IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 18, 2019, accepted June 23, 2019, date of publication June 28, 2019, date of current version July 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2925571

Location Privacy Preservation for Mobile Users in Location-Based Services

GANG SUN[1], SHUAI CAI[1], HONGFANG YU[1], SABITA MAHARJAN[2], VICTOR CHANG[3], XIAOJIANG DU[4], AND MOHSEN GUIZANI[5], (Fellow, IEEE)

[1]Key Laboratory of Optical Fiber Sensing and Communications (Ministry of Education), University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Simula Research Laboratory, University of Oslo, 0316 Oslo, Norway
[3]International Business School Suzhou, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China
[4]Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA
[5]Department of Computer Science and Engineering, Qatar University, Doha, Qatar

Corresponding authors: Gang Sun (gangsun@uestc.edu.cn) and Hongfang Yu (hfyu@uestc.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61571098, and in part by the 111 Project under Grant B14039.

ABSTRACT Because location-based cyber services are increasingly found in mobile applications (e.g., social networking and maps), user location privacy preservation is essential and remains one of the several ongoing research challenges. In this paper, we propose a region-of-interest division-based algorithm to Preserve the location Privacy of mobile device users in location-based Cyber Services (PPCS). Unlike existing methods, our proposed PPCS approach generates dummy locations while considering the semantic information of those locations. The PPCS algorithm enables the generated locations to exclude or reduce the exposure of a user's real location. In our analysis, we demonstrate that PPCS is resilient to both colluding attacks and inference attacks. We also evaluate the efficiency and demonstrate the utility of our proposed approach through extensive simulations.

INDEX TERMS Location privacy preservation, location-based service, semantic information, mobile users.

I. INTRODUCTION

Use of location-based service (LBS) applications from mobile devices and applications (apps) is rapidly increasing [1]. However, LBSs have privacy and security issues that need to be solved. For example, it has been demonstrated that user location information can be abused to facilitate nefarious activities such as cyberstalking [2], [3]. Unsurprisingly, the research community has expressed interest in designing location-based privacy techniques, including the following:

- Cloaking: sending a group of locations instead of a single real location [4];
- Dummy generation: creating fake queries and locations to hide users' true locations [5]; and
- Private information retrieval: searching data from a database without leaking query content or users' identities [6].

Because most mobile devices (e.g., smart phones, tablets, and smart entertainment systems in vehicles) include GPS modules, users can easily obtain accurate location information [7]. An LBS query can include the identity, point of interest (POI), real location, and region of interest (ROI) of the user. For example, when responding to a user's query, a service provider will deliver a POI in the user's ROI, such as a gas station, hospital, or supermarket, to the user.

The associate editor coordinating the review of this manuscript and approving it for publication was Honglong Chen.

A. RELATED WORK

Various approaches [8]–[11] have been proposed to solve the problem of location privacy preservation. For example, k-anonymity cloaking [8] is used to protect a client's location privacy in LBS and aims to make a client's location information indiscernible from other $k-1$ clients. In [12], a spatial-temporal cloaking algorithm was proposed that collects the LBS requests of k different users, forms a CR for a specific time period and then submits the k LBS requests to an LBS provider. In this scheme, a personalized k-anonymity model is used to allow a user to have different privacy demands in various contexts because different users can require different degrees of privacy in the same context. However, when the number of requests in a specified cloaking area is less than

k, a user's request is rejected. If the user density in a region is large, the k clients' locations may be close to each other; therefore, the client's location privacy may be divulged. The authors of [13] proposed a cloaking algorithm to safeguard a client's location privacy in vehicular networks in which the cloaking must include k or more different vehicles and l different road segments. However, an insufficient number of vehicles in the specified cloaking area can cause an unexpected response delay. When the number of road segments in the specified cloaking area is insufficient, the cloaking area must be enlarged, which can affect the quality of service. To better protect a client's location privacy, users interchange their pseudonyms within a mixed zone [14]. Thus, the relationship between clients' pseudonyms and their locations may be broken.

Consequently, security policies and cryptography-based approaches have become more popular. In contrast to the k-anonymity cloaking technique, the dummy location method aims to protect a client's location privacy by inserting many dummy locations in his or her LBS query without any third-party involvement [15], [16]. The authors of [17] proposed a dynamic pseudo-ID scheme in which diverse pseudo-IDs are used in different queries with the goal of unlinking the correlation between a client's real identity and the trajectory. The authors of [18] designed a framework for fine-grained privacy preservation in LBS for mobile users. The authors of [19] designed a private block retrieval protocol and proposed an efficient and secure location-based service system. In their proposal, users can retrieve information of interest associated with a current location without revealing their locations. The authors of [20] proposed a method that mixes the actual location of a user with that of other dummies and then submits a query to an LBS provider. The LBS provider searches for all the related POI locations and returns them to the user. The authors of [21] designed grid- and circle-based algorithms for generating dummy locations that consider regional privacy requirements. The authors of [22] proposed a distributed dummy client generation method to give clients control over their privacy protections. When generating dummy clients, that method selects clients with movement patterns that are close to the primary user's movement pattern based on his or her privacy requirements. The dummy location selection (DLS) algorithm [23] considers ancillary information that might be exploited by malicious users. The DLS algorithm adopts an enumeration method to select dummy locations according to an entropy metric [24] and attempts to choose the locations whose historic probability is similar to the user's true location, which enhances the entropy. The authors of [25] proposed two dummy-based solutions for privacy-aware users that preconsider the ancillary information [26] in LBS. In contrast to [23], the selected dummy locations are placed in a virtual circle or virtual grid, thus ensuring that the chosen dummy locations are not close to each other.

Obfuscation-based mechanisms send a user's real location by altering it in nonreversible ways. For example, the authors of [27] employed a time obfuscation-based scheme by sending dummy queries at leisure times to confuse adversaries by introducing extra background information. Authors in [28] introduced the N-Rand, N-Mix, and N-Dispersion techniques to add Gaussian noise when changing locations. In [29], two different obfuscation operators (R-family deobfuscation and E-family deobfuscation) were proposed to protect location privacy by manipulating the radius of the obfuscated area.

B. RESEARCH MOTIVATION

Despite the convenience and entertainment provided by LBS, a user's privacy may be compromised when LBS providers retain the user's accurate location. For example, if a dating app user near a Center for Infectious Disease requests LBS, attackers can infer both the sexual orientation of the user and that the user may have some sexually transmitted disease [30]. In addition, if an LBS provider is compromised, its attackers can take advantage of users' location information to track clients or leak users' private information to a third party for commercial gain or to facilitate hate crimes against certain populations. Thus, it is necessary to preserve the privacy of LBS users' locations. LBSs generate large amounts of data; thus, collectively, such data can be used to profile individuals for cyber threat intelligence.

However, the existing approaches have not yet considered semantic location information, i.e., the correlation between location and time. Therefore, these existing methods may have serious deficiencies; for example, the chosen dummy locations may not be similar to real locations or may represent locations with low probabilities of LBS queries. The existing dummy-location-based approaches can preserve the location privacy of a user only when the dummies cover the actual location of the client. To address this issue, Li et al. presented a geometric approach—n-CD—to preserve a client's location privacy [31]. The n-CD algorithm partitions a client's ROI into n sectors of equal size and then creates n cryptic disks (CDs) to cover the client's ROI. The n-CD algorithm sends the centers and radii of the n CDs rather than the real user location to an LBS provider. Although the query sent by this method does not contain any information about a user's real location, attackers can deduce that the user must be within a specific area.

C. RESEARCH CONTRIBUTIONS

In this research, we design a novel approach to Preserve the location Privacy of users in location-based Cyber Services (PPCS). The main contributions of this paper are as follows:

- We study location privacy preservation for mobile users in location-based cyber services and design an efficient algorithm to solve this problem.
- We propose an approach to generate dummy locations based on entropy while considering semantic location information that might be used by attackers.
- We divide the range of the user's interests into equal n sectors and then select one location in each sector with the maximum entropy for hiding the user's real location.

- We theoretically analyze the security performance of the proposed approach and conduct extensive simulation experiments under various scenarios to verify and evaluate the PPCS.

D. STRUCTURE OF THE PAPER

The remainder of this paper is organized as follows. We discuss the preliminaries and problem statement in Section II. We describe our PPCS approach, analyze and evaluate its security performance in Sections III through VI. Finally, Section VII concludes this study.

