Article

DGS-HSA: A Dummy Generation Scheme Adopting Hierarchical Structure of the Address

Mingzhen Li ^{1,2,*}, Yunfeng Wang ¹, Guangcan Yang ¹, Shoushan Luo ¹, Yang Xin ^{1,3}, Hongliang Zhu ¹, Yixian Yang ^{1,3}, Yuling Chen ³ and Fugui Luo ²

¹ National Engineering Laboratory for Disaster Backup and Recovery, Information Security Center, School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China; wangyunfeng@bupt.edu.cn (Y.W.); yangguangcan@bupt.edu.cn (G.Y.); buptlou@263.net (S.L.); yangxin@bupt.edu.cn (Y.X.); zhuhongliang@bupt.edu.cn (H.Z.); yxyang@bupt.edu.cn (Y.Y.)

² School of Computer and Information Engineering, Hechi University, Guangxi 546300, China; lfg046@163.com

³ Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guizhou 550025, China; ylchen3@gzu.edu.cn

* Correspondence: mzli@bupt.edu.cn; Tel.: +86-138-117-91091

† Current address: National Engineering Laboratory for Disaster Backup and Recovery, Information Security Center, School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China.

Received: 23 November 2019; Accepted: 8 January 2020; Published: 11 January 2020

Abstract: With the increasing convenience of location-based services (LBSs), there have been growing concerns about the risk of privacy leakage. We show that existing techniques fail to defend against a statistical attack meant to infer the user's location privacy and query privacy, which is due to continuous queries that the same user sends in the same location in a short time, causing the user's real location to appear consecutively more than once and the query content to be the same or similar in the neighboring query. They also fail to consider the hierarchical structure of the address, so locations in an anonymous group may be located in the same organization, resulting in leaking of the user's organization information and reducing the privacy protection effect. This paper presents a dummy generation scheme, considering the hierarchical structure of the address (DGS-HSA). In our scheme, we introduce a novel meshing method, which divides the historical location dataset according to the administrative region division. We also choose dummies from the historical location dataset with the two-level grid structure to realize the protection of the user's location, organization information, and query privacy. Moreover, we prove the feasibility of the presented scheme by solving the multi-objective optimization problem and give the user's privacy protection parameters recommendation settings, which balance the privacy protection level and system overhead. Finally, we evaluate the effectiveness and the correctness of the DGS-HSA through theoretical analysis and extensive simulations.

Keywords: location-based service; location privacy; query privacy

1. Introduction

With the rapid development of smart terminal technology, positioning technology, and mobile Internet technology, location-based service (LBS) is increasingly penetrating all aspects of life, such as inquiring about nearby restaurants, hotels, and so on, bringing great convenience to people. In reality, people enjoy location-based services over the cyber. Considering the network security situation, people inevitably suffer serious cyber risks, e.g., malware [1], spyware [2], encryption for malicious

purposes [3], and zero-day attacks [4]. Privacy leakage, as one of the cyber risks, is drawing more and more attention.

At present, the intrinsic privacy leakage for LBS is increasingly becoming a huge challenge for protecting a user's privacy. In applying LBS, a user submits a request, which contains locations and query content, to the LBS server. However, users' locations and query content are often leaked, due to the compromise between the provider (LP) and attackers. As a result, users are exposed to privacy leakage when using an LBS. For example, attackers can infer a user's privacy (e.g., where the user works and goes for a dinner, where the user stays overnight) from locations and query content (e.g., "which hospitals are nearby"). Even then, users still need to use LBSs in many cases. Thus a location privacy-preserving technology must settle these problems to protect locations and so query content is not disclosed. In this work, we investigate how to protect users' privacy in the scenario of a continuous query.

In this scenario, to protect users' privacy, researchers have presented a variety of methods like privacy policy [5,6], distortion, encryption [7–11], and so on. Among these methods, distortion techniques, such as pseudonym, obfuscation, and dummy, are most widely used to protect privacy. Pseudonym techniques [12,13] protect users' privacy by replacing or directly deleting users' identity identifiers with a false or temporary pseudonym. While users can use pseudonyms for privacy protection, adversaries can still infer the user's real identity by analyzing the spatial-temporal correlation of continuous queries [13]. Obfuscation techniques [14–16] protect the users' privacy by generalizing or perturbing the time and locations in LBS queries, and then the users' precise locations are not recognized. However, submitting inaccurate locations will reduce the quality of the service. An effective approach is to maintain the quality of service while ensuring that users' privacy is not leaked. Considering that the dummy techniques [13,17–27] to protect the privacy of a user are to add dummy queries to the real query and do not reduce the quality of the service. They are feasible in our work. In this paper, we focus on how to hide a user's locations or query content among dummies in the scenario of a continuous query.

While existing dummy techniques solved the privacy leakage in the LBS query, they mainly focus on the scenario neighbor query in which a user sends queries in different locations and they neglect the scenario where the same user sends a short-time continuous query in the same location (called 'scenario of this paper' for simplicity). Figure 1 shows an example of privacy leakage in the scenario of this paper.

In the example, the real user sends three queries in location L to query nearby points of interest (POI). For each query, three dummy users are selected to form an anonymous group. However, existing techniques, e.g., the distortion technique, only select one of the three queries (we assume the first query is selected) to hide the location and query content since existing techniques neglect the scenario of this paper. So, we assume not one of the users sends three queries except the real user u_1 in the example. For four query content sent by all users in each query, we assume the first query has three query categories, while there is only one query category for the second and third query, respectively.

As shown in Figure 1, by intersecting users in different anonymous groups and combining background knowledge [26], the attacker could find that only u_1 was in the three queries and then inferred u_1 was the real user who sends queries in L (only the same user has the ability to send queries continuously in a short period of time, see [13]). Moreover, attackers can deduce the user's sensitive location, e.g., home address, workplace, by collecting and analyzing the user's long-term historical query data [13]. For the query content, although the attacker cannot distinguish which one is the query content of u_1 in the first query, he was able to identify it in the second and third queries. We call this kind of attack, one meant to obtain sensitive location and query content privacy by intersection inference attacks (see [27]), intersection inference attack for location and query content (I2LC).

Furthermore, existing methods do not consider the hierarchical structure of the address (HSA), which may reveal a user's organization information. Actually, a region is comprised of different levels of region, and these different levels of region are called the HSA. For example, the HSA of Beijing is

comprised of four levels of region: City (Beijing), district (Haidian District), street (Xi Tucheng Road), organization (Beijing University of Posts and Telecommunications (BUPT)). If we do not consider the HSA, we may select $k - 1$ locations that locate in the same organization as the real location in the anonymous group. For example, the k locations in the anonymous group are all located in BUPT. Despite this not directly exposing the user's exact location, it reveals that the user is located in BUPT. For some particular users, the organization information also needs to be kept secret, such as confidential personnel, anti-drug police, etc.; once leaked, it will be used by criminals to infer their occupation and pose a life threat to the users.

Figure 1. The same user's continuous query request privacy in the same location.

In this paper, we present a dummy generation scheme considering the HSA, called DGS-HSA, to generate dummy locations and query content. In this scheme, for each query in a continuous query, we select k locations (in our scheme, the user does not send her real locations to the LBS server, see Section 4.4), from historical locations as dummy locations. In each dummy location, a user sends a dummy query selected from the basic service set. In the DGS-HSA, we first divide a city area where users locate into different grids with the two-level structure. Each grid represents an organization and all historical locations are clustered into these grids according to their organizations. Based on these grids, we then propose the degree of privacy protection $\langle L_{(k,s)}, Q_l \rangle$. Using $\langle L_{(k,s)}, Q_l \rangle$, we select dummy locations and query content to protect users' location privacy, organization information, and query privacy in the scenario of this paper. For each query in a continuous query, we select k locations distributed across s different organizations from historical locations and query content in all queries in a continuous query that has at least l query categories. It can protect a user's organization information and query privacy since it ensures that k locations do not locate in the same organization and query content in all queries in a continuous query that has at least l query categories. For all queries in a continuous query, we ensure all locations selected from historical locations are evenly distributed across all different organizations. So, our scheme can resist I2LC and protect users' location privacy. Using theoretical analysis we also prove that our scheme can meet the $\langle L_{(k,s)}, Q_l \rangle$ and resist I2LC. However, considering the privacy protection level and system overhead, the two must reach equilibrium if our scheme is feasible. So, we evaluate the feasibility and effectiveness of our scheme. Results show that our scheme can reach an equilibrium between privacy protection level and system overhead and can better protect users' location privacy, organization information, and query privacy than the scheme in [13,27].

The DGS-HSA can protect users' organization information and query privacy. Compared with [13,27], the DGS-HSA considers the HSA and divides a city area where users locate into different grids with the two-level structure. Based on it, for each query in a continuous query, the DGS-HSA selects k dummy locations distributing across s different organizations and k query content in

a continuous query that has at least l query categories. That is, attackers cannot distinguish the user's real organization and real query content.

