L2P2: A Location-Label based Approach for Privacy Preserving in LBS

Gang Sun[1, 2], Dan Liao[1], Hui Li[1], Hongfang Yu[1, 2], Victor Chang[3]
[1]Key Lab of Optical Fiber Sensing and Communications (Ministry of Education), University of Electronic Science and Technology of China, Chengdu, China
[2]Center for Cyber Security, University of Electronic Science and Technology of China, Chengdu, China
[3]Xi'an Jiaotong Liverpool University, Suzhou, China

Abstract: The developments in positioning and mobile communication technology have made the location-based service (LBS) applications more and more popular. For privacy reasons and due to lack of trust in the LBS providers, *k*-anonymity and *l*-diversity techniques have been widely used to preserve privacy of users in distributed LBS architectures in Internet of Things (IoT). However, in reality, there are scenarios where the locations of users are identical or similar/near each other in IoT. In such scenarios the *k* locations selected by *k*-anonymity technique are the same and location privacy can be easily compromised or leaked. To address the issue of privacy preservation, in this paper, we introduce the *location labels* to distinguish locations of mobile users to sensitive and ordinary locations. We design a location-label based (LLB) algorithm for protecting location privacy of users while minimizing the response time for LBS requests. We also evaluate the performance and validate the correctness of the proposed algorithm through extensive simulations.

Key words: Location-based service (LBS); *K*-anonymity; Location privacy; Location-label; Sensitive location

1. INTRODUCTION

Internet of Things (IoT) has become popular and pervasive in our day-to-day life. Since more devices and people can be connected to each other, substantial development can lead to the emerging smart cities and big data applications. With an increasing adoption in IoT, privacy preservation has become a major challenge [1, 2], since locations and actions of each user in IoT services can be tracked and even monitored. Due to the developments of mobile communication and positioning technologies, applications of location-based services (LBS) [3, 4] have been expanded rapidly and more people make use of these services. As we know, LBS application system in IoT has been involved in various fields, such as transportation, medical treatment, travel, social networking, entertainment, etc. Furthermore, the mobile communication technology is developing at a very high speed. For example, when the 2G era has quietly left us and the 3G networks have not yet fully popularized the mass, the new and fast 4G network has entered the lives of most people. And the life of people totally depends on the rapid development of the Internet. In the environment of a wireless communication network (e.g. WiFi, 3G, 4G), the users can easily request the LBS services with handheld terminals (e.g. Tablet or Smart Phone) [5].

After receiving a LBS request, the LBS provider (LP) responds to the request according to the user location information and the requested content. For example, a user submits the request "where is the nearest supermarket". Then the LP returned the address of the nearest supermarket and other relevant information to the user. The typical LBS system model [6] is shown in Figure 1.

Although users enjoy the conveniences of the services provided by the LBS providers in IoT, there is a potential security risk of losing their privacy [7, 8]. For example, the privacy of location or trajectory may be leaked to other parties [9-11]. Then they are vulnerable to be exploited by the malicious attacker, so as to damage the vital interests of the users. For example, if the malicious attackers have known users' location privacy and other privacies, they can easily get to know more comprehensive information with some analysis. Then they can defraud the property of the users through the Internet or telecom fraud. And furthermore people increasingly focuses on their own privacy security problems. Therefore, the problem of privacy protection in LBS in Internet of things needs to be solved.

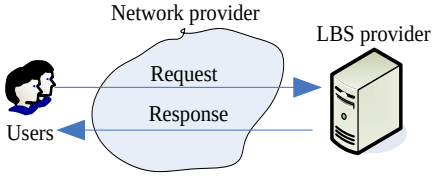


Fig. 1: The typical LBS system

There are many solutions (e.g. encryption [12], Anonymity [13]) have been proposed to protect users' personal privacy. In the existing research, the authors studied the problem of protecting users' location privacy under the scenario of single LBS query and the users located at different locations. The *k*-anonymity [14] and the pseudo-ID technique [15] are effective techniques to protect user location privacy in single LBS query. The authors in [16-18] also provided solutions to solve the problem of privacy preservation by using *k*-anonymity. In this way, before sending a query to the LP, the user merges other *k*-1 user queries and then submits the mixed query to the LP. However, the LP can easily get the requested contents of users when the requested contents of the *k* users are similar to each other. Using data analysis and data mining, the LP can infer more information about users, such as common interests and hobbies. To combat this deficiency, researchers introduced the concept of *l*-diversity [19] to protect the requested contents or preference privacy

[20]. In these method, all LBS queries can be classified into different categories (e.g., medical, traffic, entertainment, etc.) according to the requested contents. The privacy preserving framework for local-area mobile social networks (PLAM) [20] adopts *k*-anonymity and *l*-diversity to protect location and preference privacy of users. As shown in Figure 2(a), there exists 6 users (*i.e., k*=6) who are distributed in different locations requesting 3 services (*i.e., l*=3). Then the LBS provider cannot link a specific service/location to a user. Thus, the PLAM method can protect the location and preference privacy when the users' locations are different. However, consider the scenario in Figure 2(b) where the *k* users have the same location and send requests together to the LP. Although the PLAM can protect the preference privacy of users with *l*-diversity technique, the LP can know that the *k* users are in the same location and the location privacy is leaked. Therefore, PLAM cannot protect location privacy when the users have the same locations, especially in some locations such as supermarket, school and hospital where the probability of selecting the same location with *k*-anonymity technology is very high.

Furthermore, in real applications, users may send requests continuously for a period of time and the users' locations may be nearby with each other or even identical. Because of the correlation of various positions in continuous queries, it is more difficult to protect users' privacies, especially the trajectory privacy. Thus, protecting the users' privacies (e.g. location privacy, trajectory privacy, preference privacy) while the users' locations are the same both in continuous and single request is an important issue left to address.

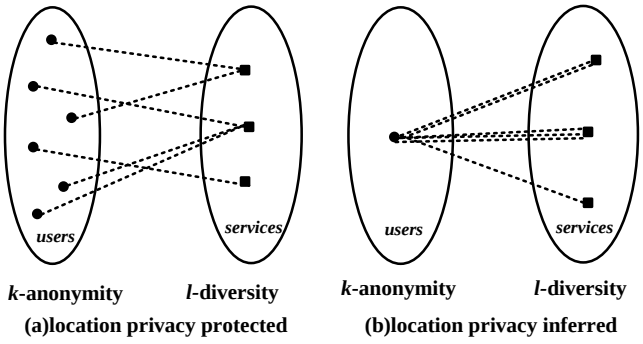


Fig. 2: Same vs. different *k* locations

In this paper, we study the problem of privacy protection for users within the same locations both in single request and continuous request. We introduce the *location label* into our proposed algorithm for efficiently protecting the location privacy, preference privacy and trajectory privacy of users. The *location label* based algorithm is not only suitable for single request, but also applicable for continuous request. The main contributions of this paper are as follows.