II. PRELIMINARIES AND PROBLEM STATEMENT

A. PRELIMINARIES

In this subsection, we provide the definitions of some terms used in our research.

1) LBS QUERY

Assume that there is a probability distribution of a user being in each possible location in an LBS system. Each user can send queries to the LBS provider. Here, an LBS query Lq is defined in Equation (1).

$$Lq = (u_{id}, \{(x, y), R, C\}), \tag{1}$$

where u_{id} denotes a user's identity; (x, y) represents the user's location information such that x and y represent latitude and longitude values, respectively; R represents the radius of the user's query scope (that is, the region of interest, ROI), which is at (x, y) with respect to the center; and C denotes the user's query content (that is, a POI).

However, the LBS provider may be malicious. Thus, the user's location may be disclosed if the user directly sends the query Lq to the LBS provider. In this paper, to preserve location privacy, we employ k-anonymity to preprocess the user's query, Lq. Thus, query Lq will be transformed to Lq^*, as follows:

$$Lq^* = (u_{id}, \{(x_1, y_1), \ldots, (x_k, y_k), R, C\}), \tag{2}$$

where $(x_1, y_1), \ldots, (x_k, y_k)$ are k dummy locations that obfuscate the user's real location (x, y). Thus, as the *LBS query* Lq is transformed to Lq^*, attackers will be unable to determine the user's real location from the k dummy locations, and the client's location privacy will remain protected.

2) LOCATION TYPE

The location information (x, y) is the real geographic location on the map (e.g., a Baidu or Google map). Typically, population density is low on roads but can be high at shopping mall, hospitals or restaurants. In this research, we assume that each location has its own location attributes and can be categorized according to their location attributes. Thus, an LBS provider can classify all the locations within its service area into different types based on location attributes. An example of various location types is shown in Figure 1. Here, we list only 5 location types. In this study, the LBS provider is responsible for disseminating the location types to the LBS users.

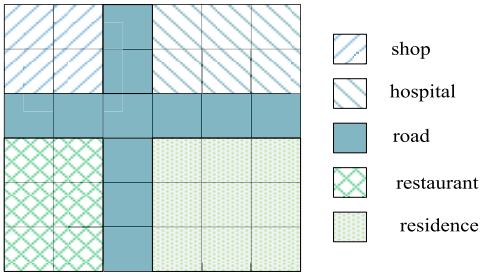


FIGURE 1. Examples of location types.

3) SEMANTIC LOCATION INFORMATION

In each location, users may request entertainment, medical treatment, transportation or other services. Thus, a correlation exists between the location type and the content of an LBS query. For example, when near a hospital a user has a higher probability of requesting medical treatment. However, the user may also request other services (e.g., transportation) from that same location (near the hospital). This correlation is called semantic location information. Through the semantic location information, we can analyze the user's request and obtain the location information to a certain degree.

To quantize the correlation between location type and content of an LBS query, a semantic parameter b_i is used to measure the semantic location information. A location type with a semantic parameter b_i that has a large value represents a high probability of users submitting LBS queries for that type of location. In this study, the users are responsible for specifying the semantic parameter to measure the semantic location information when sending requests in LBS.

4) ENTROPY

We employ entropy [24] to assess a user's privacy level. Entropy represents the uncertainty of identifying a client's true location among k dummy locations. For each location, we can calculate the corresponding historical query probability q_i [23]. We provide the definition of q_i in Equation (3).

$$q_i = \frac{number\ of\ queries\ in\ location\ i}{number\ of\ queries\ in\ all\ locations}. \tag{3}$$

Thus, based on the historical query probability q_i and the semantic parameter, we can amend the query probabilityq_i to $\overline{q_i}$ for location i:

$$\overline{q_i} = q_i \times b_i. \tag{4}$$

Based on the definition of entropy, we can compute the entropy H of k locations according to Equation (5):

$$H = -\sum_{i=1}^{k} \left\{(\log_2 \overline{q_i}) \times \overline{q_i}\right\}. \tag{5}$$

Then, from Equation (5), we can achieve the maximum entropy $H_{max} = \log_2 k$ when the k locations have the same query probability $1/k$. In addition, the sum of the k query probabilities must be 1. Thus, we need to normalize the k

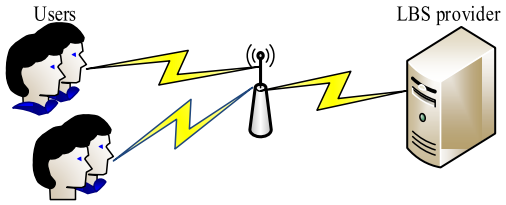


FIGURE 2. The system model.

query probability by rewriting the entropy H as

$$H = -\sum_{i=1}^{k} \{(\log_2 p_i) \times p_i\}, \quad p_i = \frac{\overline{q_i}}{\sum_{i=1}^{k} \overline{q_i}}. \tag{6}$$

B. SYSTEM MODEL

In this paper, we employ the distributed system model shown in Fig. 2, which is comprised mainly of the users and the LBS provider and no third party. Each user owns a mobile device used to request the LBS service. Furthermore, these mobile devices are capable of computing and storage. They can complete the process of transformation from Lq to Lq^* for the LBS query. However, the LBS provider is responsible for receiving and servicing the requests sent by the users. Then, the LBS provider offers query results to users.

In the distributed system model, the LBS provider can also calculate the historical query probability associated with all locations based on historical user logs and is responsible for disseminating and updating the historical query probability [23]. In addition, the LBS provider is responsible for classifying all the locations into different types based on their location properties and disseminating the location types. Therefore, attackers can compromise the LBS provider and obtain all its information. Moreover, the provider may also be an assailant due its interest in determining clients' locations for commercial gain. Thus, in the distributed system model, the LBS provider cannot be fully trusted and may be an attacker.

C. BASIC IDEA

We consider the scenario in which a location generation scheme selects dummy locations without considering the semantic location information. The location generation scheme can generate $k-1$ other dummy locations to safeguard a user's location privacy; then, the probability for leaking the actual location of a user is $1/k$. Attackers can exploit the semantic location information to filter k_f locations from the k locations, after which the probability of revealing a client's location information is enhanced to $1/(k-k_f)$. Figure 3 shows an example of selecting dummy locations using the location generation scheme. In Figure 3, the different cell shades represent different location types, and different location types have different semantic parameters. The dummy locations in the areas whose semantic parameter is less than 0.2 can be easily filtered by attackers. Thus, when using location generation schemes that do not consider the semantic location information, filtered locations can be eliminated to enhance the probability of deducing users' locations.

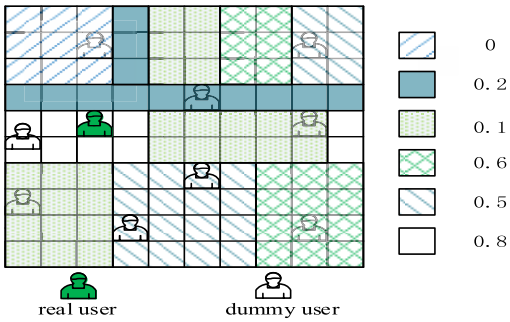


FIGURE 3. Example of dummy location generation.

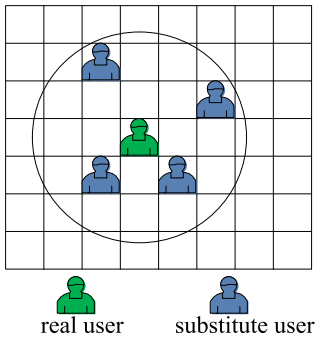


FIGURE 4. Example of replacing the user's real location.

To address this problem, our solution is to carefully select dummy locations based on the entropy value while considering the semantics of the location information, which can be used by hackers. We attempt to select dummies with similar/identical query probabilities. If we select only dummy locations that maintain a client's location privacy, the client's true location should be included among the dummies. To better protect a client's location privacy, we adopt a strategy to ensure that a client's actual location is not contained in these selected dummies. As shown in Fig. 4, we select dummy locations to replace the client's true location in a user's ROI and ensure that the client obtains the required information. This process is introduced in the PPCS algorithm described in next section.

