LPPS: Location Privacy Protection for Smartphones

Hongli Zhang, Zhikai Xu, Xiangzhan Yu
School of Computer Science Engineering,
Harbin Institute of Technology, Harbin 150001, China
Email: {zhanghongli,xuzhikai,yuxiangzhan}@nis.hit.edu.cn

Xiaojiang Du
Department of Computer and Information Sciences,
Temple University, Philadelphia, PA 19122, USA
Email: dux@temple.edu

***Abstract*—Location-based service (LBS) is useful for many applications. However, LBS has raised serious concerns about users' location privacy. Utilizing the computation and storage capacity of smart phones, we propose a novel system architecture, called Location Privacy Protection for Smartphone (LPPS), to provide a privacy-preserving top-k query. LPPS does not rely on a trust third party (TTP), nor does it requires LBS servers to change their business model. The main idea of LPPS is to rank the Points of Interests (POIs) on the client side of the application using a small amount of metadata and then to make a request to the LBS server for real-time and detailed information about the POIs. Based on LPPS, we propose a novel metric called location indistinguishability to evaluate the privacy level of users in the proposed scheme. Then, we propose two dummy-POI selection algorithms to generate a superset of the actual top-k POIs when the query cannot meet the privacy requirement. Our experimental results demonstrate the validity and practicality of the proposed schemes.**

I. Introduction

With the rapid development of GPS-enabled mobile devices and wireless communication technology, many amazing Location-based Service (LBS) applications, such as Google Maps, Foursquare, Yelp, and Dianping, have been introduced to the public. LBS is becoming an essential part of daily life. For example, with LBS applications, users can easily get information about nearby Points of Interest (POIs), e.g., the rating of restaurants and the bus stops in the vicinity. However, the potential abuse of location information by unauthorized entities is evolving into a serious concern.

Although many approaches have been proposed to address the issue of location privacy protection, none of them has yet been adopted on a mass scale. We propose two possible causes for the lack of adoption: (i) the server provider may be unwilling to take the risk of redesigning existing system architecture to meet the requirements of the approaches; (ii) the server provider may worry about that these approaches might reduce quality of service. It can be observed that the researchers are aware of the limitations of the approaches. However, due to pre-smartphone's limited computational ability, they can only offload the burden of sophisticated computations to LBS servers(e.g.,[1], [2], [3], [4]) or introduce an ideal trusted third-party (TTP)(e.g.,[5], [6]). Fortunately, as technology has advanced in the past few years, smart-phones, which are often faster than desktops from a decade ago, have become more popular. Thus, it's rational to exploit novel location privacy protection mechanism that utilizes the computation and storage ability of smart-phones.

In this paper, we propose a novel system architecture, called LPPS, to provide a privacy-preserving top-k query. LPPS does not rely on TTP, nor does it requires the LBS server to change its business model. The main idea is to rank the POIs on the client side of the application using a small amount of metadata from the server, and then to make a request to the LBS server for the real-time and detailed information about the top-k results. The user's precise location data is never transmitted outside the device. Based on LPPS, we propose a novel metric called location indistinguishability to evaluate the privacy level of users in the proposed scheme. Based on the metric, we propose two dummy-POI selection algorithms to generate a superset of the actual top-k POIs when the query cannot meet the privacy requirement. Thereafter, the query made to the LBS service contains not only the top-k POIs, but also the selected dummy-POIs. Compared with existing approaches, our methods have the following features: (i) our methods carefully select dummy locations considering that side information may be exploited by adversaries; (ii) our methods can reduce the communication cost while maintaining a certain protection level for user's privacy.

The contributions of this paper are summarized as follows:

- We exploit the computation ability of the smart phones, and design a novel system architecture, called LPPS, to provide a privacy-preserving top-k query. Our system does not rely on TTP, nor does it requires LBS to change the existing business model. Thus, it massively increases the scalability of the LBS system.
- We design two dummy POI selection algorithms (intersection based algorithm, and maximum entropy based algorithm) to generate a superset of actual top-k POIs when the actual query cannot meet the user's privacy requirement. Both of the algorithms can reduce the communication cost while continuing to protect the user's privacy.
- We conduct extensive experiments to evaluate the performance of our methods using a real-world dataset.