- We introduce the *location label* to classify all locations into sensitive locations and ordinary locations. Due to the dense population at sensitive locations, the locations selected by *k*-anonymity are much more identical than that of the scenario of ordinary locations.
- Considering the locations of *k* users are nearby with each other or identical, we propose a *location label* based (LLB) algorithm for privacy preservation under the scenario where the locations of *k* users are nearby, similar or identical. For a single request, our proposed algorithm can protect the location privacy and preference privacy of users; and for a continuous request, it can be used to protect the trajectory privacy of users.
- We propose three protocols including the *request aggregation protocol*, the *pseudo-ID exchange protocol* and the *improved PLAM protocol* in our proposed algorithm, which help in reducing the response time of the LBS system.
- We evaluate the performance of our proposed LLB algorithm by conducting extensive simulations.

The remainder of this paper is organized as follows. Section 2 discusses the related work about the privacy preserving in the LBS system. Section 3 describes the basic concepts and definition used in our approach. Section 4 presents the motivation and system model. Section 5 introduces our *location-label* based framework and gives the detailed description of the LLB algorithm. The simulation results are given in Section 6. Section 7 concludes this paper.

2. RELATED WORK

There are several studies on location privacy preservation, which focus on the possibility of losing location privacy during the location process. These location techniques in a LBS system are able to derive users' locations through anchor points [21]. Since location algorithm takes anchor points as input and outputs users' location, then the locations of anchors and users may be leaked to others. Thus in order to efficiently protect user location information during location process, the authors in [22] proposed the PriWFL algorithm, and the authors in [23] studied the problem of multi-lateral privacy preservation.

There are other studies that focus on protecting user location privacy in LBS applications. In these studies, the users' locations are calculated by local facilities, and two kinds of requests are considered: single request and continuous requests. For single request, the location privacy and preference privacy are two key contents that need to be protected. Several strategies such as *k*-anonymity [31], Mix Zones [24], *l*-diversity [19], *m*-unobservability [25] etc. have been proposed to prevent the LP from inferring the users' locations or preference privacies. Where the *k*-anonymity technology is the most commonly used and preferred techniques. It is often applied in the distributed structure. From the view of temporal and spatial, there are three types of implementations for *k*-anonymity in single request.

1) User submits a request which contains *k* locations: one true location and *k*-1 false locations. It is often applied in the centralized architecture via a trusted location anonymizer.

2) A cloaking region, stands for a vague location, contains *k* users: the real user and *k*-1 pseudo users. The *k* users

submit request to the LP together with the same vague location. This kind of *k*-anonymity is restricted by space.

3) There is a one-to-one mapping between user and his location. It is often applied for the distributed structure [26]. The *k* users joint together, and then select a representative who sends their request packets to the LP. This kind of *k*-anonymity needs extra time consumption for gathering the *k* appropriate users.

However, since the requested contents of the *k* users are similar with each other, the LP can easily infer users' preference privacies, such as common interests and hobbies of the users. To remedy this deficiency, the authors introduced the *l*-diversity [19] to protect the requested contents. The basic idea of the *l*-diversity technique is to make LBS queries of users different. Therefore, this property can ensure that there exist at least *l* services in the *k* LBS queries, where $k \geq l$. The authors in [16] proposed the DLS algorithm which takes advantage of *k*-anonymity and *l*-diversity properties for protecting location privacy and preference privacy.

When a user sends continuous requests (i.e. send requests continuously for a period of time) to the LP, the trajectory information of the user needs to be protected. Feng et al. [27] proposed an algorithm called VAvatar to protect users' locations and trajectories. Mohammed et al. [28] proposed a *Track False Data* method for the problem of protecting the privacy of continuous requests, in which users send their fake locations and track information to the LBS provider, rather than their real trajectory data. The authors in [29] provided a distributed query privacy preserving solution to protect user's trajectory privacy.

The existing studies mentioned above focus on addressing the problem of privacy preservation under the assumption that users are distributed in different locations. However, in real applications, multiple users may have the same location. In this work, we investigate the problem of location privacy preservation for users in the same location.

3. PRELIMINARIES

In this section, we give the basic concepts and definitions.

1) *Sensitive and Ordinary locations*: All locations can be statically classified into two categories: *sensitive locations* and *ordinary locations*. Sensitive locations (e.g., hospital, school or supermarket) have dense population and the ordinary locations (such as the locations on general roads) have sparse population.

Usually, there are some commonalities between sensitive locations: *i*) the sensitive location is usually in a region with heavy traffic; *ii*) they are located in an area with dense population; *iii*) the users gathered at a sensitive location have common characteristics. For example, if the users are in a hospital (a sensitive location), the possibility of that they are patients or doctors is very high. The users at a sensitive location have common characteristics (e.g., interests or needs), thus the request contents of these users may be similar. Thus we need to protect the identities of users in these locations, and do not need to pay much attention on protecting their requested contents and the preference privacy. For example, users who in hospital do not care whether the attacker knows the requested information (e.g., health information), but they expect that their identities have not been inferred by the malicious attackers. For an ordinary location, since it is just a location on general road, users expect that the location privacies and preference privacies have not been leaked. Figure 3 gives an example of location partition. The sensitive locations and ordinary locations are randomly distributed in the area shown in Figure 3. The sensitive locations including supermarket, hospital, bank, etc. Furthermore, one kind of label may not indicate only one location. For example, there are two locations with labels of Hospital A and Hospital B for labeling hospitals.

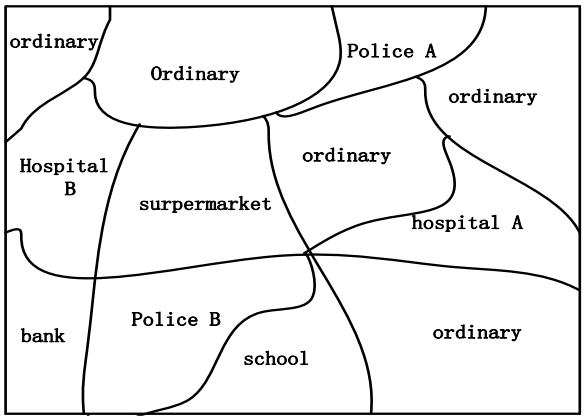


Fig. 3: An example of location partition

2) *User location:* A user location denoted as *d* (*x*, *y*, *label*), where x and y represent the latitude and longitude of a location, and $label$ represents the category of the location. According to the characteristics (e.g., population density) of actual geographical environment, we divide a large area A into a set of small and irregular cells represented as $\{A_1, A_2, \ldots . A_i \ldots\}$. The geometric center of each irregular cell is regarded as its location information. For example, when a user is in the cell A_i, we can calculate the coordinate of the geometric center of area A_i. Its latitude and longitude are used to represent the user's location information of *x* and *y*, respectively.

3) *Service category*: We classify the users' service requests into different service categories, according to the services provided by the LBS system. For example, some users query for entertainment information, while others may query for dining or dating. We use the set $Serve = \{sc_1, sc_2, \ldots, sc_i, \ldots, sc_m\}$ to denote the various service categories.

4) *Single request packet*: a single request packet is denoted as $Rq_i = \{Pid_i, d_i, serve_i, R_i, t\}$, where Pid_i represents the user's identity, d_i denotes the user's location information, $serve_i$ represents the service category, and R_i represents specific content of the corresponding request. In order to implement *k*-anonymity and protect user's privacy, the user does not directly send a request to the LP. The time *t* is used

to indicate the upper bound of response time (i.e., maximum tolerable response time) for the LBS request of a user.