Compared with [13,27], the DGS-HSA can resist I2LC and protect users' location privacy. For a continuous query, our scheme evenly selects dummy locations from different organizations. It ensures that all dummy locations are evenly distributed across these organizations. It means the probability that each location distributes across every organization is the same. That is, attackers cannot distinguish the user's real location by launching I2LC. Specifically, the major contributions of this paper are as follows:

- To protect a user's location privacy, organization information and query privacy in the scenario of this paper, we present the DGS-HSA to generate dummy locations and query content. The DGS-HSA considers the HSA and can resist I2LC.
- Considering the HSA, we introduce a novel meshing method to divide a city area where users locate into different grids with the two-level structure. Using this, our scheme ensures that, for each query in a continuous query, k dummy locations do not select from the same organization. In addition, for the query content, the probability of occurrence of each query category is the same in all queries in a continuous query. Thus, our scheme can better protect users' organization information and query privacy than the scheme in [13,27].
- To protect a user's location privacy, we propose a method to resist I2LC. Using this method, for a continuous query, our scheme ensures that each selected dummy location evenly distributes across different organizations in a continuous query. In addition, we also replace the real user's location with historical locations that are in the same organization as the real user's location. It prevents the user's exact location from being sent to the server. Thus, our scheme can resist I2LC and better protects users' location privacy than the scheme in [13,27].
- We evaluate the feasibility and effectiveness of our scheme. Results show that our scheme can reach an equilibrium between privacy protection level and system overhead and we give the recommended configuration of system parameters. They also show that our scheme can better protect users' location privacy, organization information and query privacy than the scheme in [13,27].

The rest of the paper is organized as follows. We first overview the related research in Section 2. Then, in Section 3, we introduce the adversary model, the motivation and basic idea, the privacy protection model, and the system architecture of this paper. We also describe our algorithms in detail in Section 4. Respectively, Sections 5 and 6 evaluate the effectiveness of the presented scheme from both theoretical and experimental aspects, including scheme feasibility analysis. Finally, in Section 7, we summarize the research work and the innovative results and point out the problems and future work.

2. Related Work

Location privacy issues have attracted more and more attention since Beresford and Stajano first raised them [28]. In this section, we mainly review some existing research on the privacy-preserving technique, the query request scenario, system architecture, and the attack model.

2.1. Privacy-Preserving Technique

Researchers have presented a variety of methods such as privacy policy [5,6], distortion, and encryption [7–11]. Among these methods, distortion has been the most widely used method of protecting users' privacy. It mainly contains three main methods: Pseudonym, obfuscation, and dummy. Pseudonym [12,13,28] protects users' privacy by replacing or directly deleting users' identity identifiers with a false or temporary pseudonym. In [12,28], users did not communicate with the LBS server directly but communicated with an anonymizer proxy which replaced the user's real identity to pseudonymous before the user's message was submitted to the LBS server. In [13],

for MNAME and REGP, a user stored several user names, selected randomly one user name from the name set as the current user name in every time LBS query; for SNAME, anonymizer changed the user name to the same and sent the query to LBS server in every time LBS query. However, only using pseudonyms is not enough to protect users' privacy, and adversaries can still infer the user's real identity by analyzing the spatial-temporal correlation of continuous queries [13].

Obfuscation techniques [14–16] protect the users' privacy by generalizing or perturbing the time and locations in LBS queries, and then the users' precise locations are not recognized. In those methods, the anonymizer submitted an area containing at least $k - 1$ other subjects in every time LBS query, instead of submitting the user's exact location; this is called spatial cloaking. During the above process, the LBS query would be delayed until k vehicles have visited the area chosen for the requestor, which reduces the accuracy in time; this is called temporal cloaking. However, submitting inaccurate locations or time will reduce the quality of the service in some specific services. Dummy techniques [13,17–27] to protect the privacy of a user add dummy queries to the real query and do not reduce the quality of the service.

In this paper, we focus on how to hide a user's locations or query content among dummies in the scenario of a continuous query. Different from existing dummy techniques, such as [13,27], in our method, each selected dummy location evenly distributes across different organizations in a continuous query and locations in each query are distributed across different organizations. In addition, we also replace the real user's location with a historical location that is in the same organization as the real user's location.

2.2. Scenario

Query request scenarios include the single query and the continuous query. In the single query, a user usually sends a single query request for a period of time. Many methods such as pseudonyms, perturbations, dummies, etc., are used to protect the identity, location, and query content of the user. In [14], authors first introduced k -anonymity to protect location privacy. Based on [14], the authors in [15,16] pointed out the equilibrium between the personalized privacy protection and quality of service. Methods [12,21,28] raised the anonymous group that is composed of the user's real location and $k - 1$ dummies selected from historical queries. Inspired by [12,15,16,21,28], we consider the equilibrium between privacy protection level and system overhead and select dummy locations from historical queries.

In the continuous query, a user continuously sends multiple query requests for a period of time. Most methods complete privacy protection by randomly adding dummies, using historical data to generate dummies, etc. In [19], the authors first proposed a technique of randomly generating dummies to complete anonymity, but the generated dummies have weak authenticity and can be easily identified. The methods in [20–25] study how to generate more realistic dummy trajectories. In [20], the authors presented trajectory k -anonymity, using the users' historical trajectory data to generate $k - 1$ dummy trajectories. Methods in [16,17,29–31] consider separately a user's motion patterns, physical constraints in the real world, and spatiotemporal correlation awareness to generate dummy trajectories with higher realism.

A basic assumption commonly used in the above two scenarios is that users send query requests in different locations. However, in reality, a user often sends a single or short-time continuous query in the same location. For example, a user at her home would continuously query nearby restaurants within one minute. In this paper, we mainly study the privacy protection of the same user submitting a single or continuous query scenario in the same location.

2.3. System Architecture

In existing works, there are three main system architectures: Centralized architecture, distributed architecture, and independent architecture. In centralized architecture [14–16,28], a trusted third party (a server, called TTP) is introduced into the system to complete anonymity. However, once the TTP

is untrusted or compromised, a user's information may completely leak, since the TTP has all her information. Besides, the anonymous server is a bottleneck of system performance.

In the distributed architecture [29–31], the system completes anonymity without the TTP; this is called TTP-free. In [29,30], k peer users communicate over a short distance and form a collaborative group to achieve anonymity. However, these users are required to exchange query information and to be honest and trustworthy. In addition, sometimes it may take a long waiting time for $k - 1$ users joining for completing anonymity. In [31], authors introduced an anonymous group re-clustering method to solve the problems of a long waiting time completing anonymity and anonymous failure caused by insufficient users and to improve the success rate of clustering.

To overcome the drawbacks of the TTP and the TTP-free methods. The methods in [17,18] presented the independent architecture. In [17], the authors put forward DLS and enhanced DLS to complete anonymity. This method adopted independent architecture and can effectively solve the shortcomings of the above two system architectures. In [18], the authors presented TTcloak that uses dummy techniques to achieve k -anonymity of location and l -diversity of query content.

2.4. Attack Model

Existing work focuses on how to deal with attacks in different scenarios. In [26], the authors proposed reciprocity of k -clusters to achieve strong k -anonymity. In this method, k users remain unchanged in several queries in a short time so that an attacker cannot identify users by intersecting several k -clusters of different queries. In [27], the authors studied the scenario in which different users sent query requests in the same location and proposed L2P2. The L2P2 adopted a distributed architecture and introduced location labels to classify locations of mobile users to sensitive and ordinary locations. When users' locations in the anonymous group were the same and were all located in sensitive locations, they exchanged ID with each other to achieve the anonymous effect. The method in [13] pointed out that, for long-term cumulative sporadic queries and continuous queries, attackers could deduce users' privacy by collecting and analyzing users' historical data, called LSA and RSA attacks. The method in [20] proposed MNAME and SNAME to resist LSA attacks and raised REGP to resist the RSA attack. In these methods, refs. [13,20,26] mainly focused on scenarios where a user sent a query in different locations, and [27] mainly focused on scenarios where the user sent queries in the same locations without considering short-time continuous query and difficult to resist LSA and RSA attacks. However, for continuous queries in the same location in a long time, the users in the anonymous group are not always identical, causing the real user to be able to be inferred by comparing multiple queries. Hence, in this paper, we mainly study the attack (I2LC) which a user might suffer and how to resist it in the scenario of this paper.

There are also several works related to the HSA and meshing methods to divide a city area. The method in [32] pointed out that privacy-related attributes have a hierarchical structure, and studied achieving k -anonymity by clustering in attribute hierarchical structures. The method in [33] pointed out that the level of location privacy relied on the HSA and location privacy protection needed to consider the HSA. For example, for $k - 1$ dummy locations and a real location, which all locate in BUPT, although an attacker cannot distinguish the real location, he knows that the user is in BUPT. In addition, the method in [27] divided an area based on sensitive locations and ordinary locations. The method in [13] divided an area based on the probability of locations. While they do not consider the HSA. In our paper, we consider the HSA and divide an area into different grids with the two-level structure based on the organization.

3. Preliminary

In this section, we first analyze the privacy that an attacker intends to obtain (Adversary Model). Next, we formulate the problem of how user privacy leaks (motivation) and give the basic idea (basic idea) to solve this problem in the scenario of this paper. We also give the criteria

(privacy protection model) for measuring whether a user's privacy has been leaked and the structure needed to protect the user's privacy (system architecture).