III. ALGORITHM DESIGN

A. n-CIRCULAR AREAS

In our PPCS approach, if the actual location of a user is not contained in the dummy locations, then the ROI is partitioned into n sectors with equal sizes. Then, PPCS selects one location from each sector. The PPCS algorithm carefully generates n circular areas based on n locations that cover the client's ROI. Because a user's ROI is completely covered by the generated n circular areas, the query results for the user's ROI must be included in the query results of the generated n circular areas. Although an LBS query does not contain the client's location information, users can obtain information related to their interests. We provide a detailed description

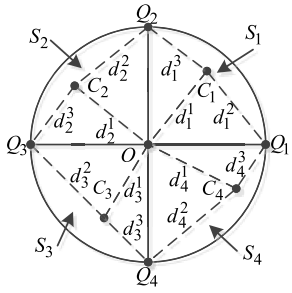


FIGURE 5. Example of dividing a user's ROI into four sectors.

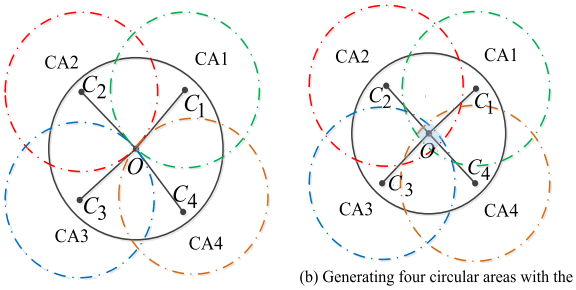


FIGURE 6. Generating circular areas to covering a user's ROI.

regarding how to cover a user's ROI by generating n circular areas using our PPCS approach. In our example, $n = 4$.

As shown in Fig. 5, our PPCS approach partitions the original ROI into four equal-sized parts, denoted by S_1, S_2, S_3 and S_4. The client's true location is the center O. In the client's ROI, the PPCS algorithm assumes that the four dummy locations have been carefully chosen based on the entropy of the four sectors, which are designated C_1, C_2, C_3 and C_4. The PPCS algorithm designates the $i^{th}(1 \leq i \leq 4)$ circular area as CA_i, and the radius and the center of CA_i are designated as r_i and C_i, respectively. To guarantee that CA_i completely covers S_i $(1 \leq i \leq 4)$, we set $r_i = \max\{d_i^1, d_i^2, d_i^3\}$, where d_i^1, d_i^2, *and* d_i^3 denote the line segments C_iO, C_iQ_i, C_iQ_{i+1} $(i \leq 3)$ and $C_iQ_1 (i = 4)$, respectively.

Although S_i can be completely covered by CA_i, a user's real location may still be exposed to adversaries. When $r_i = d_i^1 = C_iO$, the intersections of the four circles (i.e., point O) is the user's real location, as shown in Fig. 6(a). To ensure that none of the intersection points of the four circles includes the client's true location, the PPCS algorithm amends $r_i = d_i^1 \times (1 + D)$ when $r_i = d_i^1 = C_iO$, where $D \in (0.1, 0.5)$. Because the value of r_i $(1 \leq i \leq k)$ is greater than the value of the distance between O and C_i, the user's real location (i.e., the point O) must be located in the i^{th} circular area rather than on the i^{th} circle. Thus, after amending the radius of each circular area, none of the intersection points of the four circles is the user's real location, as shown in Fig. 6(b). Instead, the client's true location is in the intersection area (i.e., the shadowed area) of the four amended circles.

In this study, to prevent attackers from learning that the client's true location is included in the specific intersection area (i.e., the shadowed area in Fig. 6(b)) the value of n should be less than the anonymity degree k. Thus, to achieve k-anonymity, our PPCS approach must select an additional $(k\text{-}n)$ dummies based on their entropies. The result is that hackers and attackers cannot determine which intersection area contains the client's true location when more than one intersection area exists.

B. PPCS ALGORITHM

The basic idea of our PPCS approach is to employ the semantic position information that hackers and attackers can leverage to generate multiple dummy locations such that the query probabilities of all dummy locations are nearly equivalent. Our PPCS approach uses a greedy strategy to select dummies based on their entropies. From the definition of entropy H, the maximum entropy value of k locations is $\log_2 k$. An attacker is increasingly less able to identify the real location as the entropy increases. Thus, the greedy strategy selects each location to maximize the entropy. Without loss of generality, we introduce the entropy of the $i + 1$ iteration with the greedy strategy. First, we assume that i locations have been selected using the PPCS approach, where $k > i$. When choosing the $(i + 1)^{th}$ location, the PPCS algorithm must ensure that the query probability of the current selected location causes the function $H_{i+1}(p)$ to achieve the maximum value, in other words, the value closest to $\log_2(i+1)$. Assume that the query probability of the $(i + 1)^{th}$ location is p_{i+1} and that the query probabilities of i locations are the set $\{p_1, \ldots, p_i\}$. Then, based on Equation (6), we can compute the entropy $H_{i+1}(p)$ of the $i+1$ locations using Equation (7):

$$H_{i+1}(p) = -\sum_{j=1}^{i} \frac{p_j}{\sum_{l=1}^{i} p_l + p} \log_2 \frac{p_j}{\sum_{l=1}^{i} p_l + p} - \frac{p}{\sum_{l=1}^{i} p_l + p} \times \log_2 \frac{p}{\sum_{l=1}^{i} p_l + p}, \quad (7)$$

where p_j $(j \in [1, i])$ is a variable that represents the current query probability of a user at position j.

To achieve k-anonymity and $l/2$ diversity, we select l location types according to the semantic parameter of a client's true location. Then, we choose an additional $k - 1$ dummies based on entropy to protect the client's location privacy among the chosen location types, where a minimum of $l/2$ location types must be ensured. As shown in *Algorithm*-1, our PPCS algorithm is composed of *Procedure*-1 and *Procedure*-2.

Procedure-1 describes that the selected k locations exclude the true user location. When the k positions/locations that we choose do not include the client's true location, then the client's ROI is divided into n equal sectors. The PPCS algorithm then chooses n dummies based on the entropy from the ROI with n centers and generates n circular regions that completely cover the user's ROI. Note that n is less than k. Thus, to achieve k-anonymity, our PPCS approach must select an additional k-n dummies based on their entropy values.

Procedure-2, however, describes the situation in which the selected k locations do contain the true location of the user. The PPCS algorithm then directly chooses an additional $k-1$ dummies based on entropy. After choosing k dummies, the PPCS algorithm generates other appropriate dummy POIs to further obscure the client's true POI, if necessary. Detailed descriptions of the PPCS approaches are as follows:

(1) First, the anonymity degree (i.e., k) is determined by a user according to his or her location privacy requirement. A larger k value represents a higher user privacy requirement. Generally, the degree of anonymity k is greater than 1 because $k = 1$ indicates that the user does not care whether his or her location privacy is exposed. In this situation, users send their LBS queries directly to the LBS server.

(2) After obtaining the historical query probability that is associated with all the locations and location types in an LBS provider's service area, the user should prespecify a semantic parameter for each location type in the LBS provider's service according to the current time and his or her POI.

(3) According to the historical query probability associated with all locations and the semantic parameter of every location type, the PPCS algorithm calculates the current query probability according to the rule given in Section II. The PPCS algorithm then classifies the probabilities based on location type and sorts the current query probabilities of each type in ascending order.

(4) When the true location of a user is not included in the selected dummies according to the user's location type, the PPCS algorithm selects l different location types whose semantic parameters are similar to the user's location type. For every selected l location type, the PPCS approach chooses $k-1$ other locations whose current query probabilities are similar to those of the user's actual location. Thus, there are $(k-1)\times l$ candidate locations. To easily select dummies, our PPCS approach sorts all the candidate locations in ascending order according to their probabilities.

(5) In the sorted candidate locations, the PPCS algorithm selects $k-1$ locations whose existing query probabilities are similar to or the same as the true location of the user until it has selected a minimum of $l/2$ location types.

(6) If the dummy locations exclude the client's true location, the user must determine the value of n according to his or her privacy requirement, which determines the number of sectors into which his or her ROI should be divided. After obtaining the value of n, the PPCS approach divides the client's ROI into n parts of equal size. Then, the PPCS algorithm randomly generates one location in the client's ROI, which is part of the first sector, and applies a greedy strategy to select additional $n-1$ locations from the ROI based on their entropy values.

(7) Our PPCS approach generates n circular regions to completely cover the ROI of a user and uses the longest radius to update the radius of the ROI according to the rules introduced in Section III.A.

