A location privacy-preserving system based on query range cover-up for location-based services

Journal:	*IEEE Transactions on Vehicular Technology*
Manuscript ID	VT-2019-02147.R1
Suggested Category:	Regular Paper
Date Submitted by the Author:	10-Oct-2019
Complete List of Authors:	Wu, Zongda; Shaoxing University, Wang, Ruiqin; Huzhou University Lian, Xinze; Wenzhou University, Oujiang College Xu, Guandong; University of Technology Sydney Chen, Enhong; University of Science and Technology of China
Keywords:	Location-based service, location privacy, privacy protection

A location privacy-preserving system based on query range cover-up for location-based services

Zongda Wu, Ruiqin Wang, Xinze Lian, Guandong Xu, and Enhong Chen *Senior Member, IEEE*

***Abstract*—Location-based service (LBS) has been widely used in various fields of industry, and become a vital part of people's daily life. However, while providing great convenience for users, LBS results in a serious threat on users' location privacy, due to its more and more untrusted server-side. In this paper, we propose a location privacy-preserving system for LBS by constructing "cover-up ranges" to protect the query ranges associated with a location query sequence. Firstly, we present a client-based system framework for location privacy protection in LBS, which requires no compromise to the accuracy and usability of LBS. Secondly, based on the framework, we introduce a location privacy model to formulate the constraints that ideal cover-up ranges should satisfy, so as to improve the efficiency of location services and the security of location privacy. Finally, we describe an implementation algorithm to well meet the location privacy model. Both theoretical analysis and experimental evaluation demonstrate the effectiveness of our system, which can improve the security of users' location privacy on the untrusted server-side, without compromising the accuracy and usability of LBS.**

***Index Terms*—Location-based service, location privacy, privacy protection, security, accuracy, usability**

I. Introduction

With advances in wireless communication and mobile positioning technologies, location-based service (LBS) has been gaining increasingly popularity in recent years, and has become an important infrastructure of Industry 4.0 [1], [2], [4]. It has been reported that the revenue from LBS has reached an annual global total of over 15 billion dollars [9]. However, while bringing great convenience to users, LBS results in a serious location privacy problem [7], [8], because in order to obtain LBS, a user has to report his current location to the untrusted server-side. Obviously, the location information is sensitive, based on which an attacker can infer user's location trajectory accurately. It certainly would lead to a serious threat to users' location privacy, if the location information is released to an untrusted third party (e.g., the LBS provider). In fact, the location privacy protection in LBS is causing people's increasingly extensive concern, and has become a major barrier for the development and application of LBS in various fields of industry [5], [6].

This paper aims to construct a location privacy-preserving system for LBS, which can meet the following requirements, i.e., under the precondition of not compromising the accuracy and usability of LBS, ensuring the security of location privacy and the efficiency of location services. Specifically, the contributions of this paper are threefold.

(1) We present a framework for location privacy protection in LBS, whose basic idea is that for a location query request issued by a user, the client constructs a new request where the user query range is replaced by a well-designed "cover-up range", and submits it to the server, so as to make it difficult for the untrusted server-side to infer user's query location. Then, the client picks out the result corresponding to the user query request from the result returned by the server, so as to ensure the accuracy of the result that the user obtains finally.

(2) Based on the framework, we present a location privacy model to formulate the constraints that an ideal cover-up range constructed for a user query range should meet, so as to ensure the efficiency of location services and the security of location privacy. Moreover, in the privacy model, the relevance constraint among query locations is also taken into account, making it still difficult to identify the user locations even if an attacker has mastered the user query regularity, and further improving the security of location privacy.

(3) According to the above framework and privacy model, we present an implementation algorithm running on a trusted client. For a user location query sequence, the algorithm can construct a cover-up range sequence that well meets the privacy model. Besides, we have demonstrated the effectiveness of the privacy model and its algorithm through theoretical analysis and experimental evaluation.

The rest of this paper is organized as follows. Section 2 reviews some related work. Section 3 presents a framework and attack model for LBS privacy protection. Section 4 describes a location privacy model and its implementation algorithm. Sections 5 and 6 demonstrate the validity of the system by theoretical analysis and experimental evaluation. Finally, we conclude this paper in Section 7.

II. Related Work

In order to protect location privacy in LBS, many methods have been proposed, e.g., pseudonym methods, obfuscation methods, encryption methods and dummy-based methods. In this section, we briefly review the technical features for these methods.

Z. Wu is with Department of Computer Science and Engineering, Shaoxing University, Shaoxing 312000, Zhejiang, China.

R. Wang is with Department of Computer Science and Engineering, Huzhou University, Huzhou, Zhejiang, China.

X. Lian is with Oujiang College, Wenzhou University, Wenzhou, Zhejiang, China. E-mail: xinzelian@163.com

G. Xu is with Faculty of Engineering and IT, University of Technology, Sydney, Australia.

E. Chen is with School of Computer Science and Technology, University of Science and Technology of China, Hefei, China.

Manuscript received 2019; revised 2019.

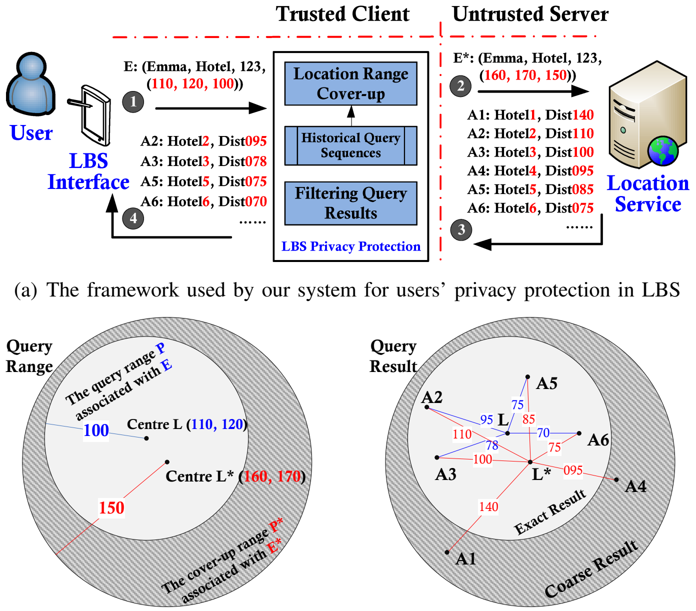
The work is supported by the National Social Science Foundation of China (No. 19BTQ056).

(1) In **pseudonym** methods, the user identity in each LBS query is replaced with a pseudonym, so as to disconnect the user from the query [11], [21]. However, for this kind of methods, although such solutions as establishing mixed zones have been designed to improve the effectiveness of pseudonyms [18], [19], [20], [2], it is still likely for an attacker to identify the user identity from query content itself, due to no change to each user query, i.e., it is difficult to resist the threat from data mining [9], [31]. For example, in [10], a novel paradigm of de-anonymization attacks based on mobility patterns of objects was proposed, which can re-identify the user trajectory accurately from a group of anonymous trajectories (where the real identities of mobile objects has been replaced with pseudonyms). In addition, it is also difficult to be applied to an environment that requires identity authentication or personalized services due to hiding user identity [3].

(2) In **obfuscation** methods, the location information in a LBS query is generalized (by using a cloaking region to replace a user location [12], [22], [23], [24]) or perturbed (by intentionally adding some errors or noises into a user query in a controllable manner [13], [25]), so as to make it difficult for an attacker to identify user's precise location. However, since each query request submitted to the server has been modified, it sometimes will lead to a compromise to the accuracy of LBS, i.e., its privacy protection needs to compromise the LBS quality to a certain degree. Moreover, the implementation of a pseudonym or obfuscation method is generally dependent on a trusted third-party server, making it easy to lead to a performance bottleneck and a privacy bottleneck [9].

