Hindawi
Security and Communication Networks
Volume 2017, Article ID 7495974, 15 pages
https://doi.org/10.1155/2017/7495974

Research Article

An Improved Privacy-Preserving Framework for Location-Based Services Based on Double Cloaking Regions with Supplementary Information Constraints

Li Kuang,[1] Yin Wang,[1] Pengju Ma,[1] Long Yu,[1] Chuanbin Li,[1] Lan Huang,[1] and Mengyao Zhu[2]

[1]*School of Software, Central South University, Changsha 410075, China*
[2]*School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China*

Correspondence should be addressed to Mengyao Zhu; zhumengyao@shu.edu.cn

Received 18 August 2017; Accepted 10 October 2017; Published 7 November 2017

Academic Editor: Lianyong Qi

With the rapid development of location-based services in the field of mobile network applications, users enjoy the convenience of location-based services on one side, while being exposed to the risk of disclosure of privacy on the other side. Attacker will make a fierce attack based on the probability of inquiry, map data, point of interest (POI), and other supplementary information. The existing location privacy protection techniques seldom consider the supplementary information held by attackers and usually only generate single cloaking region according to the protected location point, and the query efficiency is relatively low. In this paper, we improve the existing LBSs system framework, in which we generate double cloaking regions by constraining the supplementary information, and then k-anonymous task is achieved by the cooperation of the double cloaking regions; specifically speaking, k dummy points of fixed dummy positions in the double cloaking regions are generated and the LBSs query is then performed. Finally, the effectiveness of the proposed method is verified by the experiments on real datasets.

1. Introduction

In recent years, with the rapid development of cellular network and GPS (Global Positioning System) positioning technology, the use of LBSs (location-based services) devices (such as phone, PAD) became more and more popular, while driving the rapid growth of LBSs applications. The typical LBSs applications include retrieval of POI (such as MeiTuan), map (such as Google Maps), GPS navigation (such as Amap), and location-aware social networks (such as Wechat). It can be said that LBSs have penetrated into many aspects of life, and the invocation of LBSs undoubtedly brings great convenience to people's life.

At the same time, LBSs privacy risks also attract the attention of the society, because when the user requests LBSs, specific location information is needed to submit, and the locations which are involved in a large number of user's query data [1, 2] may reveal user's privacy, such as home address, living habits, and social relations. In the era of big data security, the emphasis is put on the problem of security of data being sent, data at rest, and data being processed and deleted from the system [3]. If such information is leaked to malicious attackers, the user will be exposed to a serious threat. In practice, there is no server that is absolutely secure; LBSP (location-based services providers) itself may also be an attacker, and even anonymizer on third parties cannot be trusted absolutely. In addition, the client receives a large number of results returned by anonymizer, which will increase the cost of computation, and they may wait for the service due to too many dummy positions; therefore the user may be not satisfied by the application usage experience.

The existing LBSs framework is shown as Figure 1. The client sends user's request Q_U to the anonymizer. The Q_U includes the UID (User ID), the specific location l_u, the privacy protection requirement k, and the query content con, The anonymizer then sends the processed request Q_A to the LBSPs. Q_A includes randomly generated query requests for k dummies locations. Location information is different

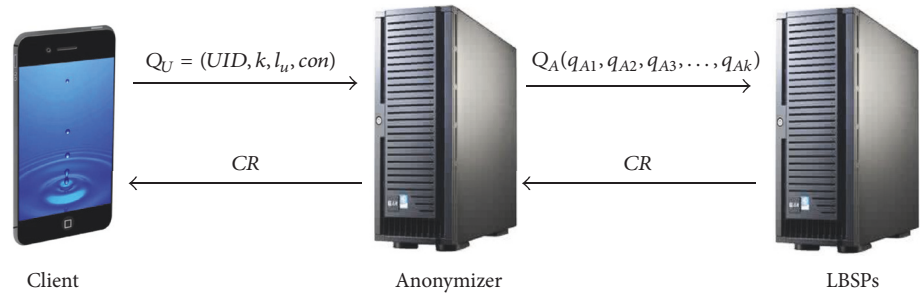


Figure 1: A centralized framework of LBSs.

from user's location in each q_{Ai}, while the remaining request information is the same. LBSP then returns the candidate results CR of the request Q_A to the anonymizer, and the anonymizer passes CR to the client, and finally the client filters the best result from the returned CR as the final result of the request.

The traditional LBSs privacy protection algorithms seldom consider that the anonymizer is not credible, so that the user's specific location information is sent directly to the anonymizer. If the data in anonymizer is leaked and used by the attackers, the user's location data will be disclosed directly. In addition, the attacker may make a fierce attack based on the probability of inquiry, map data, POI, and other supplementary information. For example, if a region is covered by a very low query number of locations such as lakes and mountains, the attacker can exclude the region with a large probability, so that the risk of user's exposure in the remaining region will be increased.

In this paper, we propose improving the existing LBSs framework and design several related algorithms within the framework. First, user's actual location which is contained in the query request is generalized into grid id, and the user's grid region is matched to another region by a dynamic matching algorithm, so that double cloaking regions are formed by considering that the attacker has a background of the number of historical queries; second, k fixed dummy positions are generated in double cloaking regions to achieve k-anonymous requirements by the proposed dummies generation algorithm; and, finally, the queries in dummy positions of double cloaking regions are sent to LBSPs and the candidate results are filtered and sent back to the user. Our proposed framework as well as the algorithms can solve the contradiction between service quality, privacy and resource overhead effectively.

The remainder of the paper is organized as follows. Related work is discussed in Section 2. The preliminaries of this paper are given in Section 3. The proposed approach is illustrated in Section 4. The experimental results are presented in Section 5. And finally, the conclusions and future work are given in Section 6.

2. Related Work

At present, researchers have put forward a lot of privacy protection methods for LBSs, and k-anonymous [4–16] is the core idea of many methods. Gruteser and Grunwald [4] propose the concept of location k-anonymity. K-anonymity requires that when a user sends a location request data to a LBSPs, the cloaking region in which a query user is located must contain at least the other $k - 1$ users, so that the probability that the location query user is identified does not exceed $1/k$. Yiu et al. [5] propose a Space Twist solution which introduces a trusted third party; after the user sends their real location information to the trusted third party, it will send a dummy coordinate to LBS service rather than user's real coordinate. Mokbel et al. [6] propose a k-anonymity protection method, which introduces a third party anonymous server; when the user sends a request to the LBSPs, the location information is sent to the anonymous server first. The anonymous server generalizes the user's location into a region of k-anonymity nature, and then the anonymous server sends the request to the LBSPs and returns the candidate result set to the user; finally, the user selects the best one.

Spatial cloaking [17–25] is a fairly popular mechanism. Chow et al. [17] propose Casper cloak algorithm, which uses a quad-tree data structure and allows users to determine the size of k and the minimum anonymous area, but the privacy can be guaranteed only when users' positions are distributed evenly. Jin and Papadimitratos [18] allow P2P responses to be validated with very low fraction of queries affected even if a significant fraction of nodes are compromised. Chen and Pang [21] propose that the cloaking region is randomly chosen from the ones with top-k largest position entropy.

Dummy position [26–30] generation is also a common method for location privacy protection. Kido et al. [26, 27] first propose a dummy position generation mechanism. Guo et al. [28] combine the dynamic pseudonym transformation mechanism with the user's personalized features to protect user's location privacy. Palanisamy and Liu [29] propose a Mix-zone approach for protecting user's privacy.

Encryption-based methods [12, 31–37] make user's location completely invisible to the server by encrypting LBSS query. Although encryption-based methods have high privacy and high quality of service, the calculation and communication costs are large, the deployment is complex, and the optimization algorithm is needed. Khoshgozaran et al. [12, 34] propose an encryption method based on the Hilbert curve to transfer user's position as well as his POI from two-dimensional coordinates to one-dimensional encryption

space; the one-dimensional encryption space transformed by two different parameters of Hilbert curve still maintains the proximity in two-dimensional space, so that k-nearest neighbor query and range query can also be performed in one-dimensional encrypted space. PIR (Private Information Retrieval) [35] method is used to protect user's query privacy, and it has the advantages of high privacy protection and good service quality. Lu et al. [36] propose the PLAM privacy protection framework, which uses homomorphic encryption to protect user privacy, but with much time overhead. Fu et al. [37] use the powerful computing power of the server to propose a retrieval encryption scheme to meet the privacy requirements of different threat models.