(8) If the chosen n locations have l^* different location types (where l^* is not less than l), the PPCS algorithm selects k-n other locations based on the entropy from the remaining locations in a greedy manner. If l^* is less than l, the PPCS algorithm has to select $(l-l^*)$ different location types, whose semantic parameters are similar to the l^* location types. For every l selected location types, PPCS selects $(k-n)$ locations whose current existing probabilities are similar to the first n chosen locations. Thus, $(k-1)\times l$ candidate locations exist. Then, the PPCS algorithm sorts all the candidate locations according to their probabilities in ascending order.

(9) From the sorted candidate locations, the PPCS selects (k–n) additional locations based on the entropy in a greedy manner while ensuring that at least $l/2$ location types exist.

(10) After selecting the k dummy locations, for users who want to preserve their POI, PPCS chooses $(\lfloor l/2\rfloor - 1)$ dummy POIs in which the users are more likely to have an interest. Then, the LBS query q^* is transported to the LBS provider. At this point, $q^* = (u_{id}, \{(x_1, y_1), \ldots, (x_k, y_k)\}, \overline{R}, \{P_1, \ldots, P_{\lfloor l/2\rfloor}\}$, where u_{id} denotes the user's identity, $\{(x_1, y_1), \ldots, (x_k, y_k)\}$ represents the coordinates of all the dummy locations, $\overline{R}$ denotes the radius of the client's ROI and the set $\{P_1, \ldots, P_{\lfloor l/2\rfloor}\}$ represents all the dummy POIs.

Algorithm 1 PPCS Algorithm

Input: (1) Historical query probability P;
(2) Parameters: *option*, k, l;
(3) LBS query $q = (u_{id}, \{(x, y), R, P\})$.
Output: LBS query q^*.
1: Compute the present query probability Q according to P and the semantic information;
2: Sort elements in Q according to location type;
3: if (*option* == 1) **then**
4: Call ***Procedure*-1**;
5: else
6: Call ***Procedure*-2**;
7: ifa user has a privacy preservation requirement,
8: Select $(\lfloor l/2\rfloor - 1)$ dummy POIs based on k dummies;
9: **return** $q^* = (u_{id}, \{(x_1, y_1), \ldots, (x_k, y_k)\}, R, \{P_1, \ldots, P_{\lfloor l/2\rfloor}\})$
10: else
11: **return** $q^* = (u_{id}, \{(x_1, y_1), \ldots, (x_k, y_k)\}, R, P)$

IV. ALGORITHM ANALYSIS

A. DEFENSE AGAINST A COLLUSION ATTACK

For nefarious purposes, passive hackers or attackers may cooperate with an LBS provider or other users to compromise some users' location privacy. Here, we demonstrate that the proposed PPCS algorithm can efficiently protect against such attacks.

Remark 1: If increasing the size of the colluding group cannot improve the possibility of identifying the true location of a user from the dummies, we consider this algorithm capable of protecting against a collusion attack.

Procedure 2 Generating Dummies That Include the True Location of a User

Input: (1) Location (x, y) of a user;
(2) Existing query probabilities;
(3) Parameters: k, l.
Output: k dummies $\{(x_1, y_1), \ldots, (x_k, y_k)\}$.
1: Select an additional $l-1$ types of locations considering the semantic location information;
2: for $(1 : l-1)$
3: Select $k-1$ candidate locations in each type of location;
4: Choose $k-1$ locations from the candidate locations based on their entropy values;
5: return

Procedure 3 Generating Dummies That Exclude the True Location of a User

Input: (1) Location (x, y) of a user;
(2) Parameters: k, l, n, R;
(3) Current query probabilities.
Output: k dummies and new radius $\{(x_1, y_1), \ldots, (x_k, y_k)\}, \overline{R})\}$.
1: Divide the ROI into n sectors;
2: Generate n circular regions to cover the ROI;
3: $l^* \leftarrow$ number of location types;
4: if $(l^* < l)$
5: Choose an additional $l - l^*$ types of locations;
6: for $(1 : l-1)$
7: Select k-n candidate locations;
8: Greedily choose additional k-n locations from the candidate locations;
9: return

The proposed PPCS approach can withstand a collusion attack. A collusion attack is a type of attack involving multiple users. The location privacy of a user can be efficiently preserved by choosing additional dummies in our PPCS approach. If a hacker or attacker (hitherto denoted as attacker) compromises user U_A, he/she can obtain the location information, which contains k locations of this user. Because the current query probabilities of these k locations are approximately the same and because any query probability may be the true location of a user, the attacker can randomly select a query probability from the k positions as the user's true location, even though the actual user location is not included in the k locations. Thus, the probability of identifying the actual location of the user is 0 or $1/k$. The attacker subsequently chooses user U_B and intercepts his or her LBS query and then obtains the location information of this user. However, the probability that the attacker can successfully infer the actual location of a user remains stable in our PPCS approach because the dummies for different users are independently generated. Therefore, among the intercepted k dummies, the attacker can only make an uncertain conjecture concerning the true location of the users, i.e., the attacker can randomly speculate each user's true location from the k dummies while colluding with numerous participants. Thus, we can conclude that the possibility of successfully identifying the real location of a user remains stable in our proposed PPCS approach.

B. DEFENSE AGAINST AN INFERENCE ATTACK

In this attack type, we assume that the LBS provider acts as an active attacker who knows the historical queries, the historical query possibility of each location, the users' current queries and the performance of a protection scheme.

Remark 2: If an attacker cannot correctly distinguish the user's true location from that user's location information, we consider this algorithm capable of defending against an inference attack.

The proposed PPCS approach can be used to defend against inference attacks. In our proposed PPCS approach, regardless of whether the k locations we select contain the user's real location, PPCS is capable of ensuring k locations with minimum differences in query probability. Thus, the LBS provider cannot be sure that the location information includes the user's true location when it receives the user's query.

In the analysis in this subsection, we assume that the location information contains the true location of a user and that the LBS provider knows this information. Although the LBS provider has the historical query possibilities for all locations, it cannot readily identify the true location of the user because the k positions from the user have almost the same current query probabilities and the location types have similar semantic parameters and historical query probabilities. Despite the efforts of the LBS provider, it will fail to reverse-engineer our PPCS scheme because it does not know the location types prespecified by users or their semantic location information. Although the LBS can predict the semantic information of each location type, that information is likely to differ from the parameters prespecified by the user, which may produce different dummy selection results.

In contrast, if we assume that the LBS provider knows that the true location of the user has not been included in the location information, then the LBS provider cannot deduce the user's true location because it does not know the number of sectors or the selected locations for a specific user because our PPCS approach randomly selects dummies from candidate locations. Although the LBS provider can attempt to deduce the true location of the user based on the spatial distribution of the selected k dummy locations, this attempt will fail because the number of sectors is smaller than the anonymity degree k in our PPCS approach. Thus, the k circular areas may generate more than one intersection area, and some intersection areas will not enclose the real user location. If the k circular areas generate more than one intersection area, the provider will be unable to identify the intersection area that includes the user's true location. Even if the k circular areas imply one intersection area, the provider knows only that the intersection area contains the true location (e.g.,

the shadowed area in Figure 5(b)), but will still be unable to identify the user's true location.

C. THEORETICAL ANALYSIS

In this subsection, we assume that an LBS provider may be acting nefariously. Although the selected k locations have equivalent or similar historical query probabilities in both the PPCS approach and the DLS approach [23], the average probabilities of successfully deducing the true location of a user differ between these two algorithms. Although the selected dummies cannot contain the true location of a user in either the PPCS or the n-CD algorithm, the average probabilities of successfully conjecturing the true location of a user using these algorithms also differs. Let event X represent the LBS provider, which filters the true location of a user from k locations, and let $P(X = 0)$ and $P(X = 1)$ denote the probability of failure and the probability of success, respectively.

In the DLS algorithm [23], although the location information contains the true user location, the LBS can only randomly conjecture the true location of the user from k locations when it obtains the user's LBS query. Thus, we have $P(X = 0) = (k-1)/k$ and $P(X = 1) = 1/k$. Then, the mathematical expectation of X can be computed by Equation (8):

$$E(X)_{\mathrm{DLS}} = \frac{1}{k}. \tag{8}$$

In our PPCS approach, the LBS provider cannot determine whether the actual location of a user is included in the location information; it can determine only whether the k locations include the true location of the user by the probability of 1/2 and conjecture the true location of the user from k locations when it receives the user's LBS query. If the LBS provider determines that the true location of a user is not contained in k locations, then it must guess which locations include the true user location. Here, we assume that an LBS provider is aware of the number of sectors of the ROI and that the LBS provider can obtain the actual location of a user once it determines the n locations. If the LBS provider determines that k locations do not contain the true location of the user, then the probability of identifying the true location of the user becomes $1/num$. If the LBS provider determines that one of the k locations contains the true user location, then the probability of selecting the true location of the user becomes $1/k$. Thus, we can obtain $P(X = 1) = (1/k + 1/num)/2$ and $P(X = 0) = 1-(1/k+1/num)/2$ for the PPCS approach. The mathematical expectation for X is defined in Equation (9):

$$E(X)_{\mathrm{PPCS}} = \frac{1}{2num} + \frac{1}{2k}. \tag{9}$$

when k is greater than n, k must be smaller than num; therefore, we know that $1/k$ must be larger than $(1/2k + 1/2\ num)$, i.e., $E(X)_{\mathrm{DLS}}$ is larger than $E(X)_{\mathrm{PPCS}}$.