The rest of the paper is organized as follows. Section II discusses the related work, Section III introduces the system model. Section IV introduces the privacy metric. We present our dummy generation algorithms in Section V. Section VI and VII show the security analysis and evaluation results, followed by the concluding remarks in Section VIII.

978-1-4799-6664-6/16/$31.00 ©2016 IEEE

II. Related Work

Many approaches have been proposed to address the location privacy in LBS. Most of them rely on a trusted third party (TTP), usually termed *location anonymizer*, which is required between the user and LBS server (e.g., [1], [2], [3], [**?**]). When a user submits a query, the location anonymizer blurs the user's real location into spatial regions to meet user's privacy requirement and reports the spatial region to LBS server. However, the TTP may become a bottleneck for both system performance and privacy. There also exist some encryption-based approaches (e.g., [5], [6]). With these approaches, the LBS server does not learn much about the user's query and location, though it can still reply the user's query. However, the implementation of these methods requires the LBS server to change its existing business model and implement their solution. The lack of incentives for LBS to do so has, to date, made such methods impractical.

Another technique proposed to protect user's location privacy is using dummy queries together with a real query (e.g., [7], [8], [9]). However, all these methods assume a thin client with limited storage and computation capability. Furthermore, in these methods the assumption about the user's knowledge about the POIs is very conservative. For example in [8], the method only assumes that the user knows the distribution of total queries. Thus, the privacy protection level achieved using these approaches is uncontrollable and unpredictable.

In contrast to the above approaches, our method has the following features: (i) No trusted third party (TTP); (ii) Applicable to existing LBS; (iii) Secure location privacy; and (iv) Low communication cost.

III. System Architecture of LLPS

In the past few years, mobile computing speed and capability have been advancing at an exponential rate. A smart phone is not just another type of handset, but a fully-fledged compute. Thus, incorporating the computational capabiity and storage capacity of smart phones, we propose a novel system architecture, called LPPS, to provide a privacy-preserving top-k query.

LLPS divides the POI's information into the following category: (i) The metadata, which includes geographic information, number of reviews, ratings, price range, etc. The metadata changes infrequently and is usually small in storage size. (ii) The real-time information, such as special offers, reservation information, etc. (iii) The detailed information, which consists of user comments, pictures, etc., and the size of the detailed information is relatively large.

The basic idea of LPPS is to rank the POIs on the client side of applications using a small amount of metadata from the server, and then to request real-time and detailed information about the top-k POIs from the LBS server. In LPPS, the user's exact location data is never transmitted outside the device. Thus, it prevents the LBS server from collecting the user's exact location information. The communication pattern in LPPS is summarized as follows:

(i) The mobile devices utilize localization infrastructure, such as GPS system, Cellular-ID look-up, or Wi-Fi positioning system, to determine the current geo-location l_{user}.

(ii) The mobile user determines a broad geographic area (say a city, or $100km^2$) that contains the location l_{user}, then sends the selected area and his interest (search keywords, such as Chinese restaurant) to the LBS server. Once the LBS server receives the query, it will found out the POIs that match the use's interest in the selected area, and then send their metadata to the user.

Note that the size of the metadata is small (usually, a few bytes), and the data changes infrequently. Thus, the user can preload the metadata information of all the POIs in the city and store them locally to rank the POIs without communicating with LBS server.

(iii) The mobile deveice determines the top-k POIs by using its current location and the value of the metadata(For example,the famous Chinese online review application DianPing's POI relevancy ranking takes into account number of user review, rating, price range). Thus, the rank of the POIs is calculated as follows:

$$Rank(a_i, l_{user}) = \lambda_0 dis(l_{user}, l_i) + \lambda_1 m_1 + \cdots + \lambda_n m_n \quad (1)$$

where l_{user} denotes the user's current location, l_i denotes the location of the POI a_i, $dis(l_{user}, a_i)$ denotes the Manhattan Distance between l_{user} and a_i, m_i denotes the value of the metadata i and λ_i denotes the weight of metadata i.