5) *The aggregated package:* Before a user u_i submitting a request to the LP, the user aggregates requests from other users. User u_i first broadcasts the aggregating message to other users. If there are other k-1 users agreeing to aggregate with u_i, they send their requests to u_i, and then u_i becomes the *representative user* for them. The representative user gathers k users' request packets and forms a new packet, denoted by $Ag=\{P_{list}, \{d_1, serve_1, R_1\}, \{d_2, serve_2, R_2\},\ldots,\{d_k, serve_k, R_k\}\}$. Where P_{list} is a list of identities of the k users and $P_{list}=\{Pid_1, Pid_2,\ldots, Pid_k\}$. And d_i, $serve_i$ and R_i denote the location information, service category and requested content of user i, respectively. Due to the randomness and uncertainty of users, the k locations and the requested services corresponding to the k users may be the same. Thus we have:

$$1 \le |d| \le k, \text{ and}$$

$$l \le |s| \le k .$$

Where d is the set of locations of k users, $|d|$ is the number of different locations, s is the set of the requested service categories of k users and $|s|$ is the number of different requested service categories.

6) *Bilinear pairings* [30]: we use G and G_T to denote the cyclic additive and multiplicative groups, both generated based on the same prime order q. Assume that p is the generator of group G, Z_q is the residual class ring with modulo q, and Z_q^* is an invertible element set relative to Z_q. There exists a mapping e: $G \times G \rightarrow G_T$ that satisfies the following conditions:

a) *Bilinear:* For any two elements $g_1, g_2 \in G$, where $a, b \in Z_q^*$, we have $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab} \in G_T$.

b) *Non-degenerated:* There exists a $P \in G$ such that $e(P, P) \neq \rho$, where ρ is the unit-element of group G.

c) *Computable:* For any two elements $g_1, g_2 \in G$, we can compute $e(g_1, g_2)$ via an efficient computational technique.

We call the mapping e that satisfies the three conditions mentioned above as bilinear pairings. By applying a bilinear mapping on the supersingular elliptic curve, we can obtain a Diffie-Hellman group. Assume that the Diffie-Hellman group is G. The Computational Diffie Hellman (CDH) problem is hard, the Decisional Diffie Hellman problem (DDH) can be easily solved. Based on the characteristic of the bilinear pairings, we can calculate a user's PID and verify whether the PID is valid [20].

4. MOTIVATION AND SYSTEM MODEL

In this section, we give the detailed descriptions on the motivation, the researched problem and the system model designed for the studied problem.

4.1 Motivation

The PLAM framework [20] adopts distributed structure combined with the k-anonymity and l-diversity to protect users' location privacy and preference privacy. However, in this work, we consider another scenario for this kind of k-anonymity in which the k users send requests together to LP in the single query application, where the locations of users are very close or even identical. Obviously, it can't protect users' location privacy even if l-diversity makes the k users' requested content various and protects users' preference privacy. Especially in some places with dense population (e.g., supermarket, school or hospital), the probability of selecting the same location with k-anonymity technology is very high. As far as we know, all of the existing research with k-anonymity have not considered about this scenario.

Furthermore, users more than just send single request to the LP and may send requests continuously for a period of time. Because of the correlation of various positions in the continuous requests, only k-anonymity technique for privacy preserving is insufficient. It cannot guarantee that the users' trajectories are not exposed, thus leaking users' trajectory privacies. Therefore, malicious attacker can easily know the paths or places the users traversed, and deduce their earnings, social class, and preferences and so on. Thus, we should take into trajectory privacy consideration in the scene of continuous queries.

4.2 System Model

Given the *location labels*, a single request packet and bilinear pairings, the problem is that how to protect users' location privacies and reduce the gathering time for k users in a distributed structured LBS system.

For preserving users' privacies, we design a LBS system, whose framework is shown in Figure 4. The LBS system consists of three key components: User Requests (USER), Pseudonym Identity Server (PIDS), LBS Provider (LP). In this work, we adopt a distributed structure without involving a trusted anonymizer. The LP operates in accordance with relevant regulations and agreements in LBS system. But it does not rule out that the LP has curious and hope to deduce the users' location, preference and trajectory privacies. Thus, the LP is honest-but-curious in our system model. Within the scope of communication, the users can communicate with each other. But they must follow the rules of corresponding agreements. So they neither collude with each other to infer other users' privacy information, nor collude with the LP.

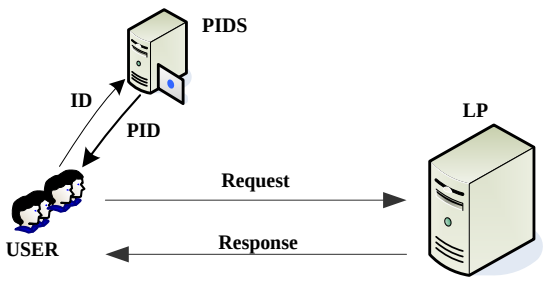


Fig. 4: System model for privacy preserving

From the Figure 4, we can see that the running of location-label framework can be divided into three main

stages: *PIDS server initialization*, *USER registration* and *Request submission*.

1) *PIDS server initialization*: For a given parameter k, the PIDS server generates a 5-tuple (q, g, G, G_T, e) about bilinear pairings, where q is a k-bit prime number. Then the PIDS server initializes the LBS system with a suitable symmetric encryption algorithm *enc*(), pseudo-random function f:$\{0,1\}^* \rightarrow Z_q^*$ and two hash functions H_1: $\{0,1\}^* \rightarrow \{0,1\}^k$ and H_2: $\{0,1\}^* \rightarrow$G. In this work, we assume that the PIDS server has held a public key and a private key (i.e., Pk_{pids} and Sk_{pids}). Finally, the PIDS server generates and publishes the system parameters (q, g, G, G_T, e, Pk_{pids}, f, H_1, H_2, *enc*()).

2) *USER registration*: User u_i registers with the PIDS server by sending registration messages. After receiving the registration message, the PIDS server computes $s=f(PK_{pids})$ and user's pseudo-ID by using the value s and the symmetric encryption algorithm *enc*(). The pseudo-ID is represented as Pid_i: $Pid_i=enc_s(u_i \| r_i)$, where $r_i \in Z_q$. Then the PIDS server calculates the corresponding private key for user u_i: $Sk_i=H_1(Pid_i)$. Finally, the PIDS server returns the Pid_i and Sk_i to the user u_i. After receiving Pid_i and Sk_i, the user u_i can verify whether they are correct by checking $e(H_1(Pid_i), Pk_{pids}) =? e(Sk_i, g)$. If they are equal, the Pid_i and Sk_i are valid. Otherwise, they are invalid and the user will register with the PIDS server again. Since the identity information of a user may change, we need to check the validity of the information after receiving identity information (e.g., *Pid*) every time, including in the initial registration process.

3) *Request submission*: If a user u_i submits a request to the LBS provider, the LBS system has to employ the LLB for protecting the user's privacy. After gathering the requests of users with k-anonymity and l-diversity properties, the user u_i becomes the *representative user*, and he/she repackages the k users' request packets and gets an aggregated packet *Ag*. Then the *representative user* sends the aggregated packet to the LBS provider. After receiving the aggregated packet, the LP processes it and returns a list of results to the k users. Each user filters the results and selects the one that is consistent with his/her own request from the list.