(3) The basic idea of **encryption** methods is to use encryption techniques to make each LBS query invisible to the untrusted server-side, typically, such as LBS privacy protection based on Private Information Retrieval (PIR) [14], [15], [26]. An encryption method would neither reveal any user location information nor compromise the usability of LBS, thereby achieving stricter privacy protection. However, this kind of methods generally requires the change to the existing LBS algorithm on the server-side and the support of additional hardware, resulting in the change to the whole LBS platform, and then decreasing the actual usability of the methods.

(4) In **dummy-based** methods, each user query would be submitted together with a group of dummy queries constructed in advance, to make it difficult for the untrusted server-side to identify user's true locations [16], [17]. A dummy method is generally developed on a client-based architecture, independent of a third-party server, resulting in its good practicability. However, although a number of good algorithms for dummy construction have been proposed [16], [17], [27], [28], [29], the security of this kind of methods is dependent on the quality of dummy construction, i.e., it is often threatened by the attack based on the feature distribution, resulting in poor security. Besides, k-anonymity refers to that for an individual, there exist k-1 individuals which are indistinguishable from it. It is often combined with pseudonym methods or dummy-based methods to measure the effect of privacy protection. However, it is actually only a kind of privacy metric, whose effectiveness is also dependent on the construction quality of pseudonyms or dummies.



(a) The framework used by our system for users' privacy protection in LBS

(b) A user query range P of E and its cover-up range P^* of E^* (where the location coordinates are not true longitude and latitude values)

Fig. 1. An example for the protection of users' location privacy in LBS

In addition, Zhang et al in [9] have presented a detailed survey on each kind of location protection methods mentioned above. From the above, we can conclude that a good method that can well protect users' location privacy in LBS should meet the following requirements. (1) Ensuring the **security** of location privacy. It should be difficult for an attacker with rich prior knowledge to identify user's location trajectory from a user query sequence. (2) Ensuring the **accuracy** of location services, i.e., the service result that a client user obtains finally should be consistent before and after the privacy method is used. (3) Ensuring the **usability** of location services, i.e., the privacy protection should neither require the change to the existing LBS algorithm, nor result in a significant impact on the **efficiency** of location services. To this end, in this paper, we aim to propose a location privacy-preserving system for LBS, which can meet the above requirements in terms of accuracy, usability and efficiency.

III. Problem Statement

A. Problem Definition

In general, LBS consists of a server-side (running the LBS algorithm) and many client-sides (running the user interfaces). In LBS, each query request issued by a user through a client interface can be denoted by $E = (U, A, T, P)$, where U is a user identity, A is a query attribute (such as Hotel and Toilet), T is a query timestamp, and P denotes a query location and its related parameter. Here, P is the key to location privacy protection in LBS, and thus the study object of this paper. According to the meaning of the parameter of P, LBS queries can be divided into range queries (e.g., querying the hotels in the range of $100m$) and nearest neighbor queries (e.g., querying the nearest 3 hotels). In this paper, we only focus on range queries (instead, we will briefly discuss nearest neighbor queries in the end of Section 4.2), so P (below, called a query range) is considered to consist of a query location and a radius. Fig. 1 describes the framework used by our system for location privacy protection, through using an example of querying the hotels in the range of $100m$. It can be seen that our location privacy-preserving system runs on the client-side as a layer of middleware between the server-side and the client interface, whose processing flow can be briefly described as follows.

- For each user location query request $E = (U, A, T, P)$, the "location range cover-up" component running on the client (located between the server and the client interface) constructs a new request $E^* = (U, A, T, P^*)$, where the user query range P is replaced by a newly constructed cover-up range P^*, and then submits the new request E^* in place of E to the server to acquire a location service.
- The "filtering query result" component picks out the exact result $\mathcal{A}$ (the interest points within the query range P, e.g., A_2, A_3, A_5, A_6 in Fig. 1(b)) located within the query range P) from the coarse result $\mathcal{A}^*$ (the interest points within P^*, e.g., $A_1 - A_6$ in Fig. 1(b)) that are returned by the LBS algorithm on the server-side, and then returns $\mathcal{A}$ to the user as the final service result.

It can be seen that the "location range cover-up" component just rewrites the query range P of E (i.e., changes the location and enlarges the radius), without changing the request message format. As a result, we can conclude that under the framework: (1) the privacy protection is transparent to both the LBS algorithm on the server-side and each user on the client-side, i.e., it requires no change to the platform architecture of LBS; (2) the coarse result returned from the server-side is certainly a superset of the user exact result, as long as the query range P is contained in the cover-up range P^*, i.e., it requires no compromise to the accuracy of LBS; and (3) the degree of the efficiency decrease caused by the privacy protection has a linearly positive correlation with the area size of the cover-up range P^*, i.e., it does not lead to a significant impact on the efficiency of LBS. Also, it can be seen that the cover-up range P^* constructed by the "location range cover-up" component plays a vital role, whose quality is the key to the security of location privacy and the efficiency and accuracy of location services. Specifically, for the construction of cover-up ranges, we should consider the following problems.

- **Problem 1**. The accuracy of location service. Each cover-up range should contain its corresponding user query range; otherwise, the accuracy of location service cannot be guaranteed.
- **Problem 2**. The efficiency of location service. In general, the bigger the area size of a cover-up range, the more the points of interest located within the cover-up range, and then the worse the efficiency of a location service. Thus, to improve the efficiency, the area size of each cover-up range should not be too big.
- **Problem 3**. The security of location privacy. In general, the bigger the area of a cover-up range and the farther the central location of the cover-up range from the user extract location, the smaller the probability of an attacker to identify the user exact location (or query range). It has a contradiction between the security of location privacy and the efficiency of location service, so how balancing the security and the efficiency to construct suitable cover-up ranges should be considered comprehensively.
- **Problem 4**. The relevance among query locations. A user often likes to periodically issue his requests, among which there may be strong location relevance. For example, the queries issued by the same user during a period of time often occur in some fixed locations or regions (e.g., near to House or Company). According to the location relevance, an attacker with rich knowledge can easily reduce the active area of the cover-up range and hence the security of location privacy. Therefore, the central locations of a cover-up range sequence constructed by the system should also exhibit the relevance accordant to the user query regularity, to make it difficult for the attacker to exclude some active area of each cover-up range.

B. *Attack Model*

In this study, we only focus on the protection of location privacy itself (i.e., the privacy related to a user query range P), although in a LBS request E, there are other kinds of user privacy information, such as user identity (U) and query attribute (A). Moreover, similar to the attack model used by most related studies in Section 2, the server-side of LBS is considered to be untrusted in this study, since the server-side has mastered a great number of users' privacy data, which is considered as the biggest potential attacker. As a result, we assume that an attacker has the following abilities. (1) The attacker can obtain all the location query sequences from the client-sides, which, however, are the cover-up query sequences generated by the privacy-preserving system (instead of users' true query sequences), so it is required for the system to prevent the attacker from identifying users' true locations or ranges from the cover-up query sequences. (2) The attacker might have mastered users' location query regularity (e.g., some fixed locations or regions that a user often likes to query around), so he can reduce the active area of each cover-up range based on the relevance regularity among users' locations. (3) The attacker might have obtained a copy of the privacy algorithm running on the client-side, so he can input each of the mastered queries to the algorithm and then analyze the output, to guess the user locations.

In summary, the attacker is powerful, but also knowledge-bounded, i.e., satisfying the following assumptions: (1) he does not know any background information on mobile users (e.g., house or company address); and (2) he does not know some right samples of users' query locations in advance. Although the above assumptions seem strong, they are reasonable in practice, because in this study, we only focus on the location privacy itself, i.e., the ability of the attacker is mainly from his mastered query range sequences themselves.

IV. Proposed Method

In this section, we describe our privacy-preserving system based on the above framework. First, we present a location privacy model to formulate the constraints that the cover-up ranges constructed by the system should meet. Then, we present an implementation algorithm for the privacy model.