In the n-CD algorithm [31], a user's ROI is directly divided into n sectors, where the value of n is equal to the value of k. Then, we have $P(X = 1) = 1$ and $P(X = 0) = 0$. As shown in Eq. (10), we obtain the expectation of X. When an LBS provider discovers that one of the n locations contains the true user location, we assume that it would be able to locate the user's true location.

$$E(X)_{\text{n-CD}} = 1. \tag{10}$$

The n-CD algorithm achieves the greatest E(X) of the three previously mentioned algorithms using the same assumptions.

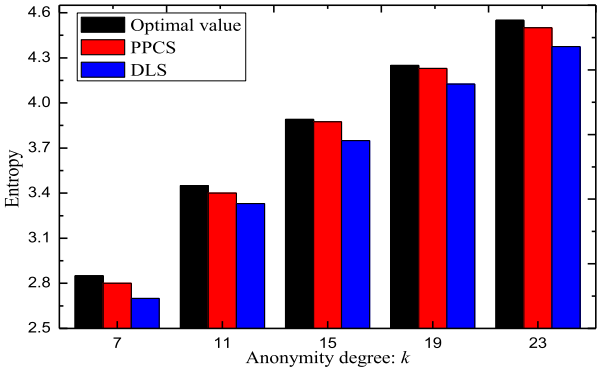


FIGURE 7. Entropy achieved by various schemes.

V. SIMULATION AND ANALYSIS

A. SIMULATION ENVIRONMENT

To assess the performance of our designed PPCS approach, we implemented it on a server with a 3.0 GHz CPU and 4 GB of memory. In this implementation, we assume that the LBS provider's service area contains 1600 equal-sized cells, in which we generated 20 uniformly distributed POIs. Each location is denoted by one cell. We associate each location with a corresponding location type. In our simulations, we assume that different users have different preferences regarding anonymity.

B. SIMULATIONS FOR PRIVACY LEVEL

In the scenario in which a user's real locations are included in the dummy locations, we compare the privacy level of our proposed PPCS algorithm and the privacy level of the DLS algorithm [23] in terms of entropy.

Fig. 7 presents the simulation results for the privacy levels of different approaches. The optimum entropy $Hmax$ can be calculated as $Hmax = \log_2 k$. Generally, the entropy increases as k increases. The optimal solution achieves the highest entropy because the k value of every submitted location has the same current query probability. Compared to the DLS scheme, the PPCS scheme achieves higher entropy because the DLS scheme considers only ancillary information that hackers can exploit, whereas PPCS considers either the semantic location information or the ancillary information of each location type. The PPCS algorithm can achieve an entropy close to the optimum value, which indicates that PPCS can efficiently protect a user's location privacy.

We investigate the relation between k and the privacy level according to entropy and running time when a user's real locations are included in the dummy locations. Fig. 8 and

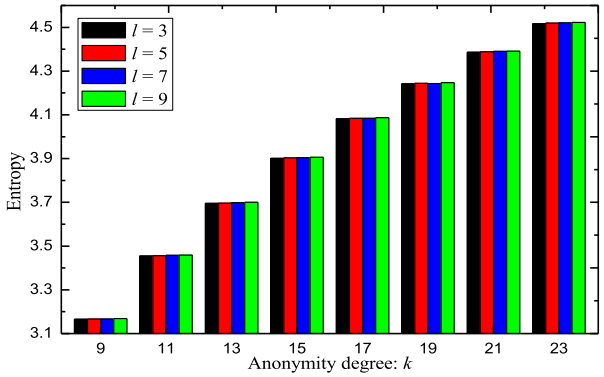


FIGURE 8. Entropy of PPCS for various l values.

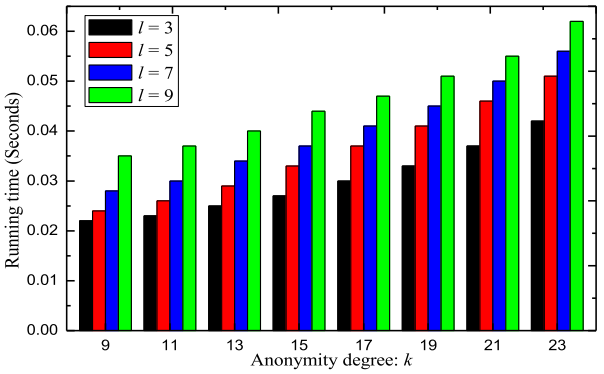


FIGURE 9. Running time of PPCS for various l values.

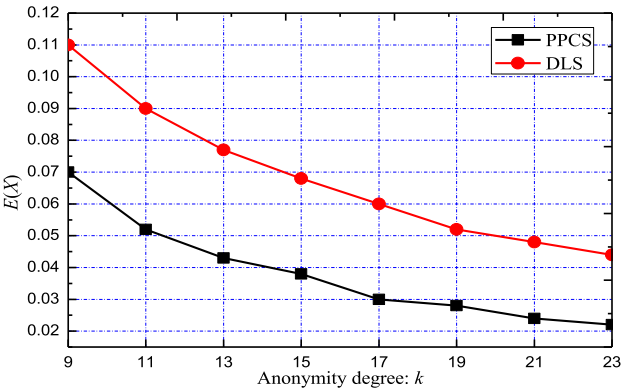


FIGURE 10. Simulation results for E(X).

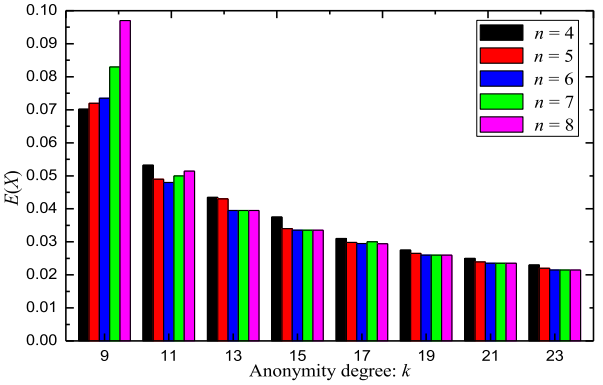


FIGURE 11. Simulation results for E(X) of PPCS.

Fig. 9 illustrate the running times and entropies of the PPCS scheme, respectively, for various l values. As shown in Fig. 8, the entropy always increases as k increases. Different l values result in different entropies (i.e., a larger l value generates a larger entropy because the value of l directly influences the number of candidate dummies selected, which enables PPCS to generate multiple suitable dummies. Fig. 9 shows the increase in the running time of PPCS for various scenarios. Additional candidate dummies must be selected by PPCS while the value of k increases, resulting in greater time consumption when generating dummies. As shown in Fig. 9, a longer running time yields a larger l because the selected k locations must belong to at least $l/2$ location types. When this condition is not satisfied, other suboptimal dummy locations must be chosen to replace dummies until the condition is satisfied.

C. SIMULATIONS FOR E(X)

From the performance analysis in Section IV, the n-CD approach achieves the highest average probabilities for successfully identifying a user's true location (i.e., $E(X)$) under our assumptions. In this subsection, we compare the $E(X)$ of the DLS and PPCS algorithms.

We explore the relationship between $E(X)$ and the value of k. In this group of simulations, we let n equal four; the $E(X)$ of DLS and PPCS are defined in Eq. (8) and Eq. (9), respectively. Note that $E(X)$ is influenced by both k and n in the PPCS algorithm. As shown in Fig. 10, $E(X)$ decreases when k increases because a larger k indicates that the LBS provider cannot readily identify the real location of a user. Our PPCS guarantees that the selected dummies do not include the real location of a user, which enhances the challenges to the LBS provider in distinguishing the true user location.