5. ALGORITHM DESIGN

In this section, we first propose three protocols including *request aggregation protocol*, *pseudo-ID exchange protocol* and the *improved Privacy-preserving framework for Local-Area Mobile social networks* (PLAM) *protocol*. We then design the *location label* based (LLB) algorithm.

5.1 The algorithm framework

According to the given security parameter k and l, the PIDS (pseudonym identity server) bootstraps the LBS system and initializes it. If user u_i wants to be served by the LBS system, he need to register himself to the LBS system. For sending request to the LP, he will firstly unite with other k-1 users by using the *request aggregation protocol*. If the k users have the same location labels, the user u_i need to make decision on whether they are sensitive locations. If they are sensitive locations, the LLB algorithm calls the *pseudo-ID* exchanging *protocol*; if they are ordinary locations, the LLB calls the *improved PLAM protocol*. Whereas the k users have different location labels, it directly calls the *improved PLAM protocol*. The framework of our proposed LLB algorithm is shown in Figure 5.

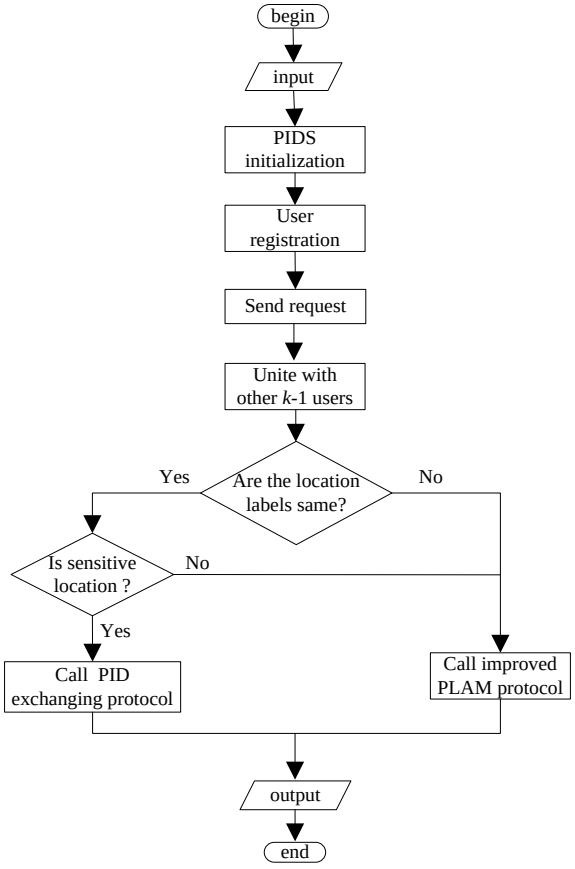


Fig. 5: The framework of LLB algorithm

5.2 Request aggregation protocol

Without loss of generality, we assume that there is a user u_a who has not received any other queries from other users and wants to launch a request to the LP. Then the user u_a will initiate request aggregating message to gather other k-1 users' requests. The *Algorithm 1* describes the pseudo code of the request aggregation protocol. The detailed requests gathering process is as follows.

User u_a first broadcasts the request aggregating message. We assume that user u_b has received the broadcast message. There are three scenarios where user u_b will ignore the broadcast message sent by user u_a: (*i*) user u_b has agreed to aggregate with other user, (*ii*) the time t is zero in the request packet of user u_b, (*iii*) the user u_b has sent aggregate request to other users and there are more than k/2 users who agree to join with user u_b.

If user u_b has neither sent aggregate request to other users, nor has agreed to aggregate with other users, and the time t is not zero, the user u_b is an ideal candidate for user u_a. If less

than $k/2$ users agree to join with user u_b (assume m ($m \leq k/2$) users have joined with user u_b, and then the user u_b is *agent user* for the m users), the agent user u_b will respond to user u_a that "m+1 users (including user u_b and other m users who have joined with user u_b) agree to join".

***Algorithm* 1: Requests aggregation**

Input: Request aggregating message, all users, k
Output: *Ag*, *representative user*

```
1: Broadcast the request aggregating message (the sender);
2: Receive the broadcast message (the receiver);
3: if (The receiver has aggregated with others || t=0 || number
     (has joined with the receiver) > k/2)
4:     Ignore the broadcast message;
5:  else
6:     Agree to aggregate with the sender;
7: end if
8: if (number (users aggregate with the sender) = k)
9:     Generate the aggregated package Ag;
10:    if (the package Ag meets the requirements)
11:        The sender becomes the representative user;
12:    end if
13: else
14:    The aggregation is unsuccessful;
15: end if
```

When user u_a has gathered k-1 users, u_a informs the corresponding k-1 users and collects their request packets. Then user u_a repackages the packets from k users and get an aggregated package *Ag*. If the aggregated package *Ag* meets the requirements (e.g. the l-diversity requirement), user u_a becomes the *representative user* who sends the aggregated packet *Ag* to LP. Otherwise, the aggregated package *Ag* will be discarded and user u_a informs the other k-1 users that their aggregation is failed. Then all the k users reset the time t and resubmit their requests.

Then we assume that user u_a has been the representative user for other k-1 users. Firstly, user u_a checks the location labels of the k users. When the k labels are different, there are three possible cases: *i*) part of the k locations are common locations, the other part are sensitive locations; *ii*) all of the k locations are sensitive locations, but the types of the k sensitive locations are different; *iii*) the k locations are sensitive locations, and their types are all the same. However, the contents of *labels* are different, i.e., the latitude and longitude may be different. When the k labels are the same, there are only two possible cases: *i*) the content of each *label* is none, so they are all ordinary locations; *ii*) they are all sensitive locations.

Theorem 1: Appropriate value for parameter k can reduce the time complexity of request aggregation protocol.

***Proof*:** As shown in the request aggregation protocol, we can see that the agent user can agree to joint for m+1 users (including himself). Assume that the average time for one user takes to respond the request user u_a is τ. Therefore, the time user u_a consumed to gather other k-1 users is $\tau \times (k\text{-}1)$. When there is an agent user, the time only need to be consumed by user u_a is $\tau \times (k\text{-}m\text{-}1)$. Therefore, lowering the value of parameter k can reduce the time complexity.

5.3 Pseudo-ID exchanging protocol

We propose the pseudo-ID exchanging protocol which can efficiently protect users' location privacies in single query when the labels of users are the same and all of the users locate at sensitive locations. This proposed protocol can also be used to protect users' trajectory privacies in continuous queries. Since the users are at sensitive location, and they do care that their identities have not been leaked rather than their request contents. Thus, the l-diversity technique is no longer need to be adopted. At this moment, the requested contents preservation has been unnecessary. We just need to protect the identities of users from the attacker's recognition through exchanging their PIDs. Figure 6 shows an example in which three users u_a, u_b and u_c come from different roads and gather at a sensitive location.