A. Privacy Model

In Section 3, we briefly describe some problems that should be taken into consideration in the construction of cover-up ranges. Below, we further formulate the constraints that ideal cover-up ranges should satisfy, to provide reference for the design of a construction algorithm of cover-up ranges. First of all, we define a query range P.

Definition 1 (Query Range): *The query range P in each location query request E can be represented by a circular region, i.e., $P = (P.L, P.R)$, where $P.L$ denotes the central location of P and $P.R$ the radius of P.*

As mentioned in Section 3, when constructing the cover-up range P^* for a user query range P, we need to consider such problems as the accuracy and efficiency of location service, the security of location privacy, and the relevance among query locations. Below, we first present Definition 2 to formulate the accuracy of a location service.

Definition 2 (Cover-up Range Accuracy): *Let P^* denote the cover-up range constructed for a user query range P by a privacy system. Then, the accuracy of location service can be guaranteed by P^*, if and only if P is contained in P^*, i.e., $P \subseteq P^*$.*

From the framework shown in Fig. 1(a), it can be known that the efficiency of a location service is dependent on the number of points of interest returned from the server-side. However, in general, the number of points of interest is positively related to the area of a cover-up range from the client-side, i.e., the bigger the area of the cover-up range, the more the points of interest located within the cover-up range. Therefore, we can use the following definition to quantify the impact of a cover-up range on the efficiency of a location service.

Definition 3 (Cover-up Range Efficiency): *Let P^* denote the cover-up range constructed for a user query range P. Then, the loss of location service efficiency caused by the privacy protection can be measured by*

$$EF\left(P, P^*\right) = \frac{\pi P.R^2}{\pi P^*.R^2} = \frac{P.R^2}{P^*.R^2}$$

In general, the security of location privacy is in inverse proportion to the probability of an attacker to identify users' location or query range from a cover-up range. However, from the attack model mentioned in Section 3, we know that the probability of obtaining users' locations is mainly dependent on the cover-up ranges themselves mastered by the attacker, i.e., the larger the area size of each cover-up range, and the farther the cover-up central location from the user location, the better the security of location privacy. Therefore, we introduce the following definition to quantify the impact of a cover-up range on the security of location privacy.

Definition 4 (Cover-up Range Security): *Let P^* denote the cover-up range constructed for a user query range P. The security of user's location privacy on the server-side can be measured by*

$$PR\left(P, P^*\right) = \left\{ \begin{array}{c} 1.0, \left|P^*.L - P.L\right| \geq \alpha^2 \\ \frac{\sqrt{\left|P^*.L - P.L\right|\left(P^*.R^2 - P.R^2\right)}}{\alpha \cdot P^*.R^2}, otherwise \end{array} \right.$$

, where $|P^*.L - P.L|$ denotes the distance between the two locations, and α^2 denotes a distance threshold.

In Definition 4, for a cover-up range P^*, if the distance between its central location $P^*.L$ and the user location $P.L$ is bigger than α^2, then it is deemed that it has no harm on the user location privacy (i.e., it is impossible for the attacker to obtain the user true query location or range from the cover-up range). At this time, the security of location privacy reaches the maximum (i.e., 1.0). Therefore, α^2 denotes a long enough safe distance, which can be determined in advance by the system according to an actual situation.

In Definition 4, we use a single location query request as the basic unit to formulate the security of a cover-up range. However, there may be some strong location relevance among the queries issued by the same user. For example, the queries from the same user during a period of time are usually around some fixed locations or regions. An attacker with rich prior knowledge can easily master user's query location regularity, to reduce the active area of each cover-up range constructed by the privacy system, and hence the security of location privacy on the untrusted server-side. Below, we use a simple example to illustrate the problem. Assume that the attacker has mastered two queries issued by the same user around the same location $P.L$ at different timestamp. Let P_1^* and P_2^* denote two cover-up ranges constructed for the queries. Because P_1^* and P_2^* meet the accuracy of location service, we have that $P.L \in P_1^* \wedge P.L \in P_2^*$. Then, according to the prior knowledge, the attacker can conclude that $P.L \in P_1^* \cap P_2^*$. Therefore, the effect of the cover-up ranges on the protection of location privacy is reduced to $P_1^* \cap P_2^*$, i.e., in the two query services, the cover-up range area that can be excluded are $P_1^* - P_1^* \cap P_2^*$ and $P_2^* - P_1^* \cap P_2^*$, respectively, resulting in that the cover-up range security in Definition 4 cannot be well guaranteed.

To solve this, we next consider the problem on the relevance among query locations by using the sequence of location query requests issued by the same user during a period of time as the basic study unit, such that each constructed cover-up range sequence exhibits the location relevance accordant to user's query regularity, thereby, making it difficult for the attacker to reduce the active area of each cover-up range. Below, we first extend Definition 1 to define a query range sequence and a query location sequence.

Definition 5 (Range Sequence): *A range sequence consists of the query ranges issued by the same user during a period of time, which is denoted by $\mathcal{P} = (P_1, P_2, ..., P_n)$, where $P_k (k = 1, 2, ..., n)$ denotes a query range.*

Definition 6 (Location Sequence): *A location sequence consists of the central locations of the query ranges from a range sequence $\mathcal{P}$, denoted by $\mathcal{L} = (L_1, L_2, ..., L_n)$, where $L_k = P_k.L$ ($P_k \in \mathcal{P}$) denotes a location.*

Below, in order to capture the relevance regularity among query locations, we define a location region that is a set of location cells (i.e., locations). Note that all the location regions located at the same level can be viewed as a partition of the whole map (i.e., they can divide the whole map into several disjoint geographical regions).

Definition 7 (Location Region): *A location region D is a set of location cells, which is associated with a level, and the higher the level of a region, the more the location cells the region contains. It is obvious that the whole map is also a location region with the highest level (denoted by r^m). A set $\mathcal{D}^r$ of all the location regions with the same level r should meet the following constraints.*

- *(C1) Any two regions with the same level are disjoint, i.e., $\forall D_1 \forall D_2 (D_1, D_2 \in \mathcal{D}^r \rightarrow D_1 \cap D_2 = \oslash)$.*
- *(C2) Any two regions with the same level have the same area, i.e., $\forall D_1 \forall D_2 (D_1, D_2 \in \mathcal{D}^r \rightarrow AR(D_1) = AR(D_2))$, where AR denotes the area size of a region.*
- *(C3) The union of all the regions with the same level is the whole map, i.e., $\mathcal{D}^{r^m} = \cup_{D \in \mathcal{D}^r} D$.*
- *(C4) Any region should be contained by another region of a higher level, i.e., $\forall D_1 \exists D_2 (D_1 \in \mathcal{D}^r \wedge D_2 \in \mathcal{D}^{r+1} \wedge D_1 \subseteq D_2)$.*

The relevance among users' query locations indicates that the queries from the same user during a period of time are usually around some fixed locations or regions. Based on the definition of a location region, we use Definition 8 to formulate the relevance constraints that the central locations of the cover-up ranges should satisfy.