Fig. 11 provides the simulation $E(X)$ results of the PPCS approach for various values of n. Generally, $E(X)$ decreases while the value of k increases for a given n. $E(X)$ first fluctuates and then remains stable. In contrast, the value of n increases for a fixed k for the following reasons: the number of cases for randomly selecting n locations from k locations is C_k^n, i.e., $num_{max} = C_k^n$. Because $C_k^n = C_k^{k-n}$, for a specific k, C_k^n increases with increasing n when n is less than $k/2$. However, C_k^n increases while n increases if n equals $k/2$. A similar trend appears when C_k^n is less than num for a given value of k.

D. SIMULATIONS FOR ANONYMITY ZONE

Hackers can know only that the true location of a user is not contained in the chosen k locations. The hackers can infer which intersection area includes the true location of a user based on the spatial distribution of the k chosen dummy locations. We simulate the worst situation, in which a hacker knows that the true location of a user is not included in the k chosen locations. Although the hacker can infer the

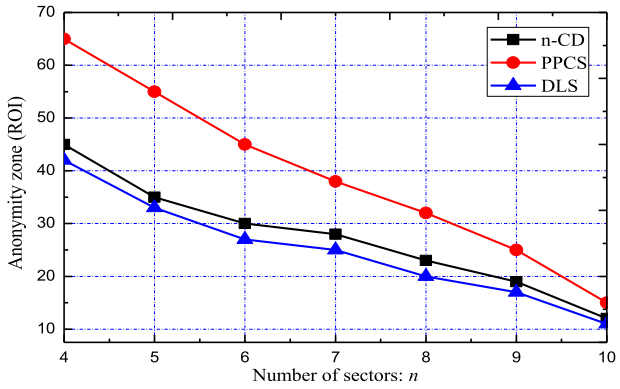


FIGURE 12. Simulation results for the anonymity zone.

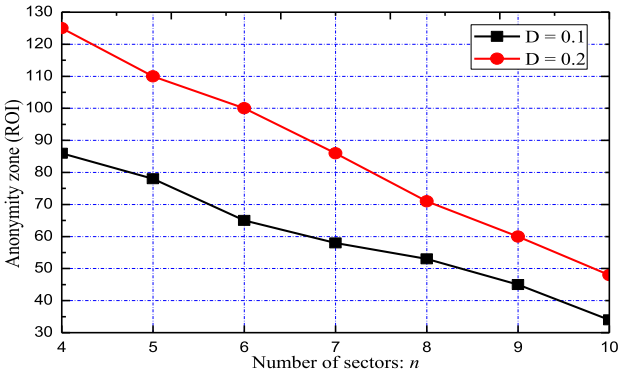


FIGURE 13. Anonymity zone achieved by PPCS.

true user location with a low probability, he or she knows that the user must be within a certain intersection region (e.g., the shadowed area in Fig. 6(b)) according to the spatial distribution of the k chosen dummy locations, which is called the hidden *anonymity zone* of a user. That is, the anonymity zone is the ROI of the target users that hackers cannot infer. In our designed PPCS approach, because the value of n is less than the anonymity degree k, the spatial distribution of the k chosen dummy locations generate anonymity zones, and some of these locations may not contain the true user location. In our simulation, we assume that the hacker knows the anonymity zone that encloses a user's real location. Based on these assumptions, we evaluate the user's hidden anonymity zone. In our designed PPCS, the anonymity zone is affected by the value of n and the amended parameter D for a specific radius of the user's ROI.

Fig. 12 provides the simulation results for the anonymity zone of our PPCS and the n-CD approach proposed in [31]. In these simulations, we fix R at ten. The size of the anonymity zone decreases as n increases. A large n produces a smaller radius of each circular area, which decreases the size of the anonymity zone (i.e., the overlapping of n circular areas). Fig. 12 shows that the anonymity zone of the PPCS approach is substantially larger than that of the n-CD approach. The size of the anonymity zone obtained by the PPCS approach decreases more rapidly than that of the n-CD algorithm as the number of sectors increases. Our PPCS approach sets the ROI radius as the maximal radius of the n generated circular regions, unlike the n-CD approach. Therefore, our PPCS approach more efficiently preserves user privacy than does the n-CD approach.

Fig. 13 shows the impact of parameter D on the anonymity zone when $R = 10$. In PPCS, to generate n circular areas, we define parameter D to amend their radii such that none of the intersection points of the n circles is the true user location. As shown in Fig. 13, a large D value yields a large anonymity zone because a large D results in a large ROI radius for a user. Thus, the size of the intersection zone of the n circular areas is large. When $D = 0.2$, the rate of decline of the anonymity zone size is higher than the rate of decline of $D = 0.1$ because the semidiameter of the n created circular areas decreases as n increases, and a large D value causes the size of the anonymity zone to decline at a faster rate.

TABLE 1. Core syntax of the LBS(Service()).

```
while LBS(service()) do
    check(status()); //to check whether the status is 0 or 1
    if (privacy == 1)
        firewall(status()); identity(status());
    else
        report(status()); //report issues to the system;
        action((status()); firewall(status());
        identity(status()); report(status()); //report the system;
    end if
end while
```

VI. EXPERIMENTS WITH LARGE-SCALE SIMULATIONS AND REAL LOCATIONS

Experiments with large-scale simulations and real locations are important for validating our PPCS approach. We developed an LBS Facebook API to test the validity of PPCS. The system design and implementation are based on the cybernetics cloud software [32]. We used extensible access control markup language (XACML) to define the privacy setting, which can be enhanced by the LBS API. The core concept is to add additional layers to hide users behind an identity-free firewall and an anonymous identity layer. Thus, two layers of defense are involved. First, a firewall layer protects all users. Second, an anonymous identity layer completely hides users' identities. The term "status ()" indicates that the LBS offers real-time privacy enforcement. All LBS commands support the functioning of these two privacy layers.

Table 1 lists the core syntax that enables the CTI LBS for cyber security to attain the maximum level of privacy. When the privacy is on (equal to 1), then the firewall service and anonymous identity service are prompted to switch on. If they cannot, the algorithm reports to the system, prepares the LBS to take control, and starts both the firewall service and the anonymous identity service before reporting to the central service that the LBS(service()) is functional.

where

- "LBS(service( ))" is used to initiate the LBS and prompt for a sequence of actions to support the LBS.

- "check(status( ))" is used to denote privacy status as 0 or 1. When the status is 1, the risk of identity exposure is high, and the identity firewall is set to the highest level to cause it to revert to 0.
- "firewall(status( ))" is used to enable the firewall (setting it to on).
- " (status( ))" is used to turn the anonymous identity to on to maintain privacy.
- "report(status( ))" is used to report to the central systems about issues regarding the service or at the end of the commands.
- "action(status( ))" is used to prepare all steps for "privacy == 1".

A. ACCURACY TESTS

One approach to testing the accuracy of the CTI LBS for cybersecurity is to adopt the F-measure, precision and recall metrics. Precision is the ratio of correctly detected positions with privacy preserved to the number of all detected positions with privacy preserved:

$$precision = \frac{t_p}{t_p + f_p}. \tag{11}$$

Recall is the ratio of the true detected positions with privacy preserved to the number of known positions:

$$recall = \frac{t_p}{t_v}, \tag{12}$$

where

- True positives (t_p) represent the number of correctly detected positions with privacy preserved;
- False positives (f_p) refer to the number of detected positions that do not actually exist; and
- True detected positions (t_v) represent the total number of correctly detected positions ($t_{p+}f_n$).

The F-measure is presented in terms of precision and recall as expressed in Eq. (13):

$$F - measure = \frac{2 \times precision \times recall}{precision + recall}. \tag{13}$$

High values F-measure values indicate better reliability [33]. To ensure a reliable measurement, we conducted two sets of experiments using private cloud services with 100 CPUs running at 3.0 GHz each with 64 GB of RAM and 2 TB of storage to perform the experiments. In the first set of experiments, a maximum of 10,000 experiments are performed to identify the extent of the F-measure over the number of experiments. In the second set of experiments, the F-measure values are tracked for 10,000 experiments over a period of time to test the robustness. We compare the LBS(service( )) = "on" and the LBS(service( )) = "off" to identify any differences between the sets of experiments. The results of the first set of experiments are presented in Figure 14. The F-measure values range between 0.89 and 0.91 for the LBS(service()) = "on" and decrease to between 0.41 and 0.44 as the number of experiments increases to

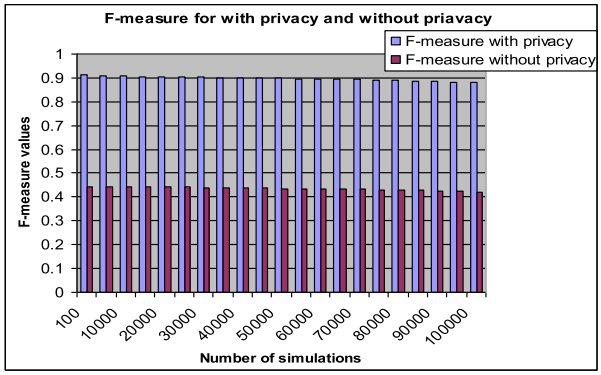


FIGURE 14. F-measure values for a maximum of 100,000 simulations between an LBS((service) that is "on" and an LBS((service) that is "off".