(iv) The mobile users request the real-time data and the detailed feature data for the POI subset from the server. Note that, in order to protect his privacy, the user may send a superset of the actual top-k POIs to the LBS. The details will be given in next section.

(v) LBS server responds with the corresponding data for the requested POI subset.

IV. Location Privacy Evaluation

In this section, we introduce the threat model and the privacy metric used in our paper. Finally, we present the formulation of the problem.

A. Threat Model

Standard cryptography techniques such as ASE, SHA can be easily used to deal with eavesdropping attacks on the wireless channel between users and other entities. Thus in this paper, we focus on analyzing the privacy of mobile users against inference attacks. We make a widely acceptable assumption that the LBS server provides "honest but curious" services. Therefore, the server correctly follows the protocol outlined in section III, but may attempt to infer the users locations from the received queries. In this paper,we assume that the LBS server has the *side information* regarding the query probability in each location. Specifically, we divide the map area into a grid of $n \times n$ cells $\{r_1, r_2, \cdots, r_{n^2}\}$. Each cell has a probability of being queried based on the query history, which is calculated as

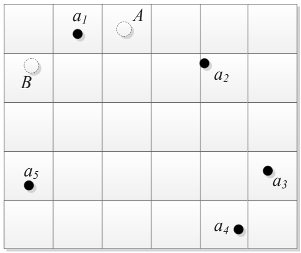


Fig. 1. Distribution of POIs

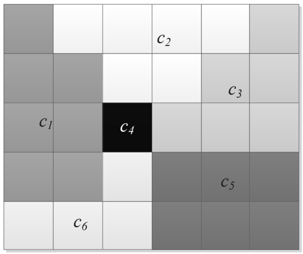


Fig. 2. Location indistinguishable set using top-2 queries

TABLE I
TOP-2 QUERY IN EACH CELL

	1	2	3	4	5	6
1	a_1,a_5	a_1,a_2	a_1,a_2	a_1,a_2	a_1,a_2	a_2,a_3
2	a_1,a_5	a_1,a_5	a_1,a_2	a_1,a_2	a_2,a_3	a_2,a_3
3	a_1,a_5	a_1,a_5	a_2,a_5	a_2,a_3	a_2,a_3	a_2,a_3
4	a_1,a_5	a_1,a_5	a_4,a_5	a_3,a_4	a_3,a_4	a_3,a_4
5	a_4,a_5	a_4,a_5	a_4,a_5	a_3,a_4	a_3,a_4	a_3,a_4

$$Pr(r_i) == \frac{the\, number\, of\, queries\, in\, cell\, r_i}{\sum_{j=1}^{n^2} the\, number\, of\, queries\, in\, cell\, r_j} \tag{2}$$

In addition, the query itself may reveal some information about the user's likely position. Therefore, once the LBS server obtains the top-k POIs from the users, it will use them to infer the user's likely position. Formally, given the user's query q, the probability that user is at location r_i is calculated as

$$Pr(r_i|q) = \frac{Pr(q|r_i)Pr(r_i)}{Pr(q)} \tag{3}$$

where $Pr(q|r_i)$ denotes the probability of generating query q if the user is at location r_i, and $Pr(r_i)$ denotes the prior probability that the user is at location r_i.

B. Location Privacy Metric

Definition 1 Location Indistinguishability: Let $Q_k(g)$ denote the results of the top-k query at location g, and $Q_k(h)$ denote the result of top-k query at location h. We say locations g and h are indistinguishable using top-k query, if $Q_k(g)$ is equivalent to $Q_k(h)$.