Definition 8 (Location Relevance): *For a user location sequence $\mathcal{L} = (L_1, L_2, ..., L_n)$, and a cover-up location sequence $\mathcal{L}^* = (L_1^*, L_2^*, ..., L_n^*)$ (L_k^* corresponds to L_k) constructed for $\mathcal{L}$ by a privacy system, to minimize the excludable area of each cover-up range, $\mathcal{L}^*$ and $\mathcal{L}$ should satisfy the following constraints.*

- *(C1) For any L_a^* and L_b^*, if L_a and L_b, which are the user query locations corresponding to L_a^* and L_b^*, are the same location, then L_a^* and L_b^* should also be the same location, i.e., $\forall L_a^* \forall L_b^* (L_a^*, L_b^* \in \mathcal{L}^* \wedge L_a = L_b \rightarrow L_a^* = L_b^*)$.*
- *(C2) For any L_a^* and L_b^*, if L_a and L_b belong to the same region D_1 with the level r, then there should exist a region D_2 with the level r, such that L_a^* and L_b^* belong to D_2, i.e., $\forall L_a^* \forall L_b^* \forall D_1 (L_a^*, L_b^* \in \mathcal{L}^* \wedge D_1 \in \mathcal{D}^r \wedge L_a, L_b \in D_1 \rightarrow \exists D_2 (D_2 \in \mathcal{D}^r \wedge L_a^*, L_b^* \in D_2))$.*

Fig. 2 illustrates the location relevance constraints of Definition 8, where the top half consists of three user query ranges (L_1, L_2 and L_3), and the lower half consists of the corresponding cover-up ranges (L_1^*, L_2^* and L_3^*). In the figure, the cover-up ranges L_1^* and L_2^* meet the first constraint of Definition 8 (i.e., they belong to the same location (9,7), as L_1 and L_2 belong to the same location (5,5)), and L_2^* and L_3^* meet the second constraint (i.e., they belong to the same region colored blue, as L_1 and L_2 belong to the same region colored red). It can be seen that the location relevance constraints make that the regularity among user query locations can be well exhibited by the cover-up ranges, so it is difficult for the attacker to reduce the active area of each cover-up range, resulting in that the cover-up range security described in Definition 4 can be guaranteed.

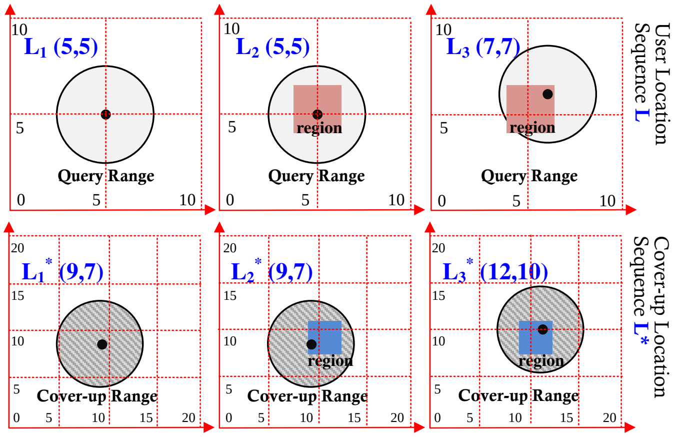


Fig. 2. An example for the location relevance constraint (where the numbers denote virtual coordinate values)

Now, according to Definition 3 (Accuracy), Definition 3 (Efficiency), Definition 4 (Security) and Definition 8 (Location Relevance), we further formulate the constraints that should be satisfied by a cover-up range sequence constructed by an effective location privacy-preserving system developed based on the framework of Fig. 1(a).

Definition 9 (Location Privacy Protection): *For a query range sequence $\mathcal{P} = (P_1, P_2, ..., P_n)$ and a cover-up range sequence $\mathcal{P}^* = (P_1^*, P_2^*, ..., P_n^*)$ constructed by a privacy system for $\mathcal{P}$, if both meet the following constraints, then it is deemed that the location privacy behind $\mathcal{P}$ can be protected effectively by $\mathcal{P}^*$ (where μ and ρ are given thresholds).*

- *(C1) Let $\mathcal{L}$ and $\mathcal{L}^*$ denote the location sequences related to $\mathcal{P}$ and $\mathcal{P}^*$, respectively. Then, $\mathcal{L}^*$ and $\mathcal{L}$ should meet the location relevance constraints of Definition 8.*
- *(C2) Any $(P_k, P_k^*) \in (\mathcal{P}, \mathcal{P}^*)$ should meet the cover-up range accuracy (P_k^* corresponds to P_k), i.e., $P_k \subseteq P_k^*$.*
- *(C3) Any $(P_k, P_k^*) \in (\mathcal{P}, \mathcal{P}^*)$ should meet the cover-up range efficiency, i.e., $EF(P_k, P_k^*) \geq \rho$.*
- *(C4) Any $(P_k, P_k^*) \in (\mathcal{P}, \mathcal{P}^*)$ should meet the cover-up range security, i.e., $PR(P_k, P_k^*) \geq \mu$.*

All the above nine definitions constitute the location privacy model, which formulates the constraints that the cover-up ranges constructed by an effective privacy-preserving system developed based on the framework of Section 3 should meet, thereby, providing reference for the design of a construction algorithm of cover-up ranges.

B. Implementation Algorithm

According to the privacy model defined in Section 4.1, we discuss its implementation algorithm, i.e., how to construct a cover-up range P^* for each user query range P, such that the finally obtained cover-up range sequence $\mathcal{P}^*$ with its user query range sequence $\mathcal{P}$ can well meet the constraints of

```
Algorithm 1: A construction algorithm for location cover-up range
Input: (1) $P$, a user query range; (2) $\mathcal{P}$, a user historical range sequence;
       (3) $\mathcal{P}^*$, a historical cover-up range sequence corresponding to $\mathcal{P}$; (4)
       $(\mu, \rho)$, threshold parameters; and (5) $B$, a search strategy.
Output: $P^*$, a cover-up range constructed for $P$.
1  begin
2      foreach $P_k \in \mathcal{P}$ do
3          if $P_k.L = P.L$ then
4              $P^* \leftarrow$ search($B, P, P_k^*.L$); /* $P_k^* \in \mathcal{P}^*$ denotes a
                   cover-up range corresponding to $P_k$ */
5              $P^*.R \leftarrow P^*.R \cdot (1 + \theta)$; // $0 \leq \theta \leq 0.2$
6              return $P^*$; // $\theta$ is an adjustable random value

7      for $r = 1, 2, ..., r^m$ do
8          foreach $D \in \mathcal{D}^r$ do
9              foreach $P_k \in \mathcal{P}$ do if $P_k.L \in D \wedge P.L \in D$ then
10                 Let $D^*$ denote a region of the same level to $D$ and
                   $P_k^*.L \in D^*$ ($P_k^*$ is a cover-up range of $P_k$);
11                 $P^* \leftarrow$ search($B, P, D^*$);
12                 $P^*.R \leftarrow P^*.R \cdot (1 + \theta)$;
13                 return $P^*$;

14 Procedure search($B, P, D^*$)
15 begin
16     $\mathcal{P}^* \leftarrow \{P^* | PR(P, P^*) \geq \mu \wedge EF(P, P^*) \geq \rho \wedge P \subseteq P^* \wedge P^*.L \in$
           $D^*\}$; /* obtain a set of range candidates */
17     if $B$ is Strategy 1 then                    /* Security First */
18         return $\arg\max_{P^* \in \mathcal{P}^*} PR(P, P^*)$;
19     if $B$ is Strategy 2 then                  /* Efficiency First */
20         return $\arg\max_{P^* \in \mathcal{P}^*} EF(P, P^*)$;
21     if $B$ is Strategy 3 then                     /* Balance First */
22         return $\arg\max_{P^* \in \mathcal{P}^*} PR(P, P^*) \cdot EF(P, P^*)$;
```

Definition 9. For given threshold parameters (μ, ρ), there may be no solution, which can meet the constraints of Definition 9 (e.g., when the security parameter μ and the efficiency parameter ρ are both set to be bigger), but there may also be many feasible solutions. Therefore, below, we introduce three strategies from different angles to search a feasible solution for cover-up range.