In summary, the existing location privacy protection mechanisms and methods still have the following problems: (1) the existing methods often do not consider the supplementary information when generating cloaking region; if the attacker has supplementary information, the success rate of the attack will be increased and the privacy security of the user will be challenged. (2) In the existing framework, the candidate results that anonymizer returns to the user often include a large number of useless dummy positions, which not only increase the computational overhead, but also reduce users' experience. (3) Dummy position coordinates are often generated randomly without considering whether it will affect the quality of final service.

The differences of this paper include the following: (1) we propose generating double cloaking regions while assuming that the attacker has a background of supplementary information, so the privacy protection can be greatly improved; (2) the proposed anonymizer in the LBSs framework will not return all candidate results but merely returns a half to the client, so that the computation overhead is reduced and user's waiting time is reduced; (3) we propose generating fixed dummy positions according to the value of k and uniform distribution rule, which can solve the contradiction between service quality, privacy, and resource overhead effectively.

3. Preliminaries

3.1. Strong Attack and Its Illustrating Examples. In this paper, we assume that the attacker is a strong attacker. LBSPs can be seen as strong attackers, since LBSPs not only have supplementary information, such as the number of historical queries, but also know the privacy protection mechanism. A strong attacker usually infers the region where the user is located and then combines the supplementary information to filter the user's region and even makes reverse attack based on the privacy protection mechanism, so that the attacker can uniquely identify the user's region, then infer the user's real location from the region, and finally access the user's privacy.

For example, as shown in Figure 3(a), if the user's region randomly matches a cloaking region with history query number of 1, obviously, it is a region with very low number of historical queries, while if the user's real location is in the region with history query number of 20, there will be a great possibility of determining the user's real region.

The strong attacker may not only have the supplementary information, but also know the privacy protection mechanism. Suppose that we simply use the region which is the closest to the user's history query as the generation mechanism of double cloaking region, and the attacker knows the mechanism. As shown in Figure 3(b), the query time in user's region is 20, and the region with query times 22 will form the double cloaking regions. If the attacker is the LBSP itself, it will analyze the two regions; if the user's real region is that with query times 22, the closest one is the region with 23. To form a double cloaking region, the region with query times 22 will select that with 23 instead of 20, so it can be determined that the user's real region is that with 20. Therefore, if attackers have supplementary information and know the privacy protection mechanisms, it will increase the risk of users to disclose the specific location.

3.2. Problem Definition and Related Concepts

Definition 1 (space division based on quad-tree). As shown in Figure 4, this paper uses the quad-tree data structure [8]. The space is divided layer by layer from top to bottom, and each layer is divided into 4^h grids; especially speaking, the 0th layer of the entire space is divided into 1 grid, the 1st layer contains 4 grids, the 2nd layer contains 16 grids, and so on, until the side length of each grid reaches the threshold L, and the space will be divided into H layers. The total number of the history query times on each layer is the same, but the entire spatial area is subdivided so that the length L of each grid is gradually reduced. The smaller the L, the lower the level of privacy protection and the higher the quality of service; on the contrary, the bigger the L, the higher the level of privacy protection and the lower the quality of service. The information in each grid is contained in the hash table, where the *GID* is the number of the grid and *NoU* is the user's history query times in each grid.

Definition 2 (query request from user Q_U (UID, k, h, GID, con)). As shown in Figure 2, the query request submitted by the user to the anonymizer is denoted as Q_U (*UID, k, h, GID, con*), where *UID* is the user's identification information; k is user's requirement on *k-anonymous* protection mechanism, which determines the number of dummies to generate; h is user's requirements on the generalization level of space, which is required to be larger than 2, since the query would be too poor if h is less than or equal to 2. Both the user and the anonymizer use the quad-tree data structure to store the spatial information. The *GID* is the grid ID which is generated according to the user's specific location; *con* is the query content, which is not the focus of this study.

Definition 3 (query request from anonymizer Q_A (q_{A1}, q_{A2}, $q_{A3}, \ldots, q_{Ak}$)). The query request passed by the anonymizer to LBSPs is denoted as $Q_A(q_{A1}, q_{A2}, q_{A3}, \ldots, q_{Ak})$, where $q_{Ai}(DID, l_{di}, con)$ is a request for each dummy and *DID* (Dummies ID) is the identification information of the k dummies generated by the anonymizer; l_{di} is the location information of k dummies; *con* is the query content.

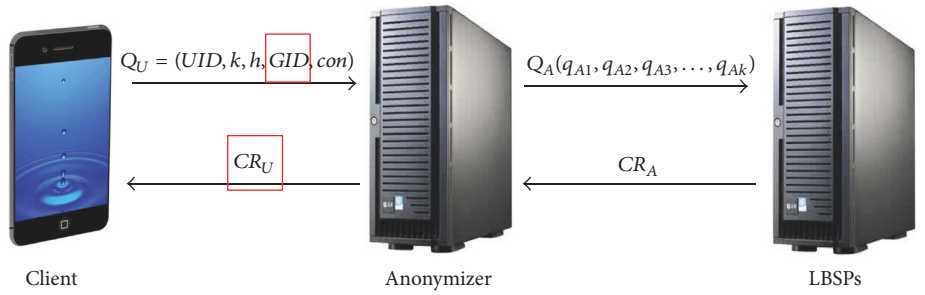


Figure 2: Improved framework of LBSs system.

1	2	25	26
18	24	33	45
28	20	22	30
51	43	23	13

(a) Cloaking region is formed by randomly matching the number of queries

1	2	25	26
18	24	33	45
28	20	22	30
51	43	23	13

(b) Cloaking region is formed according to the closest number of query

Figure 3: Two mechanisms of generating cloaking region.

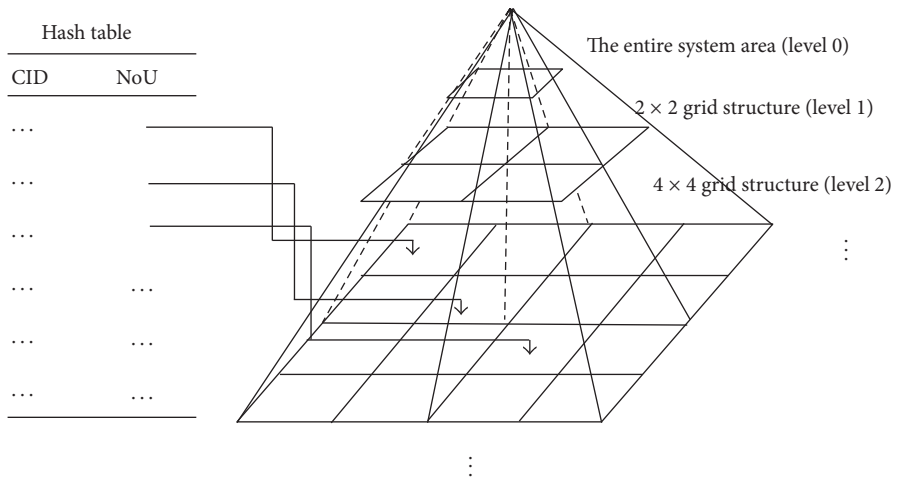


Figure 4: Data storage structure.

Definition 4 (candidate results to anonymizer CR_A). The LBSPs return the candidate results to the anonymizer as CR_A, which is the results of the request for k dummies in the double cloaking region. Each request result corresponds to the *DID* of the query request.

Definition 5 (candidate results to user CR_U). The anonymizer returns candidate results to the user as CR_U, which only includes the query results of the dummies in the *RCR*.

Definition 6 (quality of service). The quality of service obtained by the user is measured by the Euclidean distance between the dummy and the user. If the user is closer to the dummy position, the location of request and the result are more similar; therefore the service quality would be higher. Assume that the user's specific position is l_u, its latitude and longitude coordinate is (lon_u, lat_u), and l_{di} $(i = 1, 2, 3, \ldots, k)$ is the ith dummy position; its latitude and longitude coordinate is (lon_{di}, lat_{di}). r is the radius of the

earth, generally taken as 6371 km. The distance between user and dummy is calculated as formula (1):

$$\mathrm{dis}_i\left(l_u, l_{di}\right) = 2r * \arcsin\left(\sqrt{\sin\left(\frac{d_{lati}}{2}\right)^2 + \cos\left(lat_{di}\right) * \cos\left(lat_u\right) * \sin\left(\frac{d_{loni}}{2}\right)^2}\right), \tag{1}$$

where

$$d_{loni} = \frac{\pi\left|lon_{di} - lon_u\right|}{180}, \qquad d_{lati} = \frac{\pi\left|lat_{di} - lat_u\right|}{180}. \tag{2}$$

The smaller the value of $\mathrm{dis}_i(l_u, l_{di})$, the better the quality of service of ith dummy, and we can take the query result of the ith dummy as the final query result.

