Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8892079, 16 pages
https://doi.org/10.1155/2020/8892079

Research Article

Location Privacy-Preserving Method Based on Historical Proximity Location

Xueying Guo,[1] Wenming Wang,[1,2] Haiping Huang,[1,3] Qi Li,[1,3] and Reza Malekian[4]

[1]*College of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China*
[2]*School of Computer and Information, Anqing Normal University, Anqing 246011, China*
[3]*High Technology Research Key Laboratory of Wireless Sensor Network of Jiangsu Province, Nanjing 210023, China*
[4]*Department of Computer Science and Media Technology, Malmö University, Malmö 20506, Sweden*

Correspondence should be addressed to Haiping Huang; hhp@njupt.edu.cn

Received 27 March 2020; Revised 15 June 2020; Accepted 24 June 2020; Published 18 July 2020

Academic Editor: Ximeng Liu

With the rapid development of Internet services, mobile communications, and IoT applications, Location-Based Service (LBS) has become an indispensable part in our daily life in recent years. However, when users benefit from LBSs, the collection and analysis of users' location data and trajectory information may jeopardize their privacy. To address this problem, a new privacy-preserving method based on historical proximity locations is proposed. The main idea of this approach is to substitute one existing historical adjacent location around the user for his/her current location and then submit the selected location to the LBS server. This method ensures that the user can obtain location-based services without submitting the real location information to the untrusted LBS server, which can improve the privacy-preserving level while reducing the calculation and communication overhead on the server side. Furthermore, our scheme can not only provide privacy preservation in snapshot queries but also protect trajectory privacy in continuous LBSs. Compared with other location privacy-preserving methods such as *k*-anonymity and dummy location, our scheme improves the quality of LBS and query efficiency while keeping a satisfactory privacy level.

1. Introduction

With the development of Internet services, mobile communications, and IoT applications, Location-Based Service (LBS) has become one of the popular electronic applications. Users carrying mobile devices loaded with location-based applications, such as Google Maps, Wechat, and Ctrip, are able to send query requests to location service providers (LSPs) and obtain the corresponding service data. With such applications, mobile users can easily obtain information about various Point of Interests (POIs) in the vicinity; for example, users can acquire the bus schedule, the nearest restaurant providing their favorite cuisine, and the recreational facilities from a nearby edge server.

However, since the LSP is potentially untrustworthy, and the submitted queries from users usually include some personal information, such as users' locations and the queried interests, the LSP can easily infer who are doing what in which place, which may jeopardize their privacy. For example, physical destinations such as medical clinics may indicate a person's health problems. Likewise, regularly staying at certain types of places may be linked directly to one's lifestyles or political associations. Although users may be informed of the policies regarding the collection and distribution of their location data, the execution of these policies is typically beyond the users' control and relies solely on the service providers. Therefore, the privacy of users has not been truly protected and requires further technical attention. Furthermore, LSPs usually need to process a large amount of location service request messages, and the overloaded calculations may cause LSPs to become busy resulting in denial of service.

To address the privacy issue, many technical schemes [1, 2] have been proposed in the literature over recent years. Most of them are based on location perturbation and obfuscation, which employ traditional privacy techniques such as *k*-anonymity [3, 4]. However, these solutions using *k*-anonymity have some inherent flaws. First, all mobile users, regardless of whether or not they request LBSs, need to

frequently report their latest locations to the anonymity server. In addition, users without LBSs may not be willing to spend their resources to help others maintain anonymity. Second, excessive location updates from a large number of mobile users also present overwhelming communication and processing bottlenecks on the server side. Third, in addition to the issues mentioned above, another problem is that the area of cloaking regions generated by the existing approaches is highly dependent on the network density. When a user lies in an unpopulated region, its cloaking area may be very large since it needs to contain the user itself and at least $k-1$ other users. Therefore, these traditional k-anonymity schemes cannot be directly applied to the protection of location privacy due to their inherent flaws.

Trajectory privacy preservation [4] is another challenge in LBSs for the vulnerability of the spatial and temporal information contained in the continuous queries received by the LSP, which may expose users' whereabouts and other private information. It is practically impossible to support anonymity for continuous LBSs using existing techniques such as GM's OnStar services [5]. Continuous LBSs require frequent location updates from their clients. Simply ensuring that each reported location belongs to a cloaking region containing at least k users cannot really achieve the client's k -anonymity protection, and it even significantly increases the computation and communication load of servers. Therefore, how to design a secure and efficient location privacy protection scheme is worth exploring especially in the continuous LBS scenario.

To address the above problems, we propose a new privacy-preserving method based on historical proximity locations. This method ensures that the user can obtain location-based services without submitting the real location information to the untrusted LBS server, which improves the location privacy level and reduces computation and communication load on the server side. In view of the aforementioned issues, the key contributions of this work are summarized as follows:

(1) In order to avoid the computational overhead of generating pseudolocations on the server side, this paper creatively proposes a scheme that substitutes one existing historical adjacent location around the user for his current location and then submits the selected location to the LBS server

(2) Historical proximity location query model is adopted to guarantee the location privacy of snapshot queries and continuous queries. In addition, our solution is more difficult for attackers to distinguish the user's true position from historical locations, and at the same time it cannot generate unreasonable positions

(3) Finally, compared with the existing schemes, performance analysis results show that our proposal can significantly improve the query efficiency while ensuring privacy protection

The remainder of this paper is organized as follows. Related work is reviewed in Section 2. The system model and the proposed privacy-preserving method are introduced in Section 3. Section 4 presents the experimental results, performance evaluation, and privacy analysis. Finally, we conclude this paper and present future work in Section 5.

2. Related Work

During the past decades, many promising approaches for preserving location privacy in LBSs have been proposed. We roughly divide them into two categories: centralized architecture and noncentralized architecture.

In centralized/edge anonymity server architecture, a centralized entity [6–9] is introduced into the system to protect the location privacy. Under this architecture, k-anonymity is the most popular means used for protecting users' privacy in LBSs. Gruteser and Grunwald [5] originally employed this concept in LBSs. As an extension of the traditional k-anonymity model [10–13], they proposed to reduce the accuracy of users' location information along spatial and/or temporal dimensions for a certain level of anonymity protection. However, all these centralized schemes share some drawbacks: (1) The anonymity server has all the knowledge about users' locations as well as queries, thus it becomes an attractive target for the adversary; so the user's real information will be jeopardized once it is attacked. (2) All users have to continuously send their queries and update their locations to the anonymity server, which causes the anonymizer to be a performance bottleneck and the potential central point of failure for the entire system.

In the noncentralized architectures, users cloak their locations without trusting a trusted third party (TTP). Some approaches, such as obfuscation-based methods [2, 14], cryptographic-based methods [15, 16], and collaboration-based methods [17–19], were proposed to protect the user's privacy. Obfuscation is achieved by adding noise, without revealing the exact location to the LBS servers. For example, Ardagna et al. [2] presented a solution aimed at preserving the location privacy of users by perturbing location information. The main drawback of obfuscation-based methods is that the quality of services (QoS) is degraded because of the low-level accuracy of the query answers. Cryptographic methods are also used to protect privacy data in the LBS; however, they are not practical for mobile devices since they require a powerful computational capability and incur large overhead on the client side. In collaboration-based methods, each user communicates with his peers and collects their location data to generate the cloaking region. The main idea is that, before sending a request to an LSP, the mobile user forms a group with his peers via single-hop communication or multihop routing and generates a cloaking area including k users. Shokri et al. [19] designed a distributed location privacy-preserving algorithm for a collaborative group, called MobiCrowd, which allows users to answer LBS queries from neighboring peers so that querying users can protect their location privacy from the LSP. These approaches focus mainly on snapshot queries, and the problem of protecting location privacy in the continuous LBSs is not considered in the TTP-free methods. With the rise of edge computing, Wang et al. [20] proposed an edge-based model for data collection, in which the raw data from wireless sensor

networks (WSNs) is differentially processed by algorithms on edge servers for privacy computing. To avoid potential information leakage and usage, the user's exact location should not be exposed to the edge node. Tian et al. [21] proposed a stochastic location privacy protection scheme for edge computing, in which the geographical distribution of surrounding users is obtained by analyzing the proposed long-term density map and short-term density map. This scheme is practicable for the real scenario when the edge computing server is honest but curious.