An example is illustrated in Fig.1, in which there are 5 Points of Interest (POIs), such as Chinese restaurant $\{a_1, a_2, \ldots, a_5\}$ in the map. Alice is at location A and Bob is at location B. However, both of them will send query $\{a_1\}$ to the LBS server when they want to know the details about the nearest Chinese restaurant. In this case, the LBS server cannot identify the user's location from A and B by using query $\{a_1\}$. Thus, location A and B are indistinguishable using a top-1 query.

We rigorously analyze the privacy level of the users. Suppose that user u_i in cell r_i sends a top-k query to the LBS server. To analyze the privacy level of u_i, we calculate the top-k POIs in each region and place the indistinguishable locations into a set. An example is illustrated in Fig.1: there are 5 POIs in a region that consist of 6×6 cells. The top-2 POIs of each cell are shown in shown in Table I. Then, we can obtain the set of indistinguishable locations using Definition 1. As shown in Fig.2, the region is divided into 6 sub-regions, in which each cell is indistinguishable using a top-2 query, and $c_1 = \{r_i|Q_2(r_i) = \{a_1, a_5\}\}$,$c_2 = \{r_i|Q_2(r_i) = \{a_1, a_2\}\}$, $c_3 = \{r_i|Q_2(r_i) = \{a_2, a_3\}\}$,$c_4 = \{r_i|Q_2(r_i) = \{a_2, a_5\}\}$,$c_5 = \{r_i|Q_2(r_i) = \{a_3, a_4\}\}$,$c_6 = \{r_i|Q_2(r_i) = \{a_4, a_5\}\}$.

For the locations (i.e., cells) contained in an indistinguishable set c_i, each location has a conditional probability of being the real location. Let p_i denote the probability that the i^{th} location (cell) is the real location. Then $p_i = \frac{Pr(r_i)}{\sum_{r_i \in c_j} Pr(r_i)}$, and obviously $\sum_{r_i \in c_j} Pr(r_i) = 1$. The entropy $\mathcal{H}$ [10] of identifying the real location out of the indistinguishable set is calculated as follows:

$$\mathcal{H}(c_j) = \sum_{r_i \in c_j} p_i \cdot logp_i \tag{4}$$

C. Problem Formulation

Definition 2 Users privacy requirement s_i: In this paper, user's privacy is measured by entropy $\mathcal{H}$, which is indicated by the uncertainty in determining the current location of a user from an indistinguishable set. Given a user i's top-k query q and the corresponding indistinguishable location set c_i, we say the user's privacy requirement is satisfied, if and only if $\mathcal{H}(c_i)$ is greater than or equal to s_i.

Definition 3 Users QoS requirement: LPPS should be efficient both in terms of communication and computational cost since the users want the information about the POIs within a short period of time without consuming much data traffic. For simplicity, we focus on the communication cost between a user and the LBS server, as it is the most dominant cost of serving the information of POIs to mobile devices. Our results can trivially be extended to consider the computational cost on a mobile device.

Prior to sending the query q to the LBS server, the user needs to calculate the set of indistinguishable locations by using query q, and then evaluate whether their privacy requirement would be satisfied if the query q was released. If there is no privacy leakage problem, the user will send the query q to the LBS server; otherwise the user needs to re-generate the query to satisfy his privacy requirement.

A straightforward solution to re-generate the query is to add dummy POIs into the existing top-k query q and then to send the superset of its actual top-k POIs to the LBS. For example, as shown in Fig.2, Alice's top-2 query is $q = \{a_1, a_2\}$, and the corresponding indistinguishable location set is c_2. If Alice selects $\{a_3, a_4\}$ as dummy POIs, then her query will become$\{a_1, a_2, a_3, a_4\}$,and the indistinguishable location set will become $c_1 \cup c_5$. The user can receive a higher privacy protection level by adding dummy more POIs. Due to the limitation of the user's QoS requirement, the dummy POIs selection problem is formulated as

$$Minimize: \quad sizeof(query\ q) \tag{5}$$

$$s.t. \qquad \mathcal{H}(c) > s_i \tag{6}$$

where q denotes the query that sends to the LBS server and c denotes the corresponding indistinguishable locations set obtained using q. In the next section, we will propose two dummy POI generation algorithms to address the problem.