Problem Definition. We know that the user, the anonymous server, and LBSPs share the *space division based on quad-tree*. And we also know that the user submits Q_U to anonymizer, and the anonymizer passes Q_A to LBSPs, the LBSPs return CR_A to anonymizer, and the anonymizer returns CR_U to client. In this process, it is assumed that the anonymizer and LBSPs are not fully credible, so the strong attacker is most likely to guess the specific location of the user according to the background knowledge which includes the number of historical queries and privacy protection mechanism, thus causing the user's privacy to be disclosed. The problem that we want to solve is improving the *quality of service* experienced by the user and reducing the computing overhead of the user, when he accesses the LBSs, while ensuring his location security.

3.3. Symbolic Correspondence. For simplicity, we list the notations used in this paper as Notation section shows.

4. Privacy-Preserving Framework and Algorithms for LBSs

4.1. Approach Overview. In order to protect the user's real location which is contained in the query request, we employ the double cloaking region mechanism. The double cloaking region includes real cloaking region (*RCR*) and fake cloaking region (*FCR*). The *RCR* is the user's grid, and the anonymizer generates *FCR* by dynamic clustering method. *FRC* has three main functions: (1) *FCR* and *RCR* together generate k dummy positions to reach k-anonymous requirements; (2) *FCR* and *RCR* form a double cloaking region against strong-attacks; (3) when the anonymizer returns the candidate results to the client, the candidate results of the request for the dummies in the *FCR* are filtered directly. The dummies in the double cloaking region are sent to the LBSPs to request the service.

The whole process of our proposed solution is shown as Figure 5 (the system execution order is demonstrated by the numbers): (1) a *RCR* is generated according to the user's specific location; (2) the query request Q_U is submitted to the anonymizer; (3) Dynamic Matching Algorithm (*DMA*) is employed by the anonymizer to generate a *FCR* according to the *GID* in Q_U, so that a double cloaking region with similar query times is formed; (4) *k/2* dummy positions are generated in the two regions, respectively, by using the dummies generation algorithm (*DGA*), and the two dummy sets are denoted *DSs1* and *DSs2*; (5) the query request Q_A from the dummy positions in *DSs1* and *DSs2* is submitted to LBSPs together; (6) LBSP answers CR_A according to Q_A; (7) CR_A are replied to the anonymizer; (8) the anonymizer forms CR_U according to the *DIDs* (Dummy IDs) in CR_A to filter out the query results in *DSs1*; (9) CR_U is returned to the client, and the client selects the query result q_{Ai} of the dummy whose $\mathrm{dis}_i(l_u, l_{di})$ is minimum as the query result according to formula (1).

In our improved framework, there are two important algorithms, which are dynamic matching algorithm *(DMA)* and dummies generation algorithm *(DGA)*, and we will illustrate the two algorithms in the following subsections.

4.2. Dynamic Matching Algorithm. The main idea of dynamic matching algorithm (DMA for short) is to separate the regions with relatively large, relatively small, and zero number of queries, so that the two regions with obviously different number of queries will not be matched together to form a double cloaking region. As shown in Figure 6(a), the points represent the positions where users make historical requests. The position coordinates are projected into a 2D map, and the whole region is divided into 4^h grids according to the generalization level of space h. *RCR* where the user located is represented by the region with black solid line. The region is divided into 9×9 grids in Figure 6(a) as an example.

As shown in Figure 6(b), first, a 4×4 grid region which contains *RCR* is allocated randomly, and *FCR* will be matched in this region as well; second, the number of historical queries in each grid is counted and stored in a matrix represented by $G_{4\times4}$. An example of $G_{4\times4}$ is shown in Figure 7, where the number of queries in *RCR* is 25, and the number of historical queries in each grid of the 4×4 region with black line in Figure 6(b) is also shown.

And then the numbers of queries in the 4×4 region are divided into three categories, relatively large, relatively small, and zero, which are realized by the classical dynamic clustering algorithm [38]. To form a double cloaking region, we remove the region with zero number of queries firstly, and then *FCR* is only selected randomly from the regions with the number in the same category as *RCR*.

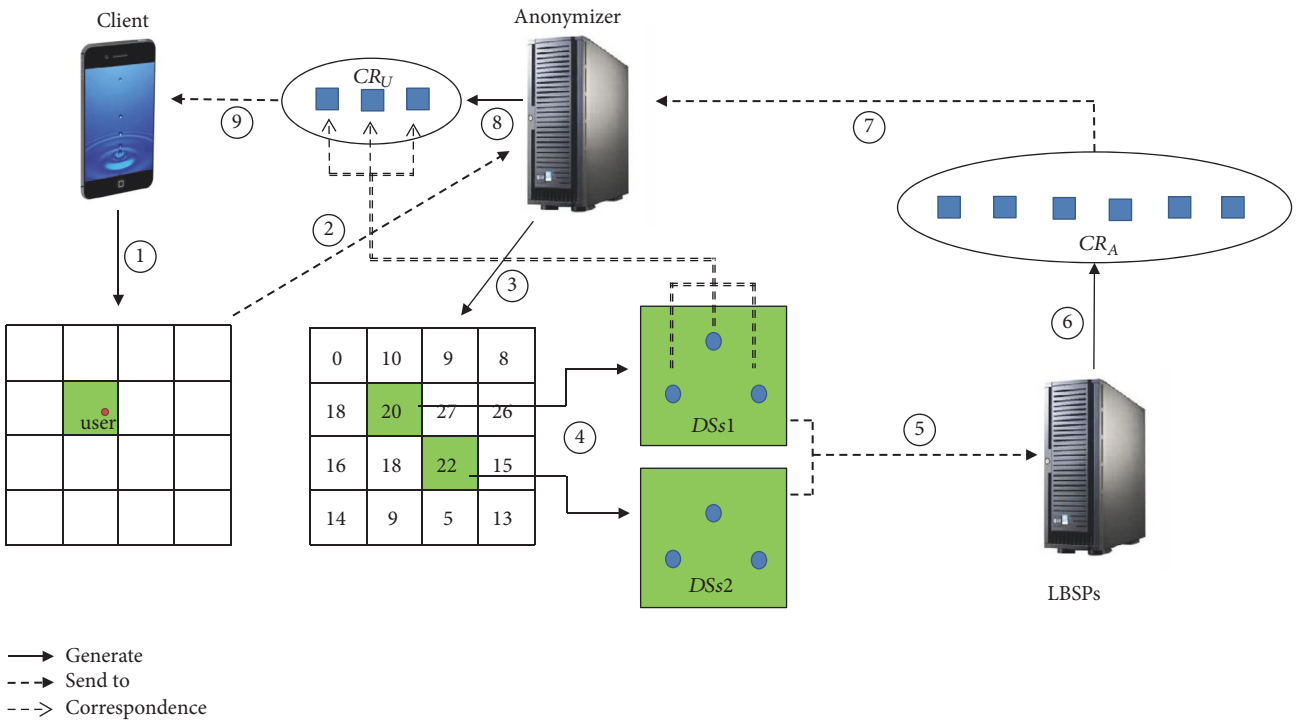
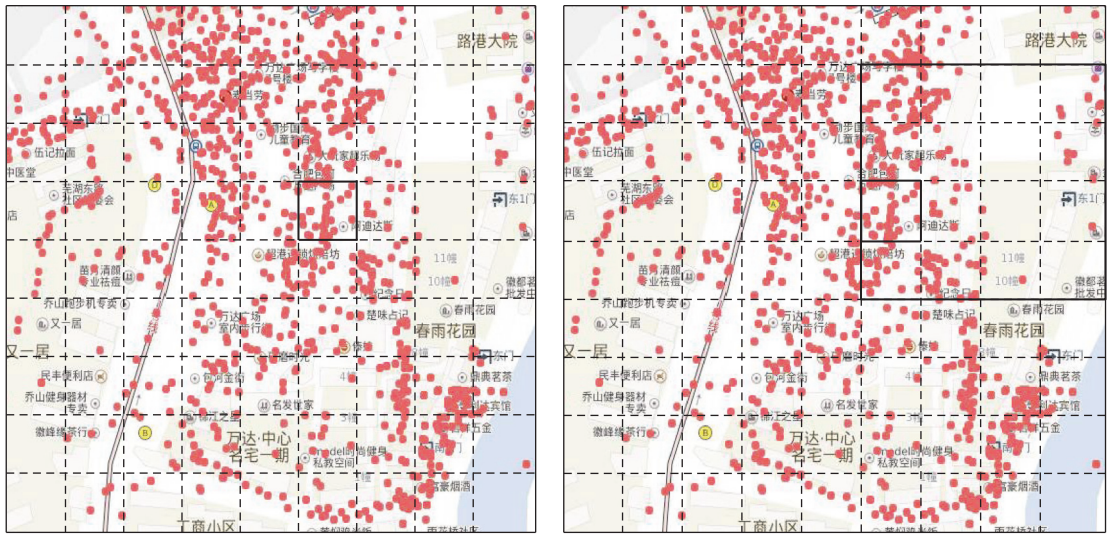


Figure 5: The improved framework.