Furthermore, in a few privacy-preserving techniques, an attempt was made to use the TTP model for continuous LBSs [22–24]. Zhang et al. [22] proposed an algorithm for k-anonymity trajectory in LBSs, the main idea of which is to continuously expand an initial cloaked area to include at least the same k users. This means that while a request for an LBS is in progress, no grouped user who participated in the original anonymity set of the requestor is allowed to leave the group, since this action would jeopardize the privacy of the requestor. Xu and Cai [24] exploited historical locations to construct the k-anonymity trajectory and then presented algorithms for spatial cloaking. However, when a user moves on the cloaked path, the LBS can still easily identify the user's actual location if no other user exists on that path.

To address the above limitations, we propose a new privacy-preserving method based on historical proximity locations to protect location privacy in both snapshot queries and continuous queries.

3. System Overview

3.1. Preliminaries

Definition 1. The requested message Q submitted by the user to LSP can be expressed as a five tuple:

$$Q = \{id, loc, t, qry, r\}, \tag{1}$$

where id represents the user's identity information; $loc = \{lx, ly\}$ is the user's location, which can be directly obtained from a Global Positioning System (GPS) or using other positioning devices; t denotes the time at which the user sends the request; qry represents the query content the user wants to submit; and r represents the user's query radius, and naturally, the corresponding query area is πr^2.

Definition 2. $d_{\min}$ denotes the minimum distance allowed between the user's current location and the historical proximity location selected to be reported to the LSP. This limited distance prevents the selected historical proximity location from being too close to the user's current one to better protect the location privacy. Likewise, in order to guarantee the query quality, $d_{\max}$ represents the maximum distance between the user's current location and the selected historical proximity location.

Definition 3. W_{true} represents the set of POIs the user can obtain under ideal conditions; W is the set of POIs returned by LSP searching according to the user's submitted locations, query content, and query radius.

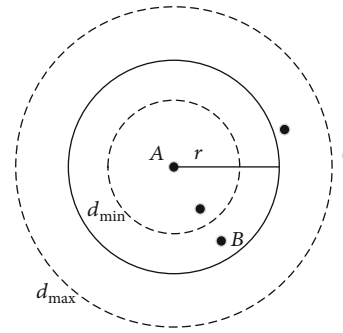


Figure 1: Location sampling phase.

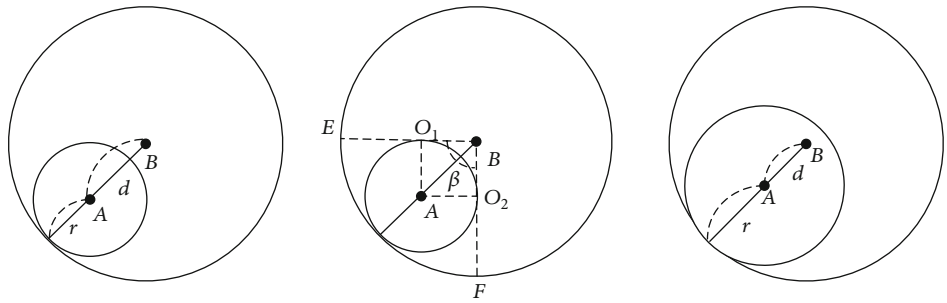
Definition 4. $P = W_{\text{true}}/W$, which represents the query quality, is the ratio of the number of POIs that the user can obtain under ideal conditions to that of POIs the user receives from the LSP.

3.2. Location Privacy Protection Model. Similar to existing work [25, 26], our system lets mobile users achieve LBSs through an anonymity server, which is considered as a TTP. However, the difference between our centralized architecture and the existing ones is that it can effectively reduce the computing and communication load based on the adopted privacy protection method.

A database that stores a large number of historical proximity locations is essential for the TTP providing privacy service in our model, and the specific characteristics of the database are given as follows:

(1) Initially, the database may be empty and the users can obtain the location service with k-anonymity protection, during which mobile users report their locations periodically to the TTP, and the k positions utilized in the anonymity process will be subsequently added to the database as historical proximity locations. Unlike existing techniques, such a periodic location update is no longer needed after the initial phase, which may last only a short time period. More location data can be obtained with more and more mobile users participating in the requests of LBSs

(2) After the initial phase, there are enough historical locations recorded in the database. As shown in Figure 1, suppose that a user is requesting location services at location A, if there are a certain number of historical proximity locations existing in the database that satisfy $d_{\min} < d < d_{\max}$, where d represents the distance between the historical location and the user's current location. In this case, the TTP will select the nearest historical location substituting the current location A and send it to the LSP. A will be subsequently added to the database as a historical location after the query process. However, if there are no historical proximity locations in the database



(a) $d > r$ (b) The actual query area when $d > r$ (c) The actual query area when $d < r$

Figure 2: The actual query area.

that satisfy $d_{\min} < d < d_{\max}$, the k-anonymity technique will be activated to provide privacy protection services for the user

Obviously, there will be a continuous increase in the number of historical proximity locations recorded in the database, and under this circumstance, the k-anonymity protection is no longer frequently needed.

Furthermore, for efficient retrieval of location data, we index the database using a simple grid-based approach. The entire domain is recursively partitioned into cells in a quadtree style. Unless a cell has been already at its minimal size (our implementation sets each cell to be at least 200×200 meter^2), it is split if the number of locations inside it exceeds a predetermined threshold. Thus, given a cell corresponding to the user's current location, we can effectively retrieve the location data and obtain historical proximity locations.

3.3. Privacy Preservation in Snapshot Queries

3.3.1. Query Area. As shown in Figure 2, the user is located at position A, the query radius is r, the nearest historical proximity location of point A is B, and d is the distance between A and B ($d_{\max} > d > d_{\min}$).

(1) As shown in Figure 2(a), when $d > r$ is satisfied, a circle is generated with point B as the center and $d + r$ as the radius. Draw two tangent lines (BE and BF) to the circle A via point B with O_1 and O_2 as the tangent points. Wherein, $\angle EBF = \beta$ (denoted in radians) is shown in Figure 2(b). To cover all the possible target positions, the fan EBF is enough as the effective query region, while the actual query region is the whole area of circle B and the area of the fan EBF can be computed as

$$S = S_{EBF} = \frac{\beta R^2}{2} = \frac{2 * \sin^{-1}(r/d)(d+r)^2}{2} = (d+r)^2 * \sin^{-1}\frac{r}{d}. \quad (2)$$

(2) As shown in Figure 2(c), when $d < r$ is satisfied, if the user wants to query all the target positions, we regard the entire circle which is centered on B as both of the effective query region and the actual query region, where $R = d + r$ is the radius, and the area of the query region is $S = \pi(d + r)^2$.

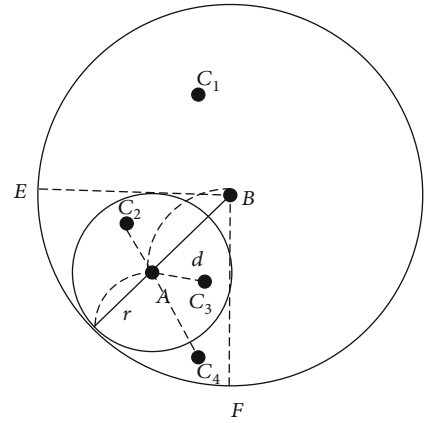


Figure 3: Filtering of query results.