3.1. Adversary Model

In this paper, we assume that adversaries are honest but curious and simply collect all the users' data that they can access. Their goal is to infer the user's privacy, including identity, location, organization, and query content, by analyzing these data. In fact, an adversary may be an LBS provider, an LBS user, or a malicious hacker. We mainly consider the LBS provider owning users' full information as an active adversary. To achieve the goal, he can obtain global information of all users' current and historical data. He can also use statistical inference methods to infer a specific user's privacy by combining with side information. Here, side information mainly refers to a city's region information, e.g., administrative region division, street, organization distribution, and so on. In this paper, we only consider protecting the user's location, organization, and query content. The reason for not protecting the user's identity is that the LBSs only providing the query service do not involve the user's identity. However, for the other LBSs needing to log in to the account, because real name Internet access is used in China, once the user logs in to the account, the service provider will know the user's identity. In both cases, protecting the user's identity is of little significance.

3.2. Motivation and Basic Idea

To protect users' privacy in the scenario of this paper, an effective approach is to add dummies. Considering the scenario of this paper, we assume a user sends m queries in a continuous query. Using dummy techniques, there will be $m \times (k - 1)$ selected dummy locations (some dummy locations are the same) and m real locations. Assume there are c kinds of locations among the $m \times k$ locations and n_i is the number of i th kinds of location. Then there are the following relationships. From the perspective of adversaries, the probability p_i that the i th kind of location is estimated as the real location is as follows:

$$p_i = \frac{n_i}{m \times k} \quad (1)$$

In particular, for the real location r (assume r is the j th kind of location), the probability p_j that it is estimated as the real location is as Equation (2).

$$p_i = \frac{m}{m \times k} \quad (2)$$

Considering the I2LC, we assume each dummy location appears once in m queries since only the real location r appears in each query. That is, for each kind of dummy location, n_i . Then we get:

$$\begin{cases} \lim_{m \rightarrow +\infty} \frac{n_i}{m \times k} = 0 (i \neq j) \\ \lim_{m \rightarrow +\infty} \frac{n_i}{m \times k} = \frac{1}{k} (i = j) \end{cases} \quad (3)$$

That is, when the user sends an unlimited number of queries, the probability that the real location is estimated as the real location is far larger than the dummy location. This means that the real location can be easily distinguished.

Hence, if a user uses dummy techniques for privacy protection, the attacker can infer her real identity by launching the I2LC. As shown in Figure 1, in the scenario of this paper, the attacker could infer that a user who appears most frequently may be the real user by intersecting users in different anonymous groups.

In addition, a basic assumption commonly used in existing methods is that all locations are geographic coordinates. For example, we assume user Alice locates in the location $L_i = (t_i, lo_i, la_i)$, where t_i , lo_i and la_i are the time, longitude, and latitude, respectively. That is, the attacker cannot obtain Alice's location privacy if he cannot infer that Alice locates in L_i . However, existing methods

ignore the HSA. In this paper, we divide the HSA into six levels based on common administrative division and use $Le_i, i = 0, 1, \dots, 5$ to denote its level, where Le_4 corresponds to the organization name and Le_5 represents a location, as shown in Figure 2. If we do not consider the HSA, we may select $k - 1$ dummies that locate in the same organization as the real location. In particular, in the methods that have the TTP-free method and collaboratively achieve anonymity through P2P network, the k users are close to each other and easily belong to the same organizations, which reveals the privacy of users' organizations.

Figure 2. Hierarchical structure of the address.

Assume that the attacker obtains k locations, which are distributed across $s (k \geq s)$ different organizations and are from one of m queries in a continuous query. Then the probability p_o that the attacker can distinguish the user's real organization is as follows:

$$p_o = \frac{1}{s} \quad (4)$$

If the k users (k locations) are close enough that they are almost in the same organization, s will be close to 1. Then we get:

$$\lim_{s \rightarrow 1} \frac{1}{s} = 1 \quad (5)$$

That is to say, if we select $k - 1$ dummies that locate in the same organization as the real location, the attacker can easily distinguish the user's real organization. In fact, the attacker can easily infer that different anonymous groups locate in the same organization. As shown in Figure 3. In Figure 3, the rectangle represents an area containing four organizations: R_1, R_2, R_3 , and R_4 . Different red circles represent different anonymous groups: A_1, A_2 , and A_3 . In the example, although users are different in A_1 and A_3 , the attacker can easily infer that the users in A_1 and A_3 are in the same organization. It is more serious in densely populated and larger organizations. According to actual need, this paper only considers location privacy protection of Le_4 and Le_5 levels.

The above formulas reveal the mechanism by which the attacker obtains users' privacy by using the I2LC and the HSA. (1) When the frequency with which each location appears in a continuous query is different, location privacy leakage occurs; (2) users (locations) in an anonymous group are concentrated in the same organization. Hence, the basic idea of our scheme is that each location evenly appears in a continuous query and locations in an anonymous group are distributed across two or more different organizations. In addition, we use the l -diversity to protect the query content. That is,

for all queries in a continuous query, the probability of occurrence of each query category is the same in multiple queries. This means that the probability that the user's query content is recognized is also the same.

To implement our basic idea, we consider two main k -anonymity methods: Cloaking and adding dummy. The cloaking method submits a cloaking area containing k users to the LBS server. (The user's real location is also in the cloaking area.) However, it is difficult to guarantee the basic idea. The reason is that users do not store the information of users who had participated in anonymity. Therefore, it is difficult to determine whether a user appears too frequently in a continuous query. For example, in the cloaking method, a user is randomly selected to participate in anonymity, resulting in the inability to control the user's participation. Another reason is that users cannot know the area of all organizations. For example, in cloaking, randomly selecting users to participate in anonymity causes users' locations to be randomly distributed across different organizations. In this paper, adding dummy is an ideal solution because it is not constrained by the real user's locations, and the dummy locations can be generated flexibly according to some conditions. While some methods, such as randomly generating dummies, selecting dummies from historical queries or historical data, to a certain extent, can guarantee that users in an anonymous group are distributed across specified organizations, there is no guarantee that all users are not in the same organization.

Figure 3. An example of users' organization information leakage.

In this paper, we use the adding dummy to realize the protection of the user's location, organization information, and query privacy in the scenario of this paper. All dummy locations and query content are selected from historical locations and the basic service set. We first divide a city area where users locate into different grids with the two-level structure and different grids represent different organizations, as shown in Figure 4. Each historical location is located in a grid. Then, we can select k dummy locations from s different grids evenly. That is, we can ensure that the k locations are different from each other in each query and distributed in s different organizations. This protects users' privacy at the organization level. In particular, the LBS provider can identify the real user by comparing historical locations (the real user's location is different from the historical locations), since he has all historical locations. Therefore, we consider replacing the real user's location with the historical locations that are in the same organization as the real user's location. This ensures that the locations in each query are different. In addition, for the query content, our idea is to ensure that the probability of occurrence of each query category is the same in all queries. This means that the probability that the user's query content is recognized is also the same.

Figure 4. Two levels of hierarchical structure grid.

3.3. Privacy Protection Model

To implement the above basic idea, the adding dummy method in our paper needs to meet two conditions. (1) The LBS server also needs to store a historical location dataset. It can guarantee that each location evenly appears in a continuous query and k dummy locations cannot locate in the same organization by controlling the user's participation; (2) the LBS server also needs to provide a service category set. This can guarantee that the probability of occurrence of each query category is the same in all queries.

Considering the above conditions, for the LBS system, it needs to provide M different service categories, e.g., restaurant and hospital. Let these service categories be $C = \{C_1, C_2, \dots, C_M\}$; this is called a basic service set. In addition, the LBS server also needs to store a historical location dataset G , used by users to generate dummy locations. The area that historical locations in G are located in is divided into N different first-level grids. So, we denote G as $G = \bigcup_{i=1}^N R_i$, where R_i denotes the i th first-level grid. For R_i , it is further divided into U_i different second-level grids (organizations). Therefore, we denote R_i as $R_i = \bigcup_{j=1}^{U_i} U_{ij} = \{L | L \in U_{i1}, L \in U_{i2}, \dots, \text{or } L \in U_{iU_i}\}$, where U_{ij} denotes the j th second-level grid in R_i and L is a location in U_{ij} . To protect users' privacy, when a user sends the i th query in location L_0 to the LBS server, the user needs to submit a query request $Req^{(i)} = ((L_1^{(i)}, C_1^{(i)}), (L_2^{(i)}, C_2^{(i)}), \dots, (L_k^{(i)}, C_k^{(i)}))$, $L_j^{(i)} \in G, C_j^{(i)} \in C, i = 1, 2, \dots, n, j = 1, 2, \dots, k$. After receiving the query request, the LBS server calculates and generates a response message $Res^{(i)} = ((L_1^{(i)}, S_1^{(i)}), (L_2^{(i)}, S_2^{(i)}), \dots, (L_k^{(i)}, S_k^{(i)}))$. Then, to implement the basic idea, the $Req^{(i)}$ and $Res^{(i)}$ in a continuous query should be bounded. To address this issue, we define the privacy protection model as $\langle L_{(k,s)}, Q_l \rangle$, called the 'degree of privacy protection', as follows:

Definition 1. In a continuous query, if the $\langle L_{(k,s)}, Q_l \rangle$ can protect a user's location privacy, organization information, and query privacy, the $Req^{(i)}$ and $Res^{(i)}$ must satisfy the following conditions:

- (1) $\forall p, q, p \neq q, L_p^{(i)} \neq L_q^{(i)}, p, q \in [1, k], i = 1, 2, \dots, n$;
- (2) The locations $L_1^{(i)}, L_2^{(i)}, \dots, L_k^{(i)}$ are located in s different second-level grids;
- (3) The locations in $Req^{(i)}$ contain at least l ($l \leq M$) different service categories. If a user sends n times queries in location L_0 , the number of times each service category selected in C is approximately equal. That is, the probability of each service category selected in C is equal.