(a) The positions where users make historical requests (b) Users are allocated randomly into the 4 × 4 grid region

Figure 6: Historical requests on the map.

Take the data in Figure 7 as an example, *DMA* first divides the number of queries into three categories, {14, 16, 22, 25, 25, 27}, {1, 1, 1, 6, 6, 8, 9}, and {0, 0, 0}, respectively. Assume that the query number of the user's region is 25; the *FCR* may be the grid with the query number 22, and the two regions will form a double cloaking region.

The pseudocode of *DMA* is shown as Algorithm 1.

4.3. Dummies Generation Algorithm. The core idea of dummies generation algorithm *(DGA)* is to generate k fixed dummy positions to approximate the user's real position, and it tries to distribute the fixed dummy positions over the region as evenly as possible, and then the answer to the query request by the user will be approximated by that from the best dummy location. It is common to generate

27	14	0	8
25	6	0	6
25	9	1	1
16	22	1	0

Figure 7: Number of historical queries in 4×4 region.

Input: Privacy protection level h, User's grid ID *GID*
Output: Double Cloaking Region (*RCR* and *FCR*)
(1) **Anonymizer selects the spatial hierarchy according to the privacy protection level *h***
(2) **Randomly match the user's grid into a grid region $G_{4\times4}$**
(3) **for query count in $G_{4\times4}$**
(4) **if query count≠0 then**
(5) **add query count to Sets**
(6) **end if**
(7) **end for**
(8) **The Sets are randomly divided into set_1 and set_2** equally
(9) **Do**
(10) **avg_1 = average(set_1); avg_2 = average(set_2);**
(11) **for** *s* **in** Sets
(12) **if $(s - avg_1)^2 < (s - avg_2)^2$ then**
(13) **s belong to c_1**
(14) **else**
(15) **s belong to c_2**
(16) **end if**
(17) **end for**
(18) **while** there are changes on the elements **in c_1 and c_2**
(19) **if** the number of ***GID*** belongs to **c_1 then**
(20) ***FCR*** isselected randomly from the regions with the number in **c_1** except ***RCR***
(21) **else *FCR*** is selected randomly from the regions with the number in **c_2** except ***RCR***
(22) **end if**
(23) **return *FCR* and *RCR***

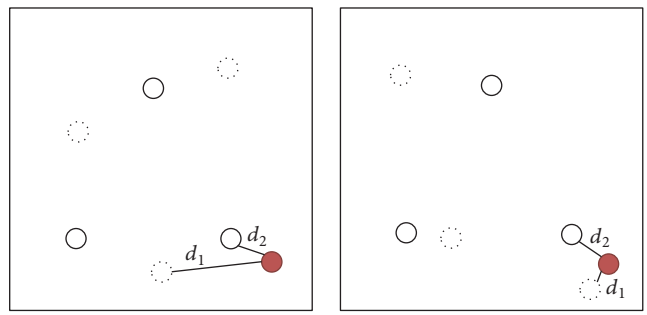
Algorithm 1: DMA (dynamic matching algorithm).

dummy positions randomly in the double cloaking regions; however, we propose defining two rules to generate fixed dummy positions according to k. In Figure 8, the red solid circle represents the user's real position and the solid circles represent the fixed dummy positions according to our rules, while the dotted circles represent the dummy positions generated randomly. As shown in Figure 8(a), when $d_1 > d_2$, that is, the shortest distance between the user and the fixed dummy is shorter than that between the user and the random dummy, we say that the quality of service of the fixed dummy positions is higher than that of the random dummy positions according to Definition 6. On the contrary, as shown in Figure 8(b), when $d_1 < d_2$, we say that the quality of service of the fixed dummy positions is lower than that of the random dummy positions. We verify that the service quality of *DGA* is higher than that of random way in Section 5.2.

The coordinate system of the two-dimensional coordinate origin is established at the lower left vertex of the grid. In the anonymizer, there is data of each grid length L in each spatial hierarchy. k_1 and k_2 both equal to $k/2$, and they represent the number of dummy positions to be generated in *RCR* and *FCR*, respectively. There are two core rules in *DGA*.

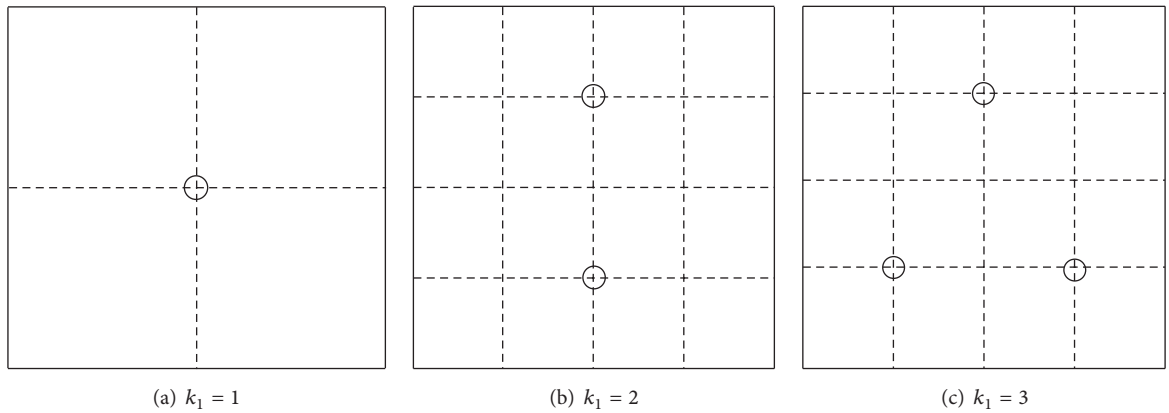
Base Rule R_1. When k_1 (or k_2) ≤ 5, the fixed dummy positions when k_1 (or k_2) = 1, 2, 3, 4, 5 are shown as Figures 9(a)–9(e), respectively.

(1) When k_1 (or k_2) = 1, the dummy position is set at $(L/2, L/2)$, as shown in Figure 9(a);



(a) $d_1 > d_2$ (b) $d_1 < d_2$

FIGURE 8: The shortest distance between the user and the random dummy d_1 compared to that between the user and the fixed dummy d_2.



(a) $k_1 = 1$ (b) $k_1 = 2$ (c) $k_1 = 3$

(d) $k_1 = 4$ (e) $k_1 = 5$

FIGURE 9: Base rule R_1.

(2) When k_1 (or k_2) = 2, the dummy positions are $(L/2, L/4)$, $(L/2, 3L/4)$, as shown in Figure 9(b);

(3) When k_1 (or k_2) = 3, the dummy positions are $(L/4, L/4)$, $(L/2, 3L/4)$, $(3L/4, L/4)$, as shown in Figure 9(c);

(4) When k_1 (or k_2) = 4, the dummy positions are $(L/4, L/4)$, $(3L/4, L/4)$, $(L/4, 3L/4)$, $(3L/4, 3L/4)$, as shown in Figure 9(d);

(5) When k_1 (or k_2) = 5, the dummy positions are $(L/4, L/4)$, $(3L/4, L/4)$, $(L/4, 3L/4)$, $(3L/4, 3L/4)$, $(L/2, L/2)$, as shown in Figure 9(e).