Strategy 1 (Security First): *For a user query range P, under the premise of meeting the constraint C1 of Definition 9, we search a solution as follows*

$$P^* = \arg\max_{P^\#} PR(P, P^\#) \textbf{\textit{s.t.}}$$
$$PR(P, P^\#) \geq \mu \wedge EF(P, P^\#) \geq \rho \wedge P \subseteq P^\#$$

Strategy 2 (Efficiency First): *For a user query range P, under the premise of meeting the constraint C1 of Definition 9, we search a solution as follows*

$$P^* = \arg\max_{P^\#} EF(P, P^\#) \textbf{\textit{s.t.}}$$
$$PR(P, P^\#) \geq \mu \wedge EF(P, P^\#) \geq \rho \wedge P \subseteq P^\#$$

Strategy 3 (Balance First): *For a user query range P, under the premise of meeting the constraint C1 of Definition 9, we search a solution as follows*

$$P^* = \arg\max_{P^\#} PR(P, P^\#) \cdot EF(P, P^\#) \textbf{\textit{s.t.}}$$
$$PR(P, P^\#) \geq \mu \wedge EF(P, P^\#) \geq \rho \wedge P \subseteq P^\#$$

According to the above three strategies, Algorithm 1 briefly describes an algorithm for constructing a cover-up range P^* for a query range P. In Algorithm 1, the main body (Lines 1 to 13) is used to search a suitable region for the central location $P^*.L$ of P^*, based on the relevance constraints among query locations. Here, the algorithm takes into consideration two cases as follows. (1) **Case 1** corresponds to the constraint C1 of Definition 8, i.e., if in the historical range sequence $\mathcal{P}$, there exists a historical user query range P_k, which has the same central location to the current range P ($P_k.L = P.L$), then the output cover-up range P^* also should have the same central location to the historical cover-up range $P_k^* \in \mathcal{P}^*$ corresponding to P_k. (2) **Case 2** corresponds to C2 of Definition 8, i.e., for any location region D with a level r, if it simultaneously contains the current range P and the historical range P_k, then the cover-up range P^* and its historical cover-up range P_k^* (corresponding to P_k) should be contained in a region D^* also with the level r.

In Algorithm 1, the **search** procedure (Lines 14 to 22) corresponds to Strategies 1 to 3, i.e., under the precondition of ensuring the location relevance constraints, the procedure searches for one feasible solution according to the principle of "security first", "efficiency first" or "balance first". Below, we introduce Theorem 1 to further demonstrate that the location relevance constraints of Definition 8 (C1 in Definition 9) can be ensured by the cover-up range sequence $\mathcal{P}^*$ obtained by running Algorithm 1 many times.

Theorem 1: *Let L_0^* denote the cover-up location constructed by Algorithm 1 for a user query location L_0. If a cover-up location sequence $\mathcal{L}^* = (L_1^*, L_2^*, ..., L_n^*)$ and a user location sequence $\mathcal{L} = (L_1, L_2, ..., L_n)$ meet the location relevance constraints of Definition 8, then the new sequences $\mathcal{L}_0^* = (L_0^*, L_1^*, ..., L_n^*)$ and $\mathcal{L}_0 = (L_0, L_1, ..., L_n)$ also meet the location relevance constraints.*

Proof. The location relevance constraints given in Definition 8 consist of C1 and C2. First, we prove that if the location sequences $\mathcal{L}^*$ and $\mathcal{L}$ meet C1, then the sequences $\mathcal{L}_0^*$ and $\mathcal{L}_0$, where L_0^* and L_0 are added, respectively, also meet C1. Let $L_k \in \mathcal{L}$ be the historical location determined by Algorithm 1 for L_0 (Line 3), i.e., L_0 and L_k are the same location ($L_k = L_0$). Then, from Algorithm 1 (Lines 4 and 16), we know that L_0^* is the same to the cover-up location $L_k^* \in \mathcal{L}^*$ corresponding to L_k, i.e., $L_k^* = L_0^*$. Now, we only need to prove that for any $L_j \in \mathcal{L} (j \neq k)$, if $L_j = L_0$, then $L_j^* = L_0^*$ ($L_j^* \in \mathcal{L}^*$ is the historical cover-up location corresponding to L_j). Since $L_j = L_k = L_0$, and $\mathcal{L}^*$ and $\mathcal{L}$ meet C1, we have that $L_j^* = L_k^*$, i.e., $L_j^* = L_0^*$.

Second, we prove that if $\mathcal{L}^*$ and $\mathcal{L}$ meet C2, then $\mathcal{L}_0^*$ and $\mathcal{L}_0$, where L_0^* and L_0 are added, respectively, also meet C2. Let $L_k \in \mathcal{L}$ be the historical location determined by Algorithm 1 for L_0, and D_1 a common region of L_0 and L_k (Lines 8 and 9), i.e., $L_0 \in D_1 \wedge L_k \in D_1$. Let D_1^* denote a location region with the same level with D_1, and it contains the cover-up location $L_k^* \in \mathcal{L}^*$ corresponding to L_k (Line 10), i.e., $L_k^* \in D_1^*$. Then, from Algorithm 1 (Line 16), we know that L_0^* constructed by the algorithm is certainly contained in the region D_1^*, i.e., $L_0^* \in D_1^*$. Below, we take into account two cases. (1) **Case 1**. We prove that if there is a region D_2, such that $L_0 \in D_2 \wedge L_k \in D_2$, then there is certainly a region D_2^* meeting that $L_0^* \in D_2^* \wedge L_k^* \in D_2^*$. From Algorithm 1, we see that D_1 is a location region with the lowest level, and it contains both L_k and L_0 (Line 6), so we have that $r_2 \geq r_1$, where r_1 and r_2 denote the levels of D_1 and D_2, respectively. After combined with the property of location regions (Definition 7), we know that there exists a region D_2^* with the same level to D_2, such that $D_1^* \subseteq D_2^*$, i.e., there exists a region D_2^*, making that $L_0^* \in D_2^* \wedge L_k^* \in D_2^*$. (2) **Case 2**. We prove that for any $L_j \in \mathcal{L}(j \neq k)$, if $L_j \in D_1 \wedge L_0 \in D_1$, then the cover-up location L_j^* corresponding to L_j is certainly contained in D_1^*, i.e., $L_j^* \in D_1^* \wedge L_0^* \in D_1^*$. Since $L_j \in D_1 \wedge L_k \in D_1$, and $\mathcal{L}^*$ and $\mathcal{L}$ meet the constraint C2, we have that $L_j^* \in D_1^* \wedge L_k^* \in D_1^*$. Finally, based on Cases 1 and 2, it is easy to further prove that $\mathcal{L}_0^*$ and $\mathcal{L}_0$ can meet C2. □

Based on Theorem 1, we conclude that the cover-up range sequence obtained by running Algorithm 1 several times can well meet the constraints of Definition 8, which enables the location privacy to be effectively protected. The time complexity of Algorithm 1 is equal to $O(r^m|\mathcal{P}|)$, where r^m is the highest level of location regions, and $\mathcal{P}$ is a user query range sequence.

In addition, it can be seen that Algorithm 1 is only targeted for a range query, without considering a nearest neighbor query. To make Algorithm 1 also suitable for a nearest neighbor query, for a nearest neighbor query $P_1 = (P_1.L, P_1.N)$ (where the parameter $P_1.N$ denotes the number of interest points), we only need to rewrite it to a range query request as $P_2 = (P_1.L, P_2.R)$ in advance, where $P_2.R$ can be obtained by an estimation function, whose input is $P_1.N$, and whose output is the radius $P_2.R$ of a minimum circular region that contains the $P_1.N$ points of interest nearest to the user.

V. Experiment

A. Experimental Setup

This section aims to evaluate our privacy-preserving system by experiment. We adopted a similar dataset used in [15], i.e., the map data was extracted from a square region of size $(80km \times 80km)$ of Connecticut. First, we divided the map into 80000^2 location cells, and divided all the regions into 4 levels, where Level 1 ($\mathcal{D}^1$) consists of 800^2 regions, Level 2 ($\mathcal{D}^2$) consists of 200^2 regions, Level 3 ($\mathcal{D}^3$) consists of 50^2 regions, and Level 4 ($\mathcal{D}^4$) is the map itself. To simplify the experiment, a user query request was represented by a query range P. We used the famous Brinkhoff road network data generator [30] to construct the location $P.L$ for each query range P, while the radius $P.R$ was set randomly within a certain range. The size of each user range sequence $\mathcal{P}$ was an experimental parameter, which can be adjusted dynamically. In addition, all the algorithms were written by the Java language. The experiments were performed on a Java virtual machine with an Intel i7 CPU and a 2G working memory.