3.3.2. Query Process and Filtering of Query Results. The LSP cannot directly search the irregular area such as the sector area mentioned above during the process of LBS. However, it is feasible to first filter the query results on the TTP side and then filter the results on the client side, which can efficiently reduce the overhead of mobile devices carried by the users. As shown in Figure 3, when $d > r$ ($d_{\max} > d > d_{\min}$) is satisfied, the user at location A, for example, is searching for gas stations nearby with the query radius r. The specific query and filtering process is as follows. Once receiving the request from the user located at A, the TTP will search the database and deliver the information of location B, which is selected carefully as a historical proximity position of A, to the LSP. And then the LSP will search the entire circle B with $d + r$ as the radius, i.e., the actual query region, for target positions meeting the request. After that, the messages related to the gas stations C_1, C_2, C_3, and C_4 will be returned to the TTP from the LSP as the results. Then, the TTP will filter out C_1 which is out of the user's query area. Finally, the user's mobile device will calculate the distance (d_2, d_3, and d_4) from location$A$to the remaining gas station candidatesC_2, C_3, and C_4, respectively, with the help of a map installed before. Compare each d_i with the query radius

r; if it is smaller than r, the corresponding information will be retained, otherwise it will be deleted, so C_4 will be filtered out as a result. Ultimately, the location information of gas stations C_2 and C_3 will be sent to the user.

When $d < r$ is satisfied, the process is similar, which is not repeated here.

It is worth noting that due to the indirect query method in our scheme, the error of the distance d between A and B will also lead to the error of the actual query radius $d + r$, which may result in the actual query area being too large or too small, possibly accompanied with a declined quality of services.

3.4. Privacy Preservation in Continuous Queries. Existing techniques mostly focus on snapshot queries. However, privacy preservation in continuous LBSs is more challenging than that in snapshot queries because adversaries could use the spatial and temporal correlations on the user trajectory to infer the user's private information. To deal with the concern, a privacy-preserving method for continuous LBSs based on historical adjacent locations is described in this section.

3.4.1. Average Query Error. As mentioned above, in snapshot queries, the error of the distance d between the user's current location and the reported location, which is selected from the historical proximity locations in the database by the TTP, may bring about a decline in the quality of queries. Similarly, it is the same in privacy preservation scenarios of continuous LBSs queries. We give a formal definition of the average error degree in continuous LBSs as follows.where d_i is the distance between A_i and B_i, $d_{\max} > d_i > d_{\min}$.

Definition 5. Given a trajectory $T = \{A_0, A_1, \cdots, A_n\}$, which is generated by the user over a period of time, where $A_i$represents the user location at the time point$i$; in response, the TTP will compute a new trajectory $T' = \{B_0, B_1, \cdots, B_n\}$ based on T using historical proximity locations, where B_i represents the historical proximity location of A_i at the time point i. The average query error can be defined as

$$\bar{d} = \frac{\sum_{i=0}^{n} d_i}{n}, \tag{3}$$

Obviously, the smaller $\bar{d}$ is, the smaller the error degree will be. For the quality of queries, $\bar{d}$ in the query process needs to be as small as possible. Therefore, in the process of the user's moving on the trajectory, it is better to select the nearest historical proximity location B_i substituting the corresponding A_i when sending it to the LSP.

3.4.2. Trajectory Privacy-Preserving Algorithm. If there are enough historical proximity locations around the user's trajectory T, it is easy to find the corresponding B_i for each A_i, and B_i will not coincide with any B_j, where $0 \le i \le j \le n$.

However, if the historical proximity locations around the trajectory T are sparse, there is a certain possibility that B_i and B_j coincide with each other. As shown in Figure 4, the directed lines denote a trajectory formed by the user over a period of time, and the solid nodes nearby denote the existing historical proximity locations. Since B_1 is both the nearest historical proximity location of A_0 and that of A_1, when the user is at the 0th time point and the 1st time point, B_1 will be selected and sent to the LSP for query results on behalf of A_0 as well as A_1, resulting in the same selection of historical proximity locations at different time points, i.e., $B_0 = B_1$. In this case, once the LSP receives the same location B_i at different time points, it will be easy to infer that the user is wandering in the vicinity of B_i during this period of time, which actually leaks the user's privacy.

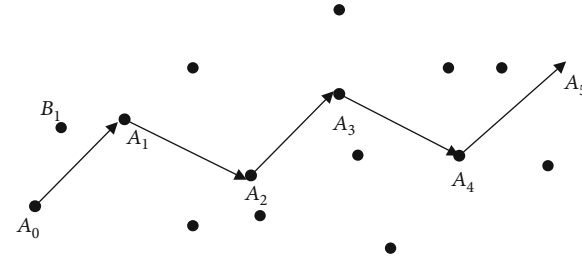


FIGURE 4: B_1 is selected as the historical location for both A_0 and A_1.

To solve this problem, we make further constraints and give Definition 6.

Definition 6. Given a trajectory $T = \{A_0, A_1, \cdots, A_n\}$, which is generated by the user over a period of time, where A_i represents the location of the user at the time point i; there is a new trajectory $T' = \{B_0, B_1, \cdots, B_n\}$ as the historical proximity trajectory (HPT) of T, where $0 \le i < j \le n$, $B_i \ne B_j$, andB_i represents the corresponding historical proximity location selected for location A_i.

Therefore, aiming at the problem for the privacy preservation of continuous LBSs, the key to our solution is how to find the corresponding historical proximity trajectory (HPT) T' for the user's trajectory T while guaranteeing the minimum value of $\bar{d}$ on the premise of satisfying both Definition 5 and Definition 6. The following is the specific solution description for this problem.

Given a trajectory $T = \{A_0, A_1, \cdots, A_n\}$ of the user, let $T'' = \{C_0, C_1, \cdots, C_m\}$ be an ordered set of historical proximity locations along the direction of trajectory T, satisfying $d_{\max} > d_k > d_{\min}$ (d_k is the distance from A_k to any location C_{k+i} among $C_k \sim C_{k+m-n}$) and $m \ge n$, where m denotes the number of historical proximity locations around the trajectory T, and n represents the number of locations the user left on trajectory T. Then, the minimum sum of error degree between the historical proximity trajectory T' and the user's trajectory T is defined as $D(n, m) = n * \bar{d}$. If C_m is selected as the historical proximity location of A_n and sent to the TTP, then the solution for getting the minimum value of $D(n-1, m-1)$ is certainly contained in the solution for getting that of $D(n, m)$. If C_m is not selected to substitute A_n, then the optimal solution for getting the minimum value of $D(n, m)$ is bound to contain the solution

Input: T, T''
Output: Array B, is used to record the locations selected from T'' and reported to LSP as historical proximity locations of user's trajectory T
for $i = 0$ to n do
  D[i][i] = D[$i-1$][$i-1$] + d[i][i]
  B[i] = i
for $i = 0$ to n do
  for $j = i + 1$ to m
  if($D[i-1][j-1] + d[i][j] > D[i][j-1]$)
    $D[i][j] = D[i][j-1]$
    $B[i] = j - 1$
  else
  $D[i][j] = D[i-1][j-1] + d[i]d[j]$
  $B[i] = j$
return B

Algorithm 1: selectHistoryLocation(T, T'').

for getting that of $D(n, m-1)$. Therefore, the recursive relationship can be denoted as follows:

$$D[n][m] = \begin{cases} \sum_{i=0}^{n} d[n][m], & n = m, \\ \min\{D[n-1][m-1] + [n][m], D[n][m-1]\}, & n < m, \end{cases} \quad (4)$$

where $d[n][m]$ represents the distance between A_n and B_m.

The pseudocode of the above procedure is given in Algorithm 1.

In Algorithm 1, array B holds the subscripts of the selected locations on T'', and the complexity of the algorithm is $O(n^3)$. Algorithm 2 is used to get the historical proximity trajectory T' that guarantees the minimum value of $\bar{d}$.

Besides, there is still a special situation needing a discussion. It is likely that the quantity of the historical proximity locations recorded in the database is not enough for the algorithm we proposed. As is shown in Figure 5, when m is much smaller than n, no matter how it is selected, it will occur that one historical proximity location is selected two or more times on the user's trajectory. Considering the peculiarity of this problem, we propose to employ a symmetry mechanism to generate dummy locations in our scheme, and the specific procedure is described as follows.