V. Dummy Generation Algorithm

In this section, we propose two privacy-aware dummy generation algorithms. The first algorithm employs the intersection to constrain all dummy POIs, while the others select the dummy POIs by employing maximum entropy.

A. Intersection based Dummy Generation Algorithm

Definition 4 Location Similarity: Given two location g and h, the location similarity between them is calculated as:

$$sim(g,h) = sim(Q_k(g), Q_k(h)) = \frac{Q_k(g) \cap Q_k(h)}{k} \tag{7}$$

where $Q_k(g)$ and $Q_k(h)$ denote the results of the top-k queries at locations g and location h, respectively. Greater $sim(g,h)(0 \leq sim(g,h) \leq 1)$ indicates higher similarity between $Q_k(g)$ and $Q_k(h)$.

The basic idea of the intersection based dummy generation (IDG) is to utilize location similarity to reduce the communication cost while satisfying the user's privacy requirement. We consider an example(Fig.2), in which user i's top-2 query is $q = \{a_1, a_5\}$, and its indistinguishable location set is sub-region $c_1 = \{r_i | Q_2(r_i) = \{a_1, a_5\}\}$. The results of top-2 query in sub-region c_1 and c_2 $(c_2 = \{r_i | Q_2(r_i) = \{a_1, a_2\}\})$ both contain POI a_1. Thus the similarity between c_1 and c_2 is 0.5, which is calculated using (7). If we then selected POI $a_2 \in c_2$ ($a_1 \in c_2$ has been included in the query) as a dummy POI, the query sent to the LBS server will become $q = \{a_1, a_2, a_5\}$. In this case, user i's indistinguishable location set contain three sub-regions $\{c_1, c_2, c_4 = \{r_i | Q_2(r_i) = \{a_2, a_5\}\}\}$ with little communication cost increase. Otherwise if we selected the POIs in unrelated sub-region $c_5 = \{r_i | Q_2(r_i) = \{a_3, a_4\}\}$ (the similarity between c_1 and c_5 is 0) as dummy POIs, the query sent to the LBS server will become $q = \{a_1, a_3, a_4, a_5\}$. In that case, the user's indistinguishable location set only contains two sub-regions $\{c_1, c_5\}$ while the user has to pay higher communication cost.

Given the top-k query q, and the corresponding indistinguishable location set c, we need to determine the dummy POIs that will be added to q. The following steps show how the IDG algorithm addresses the problem.

(i) As the first step, a particular user needs to determine his privacy requirement s_i, which is closely related to his location privacy and system overhead. Specifically, a bigger s_i, leads to a higher protection level, but also high computation and communication overhead due to the cost incurred by dummy POIs.

(ii) The user uses c to evaluate whether the his privacy requirement would be satisfied if the query was released. If there is no privacy leakage problem, then the user sends query q to the LBS server; otherwise go to (iii);

Algorithm 1: Intersection based Dummy Generation Algorithm (IDG)

```
Input: the user's top-$k$ query $q$,
the indistinguishable location set $c$ using $q$,
the users requirement $s_i$
Output: the query sent to the LBS server
while $\mathcal{H}(c) < s_i$ do
    Find the set $R = \{r_1, r_2, \ldots, r_n\}$ of the cells that is
    adjacent to $c$;
    for each $r_i$ in $R$ do
        Calculate the top-$k$ query $q_i$ at location $r_i$;
        if $sim(r_i, c) < \theta$ then
            $R = R - \{r_i\}$;
        end
    end
    Randomly select the cell $r_t$ from $R$;
    Calculate the indistinguishable location set $c_t$ using
    $q_t$;
    $c \leftarrow c \cup c_t$;
    $q \leftarrow q \cup q_t$;
end
return the query $q$ that sent to the LBS server;
```

(iii) Let $R = \{r_1, r_2, \ldots, r_n\}$ denote the set of the cells adjacent to the sub-region c. For each r_i in R, the user calculates top-k query in the cell, and then uses it to calculate the location similarity $sim(r_i, c)$ between r_i and c based on equation (7).