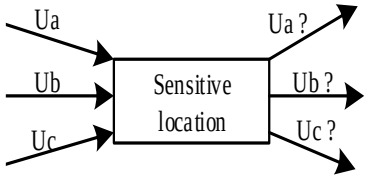


Fig. 6: An example for exchanging PIDs

Next we introduce the process of exchanging identity information between two users. Assume that two users u_a and u_b have the same sensitive location, and then they exchange PIDs with the probability of ρ. Due to the symmetry, here we only describe how user u_a changes his identity information. After receiving the Pid_b and private key Sk_b from the user u_b, user u_a re-verifies their identity by checking $e(H_2(Pid_b), PK_{pids}) =? e(Sk_b, g)$. If they are equal, user u_a will modify his/her identity as follows: Firstly, u_a abandons his/her own identity information (i.e., Pid_a and Sk_a); Then the u_a regards Pid_b as his/her own identity (u_b does the same as u_a); Finally, u_a and u_b successfully exchanging identities. *Algorithm 2* describes the pseudo code of the pseudo-ID exchanging protocol.

***Algorithm* 2: Pseudo-ID exchanging**

Input: Ag, ρ, system parameters (q, g, G, G_T, e, Pk_{pids}, f, H_1, H_2, enc())
Output: "new" identity for each user

```
1: if (the locations in Ag are identical && sensitive locations)
2:     Exchange identities with probability ρ;
3:     Receive Pid_i and Sk_i for each user
4:     if (e(H2(Pid_i), PK_pids) = e(Sk_i, g));
5:         Replace user's original identity by Pid_i;
6:     else
7:         The pseudo-ID exchanging is unsuccessful;
8:     end if
9:  Modify the Ag and send it to the LP;
10: end if
```

Theorem 2: For the users have same location, the pseudo-ID exchanging protocol can be used to protect users' location privacies, as well protect users' trajectory privacy.

Proof: When a representative user sends a LBS request at sensitive location, all of the gathered *k* users must at the same location (form the view of the characteristics of sensitive location). After exchanging the PIDs of the users in the same location, the attacker cannot link an identity to a specific user, and thus unable to distinguish the users. Therefore, it can indirectly protect users' trajectory privacy through protecting users' identities. As shown in Figure 6, we can see that PID exchanging can confuse the attacker who cannot infer which trajectory belongs to a specific user. Therefore, it can be used to protect the users' trajectory privacies.

5.4 The improved-PLAM protocol

The authors in [16] proposed the PLAM for preserving location and preference privacy in a distributed LBS system. With *k*-anonymity and *l*-diversity techniques, the PLAM protocol employed Privacy-preserving Request Aggregation to unite *k* users only considering the case where the users are in different locations. However, in real applications, their locations may be the same, especially when they are in a sensitive location may cause a location privacy leaking. In this work, we propose the improved PLAM protocol, for the scenario that the users are in same location. *Algorithm 3* describes the pseudo code of the improved PLAM protocol.

***Algorithm* 3: The improved PLAM**

Input: *Ag, k, l, representative user*
Output: "new" identity for each user
1: **if** (the locations in *Ag* are (identical && ordinary locations) || different);
2: Check the services in *Ag*;
3: **if** (the number of s*ervice categories* ≥ *l*)
4: Compare the location information (*x*, *y*) of *k* users;
5: **if** (the number of users have same location > k/2)
6: Call the *pseudo-ID exchanging protocol*;
7: **else**
8: Send the *Ag* to LP;
9: **end if**
10: **else**
11: Discard the *Ag* and the request is unsuccessful;
12: **end if**
13: **end if**

Firstly, the *representative user* compares the information *serve* in the *single packet request* of all the *k* users. If there are at least *l* services of the *k* users, it means the aggregated packet *Ag* meets the requirements of *l*-diversity. Otherwise the aggregated packet *Ag* is discarded and the *representative user* informs the other *k*-1 users that the aggregation is unsuccessful.

After ensuring that there are at least *l* services using the LBS system, the *representative user* will compare the location information (*x*, *y*) of all *k* users. If more than k/2 location information (*x*, *y*) are the same, we exchange the identities of users who have the same location by using the pseudo-ID exchanging protocol. At this time, we no longer care if the location is ordinary or sensitive. Finally, the *representative user* modifies the aggregated packet *Ag* and sends it to LP.

Theorem 3: When more than half of the users have the same location, the proposed improved-PLAM protocol can protect users' location privacies.

***Proof*:** Firstly, the improved PLAM protocol ensures the *l*-diversity, thus the preference privacy must be protected. Next, when most of users have the same location, the PLAM algorithm cannot prevent the attacker from inferring users' location privacy. For example, we consider 6 users with 3 locations: l_a, l_b, l_c. Assume that 3 users are in location l_a, and the rest users are in location l_b and location l_c, respectively. Then an attacker may guess that a user in location l_a with the probability of 1/3 in the PLAM protocol. However, since the proposed improved-PLAM protocol allows users to exchange their PIDs, the attacker only can guesses it with the much lower probability of 1/(3×6). Therefore, the location privacy can be well protected by using the improved-PLAM protocol.

5.5 The location-label based algorithm

Algorithm 4 shows the pseudo code of *location-label* based algorithm (LLB) proposed in this work. User u_i first gathers other *k*-1 users by using the *request aggregation protocol*. If there are *k*-1 users who agree to send request to the LP together with the user u_i, the user u_i becomes the *representative user* for the *k* users. Then the *representative user* compares the *k* users' location labels. There are three kinds of situations based on the results of the comparison: *i*) the *k* users location labels are the same and their locations are sensitive locations, then we use *pseudo-ID exchanging protocol* for the subsequent processing; *ii*) the *k* users' location labels are the same and their locations are ordinary location, then we use the *improved-PLAM protocol* for the subsequent processing; *iii*) the location labels are different, we use the *improved PLAM protocol* for the subsequent processing.

***Algorithm* 4: Location-Label Based (LLB) algorithm**

1: Broadcast the aggregation message;
2: Aggregate users' requests by using request aggregation protocol;
3: **if** (*k* users aggregate together)
4: compare the *k* users' location labels;
5: **if** (the locations are identical and sensitive locations)
6: Call the *pseudo-ID exchanging protocol*;
7: **else if** (the locations are identical and ordinary locations)
8: Call the *improved-PLAM protocol*;
9: **else if** (the locations are different)
10: Call the *improved-PLAM protocol*;
11: **end if**
12: **end if**

6. SIMULATION AND RESULTS

For evaluating the effectiveness of our proposed *location label* based algorithm, we have conducted extensive simulations. In this section, we first describe the simulation environment, and then give the simulation results and analysis.

6.1 Simulation Environment

We use OPNET [32] to conduct our simulations, since it can be used to construct complex network topologies and simulate the message sending/receiving. Assume that there is a region A of size $\{1.5\text{km} \times 1.5\text{km}\}$ with 10×100 locations. For simulating the locations, we construct a full-mesh network consisting of 10×100 nodes and randomly assign these nodes as sensitive or ordinary locations.

***Scenario*-1:** There are 100 users uniformly distributed in region A. We assume that user u_a in an ordinary location sends single request to the LP. The rest of users randomly send their messages about aggregation. The PID exchange probability ρ is fixed at 0.5. To ensure the l-diversity, we set $l = k/2$.