As is shown in Figure 6, when there are no other historical locations available except one existing historical proximity location B_k (for example B_1) of location A_i (for example A_1), it connects B_k to A_i and extends the connecting line to point V_j (for example V_1), making $B_kA_i = V_jA_i$, where V_j is the dummy location generated as a historical adjacent location of A_i by symmetry. However, it is possible that the dummy location generated by symmetry is unreasonable (for example, the dummy location is in a lake), so some adjustment is necessary. As shown in Figure 7, suppose that the generated dummy location V_1 is unreasonable, and the TTP will rotate V_1 and adjust the distance from V_1 to A_1 to make it meet the rationality requirements, and finally a reasonable dummy location V_1' will be obtained as the historical proximity location of A_1.

Input: T''
Output: T'
  for $i = 0$ to n do
  $T'[i] = T''[B[i]]$
return T'

Algorithm 2: getHistoryTrack(T').

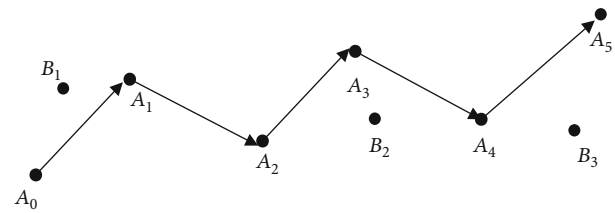


Figure 5: Sparse historical proximity locations.

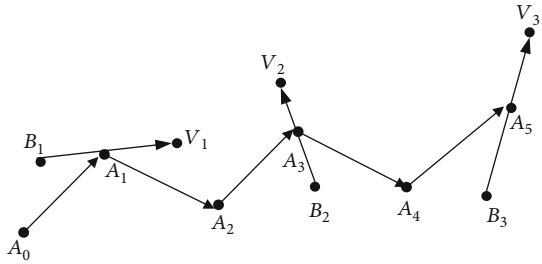


Figure 6: Generating dummy locations by symmetry.

Besides, there still exists a small possibility of $n > 2m$, in this case the number of historical proximity locations is smaller than n (the number of locations on trajectory T), even if the number of historical locations is expanded from m to $2m$ with the aid of the symmetry mechanism. To deal with this issue, we can activate the k-anonymity

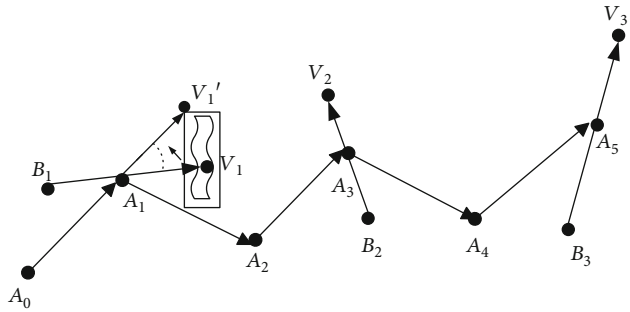


Figure 7: Generating dummy locations by symmetry and rotation.

technology to protect location privacy and add the user's locations to the database.

4. Experiment and Analysis

In this section, the experimental evaluation of the feasibility and efficiency of our proposed method under various parameter settings will be presented. Firstly, we analyze the influence of several parameters on the average query error $\bar{d}$. Secondly, we compare our method with other location privacy-preserving techniques in terms of query efficiency, query quality, and anonymity degree. The experimental region is within 10 square kilometers of the Sanpailou Campus of Nanjing University of Posts and Telecommunications. The data utilized in the experiments are captured by the coordinate pickup tool provided by Google. Our experiments are implemented with the Java Development Kit- (JDK-) 1.7 and Eclipse Integrated Development Environment (IDE), running on a local machine with an Intel Core-i5 2.8 GHz, 8 GB RAM, and Microsoft Windows 7 OS.

4.1. Influence Factors of the Average Query Error $\bar{d}$. Within the range of the experimental region, 10 coordinate points are generated randomly to construct a trajectory T of the user, i.e., let $n = 10$. And then 20~40 locations from the database are selected as historical adjacent positions around the user's trajectory generated before.

$d_{\min}$ is set by the user, and a smaller $d_{\min}$ has more probability to be taken to ensure the quality of services in a densely populated area; on the contrary, in a sparsely populated area, a larger $d_{\min}$ means better privacy level. As shown in Figure 8, it can be seen that $\bar{d}$ increases with the increase of $d_{\min}$: when selecting historical adjacent locations, it is necessary to consider whether the distance d from the user to the historical adjacent location is larger than $d_{\min}$, so as to exclude some positions that are too close to the user. The more there are historical proximity locations, the smaller$d$will be, and this results in a smaller average error degree$\bar{d}$. Besides, $\bar{d}$ approaches $d_{\min}$ infinitely as m approaches infinity.

$d_{\max}$ is also set by the user, and usually it cannot be set too small. Since $d_{\max}$ is the maximum distance between the user's current location and the historical proximity location reported to the LSP, a $d_{\max}$ that is too small will filter out most historical adjacent locations, reducing the privacy protection level. As shown in Figure 9, when m takes the value of 50 and 100, respectively, $\bar{d}$ increases as $d_{\max}$ grows in the initial phase. This is because when m and $d_{\max}$ are both small, the number of the screened historical locations m' is smaller than n, and dummy locations will be generated as historical adjacent locations by the symmetry mechanism. Therefore, there is more probability of selecting the nearest historical locations, leading to a smaller $\bar{d}$. However, when $d_{\max}$ gradually grows, m' will also increase as the screening conditions for historical locations are relaxed, so the number of historical locations generated by the symmetry mechanism will decrease, accompanied with an increase of $\bar{d}$. Until there is no need for generating symmetrical historical locations, $d_{\max}$ will no longer affect the historical locations selected. When $m = 300$ and $d_{\max} = 100$, the n locations closest to the trajectory T selected from the m historical points are not screened out, so $\bar{d}$ remains constant as $d_{\max}$ increases.

We have discussed the influence of historical adjacent location parameter selection on $\bar{d}$. The experimental results clearly show the specific effects of different values of $d_{\min}$ and $d_{\max}$ on $\bar{d}$. Therefore, in practical application scenarios, the values of parameters $d_{\min}$ and $d_{\max}$ should be selected according to specific requirements and allowable errors.

4.2. Performance Comparison

4.2.1. Performance Comparison under Snapshot Queries. In our experiments, coordinate data of 500 target positions such as hotels, hospitals, and gas stations were captured, and 5000 coordinate points were randomly selected as historical adjacent positions as well as other users' positions required when using k-anonymity and were stored in the database.

(1) Query Efficiency. The query efficiency is usually synthetically evaluated with the total time cost that contains TTP spending on generating the actual query area and the LSP spending on replying to the requested query. As shown in Figure 10(a), the Casper scheme in [25] and the anonymous-zone merging scheme in [17] generate the query region using k-anonymity, so the query domain generation time is the sum of the time of the database searching other $k - 1$ users around and that of constructing the anonymous domain containing k users. However, in most cases, the query domain generation time of our solution is almost the time to search for historical adjacent locations in the database, which has nothing to do with k. Therefore, the time to generate the query region in our scheme is relatively less and does not increase linearly with the increase of k. As shown in Figure 10(b), when the same query radius $r = 300$ m is taken and d is set to 75 m, the query region in our scheme is independent of k and its area does not exceed $\pi (d + r)^2$; the area of query region generated by the Casper scheme is theoretically no less than $k\pi r^2$, which is larger than those of our scheme and the anonymous-zone merging scheme; besides, the query areas of the two compared schemes increase significantly with the increase of k. Sufficient historical locations will ensure a smaller d in our scheme, and thus guarantee a smaller query area. Figure 10(c) illustrates that the query processing time is