The privacy protection model means that, if a user needs to protect her location, organization information, and query content, she must send k different locations to the LBS server in each query request, which these locations locate in s different second-level grids and the k query content contained in these locations contains at least l ($l \leq M$) different service categories. The first two conditions ensure

that the k locations $Req^{(i)}$ are different from each other and located in s different second-level grids. It is necessary to select k dummies from s different grids in a query request. Condition (3) guarantees that the k query contents in $Req^{(i)}$ are not exactly the same. It is necessary to protect the query content. So, $\langle L_{(k,s)}, Q_l \rangle$ can protect a user's location privacy, organization information, and query privacy.

3.4. System Architecture

Given the adversary model, the basic idea, and the privacy protection model, the problem is how to select a system architecture to select dummy locations and query content. Considering the drawbacks of the TTP and the TTP-free method, in this paper, we adopt a dummy technique that uses historical locations to generate dummy locations to protect the user's privacy, including the organization information and query content. As shown in Figure 5, the system architecture of the dummy technique is a typical independent architecture that avoids the disadvantages of centralized and distributed architectures.

Figure 5. System architecture.

In the LBS system, there are three participants: The user, the mobile terminal, and the LBS server. Additionally, the privacy protection module of the mobile terminal is the core module of the entire system and consists of an *Area* generation module, a dummy location generation module, and a dummy content generation module. The LBS server stores the historical location dataset G , provides the historical location sub-dataset G_{sub} for the user to generate dummy locations, and calculates query results Res for the user. The whole process is described as follows:

- System initialization: This mainly refers to obtaining the user's current location L_0 and initializing the historical location dataset G on the server.
- Initiating a service request: In this stage, the user needs to input initial $\langle k, s, l \rangle$ parameters as the input of the privacy protection module.
- *Area* generation: *Area* is a range of latitude and longitude sent to the LBS server by the user to indicate the latitude and longitude range of the historical location dataset that the user downloads from the LBS server. *Area* is determined by the user's current location L_0 and $L_{(k,s)}$. After the user inputs $\langle k, s, l \rangle$, the *Area* generation module gives *Area* according to L_0 and s , and sends *Area* to the LBS server for downloading G_{sub} .

- Historical location data acquisition: The LBS server sends G_{sub} which meets the requirements of *Area* to the user.
- Query request construction: After generating dummy locations and dummy query content, the privacy protection module constructs the query request message *Req* in the light of a specific format.
- Service response: According to the dummy locations and dummy query content in *Req*, the server calculates query results and constructs and feeds back response message *Res* to the user.

4. Dummy Generation Scheme

In this section, we describe our dummy generation scheme in two parts: Algorithm framework and several key algorithms.

4.1. Algorithm Framework

In our scheme, generating dummy locations and query content requires consideration of two conditions: (1) The LBS provider owning users' full information as an active adversary; (2) in our independent architecture, we select dummy locations and query content from historical location datasets and basic service sets which store on the mobile terminal and the mobile terminal has less storage. In the independent architecture, they mean that the client cannot submit the exact locations to the server and only stores fewer data. The process generating dummy locations and query content is as shown in Figure 6. The main steps are as follows:

- Step 1: The user submits the privacy protection parameters $\langle k, s, l \rangle$ and her current location L_0 to the server. To ensure that the LBS provider cannot identify L_0 , we use an algorithm, called the *Area* generation algorithm, to generate *Area* containing L_0 and submit *Area*, not the exact location L_0 .
- Step 2: After receiving *Area*, the server generates G_{sub} (a sub-dataset of G) and sends it to the mobile terminal. According to the *Area*, the server uses an algorithm, called the historical location sub-dataset generation algorithm, to generate G_{sub} . The G_{sub} is a small dataset and can be stored on the mobile terminal.
- Step 3: The mobile terminal constructs a query request *Req* and submits it to the server. In this step, the mobile terminal uses the dummy location generation algorithm to select dummy locations and the dummy query content generation algorithm to select dummy query content. The two algorithms can ensure that the constructed *Req* meets the privacy protection model.
- Step 4: The server calculates and sends the query results *Res* to the mobile terminal. To ensure that the user knows the query results and the server does not, we use the user query request construction algorithm to calculate the *Res*.

4.2. Area Generation Algorithm

To ensure that the LBS provider cannot identify L_0 , we need to blur L_0 into an area that consists of different organizations. Here, the area is denoted as $Area = (s, lo_{min}, lo_{max}, la_{min}, la_{max})$, where s indicates the number of different second-level grids, $[lo_{min}, lo_{max}]$ and $[la_{min}, la_{max}]$, respectively, point out the range of longitude and latitude of G_{sub} . The *Area* is decided by $L_{(k,s)}$. If we know the user's current location $L_0 = (t_0, lo_0, la_0)$, *Area* is also denoted as $Area = (s, lo_0 + \alpha, lo_0 + \beta, la_0 + \alpha', la_0 + \beta')$. In *Area*, α, β, α' and β' are given by the user. So, we design an algorithm to generate the *Area*. Algorithm 1 shows the formal description of it. First, it judges the number of second-level grids in the *Area*. If so, the server provides G_{sub} . Otherwise, the server increases the value of the initial parameter α, β, α' , and β' , and repeats the above judgment until it contains no less than s second-level grids.

Generally speaking, the higher the user's location privacy protection requirements are and the more grids and historical location data meeting the requirements, the better the privacy protection effect is, and of course the higher the storage and communication overhead are when generating the dummy locations.

Figure 6. Algorithm framework.

Algorithm 1: Area generation.**Input:** $L_0, s, \alpha, \beta, \alpha', \beta'$ **Output:** Area

- 1 $Area \leftarrow \emptyset$;
- 2 let $lo_{min} \leftarrow lo + \alpha, lo_{max} \leftarrow lo + \beta, la_{min} \leftarrow la + \alpha', la_{max} \leftarrow la + \beta'$;
- 3 let $Area \leftarrow \{s, lo_{min}, lo_{max}, la_{min}, la_{max}\}$;
- 4 **Return** Area

4.3. Historical Location Sub-Dataset Generation Algorithm

In our scheme, the mobile terminal is used to implement the process selecting dummy locations and query content. So, it needs to store the historical locations dataset and basic service set. Considering its lesser storage, the server must send a smaller historical location sub-dataset G_{sub} to the mobile terminal. Here, we only consider the G_{sub} , since the basic service set itself is a smaller dataset and can store on the mobile terminal. So, we design an algorithm to generate the G_{sub} . Algorithm 2 shows the formal description of it. First, Algorithm 2 determines whether the historical locations within the range of Area are distributed in s different second-level grids. If so, they meet the requirement and return G_{sub} , which are the historical locations in the given range. Otherwise, the range of the Area is extended by a degree of 10^{-3} and we get a new Area which is expressed as

$Area_1 = (s, lo_{min} - 10^{-3}, lo_{max} + 10^{-3}, la_{min} - 10^{-3}, la_{max} + 10^{-3})$. Repeat the above process until the given historical location range can contain different second-level grids. In general, iterating once will meet the requirement.

Algorithm 2: Historical location sub-dataset generation.

Input: $L_0, Area, G$
Output: G_{sub}

- 1 $G_{sub} \leftarrow \emptyset$;
- 2 Count the n_A in $Area$; // n_A is the number of the second-level grids in $Area$;
- 3 **while** $n_A < s$ **do**
- 4 let $lo_{min} \leftarrow lo_{min} - 10^{-3}, lo_{max} \leftarrow lo_{max} + 10^{-3}, la_{min} \leftarrow la_{min} - 10^{-3},$
 $la_{max} \leftarrow la_{max} + 10^{-3}$;
- 5 let $Area \leftarrow \{s, lo_{min}, lo_{max}, la_{min}, la_{max}\}$;
- 6 Count the n_A in $Area$;
- 7 Update G_{sub} by adding all locations in G that are located in the specified range of $Area$ into G_{sub} ;
- 8 **Return** G_{sub}

4.4. Dummy Location Generation Algorithm

To meet $\langle L_{(k,s)}, Q_l \rangle$, we need to achieve two goals. (1) Each location evenly appears in a continuous query; (2) in each query, k dummy locations do not locate in the same organization. Therefore, we take two measures to ensure that the two goals are achieved. Firstly, the user's real location is not submitted to the LBS server but is replaced by a historical location in the same organization as the user's real location. Secondly, we add an identifier Ide for each location in U_{ij} to ensure selected dummy locations are different in each query request Req . We set Ide to 1 each time a location is selected. When the identifiers of all locations in U_{ij} are set to 1, which represents that all locations in U_{ij} are traversed, all identifiers for each location in U_{ij} are set to 0 to start a new round of selection. While we use a historical location in the same organization to replace the user's real location to protect the real one from being identified, because of the user's real location and the dummy location that replaces the real one in the same organization, the loss of service quality is small.