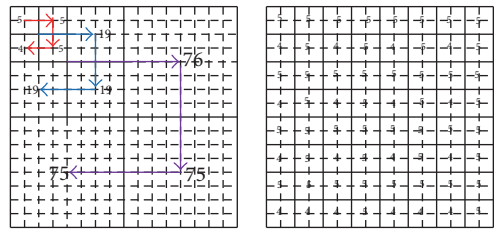
Generalization Rule R_2. When k_1 (or k_2) = n and $n > 5$, we have the following:

First, divide the whole region into 4 grids, and each value in the 4 grids is as follows:

(1) If $n\%\ 4 = 0$, $n/4$ dummy positions are assigned in each of the 4 grids;

(2) If $n\%\ 4 = 1$, $(n/4) + 1, n/4, n/4, n/4$ dummy positions are assigned in the 4 grids, respectively, starting from the left upper corner, continuing in the clockwise direction;

(3) If $n\%\ 4 = 2$, $(n/4) + 1$, $(n/4) + 1$, $n/4$, $n/4$ dummy positions are assigned in the 4 grids, respectively;



(a) An initial division when $k = 302$

(b) The final division when $k = 302$

Figure 10: An illustrating example of *DGA*.

Input: The value of k in the k-anonymity, *RCR*, *FCR*
Output: k_1 *dummies* in *RCR* and k_2 *dummies* in *FCR*, ID at each Dummy *DID*
(1) **Generate** $4^{\lceil \log_4 \lceil k_1/5 \rceil \rceil}$, $4^{\lceil \log_4 \lceil k_2/5 \rceil \rceil}$ small grids in ***RCR*** and ***FCR***
(2) **Generate** fixed dummies in ***RCR*** and ***FCR*** based on R_1, R_2
(3) **Add ID** for each dummy
(4) **return** k_1, k_2 dummies, ID at each Dummy *DID*

Algorithm 2: DGA (dummies generation algorithm).

(4) If $n\%\ 4 = 3$, $(n/4)+1$, $(n/4)+1$, $(n/4)+1$, $n/4$ dummy positions are assigned in the 4 grids, respectively.

Second, if $(n/4) + 1$ or $n/4$ is still larger than 5, repeat the first step; otherwise, follow the base rule R_1 to distribute the dummy positions.

In total, $4^{\lceil \log_4 \lceil n/5 \rceil \rceil}$ small grids will be generated in the region.

Take $k_1 = 302$ as an example, as shown in Figure 10, (1) in the first level of division; the region will be divided into 4 grids, and dummy positions in each grid are 76, 76, 75, and 75, respectively; (2) since 76 or 75 is larger than 5, in the second level of division, the 4 grids will be divided into 4 grids further; for example, the left upper grid with 76 will be divided into 4 grids with 19, 19, 19, and 19 dummy positions, respectively; and the other three grids follow the same way; (3) since 19 is larger than 5, in the third level of division, the grid with 19 will be divided into 4 grids with 5, 5, 5, and 4 dummy positions, respectively; and the other three grids follow the same way; (4) since 5 or 4 is not larger than 5, the division stops and in total $4^{\lceil \log_4 \lceil k_1/5 \rceil \rceil} = 64$ grids are generated, and the generation of fixed positions in the 64 small grids follows rule R_1.

According to *DGA*, anonymizer can store *DD* (Dummies data) that satisfies various k values in the database, so that the query requests for services can be responded to quickly. The pseudocode of *DGA* is shown as Algorithm 2.

5. Experiment and Analysis

5.1. Experiment Setting. In this paper, we use the historical GPS sampling point data within the range of 5.5 km × 3.5 km in Hefei city as historical inquiry points, which includes more than 60,000 sampling points produced by more than 30,000 people. The data consists of ID, latitude, and longitude, in which "ID" is the user's unique identifier; "longitude" and "latitude" together tell the location where the user submits a query. For convenience, the experiment selects an area of 3.2 km × 3.2 km and sets the threshold of edge length L to 50 m. The space is divided into 64 × 64 grids, and the spatial region is divided into 7 layers, from 0th to 6th layer.

We will compare the dummy algorithm (DA) and naive algorithm (NA) with our proposed *Double Cloaking Algorithm* (*DCA* for short, which consists of *DMA* and *DGA*). As shown in Figure 11, the *DA* is similar to the *DCA* process, except the red box in Figure 11. Specially speaking, *DCA* generates fixed dummy positions according to *DGA*, while *DA* generates random dummy positions in the double cloaking regions. We aim to compare the quality of service for *DA* and *DCA*.

As shown in Figure 12, *NA* is similar to the *DCA* process, except the red boxes in Figure 12. *NA* does not generate double cloaking regions; the anonymizer generates k dummy positions directly in the user's region and sends the queries in dummies to LBSPs, then receives the candidate results from LBSPs, and passes to the user without filtration. We aim to compare the time cost of *NA*, *DA*, and *DCA*.

The coding language is Python and the experiment runs with the 64 bit Windows 10 operating system configured as Intel (R), Core (TM), i5-4590, CPU, and 8 GB.

5.2. Experimental Result and Analysis

5.2.1. The Time Cost of Generating Dummies. As shown in Figure 13, when k changes within (2, 50), the time cost of *DCA* for generating dummy positions is steady, always 0.17 ms. Because *DCA* can be divided into two steps, *DMA* and then *DGA*, the time cost of *DMA* is not affected by k; moreover, the k-anonymous fixed dummy positions have already been set and stored, so it only needs to choose the fixed positions

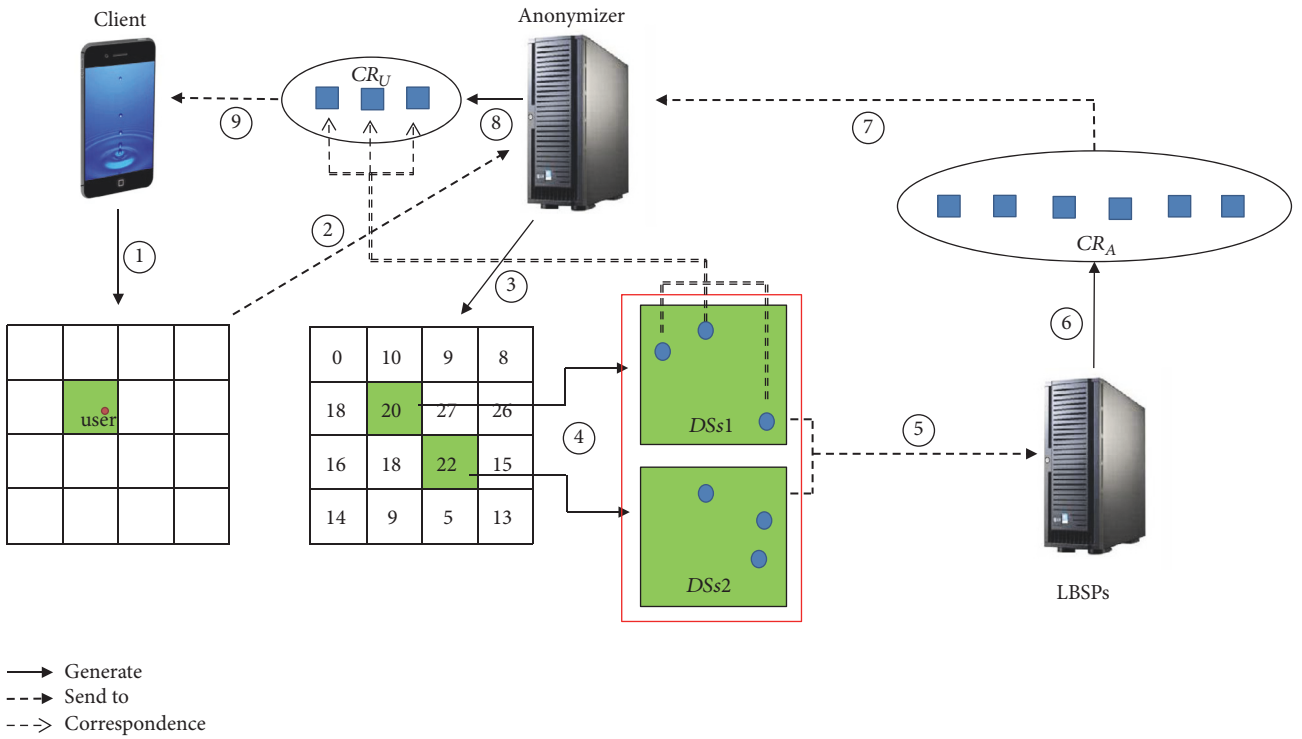


Figure 11: Dummy algorithm.