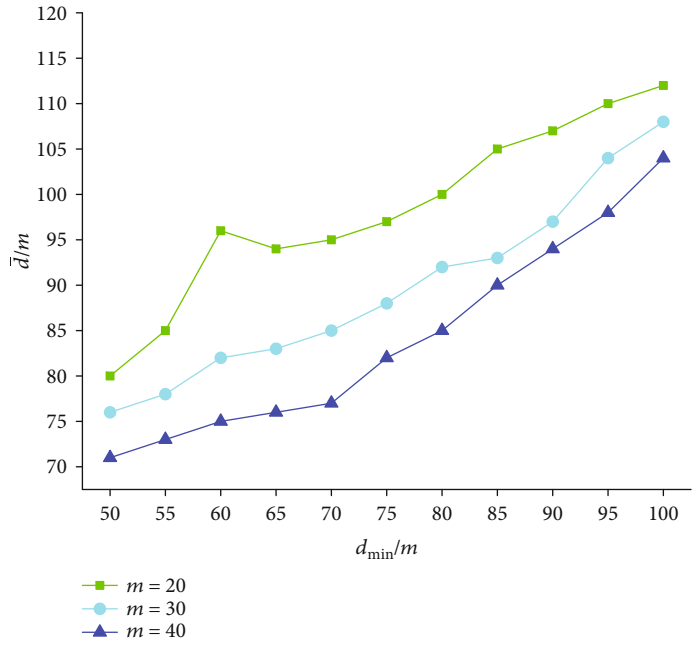


Figure 8: Impact of m and d_{min} on $\bar{d}$.

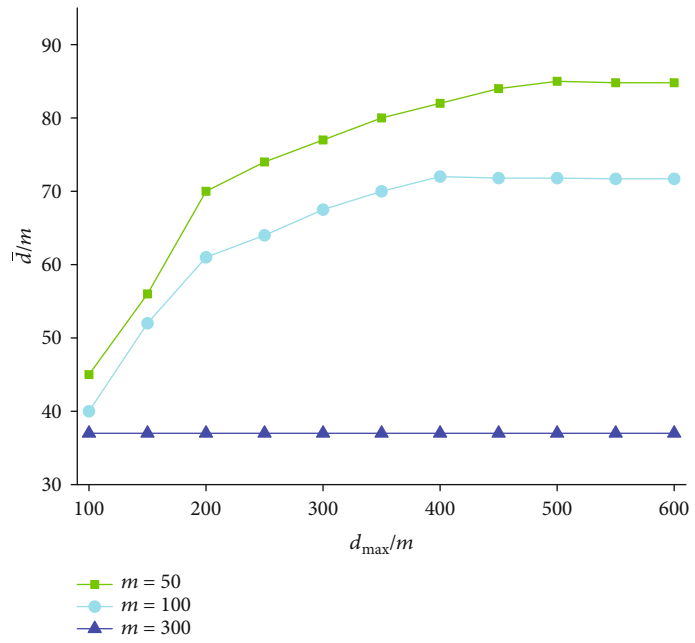
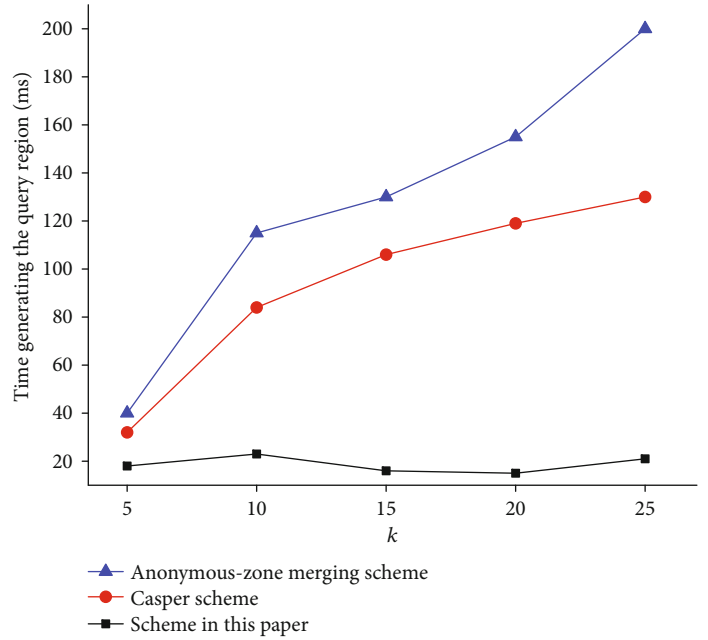


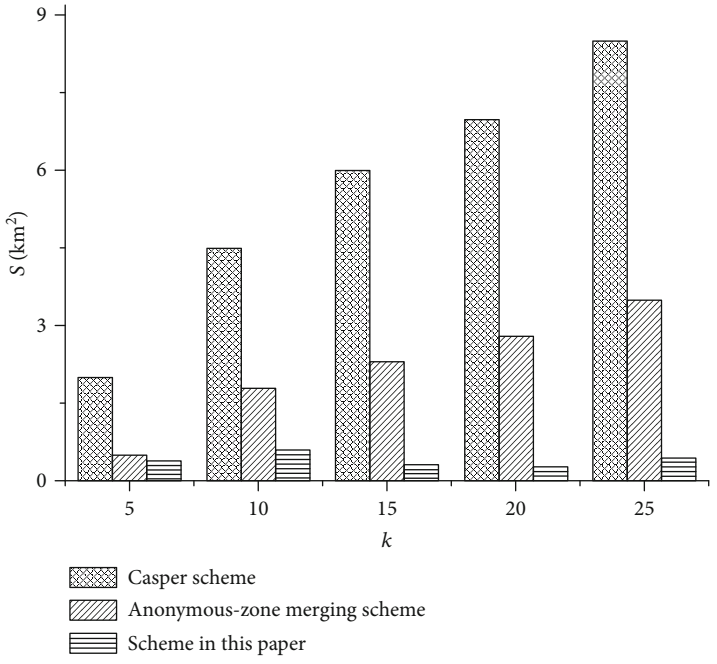
Figure 9: Impact of m and d_{max} on $\bar{d}$.

positively correlated with the query area. In order to facilitate the comparison between our scheme and the other two k -anonymity schemes, a smaller anonymity degree $k = 5$ is taken and d is set to 75 m. As shown in Figure 10(d), both the query area and the query processing time gradually increase when the radius grows. However, compared with the Casper scheme and the anonymous-zone merging scheme, the query area generated by our scheme is relatively small, which results in a shorter query processing time.

(2) Query Quality. Evaluation of the query quality is based on the ratio P of the number of POIs that the user can obtain in theory to that of positions returned by the LSP when the user requests with the query radius r, i.e., $P = W_{true}/W$, as explained in the previous study. In the experiment, we randomly select 20 points as the positions where the user can send the query. We vary the value of r, repeat the experiment 20 times, and then take the average value of P as the analysis object. Furthermore, we also compare our scheme with the



(a)



(b)

Figure 10: Continued.