Based on the above goals and measures, we design an algorithm to generate dummy locations. Algorithm 3 shows the formal description of it. First, we determine the second-level grid $U_{i_0,j}^{sub}$ which the current location L_0 belongs to and then use $U_{i_0,j}^{sub}$ as the starting point to randomly select $s - 1$ second-level grids from near and far. The s second-level grids are expressed as $SG = \{U_{i_0,j}^{sub}, U_{i_1,j+1}^{sub}, U_{i_2,j+2}^{sub}, \dots, U_{i_{s-1},j+s-1}^{sub}\}$. Then, we randomly select one location from each grid of SG and obtain s locations $L_1, L_2, \dots, L_s$ which are located in s different second-level grids. We also set each identifier Ide of the selected historical locations to 1. Second. we randomly select the next s locations $L_{s+1}, L_{s+2}, \dots, L_{2s}$ from the historical locations of $Ide = 0$ in s grids of SG , and set the identifier Ide of selected historical locations to 1. Then, we repeat the above procedure until the remaining $k - 2s$ locations are selected. Finally, k locations $L_1, L_2, \dots, L_k$ are selected and evenly distributed in s second-level grids. Among them, we use L_1 to replace the user's real location L_0 to prevent the server from distinguishing L_0 from other dummy locations by comparing with historical location data it owns. For simplicity, we assume that there are enough historical locations in the historical location dataset (in fact, we can also get enough historical locations) to ensure that users can choose different historical locations each time. The specific algorithm is described as follows.

Algorithm 3: Dummy locations generation.

Input: L_0, s, k, G_{sub}
Output: anonymous group DL

- 1 $SG \leftarrow \emptyset, IDE \leftarrow \emptyset;$
- 2 Compute the second-level grid $U_{i_0,j}^{sub}$ which L_0 belongs to;
- 3 Use $U_{i_0,j}^{sub}$ as the starting point and randomly select $s - 1$ second-level grids
 $U_{i_1,j+1}^{sub}, U_{i_2,j+2}^{sub}, \dots, U_{i_{s-1},j+s-1}^{sub}$ from near and far;
- 4 Let $SG \leftarrow \{U_{i_0,j}^{sub}, U_{i_1,j+1}^{sub}, U_{i_2,j+2}^{sub}, \dots, U_{i_{s-1},j+s-1}^{sub}\};$
- 5 Let each identifier Ide of all locations in s second-level grids in SG be 0;
- 6 Let $IDE \leftarrow \{0, 0, \dots, 0\}$, where IDE is the set of identifiers of k selected locations;
- 7 $i \leftarrow 1;$
- 8 **while** $i \leq 1 + \frac{k}{s}$ **do**
- 9 **if** $i! = 1 + \frac{k}{s}$ **then**
- 10 Randomly select one location $L_{i,t}$ from $U_{i',j+(t-1)}^{sub} \in SG$ that its identifier $Ide_{i,t} = 0$,
 where t is from 1 to s ;
- 11 Update IDE by setting the identifiers of s selected locations to 1;
- 12 $i++$;
- 13 **else**
- 14 Randomly select $k - (i - 1)s$ second-level grids from SG and randomly select one
 location that its identifier $Ide_{i,t} = 0$ from each selected $k - (i - 1)s$ second-level grids;
- 15 Update IDE by setting identifiers of $k - (i - 1)s$ selected locations to 1;
- 16 **exit**
- 17 **for all** $Ide \in IDE$ **do**
- 18 **if** $Ide_{i,t} = 1$ **then**
- 19 $DL \leftarrow L_{i,t};$
- 20 **Return** DL

4.5. Dummy Query Content Generation Algorithm

The number of locations participating in anonymity is k , so the amount of corresponding anonymous query content is also k . Assuming that the user has queried n times in a row, the number of locations and the amount of query content of participating in the anonymity are both nk . In n times continuous queries, the idea of protecting the user's query content is that the probability of each service category selected in the basic service set C is equal. In other words, the number of times each service category selected is approximately equal, and the attacker cannot guess the user's query privacy from the distribution characteristics. We denote C as $C = \{C_1, C_2, \dots, C_M\}$, and use n_i to represent the times C_i selected. n_i should meet the following constraints:

- $\forall i, j, i \neq j, i, j \in [1, M], |n_i - n_j| \rightarrow 0$
- each query contains at least l different service categories.

Then, $n_i = \frac{nk}{M}, l \leq M \leq k$.

The specific algorithm is described as follows (Algorithm 4).

Algorithm 4: Query content generation.

Input: k, n, l, C
Output: C'

```

1  $C' \leftarrow \emptyset;$ 
2  $r \leftarrow 1;$ 
3 while  $r \leq n$  do
4   for  $C_i \in C$  do
5     if  $n_i = \frac{kM}{n}$  then
6        $C \leftarrow C \setminus \{C_i\};$ 
7   Call Dummy location generation algorithm to get  $DL$ ;
8   Randomly select  $\{C'_1, C'_2, \dots, C'_k\}$  from  $C$  for  $DL$ , where they comprise no less than  $l$ 
     different types of services;
9    $C' \leftarrow \{C'_1, C'_2, \dots, C'_k\};$ 
10   $r \leftarrow r + 1;$ 
11 Return  $C'$ 

```

4.6. User Query Request Construction Algorithm

After generating k dummy locations and k dummy query content that satisfy the privacy protection requirement $\langle L_{(k,s)}, Q_l \rangle$, the user's query request Req can be expressed as $Req = ((L_1, C'_1), (L_2, C'_2), \dots, (L_k, C'_k)), C'_1, C'_2, \dots, C'_k \in C$.

4.7. Response Message Generation Algorithm

After receiving the user query request $Req = ((L_1, C'_1), (L_2, C'_2), \dots, (L_k, C'_k))$, the server calculates and generates a response message $Res = ((L_1, S_1), (L_2, S_2), \dots, (L_k, S_k))$. Finally, the user receives the response message Res and obtains the query results.

5. System Analysis

In this section, we analyze the security and feasibility of our scheme. Specifically, following the scheme, we prove the feasibility of the scheme and solve optimal solution problems. We also examine whether our scheme can achieve the desirable security and privacy requirements.

5.1. Existence of Solutions and Optimal Solution Problems

The ideal privacy protection scheme is to maximize the privacy protection level, minimize system overhead, and balance the privacy protection and overhead in specific scenarios. This section will prove the feasibility of the scheme by proofing the existence of the solution of the multi-objective optimization problem and solve the equilibrium problem of privacy protection level and system overhead balance.

In this paper, the security goal is to achieve privacy protection level $\langle L_{(k,s)}, Q_l \rangle$. We use the probability of organization information being identified p_A to measure $L_{(k,s)}$ and use the probability of query content being identified p_S to measure Q_l . The smaller p_A and p_S are, the higher the privacy protection level is. To achieve the expected security goal, we select k dummy locations from G_{sub} and make k dummy locations be located in s second-level grids (i.e., organizations). G_{sub} is related to $|U_j^{sub}|$ (the number of locations in U_j^{sub}) and s . The larger $|U_j^{sub}|$ and s are, the larger the G_{sub} is, and the larger the corresponding communication overhead and storage overhead are. The equilibrium problem of privacy protection level and system overhead can be described as minimizing p_A , p_S , and G_{sub} .

The objective functions of the above problem are denoted as $\min\{p_A(k, s)\}$, $\min\{p_S(k, s)\}$, and $\min\{c(s, G_{sub}) \cdot c(s, G_{sub})\}$ signifies the system overhead. Here, we consider the storage overhead and

communication overhead related to G_{sub} for the following two reasons. On the one hand, compared with other methods, the increase in the cost of our method chiefly comes from the increase of G_{sub} . The cost of other data involved in the service process is much smaller than G_{sub} . On the other hand, G_{sub} mostly affects the computational cost of selecting s organizations. Once s organizations are determined, the computational cost of selecting k dummy locations is basically unchanged.

Constraints are described as follows. $\frac{k}{s}$ locations for each organization correspond to $\frac{k}{s}$ query content, each query content corresponds to a different service category, and $\frac{k}{s}$ is no greater than M (in this paper, we set $M = 21$ (see Section 6), $s \geq 2$; here, let $\frac{k}{s} \leq M$ be reasonable; otherwise, it will cause k to be too large, which will greatly increase the system overhead). There is at least one solution that can make the $\frac{k}{s}$ query content corresponding to $\frac{k}{s}$ locations in each organization different, and the number of times each query category selected after n times queries equal.