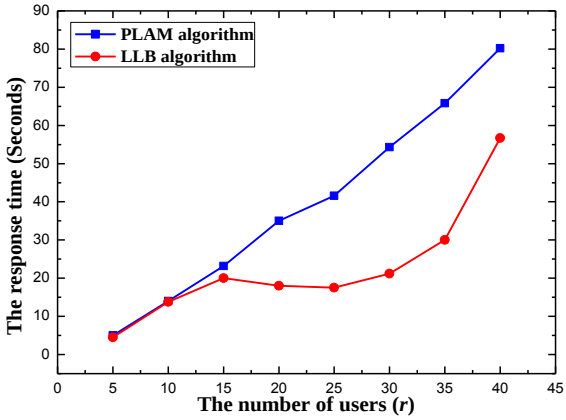
***Scenario*-2:** There are 100 users in region A, and most of them are distributed in sensitive locations, and only a few of them are distributed in ordinary locations. We assume that user u_a in a sensitive location sends their request to the LP. The rest of the conditions are the same as *Scenario-1*.

***Scenario*-3:** There are 100 users in region A. They are randomly distributed in the region A and independently moves with the same velocity $v = 1$ *m/s* in the region A. We assume that user u_a sends continuous requests to the LP in a period of time (e.g., 120seconds). The rest conditions are similar to *Scenario-1*.

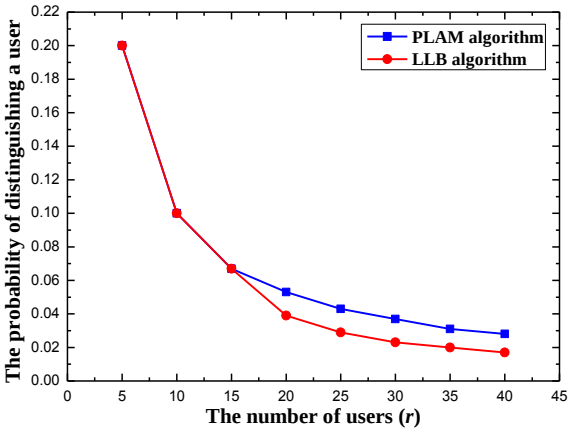
6.2 Simulation Experiment Results

Figure 7 shows the simulation results for two compared algorithms (i.e., PLAM algorithm and LLB algorithm) under *Scenario*-1. From Figure 7(a) we can see that the time for responding request of our proposed LLB algorithm is shorter than that of the PLAM proposed in [20]. Furthermore, for our LLB algorithm, the response time increases with the growth of the value of parameter r (i.e., the number of users) when $r < 20$ and reduces a little when $20 \le r \le 30$, and the response time goes up again when $r > 35$. This is because that it needs to consume more time for gathering more users. Since is an agent user who agrees to aggregate with the user u_a (i.e., the aggregation initiator) when $r \ge 20$, the response time slowly increases with the growth of the value of r in LLB algorithm whereas quickly increases with the growth of the value of r in the PLAM algorithm.

Figure 7(b) presents the relationship between the number of users (i.e., r) and the probability of distinguishing a user. From Figure 7(b), we can see that both LLB algorithm and PLAM algorithm have the same probability when $r \le 15$; and the LLB algorithm can guarantees a lower probability of distinguishing a user by attacker than PLAM algorithm does when $r \ge 20$. Since some of the r users have the same location (larger r means more users have same location), and thus our LLB algorithm can better protect location privacy of user compared to the PLAM algorithm.



(a) The response time



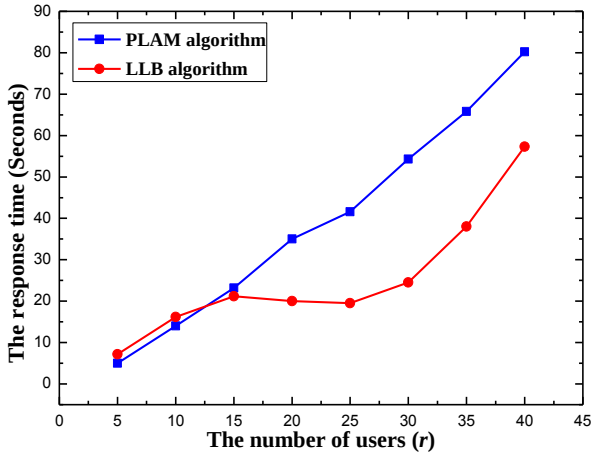
(b) The probability of distinguishing a user

Fig. 7: The simulation results for *Scenario*-1

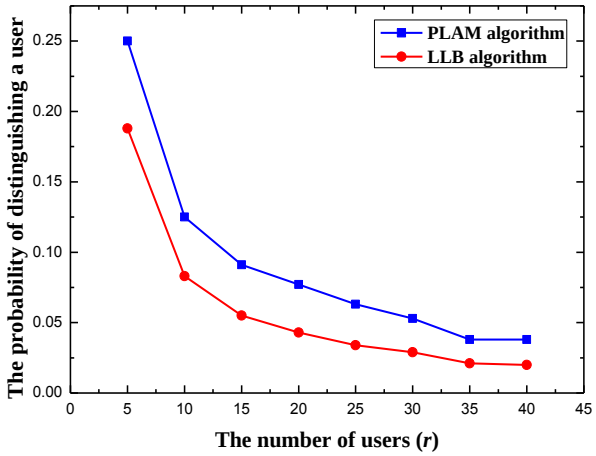
Figure 8 compares the performance of LLB algorithm and PLAM algorithm under *Scenario*-2. Figure 8(a) shows the similar simulation results with Figure 7(a). The response time of our proposed algorithm is significantly shorter than that of the PLAM algorithm when r >15. However, as user u_a (i.e., the aggregation initiator) is in a sensitive location in *Scenario*-2, the locations of k users gathered by user u_a may be identical. Then the proposed LLB algorithm will exchange the users' IDs through *pseudo-ID exchanging protocol* and the process of exchanging IDs is time consuming. Notice that when the number of users is small, e.g. $r \le 10$, the PLAM algorithm has lower time consumption.

Figure 8(b) shows that the LLB algorithm has much lower probability for distinguishing a user compared to the PLAM algorithm. Hence, the LLB algorithm can more efficiently protect user's privacy. Compared to the Figure 7(b), the probability of distinguishing a user with LLB algorithm is lower in *Scenario*-2 than that in *Scenario*-1. This is because

when users have same location, the LLB algorithm employs *pseudo-ID exchanging protocol* to reduce the probability of for distinguishing a user. Therefore, we can see from Figure 8(b) that there are more users have same location in sensitive locations than in ordinary locations.



(a) The response time



(b) The probability of distinguishing a user

Fig. 8: The simulation results for *Scenario*-2

Figure 9 shows the performance of the LLB algorithm when user u_a (i.e., the aggregation initiator) is in ordinary location (*Scenario*-1) and sensitive location (*Scenario*-2). For a user in sensitive location, before sending a request, the user gathers/aggregates with other k-1 users. The k locations of users are very likely identical and the LLB algorithm uses the *pseudo-ID exchanging protocol* for exchanging users' identities. As the process of exchanging identities is time consuming, the request gets a longer delay (i.e., response time) in *Scenario*-2 than that in *Scenario*-1.

Figure 10 shows the probabilities of distinguishing a user in *Scenario*-1 and *Scenario*-2. Form the Figure 10(a), we can see that PLAM algorithm almost has the same performance in *Scenario*-1 and *Scenario*-2. Figure 10(b) shows that LLB algorithm can better protect location privacy in sensitive location than ordinary location, this is benefit from the PID exchanging protocol.