(iv) The user randomly picks up a cell r_t, whose $sim(r_t, c)$ is higher than θ (which is determined by the user) from R, and then the user calculates the indistinguishable location set c_t using q_t. In addition, the user merges the sub-region c and c_t into an indistinguishable location set. Formally, $c \leftarrow c \cup c_t$ and $q \leftarrow q \cup q_t$. Then, the step (ii) will be re-executed.

This algorithm is detailed in Algorithm 1.

B. Maximum Entropy based Dummy Generation Algorithm

Our aim is to reduce the communication cost while satisfy the user's privacy requirement, namely, the uncertainty of identifying an individual user based on the his query. Through IDG can achieve a high protection level with little communication and computation cost increase, it still has a problem that the elements of the indistinguishable location set are located close to the real location. Thus, in this section, we propose maximum entropy based dummy generation algorithm (EDG).

Given a particular user's query $q = q_0 \cup q_1 \cup q_2 \cdots \cup q_m$ that is sent to LBS server, and the corresponding indistinguishable location set $c = c_0 \cup c_1 \cup c_2 \cdots \cup c_m$, the privacy level of the user is calculated as

$$\mathcal{H}(c) = -\sum_{r_j \in c} p_j \cdot log p_j = -\sum_{c_i \in c} \overline{p}_i \cdot log \overline{p}_i + \sum_{c_i \in c} \overline{p}_i \cdot \mathcal{H}(c_i) \tag{8}$$

$$\overline{p}_i = \frac{\sum_{r_j \in c_i} p_j}{\sum_{c_i \in c} \sum_{r_j \in c_i} p_j} \tag{9}$$

where p_j denotes the query probability at cell r_j and $\overline{p}_i$ denotes the probability that the real location is located in the sub-region c_i. The first part of the equation denotes the entropy in the location set c, which achieves the maximize value when all the m sub-regions in c have the same probability $\frac{1}{m}$, and the second part of the equation denotes the entropy in each sub-region.

Therefore, the basic idea of EDG is to find the sub-regions which have similar query probability to that of the real sub-region and then to insert the sub-regions with high entropy into the indistinguishable location set. In the following, we will show how the EDG addresses the problem.

At the beginning of our EDG algorithm, the user needs to calculate the query probabilities of all the sub-regions, then sort all cells in order of the query probability. In the sorted list, the user chooses the m (which is related to the user's privacy requirement) sub-regions right before and the m sub-regions right after the real location as $2m$ candidates(i.e., $m = 2^{s_i - \mathcal{H}(c)} + \lambda$, λ is a variable determined by users). Then, the user calculates the entropy of the candidates using equation (8). Next, the user needs to determine the indistinguishable location, and the corresponding query. To achieve higher entropy within limited dummy POIs, the user only chooses the sub-region, whose entropy is higher than threshold η (which is determined by the user). Then, in the $2m$ candidates, the user randomly selects a sub-region c_i with a entropy higher than η and repeats the process until the user's privacy requirement is satisfied.

The detail of this algorithm is shown in Algorithm 2.

Algorithm 2: Maximum Entropy based Dummy Generation Algorithm (EDG)

```
Input: the user's top-$k$ query $q$,
the indistinguishable location set $c$ using $q$,
the users requirement $s_i$
Output: the query sent to the LBS server
Calculate the query probability in each sub-region;
Sort the sub-regions based on their query probability;
Select $2m$ candidate sub-regions among in which $m$
  candidates are located prior to the real sub-region $c$ and
  $m$ candidates are located after the real location $c$ in the
  sorted list;
for $j = 1; j \leq 2m; j++$ do
    if $\mathcal{H}(c_j) > \eta$ then
        Add $c_j$ to candidate set $M$;
    end
end
while $\mathcal{H}(c) < s_i$ do
    Randomly select a sub-region $c_t$ from $M$;
    $c \leftarrow c \cup c_t$;
    $q \leftarrow q \cup q_t$;
end
return the query $q$ that is sent to the LBS server;
```

VI. Performance Evaluations

In this section, we evaluate the performance of the proposed LPPS system, together with the IDG and EDG algorithms.