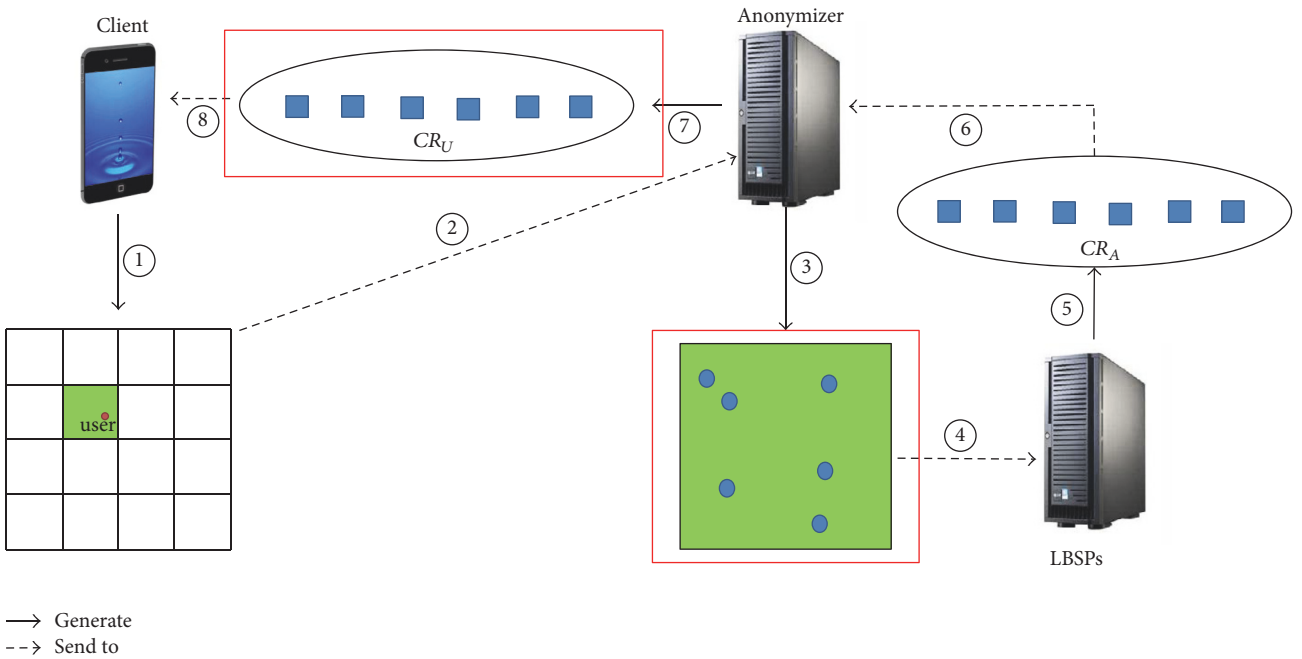


Figure 12: Naive algorithm.

according to k; therefore, the time cost of *DCA* is relatively fixed and remains a constant value.

DA can also be divided into two steps; the first step is the same as *DCA*, and the second step is to generate random dummy positions, which should be computed in real time, so it will take more time than *DCA* to generate each dummy position, and the bigger the value of k, the more time it takes.

While *NA* only takes time to generate the k dummy positions randomly, when k is less than 28, *NA* takes less time than *DCA*; when k is equal to 28, *DCA* and *NA* spend the

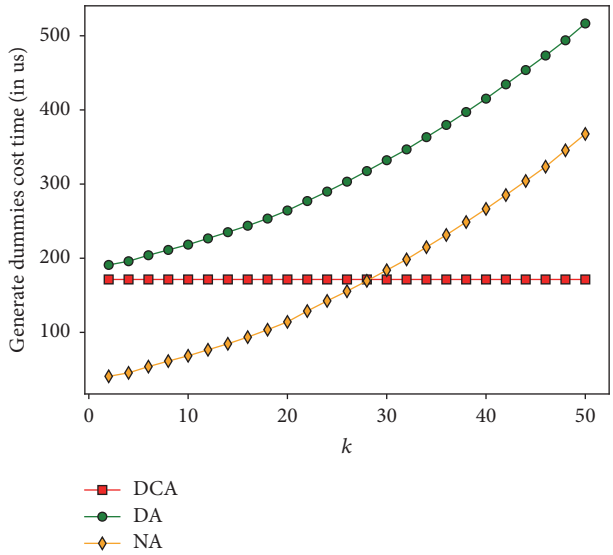


Figure 13: Comparison on time cost of generating dummies.

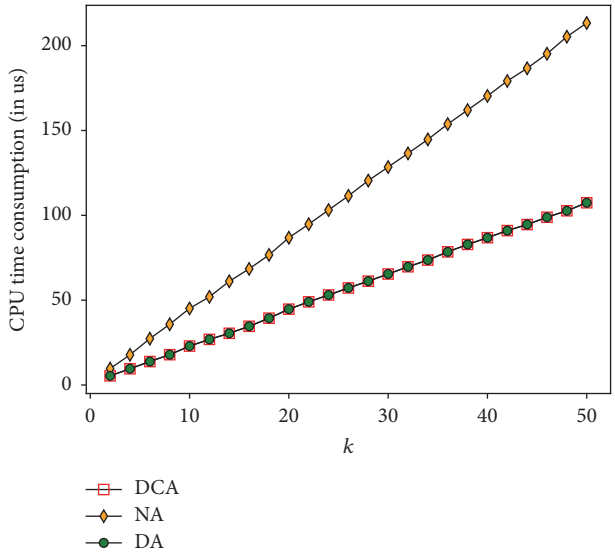


Figure 14: Comparison on time cost of result processing on client side.

same time; with the continuous increase of k, *NA* begins to take more time than *DCA*.

5.2.2. The Time Cost of Result Processing. As shown in Figure 14, the time cost of *NA* for result processing on the client side is almost twice as that in *DCA* and *DA*. Because when the anonymizer sends CR_U to the client, *DCA* and *DA* generate the double cloaking regions, the k dummy positions are equally distributed into two regions, and only the candidate results in *RCR* are returned to the client by the anonymizer; *NA* generates k dummy positions in one cloaking region; the anonymizer returns a set of candidate results for k dummy positions to the client. The number of candidate results in *NA* is twice of that in *DCA* and *DA*; in order to select the optimal dummy position, the client needs to calculate the $\text{dis}_i(l_u, l_{di})$ between all dummy positions in the candidate results and the user's specific position. So there is an obvious difference among *NA* and *DCA*/*DA* on the time cost of result processing on the client.

Please note the experiment is simulated on computer, and the unit of the experimental result is microsecond (us), but in actual environment, when the results are processed by client on smart phones, the unit of time cost will fall into millisecond (ms) level.

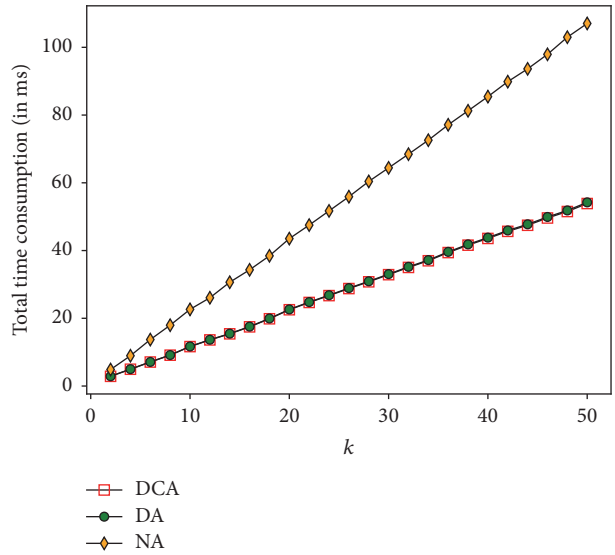


Figure 15: Comparison on total time consumption.