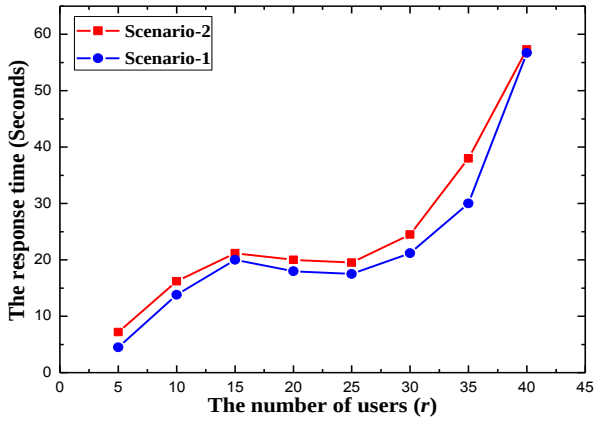
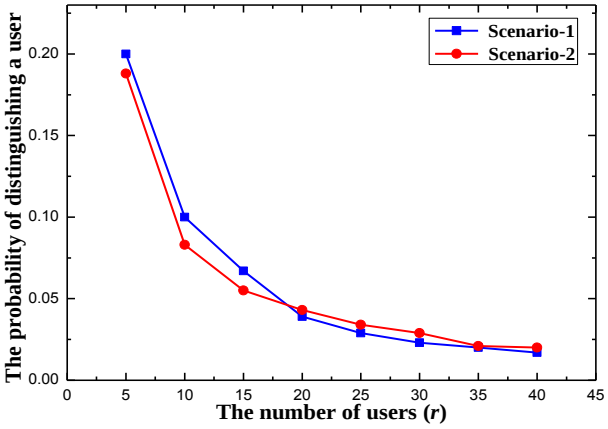
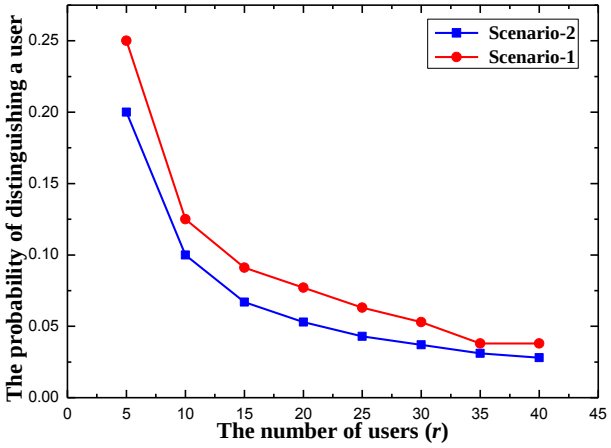


Fig. 9: The response time of LLB algorithm under different scenarios



(a) Simulation results for PLAM algorithm

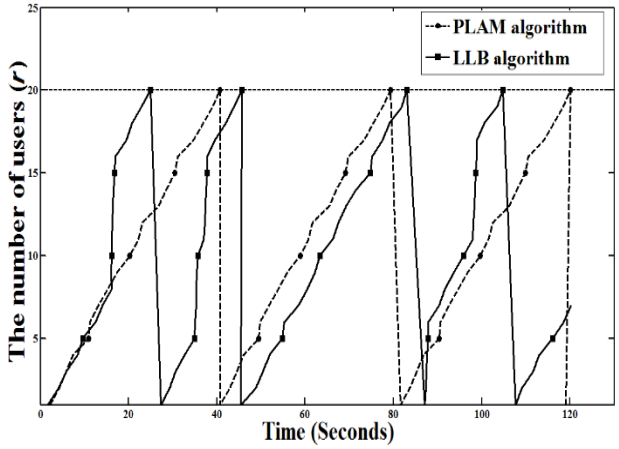


(b) Simulation results for LLB algorithm

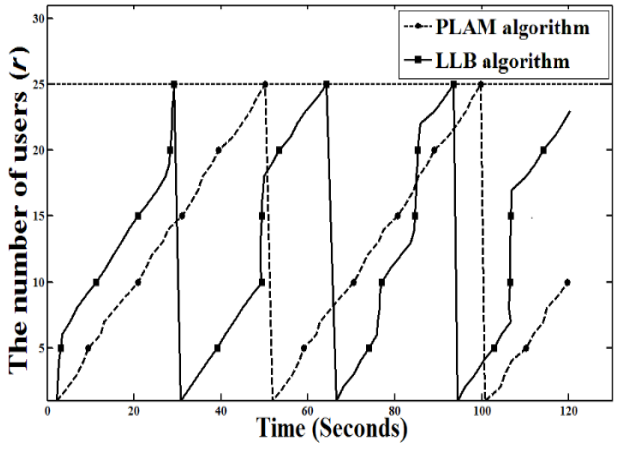
Fig. 10: The simulation results for *Scenario*-1 versus *Scenario*-2

Figure 11 shows the simulation results for continuous requests for two compared algorithms under *Scenario*-3. From Figure 11(a) (where k = 20), we can see that user u_a (i.e., the aggregation initiator) sends 4 requests in 120 seconds by using the LLB algorithm. However, in the PLAM algorithm user u_a only sends 3 requests in the same time. Thus, for aggregating 20 users in each request, the PLAM algorithm consumes more time than LLB algorithm.

Figure 11(b) (where k = 25) shows that user u_a sends 3 requests by using LLB algorithm in the same time (i.e., 120 seconds) whereas only 2 requests can be sent by using PLAM algorithm. Furthermore, the time for gathering users increases with the growth of the value of k, and LLB algorithm has better performance in term of response time for continuous request.



(a) k = 20



(b) k = 25

Fig. 11: The simulation results for *Scenario*-3

6.3 Limitations of current research

Our current focus is to ensure that location privacy simulations can be performed successfully shown in Section 6 and results match to theoretical development presented between Section 3 and 5. Current limitation is the testing of our PLAM algorithms on the road, since there are more uncontrolled factors to investigate, such as the strength of 4G network, number of sky scrapers in the tested city, interference from other services or applications and the user behaviors in different countries. Once we begin to formulate the recommended approaches to eliminate the negative impacts caused by those uncontrolled factors, the next stage is to test our PLAM algorithm on the road in Chengdu and Suzhou in China, and London and Southampton in England.

6.4 Summary of our contributions

Our work has demonstrated the contributions as follows. Firstly, LLB algorithm has been proposed, which aims to protect the location privacies and preference privacies of users; and it can also be used to protect the trajectory privacies of users for continuous request. Secondly, three protocols include *request aggregation protocol*, *pseudo-ID exchanging protocol* and *improved PLAM protocol* in our LLB algorithm, can efficiently reduce the response time of the LBS system in IoT, as well lower the possibility for hackers hijacking data in IoT, and thus allows smooth LBS privacy services for users. Finally, performance of our LLB algorithm has been evaluated by extensive simulations, and the simulation results support the validity and effectiveness of our LLB algorithm.