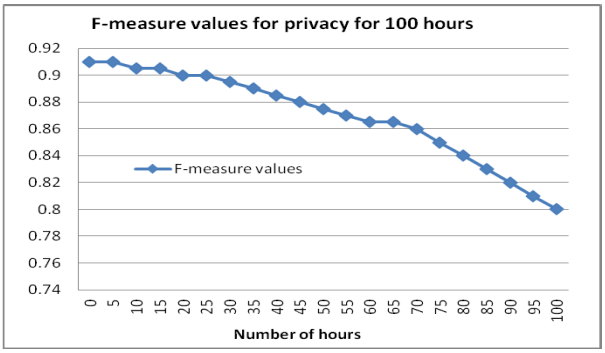


FIGURE 15. F-measure values when the LBS(service()) is "on" over a period of 100 hours.

100,000. These results indicate that when the LBS(service()) is "on", the F-measure values remain large and ensures a greater level of location privacy under the experimental conditions.

The goal for the second set of experiments is to identify the quality of service (in terms of F-measure values) over a period of time when the LBS(service()) is "on" (because the F-measure values must remain high for as long as possible to protect the users). The experiments were conducted over a period of 100 hours to test the robustness of LBS(service()). We obtained the results five times per hour and calculated the mean values. As shown in Figure 15, the F-measure values decrease from 0.91 to 0.80 by the end of 100 hours of service. The F-measure values are 0.86 after 70 hours. At this point, the quality of service (QoS) has decreased significantly.

B. EXPERIMENTS WITH REAL LOCATIONS

Experiments with real locations for the LBS for cybersecurity are required to test the robustness of our proposal by involving actual people directly in the verification process. Gedik and Liu [12] demonstrated their k-anonymity-based architecture and algorithms through simulation results but did not test with real data. We tested our work with 20 real volunteers to ensure the validity of our LBS service.

Twenty volunteers were recruited in Southampton and London. Each volunteer had an LBS setting of "on" or

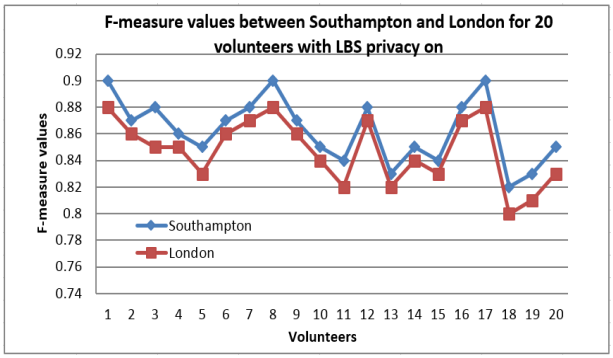


FIGURE 16. F-measure value comparison between Southampton and London with LBS privacy "on" for the first hour.

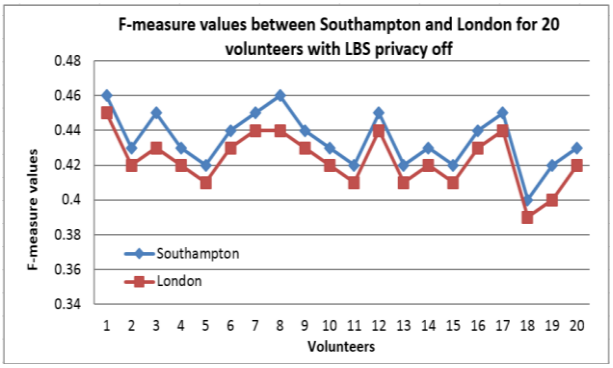


FIGURE 17. F-measure value comparison between Southampton and London with LBS privacy "off" for the first hour.

"off" and were located at least 100 meters apart to test the validity. The results from each volunteer were treated as individual results. All the experiments were performed five times, and the mean values were calculated. The F-measure values indicate the extent of the accuracy in these experiments. Figure 16 shows the F-measure values of the same volunteers/identities between Southampton and London with LBS privacy set to "on" for the first hour. Higher F-measure values were reported in Southampton. Because London has more people, buildings and traffic than Southampton, the validity and robustness of the F-measure values can be expected to differ.

Figure 17 shows the F-measure values of the same volunteers/identities between Southampton and London with the LBS privacy set to "off" for the first hour. The trend is similar to that of Figure 15, with the exception that the F-measure values are close to half of the F-measure values in Figure 15. Our proposed LBS privacy preservation approach provides significantly better location privacy and user anonymity in our real-location experiments.

VII. CONCLUSION

Ensuring the location privacy of LBS users is important in our increasingly connected society. To effectively preserve users' location privacy, we design an efficient approach named PPCS that efficiently generates dummy locations for which we consider the semantics of location information that may be used by hackers. Our proposed PPCS approach can generate dummy locations that do not contain the real location of a user and resist inference and collusion attacks. The simulation results show the effectiveness of our PPCS approach. Compared with the existing approaches, our PPCS approach has the average optimization gain of 85% and 60% on $E(X)$ and ROI metrics, respectively.

However, the trajectory privacy preservation for mobile users in LBS leaves as a challenge. In our future work, we are going to design efficient framework and algorithms to protect the trajectory privacy of users in LBS.

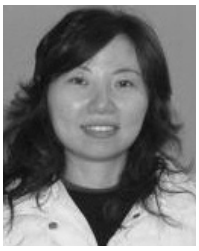
REFERENCES

[1] G. Sun, S. Sun, J. Sun, H. Yu, X. Du, and M. Guizani, "Security and privacy preservation in fog-based crowd sensing on the Internet of vehicles," *J. Netw. Comput. Appl.*, vol. 134, pp. 89–99, May 2019.
[2] R. Gupta and U. P. Rao, "An exploration to location based service and its privacy preserving techniques: A survey," *Wireless Pers. Commun., Int. J.*, vol. 96, no. 2, pp. 1973–2007, 2017.
[3] G. Sun, Y. Xie, D. Liao, H. Yu, and V. Chang, "User-defined privacy location-sharing system in mobile online social networks," *J. Netw. Comput. Appl.*, vol. 86, pp. 34–45, May 2007.
[4] C.-Y. Chow, M. F. Mokbel, and X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based service," in *Proc. 14th Annu. ACM Int. Symp. Adv. Geographic Inf. Syst.*, 2006, pp. 171–178.
[5] H. Shen, G. Bai, M. Yang, and Z. Wang, "Protecting trajectory privacy: A user-centric analysis," *J. Netw. Comput. Appl.*, vol. 82, pp. 128–139, Mar. 2017.
[6] M. Ghaffari, N. Ghadiri, M. H. Manshaei, and M. S. Lahijani, "P^4QS: A peer-to-peer privacy preserving query service for location-based mobile applications," *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 9458–9469, Oct. 2017.
[7] K. Shi, M. Xu, H. Jin, T. Qiao, X. Yang, N. Zheng, J. Xu, and K.-K. R. Choo, "A novel file carving algorithm for national marine electronics association (NMEA) logs in GPS forensics," *Digit. Invest.*, vol. 23, pp. 11–21, Dec. 2017.
[8] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. 1st Int. Conf. Mobile Syst. Appl. Services*, 2003, pp. 31–42.
[9] Z. Zhou, H. Zhang, X. Du, P. Li, and X. Yu, "Prometheus: Privacy-aware data retrieval on hybrid cloud," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2643–2651.
[10] G. Sun, D. Liao, H. Li, H. Yu, and V. Chang, "L2P2: A location-label based approach for privacy preserving in LBS," *Future Gener. Comput. Syst.*, vol. 74, pp. 375–384, Sep. 2016.
[11] R. Gupta and U. P. Rao, "Achieving location privacy through CAST in location based services," *J. Commun. Netw.*, vol. 19, no. 3, pp. 239–249, 2017.
[12] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Trans. Mobile Comput.*, vol. 7, no. 1, pp. 1–18, Jan. 2008.
[13] B. Ying and D. Makrakis, "Protecting location privacy with clustering anonymization in vehicular networks," in *Proc. IEEE INFOCOM WKSHPS*, Apr./May 2014, pp. 305–310.
[14] X. Liu, H. Zhao, M. Pan, H. Yue, X. Li, and Y. Fang, "Traffic-aware multiple mix zone placement for protecting location privacy," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 972–980.
[15] L. Wu, X. Du, and J. Wu, "Effective defense schemes for phishing attacks on mobile computing platforms," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6678–6691, Aug. 2016.
[16] S. Gang, S. Liangjun, L. Dan, Y. Hongfang, and C. Victor, "Towards privacy preservation for 'check-in' services in location-based social networks," *Inf. Sci.*, vol. 481, pp. 616–634, May 2019.
[17] X. Zhu, H. Chi, S. Jiang, X. Lei, and H. Li, "Using dynamic pseudo-IDs to protect privacy in location-based services," in *Proc. IEEE ICC*, Jun. 2014, pp. 2307–2312.
[18] J. Shao, R. Lu, and X. Lin, "FINE: A fine-grained privacy-preserving location-based service framework for mobile devices," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 244–252.