We use x_{ij}^z to indicate whether the category corresponding to the j th location in the i th organization in the z th time query is C_j . If $x_{ij}^z = 1$, it indicates that C_j is selected; otherwise, $x_{ij}^z = 0$ indicates that C_j is not selected. The constraints are expressed as follows.

$$\begin{cases} \sum_{j=1}^M x_{ij}^{(z)} = \lceil \frac{k}{s} \rceil - \eta_i \\ \sum_{i=1}^s x_{ij}^{(z)} = \lceil \frac{k}{M} \rceil - \mu_j \\ \sum_{i=1}^s (\lceil \frac{k}{s} \rceil - \eta_i) = \sum_{j=1}^M (\lceil \frac{k}{M} \rceil - \mu_j) = k \\ \sum_{z=1}^n \sum_{j=1}^M x_{ij}^{(z)} - \sum_{z=1}^n \sum_{j=1}^M x_{i'j}^{(z)} = 0 \\ \eta_i, \mu_j \in \{0, 1\} \\ i, i' = 1, 2, \dots, s, \text{ and } i \neq i'; j = 1, 2, \dots, M; 2 \leq s \leq k; z = 1, 2, \dots, n \end{cases} \quad (6)$$

In System of Linear equations (6), $\frac{k}{s}$ and $\frac{k}{M}$ are rounded up for the rigor of logic; $\sum_{i=1}^s (\lceil \frac{k}{s} \rceil - \eta_i) = \sum_{j=1}^M (\lceil \frac{k}{M} \rceil - \mu_j) = k$ ensures the number of dummy locations and dummy query content selected are both k ; $\sum_{j=1}^M x_{ij}^{(z)} = \lceil \frac{k}{s} \rceil - \eta_i$ indicates the number of locations belonging to the i th organization in the z th times query; $\sum_{i=1}^s x_{ij}^{(z)} = \lceil \frac{k}{M} \rceil - \mu_j$ represents the number of times that the service category C_j is selected in the z th times query. $\sum_{z=1}^n \sum_{j=1}^M x_{ij}^{(z)} - \sum_{z=1}^n \sum_{j=1}^M x_{i'j}^{(z)} = 0$ denotes that the number of times that any of the basic service categories is selected in the times queries are equal, which can be derived from $\sum_{i=1}^s x_{ij}^{(z)} = \lceil \frac{k}{M} \rceil - \mu_j$.

The objective function having a solution means that given k , when s changes from 2 to k , there is at least one scheme for each value of s such that it satisfies the constraints. Then k and s can be regarded as a fixed value in a specific solving process, and whether or not the objective function with a solution is transformed, the equations in the constraint conditions have a solution.

Conclusion 1: The System of Linear equations (6) with integer coefficient has multiple integer solutions.

Proof. In a certain solving process, the number of variables is $s \times m$, and the coefficient matrix and the augmented matrix of System of Linear equations (6) are, respectively, denoted as B and $\bar{B}$. Let $b'_i = \lceil \frac{k}{s} \rceil - \eta_i$ and $b''_i = \lceil \frac{k}{M} \rceil - \mu_j$, then $\bar{B}$ is expressed as follows.

$$\bar{B} = \left(\begin{array}{cccccccccccccccc|c} x_{11} & x_{12} & \cdots & x_{1M} & x_{21} & x_{22} & \cdots & x_{2M} & \cdots & x_{s1} & x_{s2} & \cdots & x_{sM} & b'_1 \\ 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 & b'_2 \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & \cdots & 0 & 0 & \cdots & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & b'_s \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 1 & 1 & \cdots & 1 & b''_1 \\ 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & \cdots & 1 & 0 & \cdots & 0 & b''_2 \\ 0 & 1 & \cdots & 0 & 0 & 1 & \cdots & 0 & \cdots & 0 & 1 & \cdots & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 1 & \cdots & 0 & 0 & \cdots & 1 & b''_M \end{array} \right) \quad (7)$$

B is composed of the first $s \times M$ columns in $\bar{B}$. We can work out the invariant factors of the matrix B and the augmented matrix $\bar{B}$ through matrix elementary transformation. The invariant factors of B and $\bar{B}$ are both $(1, 1, \dots, 1)_{M+s-1}$. In addition, the ranks of B and $\bar{B}$ are both $M + s - 1$. The number of effective equations in the constraint conditions is $s + M$ and $s + M - 1 \leq s + M \leq sM$. According to references [34,35], System of Linear equations (6) has multiple integer solutions. Hence, there must be a scheme to make the objective function reach the local or global optimum. Furthermore, the presented scheme is feasible, and the objective function can reach the local or global optimum. $\square$

5.2. Security Analysis

5.2.1. User's Privacy Protection Requirements $\langle L_{(k,s)}, Q_l \rangle$

User's privacy protection requirements in a single query: In a single query, the user submits a query request $Req = ((L_1, C'_1), (L_2, C'_2), \dots, (L_k, C'_k))$ to the LBS server. If the user wants the attacker to identify the user's real location with a probability that is not greater than $\frac{1}{k}$, and the user's organization name is identified with a probability that is no greater than $\frac{1}{s}$, the user's location privacy protection requirement is called $L_{(k,s)}$. If the user wants the attacker to identify the user's actual query content with a probability that is not greater than $\frac{1}{l}$, the user's query privacy protection requirement is called Q_l .

User's location privacy protection requirements in continuous queries in the same location: The user submits queries continuously in the same location. If the following conditions are met:

- (i) $Pr\{L_{real} = L_{i1} | Req_i, i = 1, 2, \dots, n\} \leq \frac{1}{k};$
- (ii) $Pr\{L_{real-s} = L_{i1} | Req_i, i = 1, 2, \dots, n\} \leq \frac{1}{s};$
- (iii) $Pr\{C'_{real} = C'_{i1} | Req_i, i = 1, 2, \dots, n\} \leq \frac{1}{l}$

The proposed scheme meets the user's privacy requirements $\langle L_{(k,s)}, Q_l \rangle$ in a continuous query request scenario in the same location.

5.2.2. Security Analysis of the Presented Scheme

Conclusion 2: The presented scheme can achieve privacy protection requirements $\langle L_{(k,s)}, Q_l \rangle$.

Proof. In a single query, the presented scheme is clearly able to meet the user's privacy protection requirements $\langle L_{(k,s)}, Q_l \rangle$. In n times continuous queries, the k dummy locations included in each Req are selected from s different second-level grids, so the number of organizations that nk dummy locations belong to is s and then the probability that the user's real organization is recognized is no greater than $\frac{1}{s}$. Thus, the presented scheme satisfies $L_{(k,s)}$. Moreover, The number of each basic service category selected is $\frac{n \times k}{M}$ times, so the probability of each basic service category being recognized is $(\frac{nk}{M}) / nk = \frac{1}{M} \leq \frac{1}{l}$, which satisfies Q_l . $\square$

Conclusion 3: The presented scheme can resist LSA.

Proof. In this paper, the core idea for location privacy protection is to make the locations submitted to the LBS server as scattered as possible. The locations in each Req are different. As the number of query times n increases, the nk locations involved in the n query requests are always different from each other. nk locations disperse in no less than s different second-level grids. The distribution characteristics of the nk locations are consistent with the locations distribution characteristics of the locations in G_{sub} , and there will be no situation where the user's locations are concentrated in some specific places. Therefore, the attacker cannot identify where the user often appears. In addition, because the user's real location is replaced by the other location in the same second-level grid that the real one is located in, the probability that the real location is recognized is 0. As to the query content protection, the idea of homogenization is adopted to ensure the number of times that each basic service category is selected in each query is the same, then the probabilities of each service category being selected are all $\frac{1}{M}$. From the above, it can be proved that the presented scheme can resist LSA attacks. $\square$

Conclusion 4: The presented scheme can resist RSA.

Proof. It can be seen from the proof of Conclusion 3 that the nk mutually different locations in the n queries are scattered in s different second-level grids, and there is no case where some specific locations are highly concentrated. Each query does not submit the user's real location, and the query content protection reaches l -diversity. So the attacker cannot obtain the user's real location and query content through association analysis. In a word, the presented scheme can resist RSA attacks. $\square$

5.3. Performance Analysis

- **Utility:** In the same LBS, the accuracy of the query results is determined by whether or not the user's real location and real query content are submitted to the LBS server. Both REGP and L2P2 submit the user's real location and real query content, and the query results are unaffected. In this paper, although we do not submit the user's real location, we replace the real location with the other historical location in the same second-level grid where the real location is located. The loss in utility is acceptable, and is caused by this replacement. The reason is as follows. In the actual application, the result of the neighbor query is basically around the organization where the real location is located, and the historical location that replaces the real location is located in the same organization as the real location. So there is a small loss in utility. In addition, each query submits the real query content and the dummy query content, so the obtained query results contain the query results of the real query content.
- **Communication overhead:** The communication overhead the presented method primarily includes four aspects: i) The user submits $Area$ to the LBS server. The communication overhead of submitting $Area$ is $O(1)$. ii) The LBS server sends G_{sub} to the user. G_{sub} contains no less than s different second-level grids, and each second-level grid contains approximately n records. The communication overhead of sending G_{sub} is $O(s \times n)$. iii) The same as the existing methods, each query request submitted to the LBS server by the user contains k locations and k query content. This part of the communication overhead is $O(k)$. iv) Assuming that each query returns m POI, the LBS server needs to return mk query results to the user, so the communication overhead of the query results is $O(k^a \times m^b)$. Compared with REGP and L2P2, our method has more communication overhead about $Area$ and G_{sub} , which provides the historical locations for dummy generation.
- **Computational cost:** The computational cost mostly includes three aspects: (i) After receiving $Area$, the LBS server generates a historical location sub-dataset G_{sub} , and the computational cost of this part is $O(l)$; (ii) the user generates $k - 1$ dummy locations and $k - 1$ dummy query content according to the parameter $\langle k, s, l \rangle$, then the computational cost of generating dummies is $O(k + m)$; (iii) The LBS server calculates and returns $m \times k$ query results to the user, and the computational cost about query results is $O(k^a \times m^b)$. In terms of computational cost, our method is approximately equal to REGP and L2P2.
- **Storage overhead:** In this paper, the client stores a historical location sub-dataset G_{sub} to generate dummies, which contains approximately $s \times n$ records, so the storage overhead is $O(s \times n)$. We mainly consider the storage overhead G_{sub} on the client side. Besides, the LBS server stores the historical location dataset G , which is about 1.6 GB and is negligible relative to the storage space of the server. Similarly, to defend against RSA, REGP needs to obtain and store the PLs on the map. The storage overhead is $O(r \times r)$, whereas L2P2 does not need additional storage overhead.

