for mobile sensing," in *IEEE PERCOM 2013*.
[16] ——, "Efficient privacy-preserving stream aggregation in mobile sensing with low aggregation error," in *ACM PETS 2013*.
[17] J. Meyerowitz and R. Roy Choudhury, "Hiding stars with fireworks: location privacy through camouflage," in *ACM MobiCom 2009*.
[18] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: privacy beyond k-anonymity," in *IEEE ICDE 2006*.
[19] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *IEEE ICDE 2007*.
[20] B. Niu, X. Zhu, H. Chi, and H. Li, "3plus: Privacy-preserving pseudo-location updating system in location-based services," in *IEEE WCNC 2013*.
[21] B. Niu, X. Zhu, X. Lei, W. Zhang, and H. Li, "Eps: Encounter-based privacy-preserving scheme for location-based services," in *IEEE GLOBECOM 2013*.
[22] X. Zhu, H. Chi, B. Niu, W. Zhang, Z. Li, and H. Li, "Mobicache: When k-anonymity meets cache," in *IEEE GLOBECOM 2013*.
[23] B. Niu, Z. Zhang, X. Li, and H. Li, "Privacy-area aware dummy generation algorithms for location-based services," in *IEEE ICC 2014*.
[24] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion," in *IEEE SECURECOMM 2005*.
[25] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *IEEE Security and Privacy 2011*.
[26] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, Jan. 2008.
[27] W3C. (2011, Apr.) Platform for privacy preferences (p3p) project. [Online]. Available: http://www.w3.org/P3P/
[28] I. Bilogrevic, M. Jadliwala, K. Kalkan, J.-P. Hubaux, and I. Aad, "Privacy in mobile computing for location-sharing-based services," in *Springer PETS 2011*.
[29] E. Frejinger, "Route choice analysis: data, models, algorithms and applications," Ph.D. dissertation, Lausanne, 2008.
[30] S. Saroiu and A. Wolman, "Enabling new mobile applications with location proofs," in *ACM HotMobile 2009*.
[31] W. Luo and U. Hengartner, "Veriplace: a privacy-aware location proof architecture," in *ACM GIS 2010*.
[32] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong, "On the levy-walk nature of human mobility," in *IEEE INFOCOM 2008*.
[33] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "Slaw: A new mobility model for human walks," in *IEEE INFOCOM 2009*.