A. Simulation Setup

We evaluated our system using a real-world Dianping dataset that was made available by [11]. We consider a 80×80 grid over a $8 \times 8km^2$ broad area (each cell is $100 \times 100m^2$) centered at the city of Beijing. The query probability at each cell is calculated using the total number of reviews about each POI in the cell. As shown in Table II, we classify the POIs into the following categories according to density: low-density POIs, such as the hospital, the church, and the railway station; middle-density POIs, such as the restaurant; and high-density POIs, such as the ATM, caf, and bakery. In the experiment, the Starbucks, gas station, and ATM were chosen for study because they are typical representatives of their respective densities.

TABLE II
The Types of The POIs

POI type	Number	POI Name	Number
High-density	≥200	ATM	247
Middle-density	50-200	Gas-station	108
Low-density	≤50	Starbucks	27

There are several parameters used in our evaluation. k is related to top-k query, and is commonly set from 1 to 20. We assume that the users privacy requirement s_i is 3 (the entropy of the indistinguishing location set is 3, calculated using Eq.8). For simplicity, we only consider the distance factor in our experiment, but in our methods the users can rank the POIs by using multiple factors.

We compare our LPPS with IDG algorithm, and the EDG algorithm with two other schemes. The "Baseline" scheme represents the existing LBS without privacy protection. The "Rand" scheme represents the LPPS with a randomly chosen dummy strategy.

B. Evaluation Results

1) *Indistinguishable location set vs. k:* We evaluate the relationship between k and the number of indistinguishable location sets. As shown in Fig.3, at first the number of indistinguishable sets increases with k. However, when k is higher than half of the POIs, the number decreases with k. This is because when k is lower than half the POIs, the probability that the query result is the same in different cells decreases with k, whereas the probability increases with k, when k is higher than one half the POIs. For the same reason, the number of indistinguishable location sets increases with the number of the POIs.

2) *Communication cost vs. k:* We evaluate the relationship between k and the communication level. Fig. 5 plots the communication cost when k varies from 1 to 20. Generally, the communication cost increases with k. This is because the average size of top-k query's indistinguishable location set

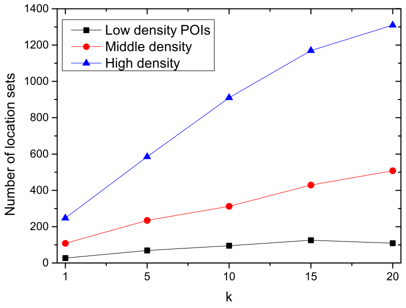
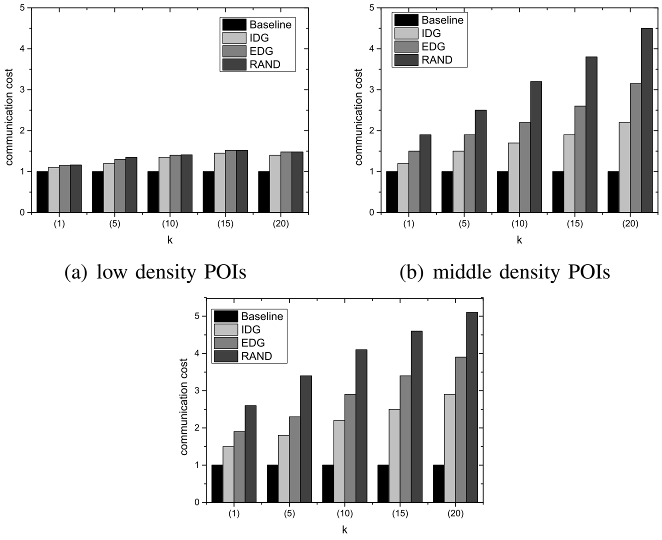


Fig. 3. The number of indistinguishable set vs. k



(a) low density POIs

(b) middle density POIs

(c) high density POIs

Fig. 4. Communication cost vs. k under different types of POIs

decreases with k, and the user needs to add more dummy POIs into the real top-k query. However, as shown in Fig.5 (a), the communication cost drops off when k (k=20) is higher than half of the POIs, because the size of the indistinguishable set increase in this case.