6. Experiment

L2P2 study privacy protection for users is within the same locations both in single request and continuous request, the scene of which is similar to ours. Besides, REGP aims at resisting the attacks of RSA and LSA; we also try to solve the same problem. So we compare DGS-HSA with L2P2 and REGP from privacy protection effects and system overhead to evaluate the effectiveness of the presented

scheme, DGS-HSA. The privacy protection level is measured by the probability P_A that the user's real organization is recognized and the probability P_S that the user's real query content is recognized. Among those, P_A reflects after considering the factor of the hierarchical structure of the address that the scheme not only protects the specific location but also protects the organization information corresponding to the location. P_S reflects after adopting l -diversity that the scheme can protect the user's query content. The smaller P_A and P_S are, the better the privacy protection effect is. The system overhead $C(G_{sub})$ primarily refers to the storage overhead and communication overhead, which is related to the historical location sub-dataset G_{sub} .

Below, we first describe the dataset and the experimental setup and then give out the experiment results and analysis after conducting extensive simulations.

6.1. Dataset

We choose the Geolife Trajectories 1.3 dataset of Microsoft Research Asia [36–38] as the historical location dataset and use the POI dataset of Amap [39] to provide the query results. Amap is a free map product in China, and also a very comprehensive and informative map application based on location.

REGP divided the map into grids of 1000×1000 . We preprocess and mesh the original Geolife Trajectories 1.3 dataset according to the administrative region division. Firstly, we delete the sequence number and time field in the dataset and retain the longitude and latitude of the location. Then we add the district-level administrative region names (called district names for short) and organization names to each location of the dataset by using the open developer interface of Amap. At last, we obtain G which contains records corresponding to the historical locations and is used to generate dummies. Each record of G has four fields of longitude, latitude, district name, and organization name. More specifically, we sort G according to the district names, and the records that have the same district name field are regarded as the same first-level grid R_i . Then we sort each first-level grid R_i by organization name, and the records that have the same organization name are regarded as the same second-level grid U_{ij} . After twice sorting and meshing, we obtain a grid dataset with a two-level structure. The grid dataset is stored on the LBS server and maintained by the LBS provider, using a quadtree for data indexing.

6.2. Experimental Setup

In the experiment of DGS-HSA, we select the trajectory data within three kilometers of BUPT in the Geolife Trajectories 1.3 dataset as the historical location dataset G . The processed historical location dataset has three first-level grids ($|R^{sub}| = 3$) and approximately 1300 second-level grids ($|U^{sub}| = 1300$). One hundred users are randomly distributed in different second-level grids. The basic service categories of Amap are used as the basic service set of the experiments, which has about 21 categories ($M = 21$). Set ($k = 45$), then ($s = 1, 2, \dots, k$). Each user sends 100 query requests at a frequency of once per minute.

In the experiment of L2P2, the second-level grids where the residential quarters and hospitals are located are set as sensitive areas. In order to ensure the privacy protection effect, we set the PID exchange probability $\rho = 0.5$ and $l = \frac{k}{2}$.

In the experiment of REGP, we divide all second-level grids into four PLs. The second-level grids where the confidential organizations are located are set as the first-level privacy zone w_1 , such as the military and scientific research departments. The second-level grids where the hospital and residential community are located are set as the second-level privacy zone w_2 . In w_2 , personal privacy is easily leaked. The second-level grids where the parks are located have relatively fewer people and are set as the third-level privacy zone w_3 . The second-level grids that the malls and the schools are located in are densely populated, and are set as the fourth-level privacy zone w_4 . We also set the parameter $\lambda = 4$ and $w_1 : w_2 : w_3 : w_4 = 8 : 6 : 4 : 1$.

6.3. Experimental Results

6.3.1. The Probability P_A that the User's Real Organization is Recognized

The presented scheme does not submit the user's real location, and the probability that the real location is recognized is zero. Therefore, this section discusses the relationship between P_A and k . The results are plotted in Figure 7. When k increases, the P_A of all three methods shows a downward trend. The larger k is, the more the number of different organizations of k locations are located, and the smaller P_A is. Among the three methods, DGS-HSA has the best privacy protection effect, and L2P2 is the worst. This is because DGS-HSA considers the hierarchical structure of the address. When we select the dummies, k locations need to be located in s different organizations; however, L2P2 realizes anonymity through an ad-hoc network in which the communication distance is limited, so the distribution of k locations is relatively concentrated. In other words, s is smaller, and in extreme cases (populated dense area) $s = 1$; when REGP selects dummy locations based on the probability of the historical queries and the locations are scattered as much as possible, s can increase to some extent. Compared with L2P2, REGP has a larger s , so it has a better privacy protection effect. However, when the same historical query probability occurs frequently in some areas, there is no guarantee that k locations will not be in fewer organizations (i.e., s is relatively smaller) and the privacy protection effect will be weakened.

Figure 7. The probability P_A that the user's real organization is recognized varies with k .

6.3.2. The Probability P_S that the User's Real Query Content is Recognized

It can be seen from Figure 8 that for different k , the P_S of REGP is larger than that of L2P2 and DGS-HSA, indicating that the privacy protection effect of REGP is the worst. This is because both L2P2 and DGS-HSA use l -diversity to protect the query content, while REGP does not use l -diversity. In addition, DGS-HSA requires k locations to be evenly distributed among s different organizations, and the categories of query content in each organization are as different as possible, so DGS-HSA has a better privacy protection effect than L2P2.

Figure 8. The probability P_S that the user's real query content is recognized varies with k .

6.3.3. The Equilibrium Problem of Privacy Protection and System Overhead

Next, we discuss the equilibrium problem of privacy protection and system overhead for DGS-HSA. From the attacker's point of view, when the attacker receives a query request Req , $P_A = \frac{1}{s}$ and $P_S = P_A / \frac{k}{s} = \frac{1}{k}$. Given k , then P_S is also determined. Hence, we mainly discuss, when $k \in [2, 42]$, with the change of k , how the value of s can balance the privacy protection and system overhead. When we calculate system overhead, for simplicity, each second-level grid contains about 3000 historical locations.

After normalization, P_A and $C(G_{sub})$ can be represented in the same coordinate system. As shown in Figure 9, when $s = 7$, the two curves have an intersection, i.e., the proposed scheme achieves the balance between privacy protection and system overhead at the intersection. At this moment, the storage overhead and communication overhead generated by each submission of the query are about 1.6 Mb. Moreover, as k changes, the value of s changes as shown in Figure 10, which is the equilibrium point of privacy protection and system overhead. We can see from Figure 10 that when $k \geq 7$, the equilibrium is reached at $s = 7$; when $k < 7$, the local optimum is reached at $s = k$. Furthermore, the above results show that the presented scheme is feasible.

Figure 9. The balance of privacy protection level and system overhead.

Figure 10. The value of s that balances the privacy protection and system overhead varies with k .

7. Conclusions

In this paper, we point out two problems of the existing location privacy protection schemes. The continuous query request scenario for the same user in the same location and the hierarchical structure of the address corresponding to the location are not considered, which makes it difficult for the privacy protection scheme to resist statistical attacks and hard to achieve the ideal privacy protection effect. To protect location privacy and privacy preferences, and improve anti-attack capabilities, we present the privacy protection scheme of DGS-HSA. The scheme divides the historical location dataset into a grid with a two-level structure according to the hierarchical structure of the address. Then we select the dummy locations from the structured historical location dataset to achieve the privacy protection level $< L_{(k,s)}, Q_l >$. In addition, we formalize the equilibrium problem of privacy protection and system overhead. By solving the multi-objective optimization problem, we give the recommended values of the system parameters. The effectiveness of the presented scheme is evaluated from both theoretical and experimental aspects. The results show that the presented scheme achieves the expected goals.

Nevertheless, the effect of the presented scheme in this paper depends largely on whether the historical location dataset covers enough areas, especially special areas such as sparsely populated or inaccessible areas, which easily lead to anonymous failure due to an insufficient number of organizations. How to process, update, and distribute the historical location dataset more effectively is the focus of future research efforts.