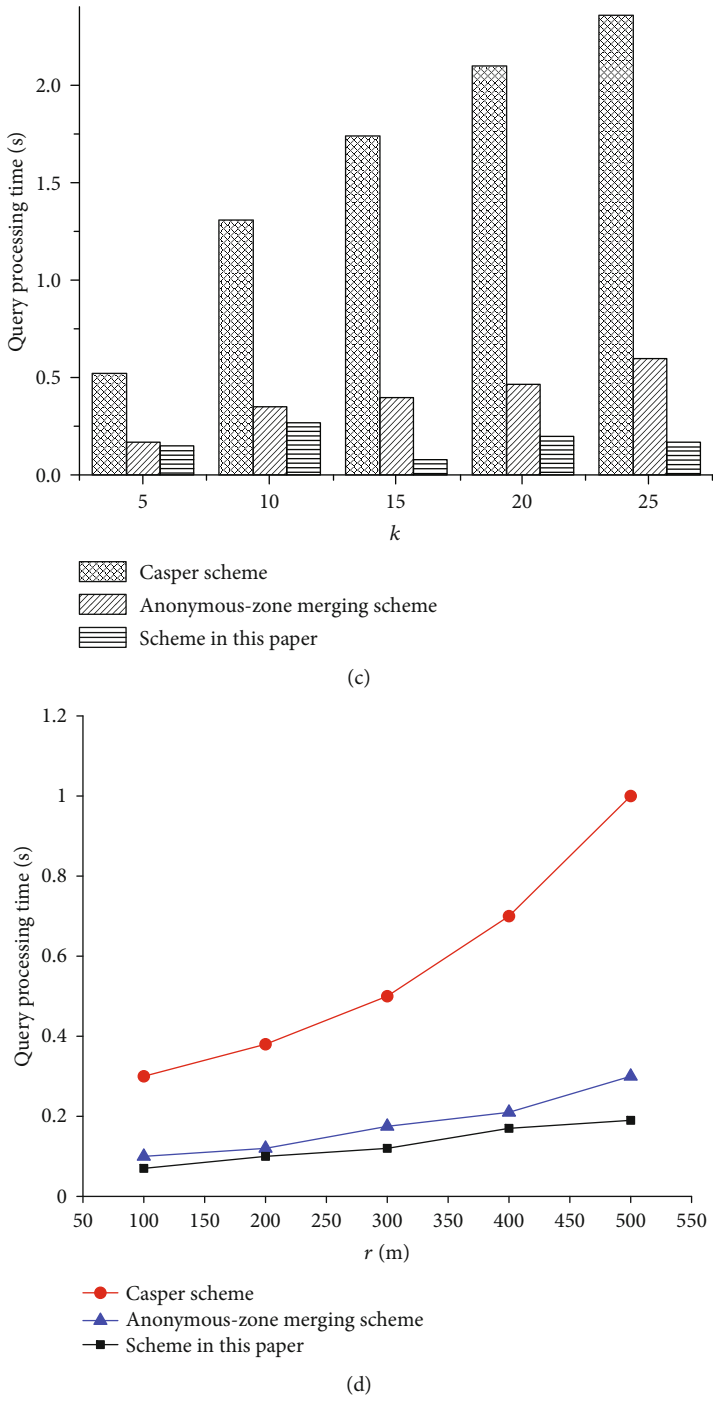


FIGURE 10: Query efficiency comparison. (a) Time comparison of generating the query area. (b) Query area comparison. (c) Query processing time comparison. (d) Impact of r on query processing time.

enhanced pseudonym selection scheme in [26] besides the other two schemes mentioned before. As shown in Figure 11, our scheme maintains satisfactory query quality and stability with the increase of the query radius. In contrast, as for the Casper scheme and the anonymous-zone merging scheme, the query area increases significantly as r becomes larger, which indicates that a great number of POIs cross the user's query area, resulting in the decline of query quality. In addition, since the enhanced pseudonym selection scheme cannot flexibly adjust the query region to cover all the target positions, it is difficult for it to guarantee high query quality.

Experimental results show that, our scheme can effectively improve the query efficiency while guaranteeing satisfactory query quality in snapshot queries.

4.2.2. Performance Comparison under Continuous Queries. In the experiment, we select $\bar{d}$, defined as the average query

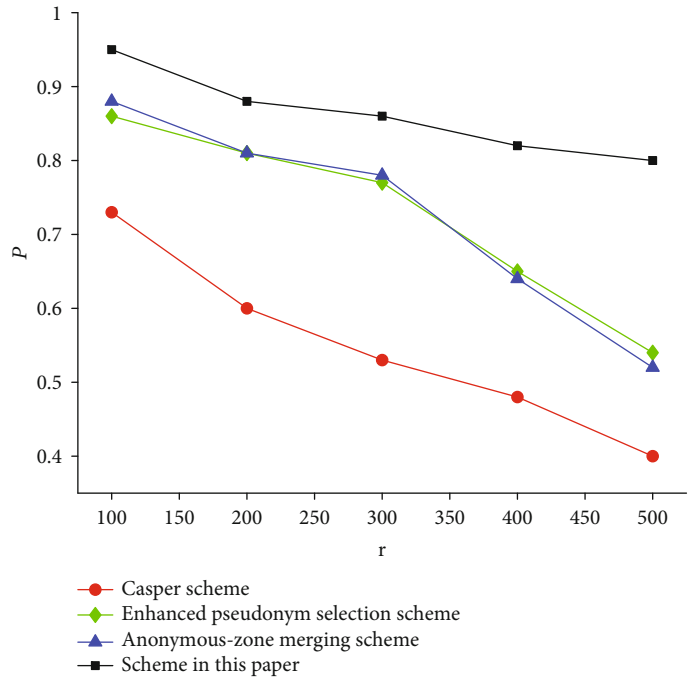


Figure 11: Query quality comparison.

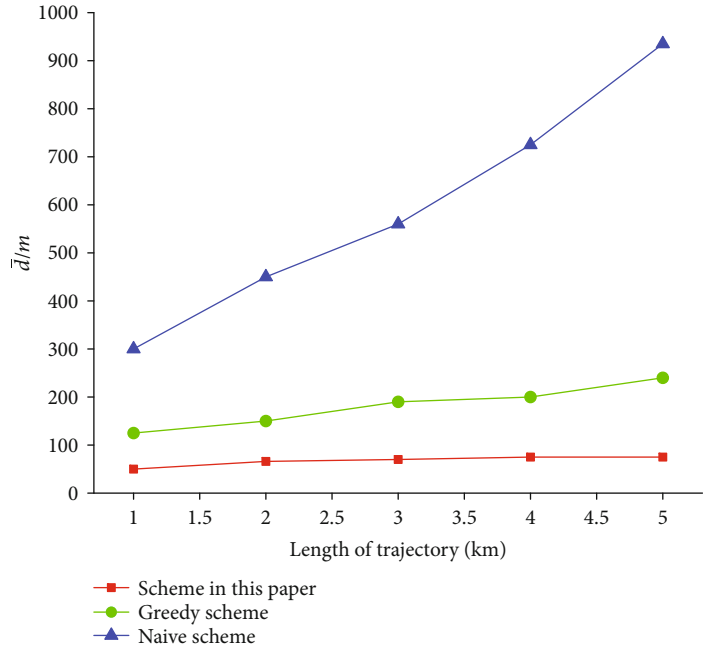


Figure 12: Impact of the length of trajectory on $\bar{d}$.

error for a user's trajectory in continuous LBSs, as our performance evaluation metric. And obviously, the smaller $\bar{d}$ is, the better the quality of services will be in continuous LBSs. We compare our scheme with two other existing schemes, the Native scheme in [22] and the Greedy scheme in [24], which are both extensions of the k-anonymity method. Within the experimental region, the length of the user's trajectory was set to 1-5 km, and 500-2500 coordinate points were captured within the radius of 200 m around the trajectory as historical proximity locations and added to the database.

We set $m = 1000$, and the experiment results are shown in Figure 12. In the Native scheme, the cloaking area will become increasingly large since the traditional trajectory k-anonymity method expands an initial cloaking region to cover at least the same k users who may move in different directions, resulting in a sharp increase of $\bar{d}$. Moreover, the

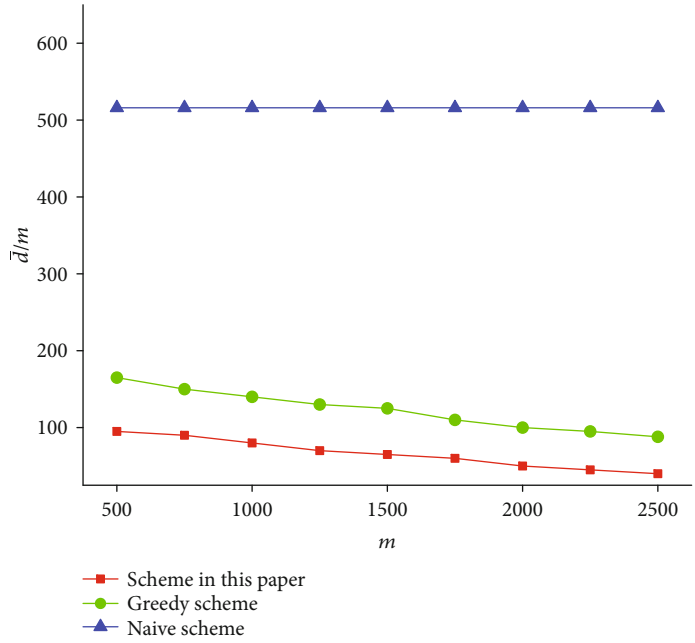


Figure 13: Impact of m on $\bar{d}$.