5.2.3. Total Time Consumption. In this section, we will compare the total time cost of the three algorithms, taking into account the device performance of the anonymizer and the client. In general, the computing power of our PC is much better than that of phones used by the client. Theoretically the floating-point computing power of 1.3 GHz frequency quad-core ARM processor is about 10 MFLOPs/s, and that of 2.5 GHz frequency Intel quad-core Q8300 is 25GFLOPs/s; the two differ 2500 times. Due to the different computing power of different devices, we deem conservatively that the computing power of PC is 500 times as much as the client device, while the computing power of anonymizer is the same as PC; therefore the total time consumption is

$$\text{Total Time} = \text{Time of Generating Dummies} + 500 \times \text{Time of Result Processing}. \quad (3)$$

According to formula (3), it can be known that the time cost on the client side is much larger than that on the anonymizer, and the result processing accounts for the majority percentage of the total time consumption. As shown in Figure 15, the total time consumed by *NA* is about twice as much as *DCA* and *DA* when k is given. In terms of time

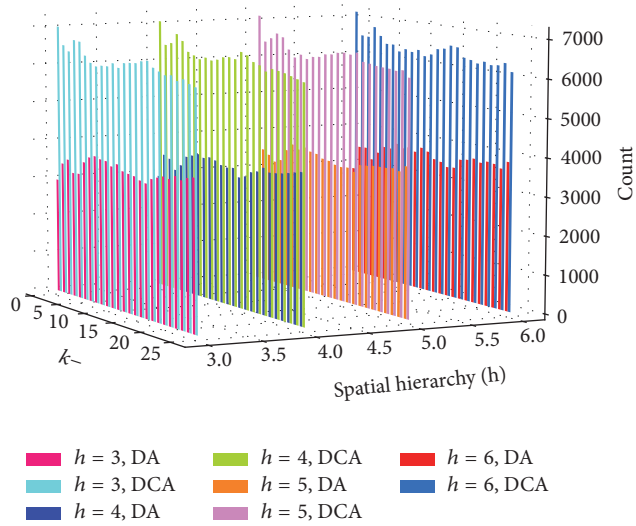


Figure 16: Comparison on the probability of getting better quality of service.

efficiency, *DCA* and *DA* are better than *NA*. Since *NA* does not generate double cloaking regions, its privacy protection capability is weaker than DCA and DA, and in the next experiment we only compare *DCA* and *DA*.

5.2.4. Comparison on Quality of Service between DCA and DA. In order to compare *DCA* and *DA* on the quality of service, we first conduct 10,000 times of experiments on different values of h and k and count the times when $d_1 > d_2$and $d_1 < d_2$, as shown in Figures 8(a) and 8(b). As shown in Figure 16, given k and h, for the 10000 experiment, if $d_1 > d_2$, the count of *DCA* adds 1, if $d_1 < d_2$, the count of *DA* adds 1.

When h is specific while k varies between (1, 25), the count of *DGA* ranges between 6000 and 7200, and the count of *DA* ranges between 2,800 and 4000; when h varies, the count range of *DGA* and *DA* does not change much, because although h becomes larger and L becomes smaller in the cloaking region, the ratio of the fixed dummy positions to L stays the same, and the position of the random dummy positions is also independent of L.

In summary, *DCA* has a greater probability of getting better quality of service than *DA*.

We further compare the average quality of service of *DCA* and *DA*. We conduct 10,000 times of experiments on different values of h and k and compute the average quality of service. In the experiment, k ranges from 1 to 25 and h ranges from 3 to 6. According to Definition 6, we can see that the smaller the distance from the user, the better the quality of service. In Figures 17(a)–17(d), with the decrease of h, the average quality of service of *DCA* and *DA* is decreasing, but the average quality of service of *DCA* is always better than *DA*. With the increase of k, the average quality of service of *DCA* and *DA* is increasing, and when k is larger than 15, the trend of increasing becomes slow. In summary, *DCA* has a better average quality of service than *DA*.

6. Conclusion

In this paper, we propose an improved privacy-preserving framework for location-based services based on double cloaking regions with supplementary information constraints. Compared to previous work, our method is effective in solving the strong attack with supplementary information, and, comparing to generating random dummy positions, generating fixed ones improves the service quality but reduces the computational overhead for the client. However, when the distribution of the information data is extremely nonuniform, the dynamic matching algorithm is difficult to match the region of similar information and forms double cloaking regions with the user's region. In the future, we plan to improve the dynamic matching algorithm; in addition, we will consider the continuous query requests of the mobile user.

Notations

LBSPs: Location-Based Services Providers
GID: Grid ID
DIDs: Dummies IDs
UID: User ID
Q_U: A set of query requests submitted by the user to the Anonymizer
Q_A: A set of query requests passed by the Anonymizer to LBSPs
CR_A: The set of candidate results sent by LBSPs to Anonymizer
CR_U: The set of candidate results sent by Anonymizer to Client
$G_{4\times4}$: RCR randomly matches into the grid of 4×4
h: Spatial hierarchy
LS: Level saturated

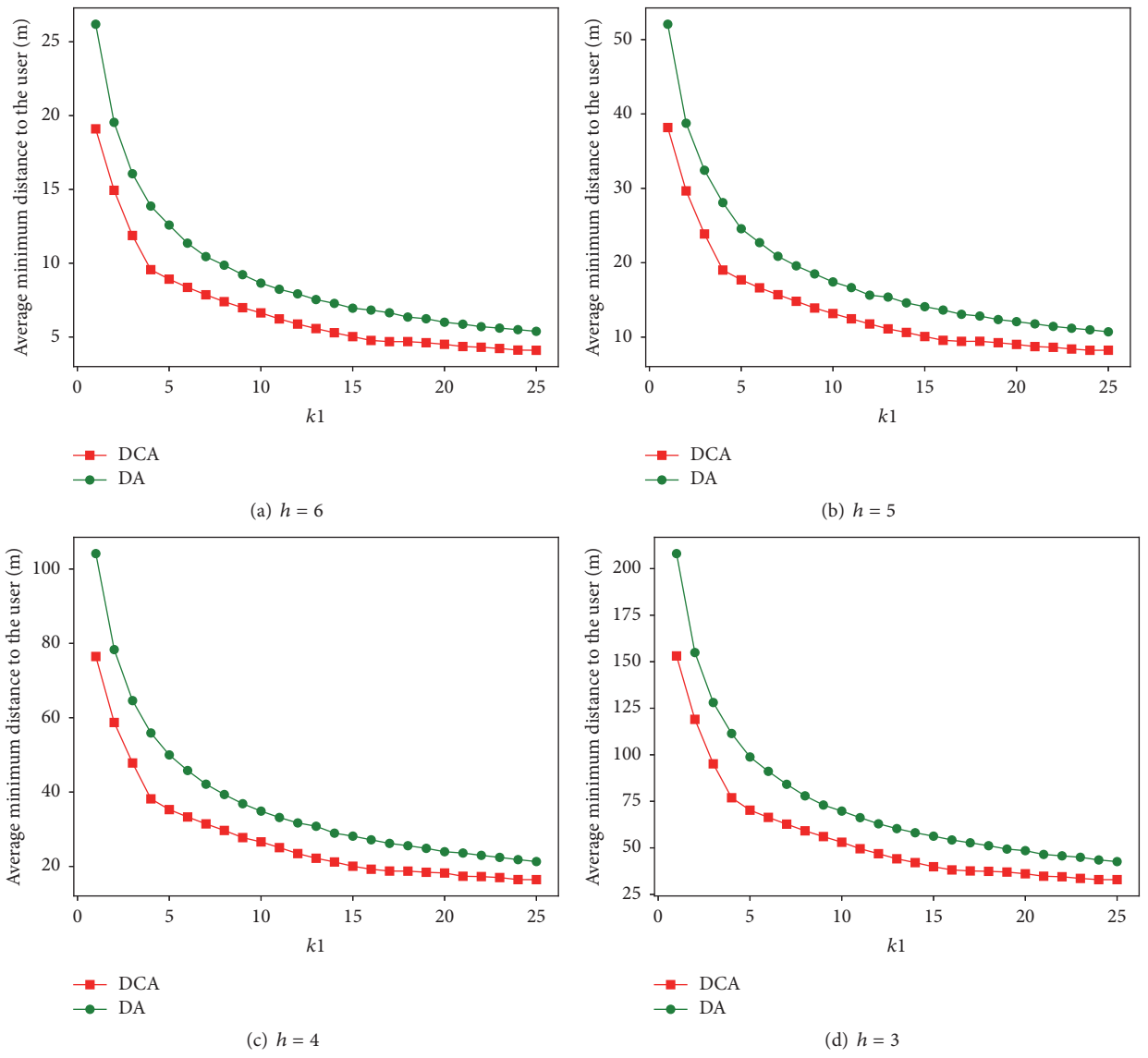


Figure 17: Comparison on the average quality of service.