[19] X. Zhao, H. Gao, L. Li, H. Liu, and G. Xue, "An efficient privacy preserving location based service system," in *Proc. IEEE GLOBECOM*, Dec. 2014, pp. 576–581.
[20] G. Sun, V. Chang, M. Ramachandran, Z. Sun, G. Li, H. Yu, and D. Liao, "Efficient location privacy algorithm for Internet of Things (IoT) services and applications," *J. Netw. Comput. Appl.*, vol. 89, pp. 3–13, Jul. 2017.
[21] H. Lu, C. S. Jensen, and M. L. Yiu, "Pad: Privacy-area aware, dummy-based location privacy in mobile services," in *Proc. ACM MobiDE*, 2008, pp. 16–23.
[22] X. Liu, K. Liu, L. Guo, X. Li, and Y. Fang, "A game-theoretic approach for achieving k-anonymity in location based services," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2985–2993.
[23] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 754–762.
[24] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Proc. Int. Workshop Privacy Enhancing Technol.*, 2002, pp. 41–53.
[25] B. Niu, Z. Zhang, X. Li, and H. Li, "Privacy-area aware dummy generation algorithms for location-based services," in *Proc. IEEE ICC*, Jun. 2014, pp. 957–962.
[26] Y. M. Sun, M. Chen, L. Hu, Y. F. Qian, and M. M. Hassan, "ASA: Against statistical attacks for privacy-aware users in location based service," *Future Gener. Comput. Syst.*, vol. 70, no. 2017, pp. 48–58, May 2017.
[27] F. Li, S. Wan, B. Niu, H. Li, and Y. He, "Time obfuscation-based privacy-preserving scheme for location-based services," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Apr. 2016, pp. 465–470.
[28] I. Ullah and M. A. Shah, "A novel model for preserving location privacy in Internet of Things," in *Proc. 22nd Int. Conf. Automat. Comput. (ICAC)*, Sep. 2016, pp. 542–547.
[29] C. A. Ardagna, M. Cremonini, S. D. C. D. Vimercati, and P. Samarati, "An obfuscation-based approach for protecting location privacy," *IEEE Trans. Dependable Secure Comput.*, vol. 8, no. 1, pp. 13–27, Jan. 2011.
[30] R. Shetty, G. Grispos, and K.-K. R. Choo, "Are you dating danger? an interdisciplinary approach to evaluating the (in) security of Android dating apps," *IEEE Trans. Sustain. Comput.*, to be published.
[31] M. Li, S. Salinas, A. Thapa, and P. Li, "N-CD: A geometric approach to preserving location privacy in location-based services," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 3012–3020.
[32] V. Chang, "A cybernetics social cloud," *J. Syst. Softw.*, vol. 124, pp. 195–211, Feb. 2017.
[33] V. Chang and M. Ramachandran, "Towards achieving data security with the cloud computing adoption framework," *IEEE Trans. Services Comput.*, vol. 9, no. 1, pp. 138–151, Jan. 2016.

GANG SUN is currently an Associate Professor with the University of Electronic Science and Technology of China. He has coauthored 80 technical publications, including papers in refereed journals and conferences, invited papers, and presentations and book chapters. His research interests include network virtualization, cloud computing, parallel and distributed systems, ubiquitous/pervasive computing, and intelligence and cybersecurity.

SHUAI CAI is currently pursuing the master's degree in communication and information system with the University of Electronic Science and Technology of China. His research interests include privacy preservation and cybersecurity.



HONGFANG YU received the B.S. degree in electrical engineering from Xidian University, in 1996, and the M.S. and Ph.D. degrees in communication and information engineering from the University of Electronic Science and Technology of China, in 1999 and 2006, respectively. Her research interests include network survivability, network security, and the next-generation Internet.

SABITA MAHARJAN received the Ph.D. degree in networks and distributed systems from the University of Oslo, Norway, and the Simula Research Laboratory, Norway, in 2013. She is currently a Senior Research Scientist with the Simula Metropolitan Center for Digital Engineering, Norway, and also an Associate Professor with the University of Oslo. Her current research interests include wireless networks, network security and resilience, smart grid communications, the Internet of Things, machine-to-machine communications, software-defined wireless networking, the Internet of Vehicles, and artificial intelligence for next-generation networks.

VICTOR CHANG is currently an Associate Professor with Xi'an Jiaotong-Liverpool University. He is a leading Expert in big data/cloud/security, a Visiting Scholar/Ph.D. Examiner at several universities, an Editor-in-Chief of IJOCI and OJBD journals, an Editor of FGCS, the Founding Chair of two international workshops, and the Founding Conference Chair of IoTBD 2016 www.iotbd.org and COMPLEXIS 2016 www.complexis.org. He has given 14 keynotes in international conferences and received numerous awards, since 2011, including the 2016 European Award: Best Project in Research, the Outstanding Young Scientist 2017, the INSTICC Service Award, and the ICGI 2017 Special Award.

XIAOJIANG DU received the B.S. and M.S. degrees from Tsinghua University, Beijing, China, in 1996 and 1998, respectively, and the M.S. and Ph.D. degrees from the University of Maryland College Park, in 2002 and 2003, respectively, all in electrical engineering. He is a currently a Professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, USA. His research interests include security, wireless networks, and systems. He has authored over 250 journal and conference papers in these areas, and a book published by Springer. He has been awarded more than 5 million U.S. dollars research grants from the U.S. National Science Foundation (NSF), the Army Research Office, the Air Force, the NASA, the State of Pennsylvania, and Amazon. He is a Life Member of ACM. He received the Best Paper Award at the IEEE GLOBECOM 2014 and the Best Poster Runner-Up Award at the ACM MobiHoc 2014. He served as the Lead Chair for the Communication and Information Security Symposium of the IEEE International Communication Conference (ICC) 2015 and a Co-Chair for the Mobile and Wireless Networks Track of IEEE the Wireless Communications and Networking Conference (WCNC) 2015.

MOHSEN GUIZANI (S'85–M'89–SM'99–F'09) received the B.S. (Hons.) and M.S. degrees in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He served in different academic and administrative positions at the University of Idaho, Western Michigan University, the University of West Florida, the University of Missouri–Kansas City, the University of Colorado at Boulder, and Syracuse University. He is currently a Professor with the Computer Science and Engineering Department, Qatar University, Qatar. He has authored nine books and more than 500 publications in refereed journals and conferences. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He is a Senior Member of ACM. He also served as a member, the Chair, and the General Chair for a number of international conferences. Throughout his career, he received three teaching awards and four research awards. He also received the 2017 IEEE Communications Society WTC Recognition Award and the 2018 AdHoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and ad hoc sensor networks. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer. He guest edited a number of special issues in the IEEE journals and magazines. He is currently the Editor-in-Chief of the *IEEE Network Magazine*, serves on the editorial boards of several international technical journals, and is the Founder and an Editor-in-Chief of the *Wireless Communications and Mobile Computing* journal (Wiley).