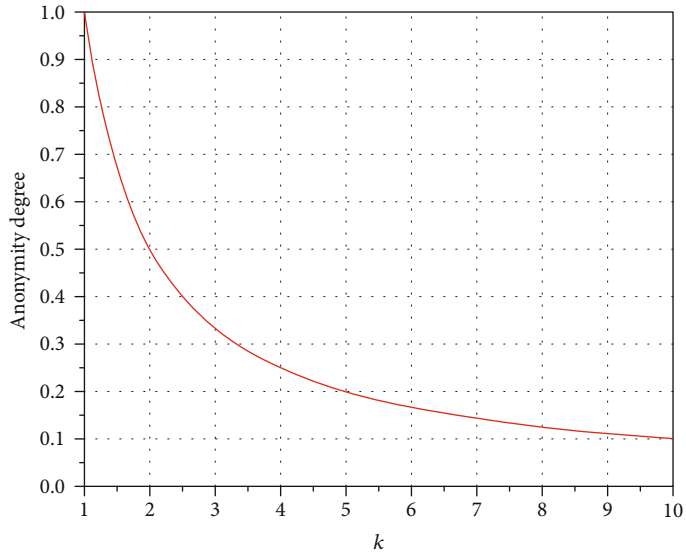


Figure 14: Anonymity degree about k.

query sequence is consistent with the user's movement direction, which may provide some valuable information for the adversary to infer the user's trajectory. In the Greedy scheme, the Greedy algorithm is utilized to verify that each node on the candidate $k-1$ trajectories is as close to the user's trajectory as possible; however, since a complete historical trajectory will be finally selected from the $k-1$ candidates, it cannot guarantee that each position on the selected historical trajectory is the nearest point for each node on the user's real trajectory.

The following is the analysis of the impact of m (the number of historical proximity locations) on $\bar{d}$ of the three schemes, and the length of the user's trajectory is set to 3 km. As shown in Figure 13, for the Native scheme, m has no effect on $\bar{d}$ since the generated cloaking area is only relevant to the current locations of the other users. However, $\bar{d}$ declines with the increase of$m$for the Greedy scheme and our scheme, since historical trajectories and historical proximity positions are utilized in the two schemes, respectively.

4.3. Privacy Analysis. In this section, we will evaluate the privacy degree of our solution by comparing it with the k-anonymity and dummy location technology.

In the process of k-anonymity protection, the LSP receives the locations of$k$users involved with the service requestor, so the probability of identifying the user's real location is $1/k$. As shown in Figure 14, the larger the value

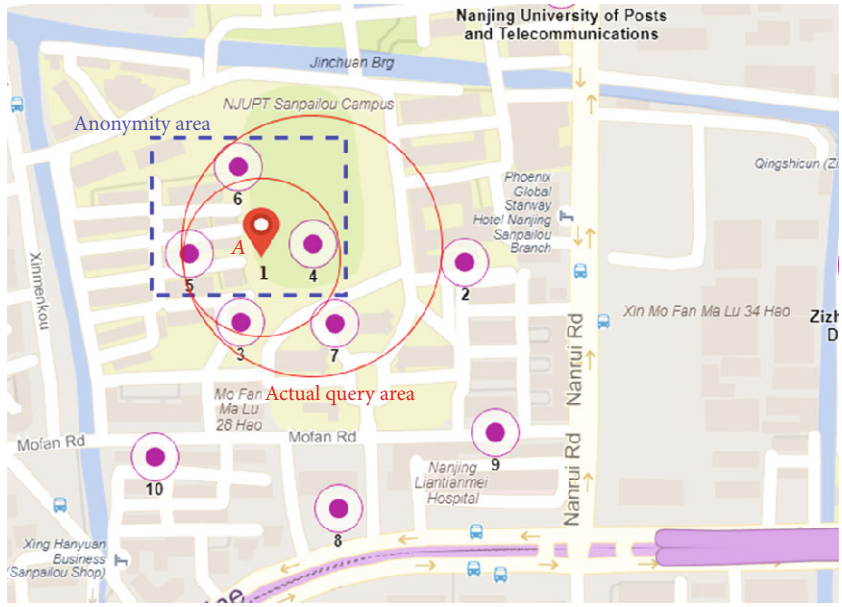
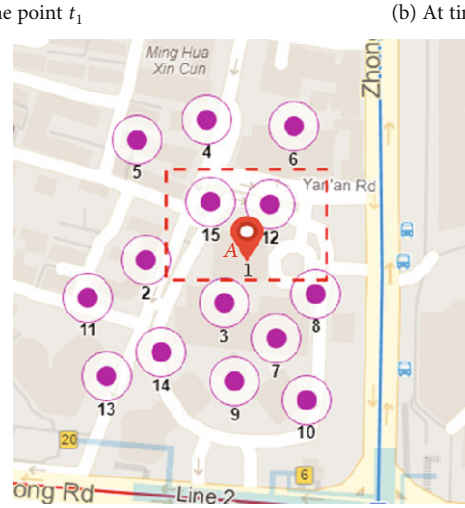


FIGURE 15: Privacy analysis in snapshot queries.

(a) At time point t_1

(b) At time point t_2



(c) At time point t_3

FIGURE 16: Privacy analysis in continuous queries.

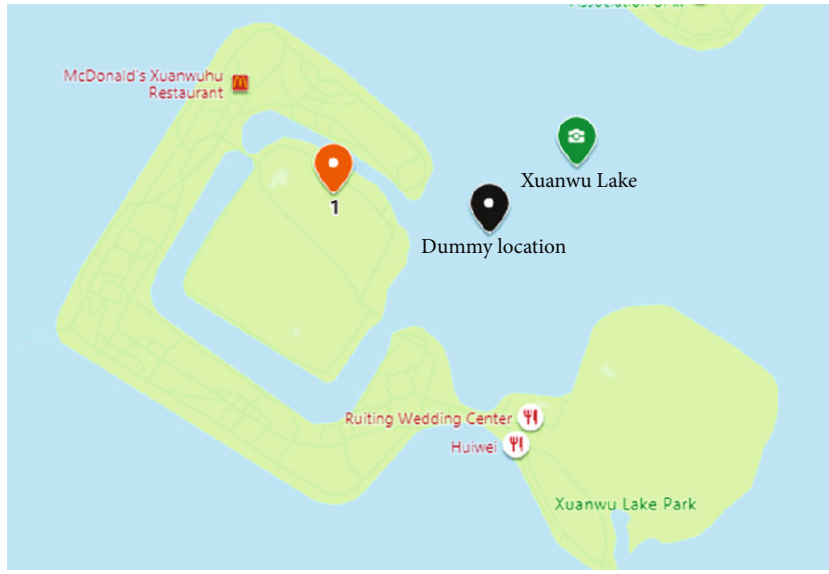


Figure 17: Unreasonable locations generated by dummy location.

of k, the higher the privacy degree will be; however, it will also lead to a decrease in query efficiency and quality. As shown in Figure 15, a user A, located at position 1 in the Sanpailou Campus of Nanjing University of Posts and Telecommunications, wants to request the location service together with 9 other users in the vicinity who also request services. Suppose that user A obtains location service through k-anonymity with a cloaking area containing users in positions 4, 5, and 6, the probability that the adversary recognizes user U will be 1/4. However, if user A adopts the scheme as described in this paper, point 4 will be treated as a historical adjacent position to be queried. By the description of the proposed scheme, the actual query area covers a total of 6 points including user A and points 1, 3, 4, 5, 6, and 7, which is denoted by the big red circle in Figure 15. Therefore, the probability of identifying user A is only 1/6.