$\text{dis}_i(l_u, l_{di})$: Euclidean distance between user and dummy
DD: Dummies Data.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The research is supported by "National Natural Science Foundation of China" (no. 61772560), "Natural Science Foundation of Hunan Province" (no. 2016JJ3154), "Key Support Projects of Shanghai Science and Technology Committee" (no. 16010500100), "Scientific Research Project for Professors in Central South University, China" (no. 904010001), and "Innovation Project for Graduate Students in Central South University" (no. 1053320170313).

References

[1] D. Vatsalan, Z. Sehili, P. Christen et al., *Privacy-Preserving Record Linkage for Big Data: Current Approaches and Research Challenges*, Handbook of Big Data Technologies, Springer International Publishing, 2017, 851–895.

[2] H. Ye, X. Cheng, M. Yuan, L. Xu, J. Gao, and C. Cheng, "A survey of security and privacy in big data," in *Proceedings of the 16th International Symposium on Communications and Information Technologies, ISCIT 2016*, pp. 268–272, Qingdao, China, September 2016.

[3] A. Jakóbik, *Big Data Security*, Resource Management for Big Data Platforms, Springer International Publishing, 2016, 241–261.

[4] 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, ACM, San Francisco, Calif, USA, May 2003.

[5] M. L. Yiu, C. S. Jensen, X. Huang, and H. Lu, "SpaceTwist: managing the trade-offs among location privacy, query performance, and query accuracy in mobile services," in *Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE '08)*, pp. 366–375, IEEE Press, Cancun, Mexico, April 2008.

[6] M. Mokbel F, C. Chow Y, and G. Aref W, "The new casper: Query processing for location services without compromising privacy," in *Proceedings of the 32nd international conference on Very large data bases*, pp. 763–774, VLDB Endowment, 2006.

[7] X. Pan, J. L. Xu, and X. F. Meng, "Protecting location privacy against location-dependent attacks in mobile services," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1506–1519, 2012.

[8] X. Zhu, H. Chi, B. Niu, W. Zhang, Z. Li, and H. Li, "MobiCache: When k-anonymity meets cache," in *Proceedings of the 2013 IEEE Global Communications Conference, GLOBECOM 2013*, pp. 820–825, IEEE, Atlanta, GA, USA, December 2013.

[9] Y. Wang, D. Xu, X. He, C. Zhang, F. Li, and B. Xu, "L2P2: location-aware location privacy protection for location-based services," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '12)*, pp. 1996–2004, IEEE, Orlando, Fla, USA, March 2012.

[10] T. Hashem, L. Kulik, and R. Zhang, "Countering overlapping rectangle privacy attack for moving kNN queries," *Information Systems*, vol. 38, no. 3, pp. 430–453, 2013.

[11] B. Yao, F. Li, and X. Xiao, "Secure nearest neighbor revisited," in *Proceedings of the IEEE 29th International Conference on Data Engineering (ICDE '13)*, pp. 733–744, IEEE, Brisbane, Australia, April 2013.

[12] A. Khoshgozaran, C. Shahabi, and H. Shirani-Mehr, "Location privacy: Going beyond K-anonymity, cloaking and anonymizers," *Knowledge and Information Systems*, vol. 26, no. 3, pp. 435–465, 2011.

[13] 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.

[14] Y. Elmehdwi, B. K. Samanthula, and W. Jiang, "Secure k-nearest neighbor query over encrypted data in outsourced environments," in *Proceedings of the 30th IEEE International Conference on Data Engineering (ICDE '14)*, pp. 664–675, Chicago, IL, USA, April 2014.

[15] H. Hu and J. Xu, "Non-exposure location anonymity," in *Proceedings of the 25th IEEE International Conference on Data Engineering, ICDE 2009*, pp. 1120–1131, Shanghai, China, April 2009.

[16] B. Gedik and L. Liu, *A Customizable k-Anonymity Model for Protecting Location Privacy*, Georgia Institute of Technology, 2004.

[17] C.-Y. Chow, M. F. Mokbel, and W. G. Aref, "Casper*: query processing for location services without compromising privacy," *ACM Transactions on Database Systems (TODS)*, vol. 34, no. 4, article 24, 2009.

[18] H. Jin and P. Papadimitratos, "Resilient privacy protection for location-based services through decentralization," in *Proceedings of the the 10th ACM Conference*, pp. 253–258, Boston, Mass, USA, July 2017.

[19] H. Jadallah and Z. Al Aghbari, "Aman: Spatial cloaking for privacy-aware location-based queries in the cloud," in *Proceedings of the International Conference on Internet of Things and Cloud Computing, ICC 2016*, New York, NY, USA, March 2016.

[20] F.-Y. Tai, J.-K. Song, Y.-C. Tsai, and H.-P. Tsai, "Cloaking sensitive patterns Td preserve location privacy for LBS applications," in *Proceedings of the 3rd IEEE International Conference on Consumer Electronics-Taiwan, ICCE-TW 2016*, Nantou, Taiwan, May 2016.

[21] X. Chen and J. Pang, "Measuring query privacy in location-based services," in *Proceedings of the the second ACM conference*, pp. 49–60, San Antonio, Tex, USA, Feburary 2012.

[22] M. Li, Z. Qin, and C. Wang, "Sensitive semantics-aware personality cloaking on road-network environment," *International Journal of Security and Its Applications*, vol. 8, no. 1, pp. 133–146, 2014.

[23] Y. Huang, Z. Huo, and X.-F. Meng, "Coprivacy: a collaborative location privacy-preserving method without cloaking region," *Chinese Journal of Computers*, vol. 34, no. 10, pp. 1976–1985, 2011.

[24] Y. Cai and G. Xu, *Cloaking with Footprints to Provide Location Privacy Protection in Location-Based Services*, U.S. Patent, 2017.

[25] C. Li and B. Palanisamy, "De-anonymizable location cloaking for privacy-controlled mobile systems," in *Proceedings of the International Conference on Network and System Security*, pp. 449–458, Springer, Cham, Switzerland.

[26] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proceedings of the 2nd International Conference on Pervasive Services (ICPS '05)*, pp. 88–97, IEEE Press, Santorini, Greece, July 2005.

[27] H. Kido, Y. Yanagisawa, and T. Satoh, "Protection of location privacy using dummies for location-based services," in *Proceedings of the 21st International Conference on Data Engineering Workshops (ICDEW '05)*, p. 1248, Tokyo, Japan, April 2005.

[28] M. Guo, N. Pissinou, and S. S. Iyengar, "Pseudonym-based anonymity zone generation for mobile service with strong adversary model," in *Proceedings of the 2015 12th Annual IEEE Consumer Communications and Networking Conference, CCNC 2015*, pp. 335–340, Las Vegas, NV, USA, January 2015.

[29] B. Palanisamy and L. Liu, "Attack-resilient mix-zones over road networks: Architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 14, no. 3, pp. 495–508, 2015.

[30] B. Niu, Z. Zhang, X. Li, and H. Li, "Privacy-area aware dummy generation algorithms for location-based services," in *Proceedings of the1st IEEE International Conference on Communications, ICC ('14)*, pp. 957–962, Sydney, Australia, June 2014.

[31] S. Papadopoulos, S. Bakiras, and D. Papadias, "Nearest neighbor search with strong location privacy," in *Proceedings of the VLDB Endowment*, pp. 619–629.

[32] K. Mouratidis and M. L. Yiu, "Shortest path computation with no information leakage," in *Proceedings of the VLDB Endowment*, pp. 692–703.

[33] Z. Liao, L. Kong, X. Wang et al., "A visual analytics approach for detecting and understanding anomalous resident behaviors in smart healthcare," *Applied Sciences (Switzerland)*, vol. 7, no. 3, article no. 254, 2017.

[34] A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy," *Advances in Spatial and Temporal Databases*, pp. 239–257, 2007.

[35] R. Paulet, M. G. Kaosar, X. Yi, and E. Bertino, "Privacy-preserving and content-protecting location based queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1200–1210, 2014.

[36] R. Lu, X. Lin, Z. Shi, and J. Shao, "PLAM: A privacy-preserving framework for local-area mobile social networks," in *Proceedings of the 33rd IEEE Conference on Computer Communications, IEEE INFOCOM 2014*, pp. 763–771, Canada, May 2014.

[37] Z. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, "Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Transactions on Communications*, vol. 98, no. 1, pp. 190–200, 2015.

[38] R. Verde, F. A. T. Carvalho, and Y. Lechevallier, *A Dynamical Clustering Algorithm for Multi-Nominal Data*, Data Analysis, Classification, and Related Methods, Springer, Berlin, Germany, 2000.