In the experiment, the candidate algorithms that we used include: (1) LBS1_PR, i.e., the algorithm based on the security-first strategy; (2) LBS2_EF, i.e., the algorithm based on the efficiency-first strategy; and (3) LBS3_BA, i.e., the algorithm based on the balance-first strategy. It is obvious that the three candidates are proposed in this paper. In addition, for comparisons, we also introduced other candidates: (1) DUMMY [17], i.e., for a user location L, k dummy locations (where k is a parameter) are constructed and submitted along with L to the server; (2) NOISE [25], whose basic framework is similar to our method (i.e., using a random location L^* to replace the user location L), but L^* and L should meet geo-indistinguishability (a notion built on differential privacy); and (3) RANDOM (i.e., a random method), whose framework is identical to Algorithm 1, but the central location $P^*.L$ of each cover-up range P^* is randomly constructed. In the experiment, we did not compare against other algorithms mentioned in Section 2, since they are proposed under different system models or privacy models, thus they are incomparable to our method. Instead, in Section 6, we will briefly analyze the advantages and disadvantages of these algorithms.

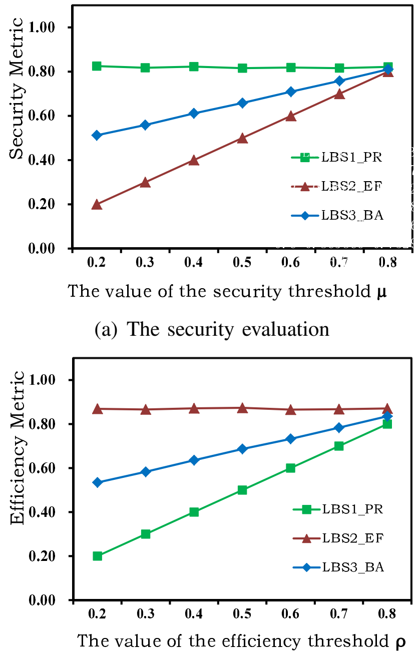
B. Experimental Result

In the first group of experiments, we aim to evaluate the impact of the cover-up ranges constructed by each strategy of our method in terms of location privacy security and location service efficiency. Based on Definitions 3 and 4, we define the metrics of efficiency and security of a cover-up range sequence $\mathcal{P}^*$ as follows

$$EF(\mathcal{P}^*) = \min_{P^* \in \mathcal{P}^*} EF(P, P^*)$$
$$PR(\mathcal{P}^*) = \min_{P^* \in \mathcal{P}^*} PR(P, P^*)$$

It is obvious that for the above two metrics, the bigger the values the better, which means that the better the effect of the cover-up ranges to the location privacy protection, and the smaller the impact on the location service efficiency. It can be seen that the measures mainly depend on the security and efficiency parameters (i.e., μ and ρ). It was found that the length of a cover-up range sequence $\mathcal{P}^*$ had a little impact on the metric values, so the length of each sequence was fixed to 1000. The experimental results are shown in Fig. 3, where the efficiency parameter ρ in Fig. 3(a) is set to 0.1, and the security parameter μ in Fig. 3(b) is set to 0.1.

From Fig. 3(a), we see that with the increasing of the value of μ, the security metric values of the cover-up range sequence constructed by LBS2_EF or LBS3_BA also increase, where the increase for LBS2_EF is almost linear. This is because for the efficiency-first strategy (LBS2_EF), the area of each of its constructed cover-up ranges is controlled only by the security parameter μ, not affected by the efficiency parameter ρ. In addition, the security of each cover-up range constructed by LBS1_PR is unchanged with the increasing of the value of μ. This is because the security-first strategy used by LBS1_PR would maximize the area of each cover-up range as much as possible, under the precondition of meeting the efficiency threshold ρ, such that the cover-up range area is controlled only by ρ, while the value of ρ is fixed in this group of experiments. From Fig. 3(b), we see that with the increasing of the value of ρ, the efficiency metric values of the cover-up range sequence constructed by LBS1_PR or LBS3_BA also increase (where LBS1_PR increases linearly), because the cover-up area is affected only by ρ at this time. However, for



(a) The security evaluation

(b) The efficiency evaluation

Fig. 3. The evaluation results for the constraints of security and efficiency

LBS2_EF, the efficiency metric values are unchanged with the change of ρ, because at this time, the construction of cover-up ranges is based on the efficiency-first strategy. In summary, the cover-up range sequence constructed by our method regardless of what strategy we used can well meet the efficiency and security constraints presented in Definition 9.

In the second group of experiments, through comparing the RANDOM method, we aim to evaluate whether the cover-up ranges constructed by our method can well meet the location relevance constraints. Therefore, we count the locations in a cover-up range sequence $\mathcal{P}^*$, which cannot meet the constraint C1 or C2 in Definition 8, i.e.,

$$RN1(\mathcal{P}^*) = |\{P \,|\, P \in \mathcal{P}^*, P \text{ cannot meet C1.}\}|$$
$$RN2(\mathcal{P}^*) = |\{P \,|\, P \in \mathcal{P}^*, P \text{ cannot meet C2.}\}|$$

For the metric $RN2$, to make the values not too large, we only count the locations that cannot meet C2 on the region level 1. As you can see, the metric mainly depends on the length of the cover-up range sequence $\mathcal{P}^*$. In this group of experiments, we randomly selected one strategy for our system, since the cover-up locations determined by each of the three strategies are the same. In addition, the length of a cover-up range sequence is set from 200 to 2000. The experimental results are shown in Fig. 4. From Fig. 4, we see that each cover-up range sequence constructed by our system can well meet the location relevance constraints (i.e., the number of locations that cannot meet C1 or C2 is close to 0), even if the length of a cover-up sequence has increased to a larger value (e.g., 2000). In addition, compared to our system, the cover-up range sequence from RANDOM exhibits an unsatisfactory effect on the location relevance constraints (especially for C1), which is mainly because for any locations, if they cannot meet C2, then they certainly cannot meet C1, and thus we have that $RN1(\mathcal{P}^*) \geq RN2(\mathcal{P}^*)$.

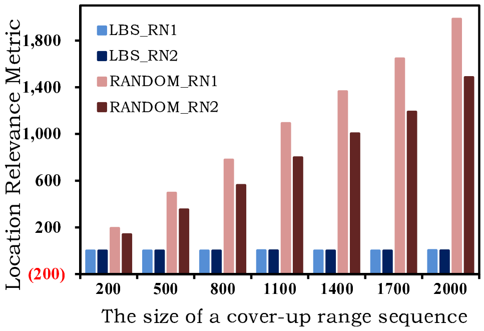


Fig. 4. The evaluation results for the location relevance constraints

In the third group of experiments, by comparing DUMMY and NOISE, we aim to evaluate the overall performance of our method in terms of location privacy security and location service efficiency. To make the comparison fairer, we redefine the location privacy metrics, namely, the impacts of a cover-up range P^* and k dummy locations on the location privacy are calculated as follows (where L_i^* denotes a dummy location):

$$PR(P^*) = |P^*|\sqrt{|P^*.L - P.L|}$$
$$PR(k) = \frac{k+1}{k}\sum_{i=1}^{k}\sqrt{|L_i^* - P.L|}$$