7. CONCLUSION AND FUTURE WORK

In this paper we study the problem of privacy preservation for LBS users have same location in IoT. To protect the location privacy, preference privacy and trajectory privacy of users in a distributed structure of IoT LBS system, we proposed a *location label* based algorithm that includes three key protocols: the *user requests aggregation protocol*, the *pseudo-ID exchange protocol* and the *improved PLAM protocol*. We conduct extensive simulation experiments to evaluate the performance of our proposed algorithm. The simulation results show that the proposed algorithm outperforms the existing approach. Therefore, our work for LBS privacy preservation can be used to ensure the locations of IoT users remain private.

In the future work, we plan to test our proposed approach with real volunteers with real locations in selected cities to further consolidate our contributions. We will integrate other security services such as [33] to ensure that our LBS privacy service can be protected from threats by worms in IoT. We will test our PLAM algorithm on the road in China and England.

ACKNOWLEDGEMENT

This work was partially supported by the National Grand Fundamental Research 973 Program of China under Grant (No. 2013CB329103), Natural Science Foundation of China grant (No.61571098), China Postdoctoral Science Foundation (No. 2015M570778), Guangdong Science and Technology Project (2012B090400031, 2012B090500003, 2012B091000163), and National Development and Reform Commission Project.

REFERENCES

[1] Y. Sun, M. Chen, L. Hu, Y. Qian. ASA: Against statistical attacks for privacy-aware users in Location Based Service. Future Generation Computer Systems, 2016.

[2] B. Niu, X. Zhu, L. Hu, et al. A novel attack to spatial cloaking schemes in location-based services. Future Generation Computer Systems, 49:125-132, 2015.

[3] Y. Li, M. Yiu. Route-Saver: Leveraging Route APIs for Accurate and Efficient Query Processing at Location Based Services. IEEE transactions on knowledge and data engineering (TKDE), 1-15, 2015.

[4] M. Xin, M. Lu, W. Li. An adaptive collaboration evaluation model and its algorithm oriented to multi-domain location-based services. Expert Systems with Applications, 42: 2798-2807, 2015.

[5] T. Shu, Y. Chen, J. Yang. Protecting Multi-Lateral Localization Privacy in Pervasive Environments. IEEE/ACM Transactions on Networking, 1688-1701, 2015.

[6] X. Liu, K. Liu, L. Guo, et al. A game-theoretic approach for achieving k-anonymity in Location Based Services. IEEE INFOCOM, 2985-2993, 2013.

[7] B. Niu, Q. Li, X. Zhu, et al. Enhancing Privacy through Caching in Location-Based Services. IEEE INFOCOM, 1017-1025, 2015.

[8] X. Liu, H. Zhao, M. Pan, et al. Traffic aware multiple mix zone placement for protecting location privacy. IEEE INFOCOM, 972-980, 2012.

[9] D. Chen, P. Zhang, C. Hu, et al. PAPERS: Private and Precise Range Search for Location Based Services. IEEE International Conference on Communications (ICC), 7347-7352, 2015.

[10] G. Zhuo, Q. Jia, L. Guo, et al. Privacy-Preserving Verifiable Proximity Test for Location-Based Services. IEEE GLOBECOM, 2015.

[11] X. Chen, A. Mizera, J. Pang. Activity tracking: A new attack on location privacy. IEEE Conference on Communications and Network Security (CNS), 22-30, 2015.

[12] R. Jiang, R. Lu, K. Choo. Achieving high performance and privacy-preserving query over encrypted multidimensional big metering data. Future Generation Computer Systems, 2016.

[13] S. Zhang, Q. Liu, Y. Lin. Anonymizing popularity in online social networks with full utility. Future Generation Computer Systems, 2016.

[14] D. Yang, X. Fang, G. Xue. Truthful incentive mechanisms for k-anonymity location privacy. IEEE INFOCOM, 2994-3002, 2013.

[15] C. Bettini, S. Mascetti, X. Wang, et al. Anonymity in location based services: Towards a general framework. IEEE International Conference on Mobile Data Management, 69-76, 2007.

[16] B. Niu, Q. Li, X. Zhu, et al. Achieving K-anonymity in Privacy Aware Location Based Services. IEEE INFOCOM, 754-762, 2014.

[17] X. Zhu, H. Chi, B. Niu. When k-anonymity meets cache. IEEE GLOBECOM, 820-825, 2013.

[18] J. Shao, R. Lu, X. Lin. FINE: A Fine-Grained Privacy-Preserving Location Based Service Framework for Mobile Devices. IEEE INFOCOM, 244-252, 2014.

[19] B. Niu, X. Zhu, W. Li, et al. A Personalized Two-Tier Cloaking Scheme for Privacy-Aware Location-Based Services. International Conference on Computing, Networking and Communications, Communications and Information Security Symposium, 94-98, 2015.

[20] R. Lu, X. Lin, Z. Shi, et al. PLAM: A Privacy-Preserving Framework for Local-Area Mobile Social Networks. IEEE INFOCOM, 763-771, 2014.

[21] A. Uchiyama, S. Fujii, K. Maeda, et al. UPL: opportunistic localization in urban districts. IEEE Transactions on Mobile Computing, 12(5): 1009-1022, 2013.

[22] H. Li, L. Sun, H. Zhu, et al. Achieving Privacy Preservation in WiFi Fingerprint Based Localization. IEEE INFOCOM, 2337-2345, 2014.

[23] T. Shu, Y. Chen, J. Yang. Multi-lateral Privacy-Preserving Localization in Pervasive Environments. IEEE INFOCOM, 2319-2327, 2014.

[24] A. Beresford, F. Stajano. Mix Zones: User Privacy in Location-aware Services. IEEE Conference on Pervasive Computing and Communications Workshops, 127-131, 2004.

[25] Z. Chen, X. Hu, X. Ju, et al. LISA: location information scrambler for privacy protection on smartphones. IEEE Conference on Communications and Network Security, 296-304, 2013.

[26] J. Abawajy, G. Wang, L. Yang, et al. Trust, Security and Privacy in Emerging Distributed Systems. Future Generation Computer Systems, 55: 224-226, 2016.

[27] Y. Feng, P. Liu, J. Zhang. A Mobile Terminal Based Trajectory Preserving Strategy for Continuous Querying LBS Users. IEEE International Conference on Distributed Computing in Sensor Systems, 92-98, 2012.

[28] N. Mohammed, B. Fung, M. Debbabi. Walking in the crowd, Anonymizing trajectory data for pattern analysis. The 18th ACM Conference on Information and Knowledge, 1441-1444, 2009.

[29] Y. Wang, J. Peng, L. He, et al. LBSs Privacy Preserving for Continuous Query based on Semi-Honest Third Parties. IEEE International Performance Computing and Communications Conference, 384-391, 2012.

[30] Y. Liao, C. Hsiao. The improvement of ID-based remote user authentication scheme using bilinear pairings. IEEE International Conference on Consumer Electronics, Communication and Networks, 865-869, 2011.

[31] D. Liao, X. Huang, V. Anand, et al. k-DLCA: An efficient approach for location privacy preservation in location based services. IEEE International Conference on Communications (ICC), 1-6, 2016

[32] OPNET, https://www.opnet.com/

[33] S. Aljawarneh, R. Moftah, A. Maatuk. Investigations of automatic methods for detecting the polymorphic worms signatures. Future Generation Computer Systems, 60: 67-77, 2016.