Zhang *et al. EURASIP Journal on Wireless Communications and Networking* (2019) 2019:93
https://doi.org/10.1186/s13638-019-1406-4

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

An anonymous entropy-based location privacy protection scheme in mobile social networks

Lina Ni[1], Fulong Tian[1], Qinghang Ni[2], Yan Yan[1] and Jinquan Zhang[1*]

Abstract

The popularization of mobile communication devices and location technology has spurred the increasing demand for location-based services (LBSs). While enjoying the convenience provided by LBS, users may be confronted with the risk of privacy leakage. It is very crucial to devise a secure scheme to protect the location privacy of users. In this paper, we propose an anonymous entropy-based location privacy protection scheme in mobile social networks (MSN), which includes two algorithms K-DDCA in a densely populated region and K-SDCA in a sparsely populated region to tackle the problem of location privacy leakage. The K-DDCA algorithm employs anonymous entropy method to select user groups and construct anonymous regions which can guarantee the area of the anonymous region formed be moderate and the diversity of the request content. The K-SDCA algorithm generates a set of similar dummy locations which can resist the attack of adversaries with background information. Particularly, we present the anonymous entropy method based on the location distance and request contents. The effectiveness of our scheme is validated through extensive simulations, which show that our scheme can achieve enhanced privacy preservation and better efficiency.

Keywords: Location privacy protection, Location-based service, *k*-anonymity, Anonymous entropy, kd-tree

1 Introduction

Nowadays, the Internet of Things (IoT) is building a connected world seamlessly and enhancing the quality of our daily life throughout applications coming from consumer, commercial, industrial, and infrastructure spaces [1–3]. It provides more intelligent services and makes them more efficient via accessing to and storage as well as processing of data [4–7]. An increasing number of people would prefer making use of smart mobile devices to access the Internet for social activities; thus, the emerging mobile social networks (MSN) have promoted the development of a new application pattern of location-based services (LBS) which is a location-based value-added service provided by a location service provider and brings great convenience to our lives [8–10].

As we know, LBS can be used in location-based point of interest retrieval service, navigation service, social service, motion detection service, advertisement push service, and so on [11, 12]. When using LBS, users may need to send the personal identification information, the location information, or the request content to the LBS server. The service provider receives the request and processes it to provide the user location-based services, such as Meituan WaiMai, Didi ChuXing, and BaiDu Map. Unfortunately, LBS could explore the preferences and behavior patterns of users by analyzing the users' location information [13, 14]. If these information is abused or resold by the service provider or intercepted by the attacker, the users' location information may be disclosed and the potential threat may be posed to them. Therefore, the location privacy protection is of crucial challenge [15, 16].

Current system architectures for location privacy protection include standalone architecture, distributed peer-to-peer architecture, and central server architecture. At present, most of the location privacy protection methods based on the *k*-anonymity model adopt the central server

*Correspondence: tjzhangjinquan@126.com
[1]College of Computer Science and Engineering and Shandong Province Key Laboratory of Wisdom Mine Information Technology, Shandong University of Science and Technology, Qingdao, Shandong, China
Full list of author information is available at the end of the article

architecture. In this paper, we employ this architecture in our scheme as well and assume that the central anonymous server is trusted as in many methods [13, 15, 17]. The central server architecture consists of mobile users, LBS servers, and location anonymous server [17], as depicted in Fig. 1. As the core of the whole architecture, the anonymous server is responsible for processing anonymously the information such as the users' real location, filtering the candidate query result set returned by the service provider and then returning them to the users who send LBS queries.

In the real MSN scenario, areas where many mobile users using LBS are divided into densely populated regions and sparsely populated regions. In densely populated regions, numerous mobile users send LBS requests simultaneously, while few mobile users send LBS requests or the content requested by the user is single in sparsely populated regions. As for the location privacy protection of mobile users in these two regions, it is of equal importance. The existing location privacy protection methods generally consider the location privacy protection in sparsely populated regions or in densely populated regions separately or do not explicitly indicate the specific region. In fact, the location privacy protection method is different in these two regions. In this work, we propose the K-DDCA algorithm in densely populated regions and the K-SDCA algorithm in sparsely populated regions respectively to protect the location privacy of mobile users, comprehensively considering the user location distance and request content and combining with the kd-tree algorithm.