Since NOISE [25] has a basic framework similar to our method (using P^* to replace P), so its privacy metric is also calculated by $PR(P^*)$. It is obvious that the bigger the metric values the better, which means the better the location privacy is protected. In the experiment, we used the efficiency-first strategy LBS2_EF, since the cover-up range constructed by LBS2_EF has the worst security among the three strategies. In the experiments, we adjusted the related parameters to make the loss of service efficiency caused by the candidates gradually increased (compared to that of no privacy protection), and then observed and calculated the security metric values. The experimental results are shown in Fig. 5. As you can see from Fig. 5, with the increasing of the loss of efficiency caused by the privacy protection, the security metric values of the three candidates can be improved to a certain degree, but our method and NOISE has better performance than DUMMY. This is mainly because under the same loss of efficiency, compared to DUMMY, a cover-up range P^* constructed by our method or NOISE can cover more location cells. For example, when the efficiency loss is equal to 1.0, the number of dummy locations from DUMMY is equal to 1, so the probability of an attacker to guess the user true location is 0.5. However, at this time, the cover-up range can cover hundreds of location cells, so the probability that the attacker can guess the user location is less than one-percent, which greatly improves the effect on the location privacy protection. Also, it can be seen that our method and NOISE have similar performance in terms of location privacy security and location service efficiency. However, in the NOISE method, the LBS efficiency cannot be adjusted dynamically, and the LBS accuracy cannot be guaranteed (i.e., it cannot ensure that P is contained in P^*), so

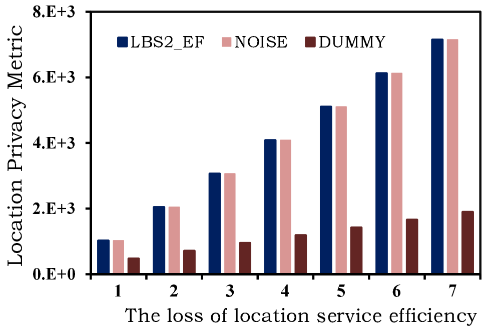


Fig. 5. The overall evaluation results for security and efficiency

our method has better overall performance in terms of security, efficiency and accuracy.

VI. Discussion

From the framework of Section 3.1, it can be seen that our system requires no change to the LBS accuracy, the LBS algorithm and the LBS platform. In addition, it can be also seen that the impact of the privacy protection on the efficiency is mainly dependent on the active area of each cover-up range, i.e., the greater the area size, the worse the location service efficiency. However, the active area of each cover-up range also affects the security of users' location privacy, i.e., the greater the area size, the smaller the risk of privacy exposure. As a result, the efficiency loss caused by the privacy protection is linearly positive to the level of users' location privacy protection, i.e., our system does not lead to a significant impact on the service efficiency. Below, we introduce two remarks to formally analyze the security of our method.

Definition 10 (Level I Privacy): *A privacy-preserving system developed based on the framework shown in Fig. 1(a) can meet the Level I privacy, if the attacker cannot immediately identify the user location $P.L$ or the query range P from any isolated cover-up range P^* constructed by the system for P.*

Remark 1: *The location privacy-preserving system developed based on our method can meet the Level I privacy.*

Rationale. In the cover-up range P^*, the central location $P.L$ has been replaced by $P^*.L$, making the probability of the attacker to identify $P.L$ from P^* equal to $\frac{1}{|P^*|}$ (obviously, it is a smaller value). Then, because the attacker does not know the radius $P.R$ of P, the probability of obtaining P is even smaller. Of course, if the attacker has mastered a copy of our algorithm (i.e., he has mastered the algorithm principle, the used strategy and the parameter setting) and known the radius $P.R$, he can identify a rough outer region within which $P.L$ is probably located, from the cover-up range P^*. However, in the algorithm, when determining the cover-up range radius (Lines 5 and 12), we introduced a random value θ, resulting in a still small probability of the attacker to obtain $P.L$, which is approximately equal to $\frac{1}{\theta \cdot |P^*|}$. In summary, the attacker cannot identify the user location from P^*. □

Definition 11 (Level II Privacy): *A privacy-preserving system developed based on the framework shown in Fig. 1(a) can meet the Level II privacy, if it can meet the Level I privacy, and the attacker cannot identify the user location $P.L$ or the user query range P from the cover-up range sequence $\mathcal{P}^*$ issued by the same user during a period of time.*

TABLE I

The effectiveness comparison between our method and other related ones, where $\surd$ denotes "good", $\odot$ "general" and $\otimes$ "not good".

Candidates	Security	Accuracy	Efficiency	Usability
Our method	$\surd$	$\surd$	$\odot$	$\surd$
Pseudonym	$\odot$	$\surd$	$\surd$	$\otimes$
Obfuscation	$\odot$	$\otimes$	$\surd$	$\surd$
Encryption	$\surd$	$\surd$	$\surd$	$\otimes$
Dummy	$\odot$	$\surd$	$\odot$	$\surd$

Remark 2: *The location privacy-preserving system developed based on our method can meet the Level II privacy.*

Rationale. At this time, the attacker can shrink the active area of a cover-up range based on his mastered location relevance regularity, to improve the probability of guessing user locations. Let us consider the example of two cover-up ranges P_1^* and P_2^* mentioned in Section 4.1 for location relevance. In the example, the attacker can conclude that the user location $P.L$ is certainly within P_1^* and P_2^*, such that the effect of the cover-up ranges to location privacy protection is reduced to $P_1^* \cap P_2^*$, i.e., the probability of the attacker to guess $P.L$ is equal to $\frac{1}{|P_1^* \cap P_2^*|}$. It results in a serious impact on the cover-up effect to user's location privacy, especially, if the area of $P_1^* \cap P_2^*$ is much smaller than that of P_1^* or P_2^*. However, such a situation has been considered by our system, i.e., the cover-up ranges can well reflect the location relevance regularity (e.g., the cover-up ranges P_1^* and P_2^* can meet that $P_1^*.L = P_2^*.L$), making that $P_1^* \cap P_2^* \approx P_1^*$ or P_2^*, so the cover-up effect can be well guaranteed.

Furthermore, if the attacker has mastered a copy of our algorithm running on the client-side, then he can input each location $L_k \in \mathcal{P}^*$ to the algorithm one by one, and then observe whether the output is the cover-up range P^*. If successful, then it indicates that L_k is the user location. However, such attempt will be unsuccessful, because in the algorithm, when determining the radius $P^*.R$ for a cover-up range P^*, we introduced a random value θ (Lines 5 and 12), which makes that the same input will lead to different output, and even if obtaining the same output, we still cannot conclude that the input is the same. From the above, we conclude that the attacker cannot identify the user locations from $\mathcal{P}^*$. □

In summary, although the attacker has mastered rich prior knowledge, it is still difficult to identify the user locations or ranges from the historical query sequence recorded by the server, so our system has good security. In addition, we know that: (1) for a pseudonym method, it is difficult to resist the threat from data mining, and hiding user identity also reduces the usability of the method; (2) an obfuscation method generally requires a compromise to the accuracy of LBS, and its implementation is dependent on a trusted third-party server, making it easy to lead to a privacy or efficiency bottleneck;

(3) an encryption method generally requires the change to the existing LBS algorithm, and the support of additional hardware, thereby, reducing the usability of the method; and (4) for a dummy-based method, the security is dependent on the quality of dummy construction, i.e., it is often threatened by the attack based on the feature distribution, resulting in poor security. In Table 1, we present a brief comparison between our method and other related ones. From the table, we see that compared with others, our system has better overall performance in terms of security, accuracy, efficiency and usability, so it can achieve the goal presented in Section 2.