Author Contributions: Conceptualization, M.L.; Data curation, Y.W., G.Y. and Y.X.; Formal analysis, M.L.; Funding acquisition, Y.X.; Investigation, M.L.; Methodology, M.L. and Y.Y.; Supervision, S.L.; Validation, S.L., H.Z. and Y.Y.; Writing—original draft, M.L.; Writing – review & editing, Y.W., G.Y., Y.C. and F.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Key R&D Program of China under Grant 2017YFB0802300, in part Major Scientific and Technological Special Project of Guizhou Province under Grant 20183001, in part Foundation of Guizhou Provincial Key Laboratory of Public Big Data under Grant 2018BDKFJJ021, and in part Research Project of Hechi University under Grant XJ2017ZD08 and XJ2016ZD007.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Demertzis, K.; Iliadis, L. A Computational Intelligence System Identifying Cyber-Attacks on Smart Energy Grids. *Mod. Discret. Math. Anal.* **2018**, *131*, 97–116.
- Demertzis, K.; Iliadis, L.; Spartalis, S. A spiking one-class anomaly detection framework for cyber-security on industrial control systems. In Proceedings of the International Conference on Engineering Applications of Neural Networks, Athens, Greece, 25–27 August 2017; pp. 122–134.
- Luo, S.F.; Seideman, J.D.; Dietrich, S. Fingerprinting Cryptographic Protocols with Key Exchange using an Entropy Measure. In Proceedings of the IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24 May 2018; pp. 170–179.
- Duessel, P.; Gehl, C.; Flegel, U.; Dietrich, S.; Meier, M. Detecting zero-day attacks using context-aware anomaly detection at the application-layer. *Int. J. Inf. Secur.* **2017**, *16*, 475–490. [\[CrossRef\]](#)
- IETF. RFC 5870. A Uniform Resource Identifier for Geographic Locations ('geo' URI). Available online: <https://datatracker.ietf.org/doc/rfc5870/> (accessed on 14 October 2015).
- W3C. Platform for Privacy Preferences (P3P) Project. Enabling Smarter Privacy Tools for the Web. Available online: <https://www.w3.org/P3P/> (accessed on 2 February 2018).
- Ghinita, G.; Kalnis, P.; Khoshgozaran, A.; Shahabi, C.; Tan, K.L. Private queries in location based services: Anonymizers are not necessary. In Proceedings of the 2008 ACM SIGMOD international conference on Management of Data, Vancouver, BC, Canada, 9–12 June 2008; pp. 121–132.
- Khoshgozaran, A.; Shahabi, C.; Shirani-Mehr, H. Location privacy: Going beyond K-anonymity, cloaking and anonymizers. *Knowl. Inf. Syst.* **2011**, *26*, 435–465. [\[CrossRef\]](#)
- Li, L.; Lu, R.; Huang, C. EPLQ: Efficient privacy-preserving location-based query over outsourced encrypted data. *IEEE Internet Things J.* **2015**, *3*, 206–218. [\[CrossRef\]](#)
- Zhu, H.; Liu, F.; Li, H. Efficient and privacy-preserving polygons spatial query framework for location-based services. *IEEE Internet Things J.* **2016**, *4*, 536–545. [\[CrossRef\]](#)
- Lu, R.; Lin, X.; Shi, Z.; Shao, J. PLAM: A privacy-preserving framework for local-area mobile social networks. In Proceedings of the IEEE INFOCOM 2014-IEEE Conference on Computer Communications, 27 April–2 May 2014, Toronto, ON, Canada.
- Beresford, A.R.; Stajano, F. Mix zones: User privacy in location-aware services. In Proceedings of the IEEE Annual Conference on Pervasive Computing and Communications Workshops, Budapest, Hungary, 24–28 March 2004; pp. 127–131.
- Sun, Y.M.; Chen, M.; Hu, L.; Qian, Y.F.; Hassan, M.M. ASA: Against statistical attacks for privacy-aware users in Location Based Service. *Future Gener. Comput. Syst.* **2017**, *70*, 48–58. [\[CrossRef\]](#)
- Gruteser, M.; Grunwald, D. Anonymous usage of location-based services through spatial and temporal cloaking. In Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, San Francisco, CA, USA, 5–8 May 2003; pp. 31–42.
- Gedik, B.; Liu, L. Location privacy in mobile systems: A personalized anonymization model. In Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05), Columbus, OH, USA, 6–10 June 2005; pp. 620–629.
- Gedik, B.; Liu, L. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Trans. Mob. Comput.* **2007**, *7*, 1–18. [\[CrossRef\]](#)
- Niu, B.; Li, Q.; Zhu, X.Y.; Cao, G.; Li, H. Achieving k-anonymity in privacy-aware location-based services. In Proceedings of the IEEE INFOCOM 2014-IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 754–762.
- Niu, B.; Zhu, X.; Li, W.H. A personalized two-tier cloaking scheme for privacy-aware location-based services. In Proceedings of the 2015 International Conference on Computing, Networking and Communications (ICNC), Garden Grove, CA, USA, 16–19 February 2015; pp. 94–98.
- Kido, H.; Yanagisawa, Y.; Satoh, T. An anonymous communication technique using dummies for location-based services. In Proceedings of the ICPS'05. Proceedings. International Conference on Pervasive Services, Santorini, Greece, 11–14 July 2005; pp. 88–97.
- Xu, T.; Cai, Y. Exploring historical location data for anonymity preservation in location-based services. In Proceedings of the IEEE INFOCOM 2008-The 27th Conference on Computer Communications, Phoenix, AZ, USA, 13–18 April 2008; pp. 547–555.

21. Suzuki, A.; Iwata, M.; Arase, Y.; Hara, T.; Xie, X.; Nishio, S. A user location anonymization method for location based services in a real environment. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 3–5 November 2010; pp. 398–401.
22. Kato, R.; Iwata, M.; Hara, T.; Suzuki, A. A dummy-based anonymization method based on user trajectory with pauses. In Proceedings of the 20th International Conference on Advances in Geographic Information Systems, Redondo Beach, CA, USA, 7–9 November 2012; pp. 249–258.
23. Hara, T.; Suzuki, A.; Iwata, M. Dummy-based user location anonymization under real-world constraints. *IEEE Access* **2016**, *4*, 673–687. [\[CrossRef\]](#)
24. Liu, H.; Li, X.; Li, H. Spatiotemporal correlation-aware dummy-based privacy protection scheme for location-based services. In Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017.
25. Hayashida, S.; Amagata, D.; Hara, T.; Xie, X. Dummy generation based on user-movement estimation for location privacy protection. *IEEE Access* **2018**, *6*, 22958–22969. [\[CrossRef\]](#)
26. Zhang, C.; Huang, Y. Cloaking locations for anonymous location based services: A hybrid approach. *GeoInformatica* **2009**, *13*, 159–182. [\[CrossRef\]](#)
27. Sun, G.; Liao, D.; Li, H.; Yu, H.F.; Chang, V. L2P2: A location-label based approach for privacy preserving in LBS. *Future Gener. Comput. Syst.* **2017**, *74*, 375–384. [\[CrossRef\]](#)
28. Beresford, A.R.; Stajano, F. Location privacy in pervasive computing. *IEEE Pervasive Comput.* **2003**, *1*, 46–55. [\[CrossRef\]](#)
29. Chow, C.Y.; Mokbel, M.F.; Liu, X. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems, Arlington, VA, USA, 10–11 November 2006; pp. 171–178.
30. Chow, C.Y.; Mokbel, M.F.; Liu, X. Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. *GeoInformatica* **2011**, *15*, 351–380. [\[CrossRef\]](#)
31. Yang, G.; Luo, S.; Zhu, H. An Efficient Approach for LBS Privacy Preservation in Mobile Social Networks. *Appl. Sci.* **2019**, *9*, 316. [\[CrossRef\]](#)
32. Li, J.; Wong, R.C.W.; Fu, A.W.C.; Pei, Jian. Achieving k-anonymity by clustering in attribute hierarchical structures. In *International Conference on Data Warehousing and Knowledge Discovery*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 405–416.
33. Andrés, M.E.; Bordenabe, N.E.; Chatzikokolakis, K.; Palamidessi, C. Geo-indistinguishability: Differential privacy for location-based systems. *arXiv* **2017**, *1984*, 901–914.
34. Gomory, R.E. Outline of an algorithm for integer solutions to linear programs. *Bull. Am. Math. Soc.* **1958**, *64*, 275–278. [\[CrossRef\]](#)
35. Mangasarian, O.L.; Recht, B. Probability of unique integer solution to a system of linear equations. *Eur. J. Oper. Res.* **2011**, *214*, 27–30. [\[CrossRef\]](#)
36. Zheng, Y. GeoLife GPS Trajectories 1.3. Available online: <https://www.microsoft.com/enus/download/confirmation.aspx?id=52367> (accessed on 13 November 2018).
37. Zheng, Y.; Zhang, L.; Xie, X.; Ma, W.Y. Mining interesting locations and travel sequences from GPS trajectories. In Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, 20–24 April 2009; pp. 791–800.
38. Zheng, Y.; Xie, X.; Ma, W.Y. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull* **2010**, *33*, 32–39.
39. Peking University Opens Research Data. Amap POI Data. Available online: <http://opendata.pku.edu.cn/dataset.xhtml?persistentId=doi:10.18170/DVN/WSXCNM> (accessed on 13 November 2018).