In recent years, many techniques have been proposed to solve the problem of location privacy protection, such as location perturbation and obfuscation [18, 19], region anonymization [20–22], and dummy location [23]. The region anonymization technique, which reduces the probability of identifying the real user to $1/k$, is an important one for location privacy protection. However, there are some problems with this technique. First, the region anonymization technique may form redundant regions in the process of constructing an anonymous region without considering the diversity of the request contents. Second, the neighbors constructing anonymous region may be in the vicinity of real users, which may expose the real users' location and reduce the user's experience. Meanwhile, this technique fails to solve the issue that the anonymity cannot satisfy the user's demand, resulting in privacy leakage of mobile user locations in sparsely populated regions, which poses a challenge to location privacy protection.

In this paper, we propose an anonymous region constructing algorithm based on kd-trees in densely populated regions (K-DDCA), which employs the kd-tree algorithm [24, 25] to search neighbor users. Compared with other neighbor search algorithms, the kd-tree algorithm can improve the search efficiency. Furthermore, according to our proposed anonymity entropy, it selects the nearest neighbor users that meet the requirements and achieve k-anonymity among the returned candidate sets of neighbor users together with the real users to construct an anonymous region. When constructing an anonymous region, we comprehensively consider the distance between the real requesting users and their neighbor users as well as the difference of the content requested by them. Thus, in a densely populated region like a school, the distance between neighbor users may be small, making it easier for an attacker to infer the user's region. However, since we consider the distance and the content of the user's requests and add randomness when constructing the user groups, it can effectively protect the user's identity information and request contents, and the attacker cannot associate a request with a user.

The dummy location generation technique [26, 27] solves the problem of user location privacy leakage caused by neighbor users in sparsely populated regions without satisfying the requirement of anonymity. This technique

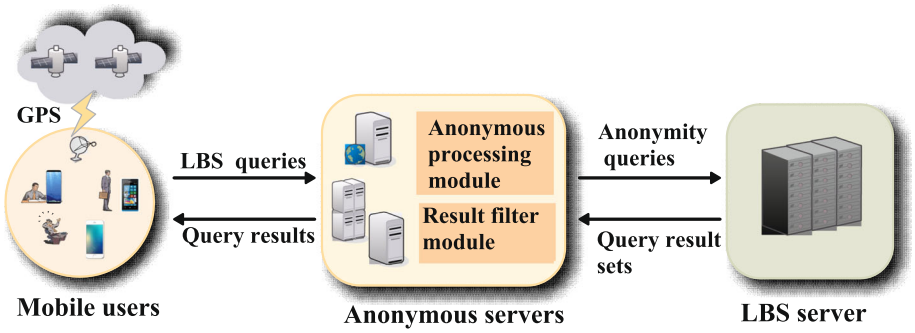


Fig. 1 Central server architecture. It consists of mobile users, LBS servers, and location anonymous server. As the core of the whole architecture, the anonymous server is responsible for processing anonymously the information such as the users' real location, filtering the candidate query result set returned by the service provider and then returning them to the users who send LBS queries

does not rely on trusted third-party servers to build an anonymous region with the dummy location data and the real location of the requesting users, reducing the communication overhead. However, this technique does not fully consider the context when generating data. For example, the generated dummy location points may be located in sparsely populated regions such as lakes, rivers, and swamps. If the attacker has mastered certain background knowledge (such as maps and historical query records), the dummy location can be easily filtered out by the adversaries, which cannot satisfy the user's anonymity requirement.