Generally, as shown in Fig.5, LPPS achieves a high protection level for all the POIs with little increase in communication cost. Compared with the baseline, the additional communication cost of all schemes for low-density POIs is less than 50%, because the size of the indistinguishable location set in the low-density POIs is relatively higher than the size of the indistinguishable location set in other POIs, as shown in Fig.5(a). Among all these schemes, LPPS with IDG has the lowest communication cost dut to the high location similarity between its selected sub-regions. EDG performs much better than the Rand becuase all the candidate sub-regions using the queries have the same probability to be treated as the real location. Although the communication overhead of the EDG algorithm is higher than the communication cost of IDG algorithm, the distribution of the indistinguishable set is more extensive and uniform on the entire map. The users can decide which methods they will use according to the remaining data traffic.

VII. Conclusion

In this paper, we propose a novel system architecture, LPPS, which takes advantage of smartphones' computational capability and storage capacity, does not require TTP, and provides a privacy-preserving top-k query. The basic idea of LPPS is to rank the POIs on the client side of applications using a small amount of metadata from the server, then to request real-time and detailed information about the top-k POIs from the LBS server. On the basis of LPPS, a novel metric called location indistinguishability was proposed to evaluate the privacy level of users in the proposed scheme. Based on the metric, we proposed two dummy POI generation algorithms. The IDG algorithm utilizes location similarity to reduce the communication cost while satisfying the users privacy requirement. The EDG algorithm considers both the communication cost and the distribution of the dummy POIs. Evaluation results indicate that LPPS achieve a high protection level with little communication cost increase.

Acknowledgment

This research was partially supported by the National Basic Research Program of China under grant No. 2011CB302605, No. 2013CB329602, and the National Natural Science Foundation of China (NSFC) under grants No. 61202457, No. 61173144, No. 61402137, and No. 61402149.

References

[1] H. P. Li, H. Hu, and J. Xu, "Nearby friend alert: Location anonymity in mobile geosocial networks," *Pervasive Computing, IEEE*, vol. 12, no. 4, pp. 62–70, 2013.

[2] Y. Wang, D. Xu, X. He, C. Zhang, F. Li, and D. Xu, "L2p2: Location-aware location privacy protection for location-based services," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1996–2004.

[3] H. Kang and W. Meng, "Protecting location privacy with personalized k-anonymity," *Journal of Nanjing University of Posts and Telecommunications (Natural Science)*, vol. 6, p. 014, 2012.

[4] H. Zhang, Z. Xu, Z. Zhou, J. Shi, and X. Du, "Clpp: Context-aware location privacy protection for location-based social network," in *Communications (ICC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1164–1169.

[5] X. Zhao, L. Li, and G. Xue, "Checking in without worries: Location privacy in location based social networks," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 3003–3011.

[6] X.-Y. Li and T. Jung, "Search me if you can: privacy-preserving location query service," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2760–2768.

[7] H. Lu, C. S. Jensen, and M. L. Yiu, "Pad: privacy-area aware, dummy-based location privacy in mobile services," in *Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*. ACM, 2008, pp. 16–23.

[8] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 754–762.

[9] ——, "Enhancing privacy through caching in location-based services," in *Proc. of IEEE INFOCOM*, 2015.

[10] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Privacy Enhancing Technologies*. Springer, 2003, pp. 41–53.

[11] Y. Zhang, M. Zhang, Y. Liu, S. Ma, and S. Feng, "Localized matrix factorization for recommendation based on matrix block diagonal forms," in *Proceedings of the 22nd international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2013, pp. 1511–1520.