VII. Conclusion

In this paper, we propose a location privacy-preserving system for LBS, whose basic idea is to construct high-quality "cover-up ranges" to make it difficult for an attacker on the untrusted server-side to learn users' query locations or query ranges. Specifically, the system consists of: (1) a client-based framework, which requires no compromise to the accuracy and usability of LBS; (2) a location privacy model, which formulates the constraints that ideal cover-up ranges should meet so as to ensure the security of users' location privacy and the efficiency of LBS; and (3) a privacy algorithm deployed on a client-side, which can construct cover-up ranges that well meet the constraints presented in the privacy model. Finally, both theoretical analysis and experimental evaluation demonstrate the effectiveness of the system, which can improve the security of location privacy, without compromising the accuracy and usability of LBS.

References

[1] M. Ghaffari, N. Ghadiri N, M. H. Manshaei et al. P4QS: A peer to peer privacy preserving query service for location-based mobile applications. IEEE Transactions on Vehicular Technology, 2017, 66(10): 9458–9469

[2] Z. Li, Q. Pei, I. Markwood et al. Location privacy violation via GPS-agnostic smart phone car tracking. IEEE Transactions on Vehicular Technology, 2018:1-1., 2018, 67 (6): 5042–5053

[3] T. Peng, Q. Liu, G. Wang. Enhanced location privacy preserving scheme in location-based services. IEEE Systems Journal, 2017, 11 (1): 219–230

[4] C. Anagnostopoulos, S. Hadjiefthymiades. Intelligent trajectory classification for improved movement prediction. IEEE Transactions on Systems Man and Cybernetics: Systems, 2014, 44 (10): 1301–1314

[5] H. Zhu, R. Lu, C. Huang et al. An efficient privacy-preserving location-based services query scheme in outsourced Cloud. IEEE Transactions on Vehicular Technology, 2016, 65 (9): 7729–7739

[6] T. Peng, Q. Liu, G. Wang. Enhanced location privacy preserving scheme in location-based services. IEEE Systems Journal, 2017, 11 (1): 219–230

[7] M. Damiani. Location privacy models in mobile applications: Conceptual view research directions. Geoinformatica, 2014, 18(4): 819–842

[8] C. Yin, J. Xi, R. Sun et al. Location privacy protection based on differential privacy strategy for big data in industrial internet-of-things. IEEE Transactions on Industrial Informatics, 2017

[9] X. Zhang, X. Gui, Z. Wu. Privacy preservation for location-based services: A survey. Chinese Journal of Software, 2015, 26(9): 2373–2395

[10] S. Chang, C. Li, H. Zhu et al. Revealing privacy vulnerabilities of anonymous trajectories. IEEE Transactions on Vehicular Technology, 2018, 67 (12): 12061–12071

[11] S. Gao, J. Ma, W. Shi et al. TrPF: A trajectory privacy preserving framework for participatory sensing. IEEE Transactions on Information Forensics and Security, 2013, 8(6): 874–887

[12] C. Chow, M. Mokbel, W. Aref. Casper: Query processing for location services without compromising privacy. ACM Transactions on Database Systems, 2009, 34(4): 1–48

[13] R. Dewri, R. Thurimella. Mobile local search with noisy locations. Pervasive and Mobile Computing, 2016, 32: 78–92

[14] S. Papadopoulos, S. Bakiras, D. Papadias. Nearest neighbor search with strong location privacy. PVLDB Endowment, 2010, 3(1): 619–629

[15] K. Mouratidis, M. Yiu. Shortest path computation with no information leakage. PVLDB Endowment, 2012, 5(8): 692–703

[16] A. Pingley, N. Zhang, X. Fu et al. Protection of query privacy for continuous location based services. Proc. of INFOCOM, 2011: 1710–1718

[17] B. Niu, Q. Li, Q. Zhu et al. Achieving k-anonymity in privacy-aware location-based services. Proc. of INFOCOM, 2014: 754–762

[18] B. Palanisamy, L. Liu. MobiMix: Protecting location privacy with mix-zones over road networks. Proc. of ICDE, 2011, 6791(4): 494–505

[19] B. Palanisamy, L. Liu. Attack-Rresilient mix-zones over road networks: Architecture and algorithms. IEEE Transactions on Mobile Computing, 2015, 14(3): 495–508

[20] B. Palanisamy, L. Liu, K. Lee et al. Anonymizing continuous queries with delay-tolerant mix-zones over road networks, Distributed and Parallel Databases, 2014, 32(1): 91–118

[21] R. Yu, J. Kang, X. Huang et al. MixGroup: Accumulative pseudonym exchanging for location privacy preservation in vehicular social networks. IEEE Transactions on Dependable and Secure Computing, 2016, 13(1): 93–105

[22] A. Xue, R. Zhang, Y. Zheng et al. Destination prediction by sub-trajectory synthesis and privacy protection against such prediction. Proc. of ICDE, 2013: 254–265

[23] A. Xue, R. Zhang, Y. Zheng et al. DesTeller: A system for destination prediction based on trajectories with privacy protection. PVLDB Endowment, 2013, 6(12): 1198–1201

[24] B. Agir, T. Papaioannou, R. Narendula et al. User-side adaptive protection of location privacy in participatory sensing. GeoInformatica, 2014, 18(1): 165–191

[25] M. E. Andres, N. E. Bordenabe, K. Chatzikokolakis et al. Geo-indistinguishability: Differential privacy for location-based systems. Proc. of CCS, 2013: 901–914

[26] G. Ghinita, P. Kalnis, A. Khoshgozaran et al. Private queries in location based services: Anonymizers are not necessary. Proc. of SIGMOD, 2008: 121–132

[27] Z. Wu, R. Li, Z. Zhou et al. A user sensitive subject protection approach for book search service. Journal of the Association for Information Science and Technology, 2019: doi.org/10.1002/asi.24227

[28] A. Suzuki, M. Iwata, Y. Arase et al. A user location anonymization method for location based services in a real environment. Proc. of SIGSPATIAL, 2010: 398–401

[29] R. Kato, M. Iwata, T. Hara et al. A dummy-based anonymization method based on user trajectory with pauses. Proc. of SIGSPATIAL, 2012: 289–300

[30] K. Chatzikokolakis, C. Palamidessi, M. Stronati. Constructing elastic distinguishability metrics for location privacy. Proc. on Privacy Enhancing Technologies, 2015(2): 156–170

[31] Y. Sei, A. Ohsuga. Location anonymization wth considering errors and existence probability. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017, 47 (12): 3207–3218

Zongda Wu is a full professor at Shaoxing University. He received his Ph.D. degree in Computer Science from Huazhong University of Science and Technology (HUST) in 2009. From 2019, he worked as a postdoctoral research fellow at Nanjing University. He has published more than 50 papers on many journals (such as IEEE TSC, JASIST, KIS, INS and WWW) and conferences (such as CIKM, ICDM and IJCNN).

Ruiqin Wang is an associate professor at Huzhou University. She received her Ph.D. degree in Computer Science from Zhejiang University (ZJU) in 2013. From 2019. She has published more than 20 papers on many journals. Her main research interests include data mining and social recommendation.

Xinze Lian received the Ph.D. degree iin Computer Science from Chinese Academy of Sciences in 2014. Now, he is an associate professor at Oujiang College, Wenzhou University. His research interests include information management and information processing. He has published more than 30 papers in many journals.

Guandong Xu is a research fellow in Faculty of Engineering and Information Technology, University of Technology, Sydney. His research interests include web information retrieval, web mining, web services etc. He has been actively involved in the research community by serving as a PC member for more than 50 conferences. He has published more than 100 papers on many journals (such as IEEE TKDE, IEEE TIFS and WWWJ) and conferences (such as CIKM, ICDM and SIGIR).

Enhong Chen is a professor and a vice dean at School of Computer Science and Technology, University of Science and Technology of China (USTC). He received his Ph.D. degree in Computer Science from USTC in 1996. He has been actively involved in the research community by serving as a PC member for more than 50 conferences, such as KDD, AAAI, ICDM and SDM. His research interests include semantic web, machine learning, data mining, web information processing etc.