Considering this problem of typical dummy location generation technique, we propose an anonymous region constructing algorithm based on kd-trees in sparsely populated regions (K-SDCA). This algorithm takes into account the temporal and spatial factors and finds out multiple candidate users based on historical access records, thus preventing the selected users from clustering together. K-SDCA utilizes anonymous entropy to select user groups with uniform geographical distribution and large differences in request content. In this way, even if the attacker has mastered a certain background knowledge, the dummy users cannot be easily filtered out, thereby achieving the user's anonymity requirement.

Our main contributions are listed as follows:

1) We present a more efficient location privacy protection scheme based on anonymous entropy in MSN, which utilizes central server architecture and local area anonymization to protect the location privacy in both dense region and sparse region.
2) We propose two location privacy protection algorithms K-DDCA in a densely populated region and K-SDCA in a sparsely populated region to construct anonymous regions.
3) We propose the anonymous entropy method to effectively and securely select user groups based on the location distance and request contents and further construct anonymous regions, which can guarantee that the area of the anonymous region formed be moderate, and ensure the diversity of the request content.

The rest of the paper is organized as follows: Section 2 introduces the problems we have studied, the proposed methods, and the simulation environment. We discuss the related work in Section 3. Section 4 introduces the basic concepts and the kd-tree algorithm. Section 5 introduces the system structure model, together with our algorithm design, and the security analysis. Section 6 shows the evaluation results and discussion. We conclude this paper in Section 7.

2 Methods/experimental

In this paper, we study the issue of location privacy protection in MSN employing kd-tree as the storage structure. Specially, we put forward an *anonymous entropy* method based on on the location distance and request contents to tackle the problem of location privacy leakage. Our scheme consists of two algorithms K-DDCA in a densely populated region and K-SDCA in a sparsely populated region.

Our experimental environment is 64 bit Windows 7 system with Intel (R) Core i7-8700k CPU @ 3.70 GHz, and the RAM of 32G. The programming language is Python on PyCharm. We adopt OPNET [28, 29] to generate data and execute simulation experiments. A large number of experiments and security analysis prove that our scheme has high security and better efficiency.

3 Related work

The privacy protection has been well studied in MSN. In this section, we introduce some work on location privacy protection in LBS related to our scheme. At present, the privacy-preserving schemes are categorized into three methods including the spatial cloaking method [30–32], dummy location method [33–35], and cryptography primitive-based method [36–38].

3.1 Spatial cloaking method

The spatial cloaking technology [30] forms a hiding area that contains k real users for each user employing LBS, which makes it difficult for service providers to determine the real identities and accurate locations of users from the hiding region.

In [30], Gruteser et al. first introduced k-anonymity into location privacy and proposed an anonymous usage method of LBS, which obscures the exact location of the real user into an anonymous region. It reduces the probability that the attacker accurately infers the actual location of the user to $1/k$. In [31], Abul et al. proposed a quad-tree-based anonymity algorithm which adopts a recursive method to continuously divide the space region in which the mobile user resides into four quadrants. During constructing an anonymous region, it starts from the quadrant where the real user requests the service and expands to the parent quadrant until the user's anonymity attains k (i.e., at least k users in the anonymous region). However, the anonymous region generated by this method is too large and there may exist a large amount of redundant space, resulting in a decline in service quality. The reason for the existence of redundant space is that the algorithm does not consider the distribution of users with their adjacent quadrants in the formation of anonymous region. Meanwhile, due to lack of users in sparsely populated regions, the anonymous region will fail to be constructed.

In [32], Mokbel et al. proposed an anonymous algorithm based on the Casper model, which effectively improves the performance of the anonymity algorithm in [30]. When the user's quadrant does not satisfy k-anonymity, it first expands the anonymous region to the adjacent quadrant and then expands to the parent quadrant if k-anonymity has not been reached yet. Unfortunately, when the real user is located in a sparsely populated region, the Casper algorithm will fail to construct the anonymous region due to lack of sufficient dummy locations. Since the Casper algorithm still uses quad-tree to partition the spatial region and fails to consider the distribution of the target user's adjacent users when it merges the quadrants, there exists redundant region in the anonymity region constructed. Meanwhile, the Casper anonymous algorithm will gradually increase the area of the merged quadrant after it is recursively extended to the parent quadrant; thus, the next merge may be incorporated into the redundant region.