The advantages of our proposal will become more obvious in continuous queries under densely populated areas. We take Xinjiekou, the commercial center of Nanjing City, as an example. As shown in Figure 16, user A sends a service request with $k = 3$, the anonymous set of which is $\{1, 2, 11\}$ at the initial time t_1; while that updates to $\{1, 2, 6\}$ and $\{1,12,15\}$ separately at t_2 and t_3. At each moment, the probability of identifying user A is 1/3. However, if an attacker obtains the user's anonymous sets at the three moments and then performs an intersection operation, then the true identity of A can be obtained. In our proposed scheme, point 2 is selected as a historical adjacent position at time t_1. At time t_2, considering that point 2 remains closest to A and to block the attacker from speculating that$A$is located near point 2, we chose point 7 as the historical adjacent position according to the historical proximity selection method described in the scheme ($B_i \neq B_j$). Due to population density, there are more historical adjacent locations around the user, so the possibility of the users' real identity being exposed is lower.

In addition, it is more reasonable to utilize historical proximity locations instead of the randomly generated locations by the traditional dummy location technology. As shown in Figure 17, when the user requests LBSs at position 1 with the dummy location technology, it is possible that the pseudonym-location generated is in the lake. In this case the adversary can easily identify it with background knowledge and filter out other unreasonable locations, which definitely decreases the privacy degree. Fortunately, using the historical proximity locations proposed in this paper can address this problem.

Compared with most of the existing methods such as k-anonymity and dummy location, which have to report the user's true position to the service providers, our approach submits the historical proximity location to substitute the user's current location, which improves the privacy level.

5. Conclusion

This paper proposed a solution for location privacy protection in both snapshot queries and continuous queries. With historical proximity locations around the user submitted to the LSP for query instead of the true location, the user's location information is completely anonymous from the LSP in the whole process of requesting the LBSs, and thus a high privacy level is achieved. Compared with k-anonymity and the enhanced pseudonym selection scheme, our scheme is simple and feasible, and can achieve better query efficiency and higher query quality. Our proposal also provides a new solution for dealing with the problem of maintaining the equilibrium among the privacy level, query efficiency, and quality of services. However, in continuous queries, we have not succeeded in achieving sufficient anonymity protection level for a user's movement trajectory. It is still possible for an attacker to infer the general direction of the user's movement by analyzing the changes of the user's query range recorded in the LSP. This is a difficulty in the current techniques of trajectory privacy preservation, and it is also the focus of future research work.

Data Availability

No data were used to support this study.

Disclosure

X. Guo and W. Wang are co-first authors.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Authors' Contributions

X. Guo and W. Wang contributed equally to this work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant number 61672297), the Key Research and Development Program of Jiangsu Province (grant number BE2017742), the Postgraduate Research & Practice Innovation Program of Jiangsu Province (grant number KYCX19_0908), and the Key Project on Anhui Provincial Natural Science Study by Colleges and Universities (grant numbers KJ2019A0579 and KJ2019A0554).

References

[1] C. Yin, J. Xi, R. Sun, and J. Wang, "Location privacy protection based on differential privacy strategy for big data in industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3628–3636, 2017.

[2] C. Ardagna, M. Cremonini, S. D. C. di Vimercati, and P. Samarati, "An obfuscation-based approach for protecting location privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 1, pp. 13–27, 2011.

[3] S. Zhang, X. Li, Z. Tan, T. Peng, and G. Wang, "A caching and spatial K-anonymity driven privacy enhancement scheme in continuous location-based services," *Future Generation Computer Systems*, vol. 94, pp. 40–50, 2019.

[4] Y. Dong and D. Pi, "Novel privacy-preserving algorithm based on frequent path for trajectory data publishing," *Knowledge-Based Systems*, vol. 148, pp. 55–65, 2018.

[5] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services*, pp. 31–42, 2003.

[6] Y. Zhang and Q. Zhang, "A *k*-anonymous location privacy protection method of dummy based on geographical semantics," *International Journal of Network Security*, vol. 21, no. 6, pp. 937–946, 2019.

[7] J. Wang, J. Luo, X. Liu, Y. Li, and S. Liu, "Improved Kalman filter based differentially private streaming data release in cognitive computing," *Future Generation Computer Systems*, vol. 98, pp. 541–549, 2019.

[8] T. Nakagawa and H. Arai, "Personalized anonymization for set-valued data by partial suppression," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 1003–1010, IEEE, 2017.

[9] G. Ghinita, K. Zhao, D. Papadias, and P. Kalnis, "A reciprocal framework for spatial *K*-anonymity," *Information Systems*, vol. 35, no. 3, pp. 299–314, 2010.

[10] W. Zhang, M. Li, R. Tandon, and H. Li, "Online location trace privacy: an information theoretic approach," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 235–250, 2019.

[11] G. Li, L. Li, J. Li, and Y. Li, "Network Voronoi diagram on uncertain objects for nearest neighbor queries," *Information Sciences*, vol. 301, pp. 241–261, 2015.

[12] S. Tang, S. Liu, X. Huang, and Z. Liu, "Privacy-preserving location-based service protocols with flexible access," *International Journal of Computational Science and Engineering*, vol. 20, no. 3, pp. 412–423, 2019.

[13] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "Adaptive and channel-aware detection of selective forwarding attacks in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 5, pp. 3718–3731, 2016.

[14] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *International conference on pervasive computing*, pp. 152–170, Springer, Berlin, Heidelberg, 2005.

[15] X. Li, E. Wang, W. Yang, and J. Ma, "DALP: a demand-aware location privacy protection scheme in continuous location-based services," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 4, pp. 1219–1236, 2016.

[16] J. Zhang, X. Zhao, and C. Ji, "A novel authenticated encryption scheme and its extension," *Information Sciences*, vol. 317, pp. 196–201, 2015.

[17] T. Peng, Q. Liu, D. Meng, and G. Wang, "Collaborative trajectory privacy preserving scheme in location-based services," *Information Sciences*, vol. 387, pp. 165–179, 2017.

[18] B. Wang, L. Zhang, and G. Zhang, "A fine granularity based user collaboration algorithm for location privacy protection," *PloS one*, vol. 14, no. 7, p. e0220278, 2019.

[19] R. Shokri, G. Theodorakopoulos, P. Papadimitratos, E. Kazemi, and J.-P. Hubaux, "Hiding in the mobile crowd: location privacy through collaboration," *IEEE transactions on dependable and secure computing*, vol. 11, no. 3, pp. 266–279, 2014.

[20] T. Wang, Y. Mei, W. Jia, X. Zheng, G. Wang, and M. Xie, "Edge-based differential privacy computing for sensor–cloud systems," *Journal of Parallel and Distributed Computing*, vol. 136, pp. 75–85, 2020.

[21] Y. Tian, B. Song, M. Al Rodhaan et al., "A stochasticlocation privacy protection scheme for edge computing," *Mathematical Biosciences and Engineering*, vol. 17, no. 3, pp. 2636–2649, 2020.

[22] S. Zhang, G. Wang, M. Z. A. Bhuiyan, and Q. Liu, "A dual privacy preserving scheme in continuous location-based services," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4191–4200, 2018.

[23] S. Zhang, G. Wang, Q. Liu, and J. H. Abawajy, "A trajectory privacy-preserving scheme based on query exchange in mobile social networks," *Soft Computing*, vol. 22, no. 18, pp. 6121–6133, 2018.

[24] T. Xu and Y. Cai, "Exploring historical location data for anonymity preservation in location-based services," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pp. 1220–1228, Phoenix, AZ, USA, 2008.

[25] C.-Y. Chow, M. F. Mokbel, and W. G. Aref, "Casper∗," *ACM Transactions on Database Systems*, vol. 34, no. 4, pp. 1–48, 2009.

[26] X. Li, M. Miao, H. Liu, J. Ma, and K. C. Li, "An incentive mechanism for K-anonymity in LBS privacy protection based on credit mechanism," *Soft Computing*, vol. 21, no. 14, pp. 3907–3917, 2017.