3.2 Dummy location method

The dummy location approach [33] generates multiple dummy locations and integrates the users' real locations into the dummy ones and sends them to the service provider for privacy protection. Since the service providers cannot distinguish between real locations and dummy ones, they can only provide the required services for each submitted location.

In [33], Niu et al. proposed DLS and enhanced DLS algorithms by selecting several candidate dummy locations with similar query probabilities. In [34], Wu et al. formulated a multi-objective optimization (MOS) algorithm considering both the query probability and the area of the anonymous region to generate $k-1$ dummy locations and achieve k-anonymity. Although these methods reduce the possibility that some location points are filtered out, they do not take into account the inherent differences between the requests. Moreover, since the procedure of the algorithms takes place on mobile clients, the amount of computing is relatively large and they have high requirements on computing power and storage space of mobile clients.

In [35], Liao et al. proposed the K-DLCA algorithm considering the semantic information and the historical query probability of the locations and using greedy strategy to select each dummy location that can maximize the current entropy. However, since the selected dummy locations are located near the real user, this algorithm may expose the real user's location.

3.3 Cryptography primitive-based method

The cryptography primitive-based approach [36–38] mainly employs related cryptography to process the requests sent by users, which can protect users' privacy information and obtain the service data. This type of scenario does not require an anonymous server and just needs to encrypt the user's location information and send it to the service provider. The service provider decrypts and executes the corresponding query, then encrypts and returns the results. Thus, the cryptography primitive-based approach has excellent security. However, the obvious drawback is that the computational overhead is comparatively large. Therefore, the feasibility of such schemes is relatively poor.

In our strategy, the K-DDCA algorithm employs kd-tree to search nearby users in the densely populated region. During the generation of anonymous regions, neighbor users are selected according to the distance and the request content among users without generating redundant regions, reducing the time complexity and improving the quality of service. In sparsely populated regions, considering the historical records and geographical distribution, the K-SDCA algorithm achieves k-anonymity by selecting dummy users with high historical query probability, relatively uniform geographical distribution, and large difference in request contents.

4 Preliminaries

In this section, we first present some concepts and then introduce the idea and procedure of the kd-tree algorithm. Specifically, we devise an anonymous entropy method to construct the anonymous region.

4.1 Basic concepts

Here, we give some basic concepts to lay a foundation for our scheme.

Definition 1 *(k-anonymity [39]) k-anonymity is proposed by Sweeney et al. as a privacy protection mechanism in data publishing. A data object is said to have the k-anonymity property if the information for this target data object cannot be distinguished from at least the other* $k-1$ *individuals whose information also appear in the data publishing. Thereafter, k-anonymity is considerablely extended to protect location privacy and query privacy in LBS.*

In this paper, we adopt the classical notion of *Euclidean distance* as our *distance measure*.

Definition 2 *(Distance measure) Assume that the locations of two users* u_i *and* u_j *are* (x_i, y_i) *and* (x_j, y_j)*, respectively, the distance measure (i.e., Euclidean distance) between* u_i *and* u_j*, denoted by dist, is defined as*

$$dist\left(u_i, u_j\right) = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}, \tag{1}$$

where x_i, y_i represent the longitude and latitude of the location of user u_i, respectively.

Definition 3 *(Distance between user and line) Assume that two users u_i and u_j form a straight line, denoted by $line_{ij}$, then the distance between user u_l and line $line_{ij}$, denoted by d_{ul}, is defined as*

$$d_{ul} = \frac{|\left(y_j - y_i\right) x_l + \left(x_i - x_j\right) y_l + \left(x_j y_i - x_i y_j\right)|}{\sqrt{\left(y_j - y_i\right)^2 + \left(x_i - x_j\right)^2}}, \quad (2)$$

where (x_i, y_i), $\left(x_j, y_j\right)$, and (x_l, y_l) are the locations of users u_i, u_j, and u_l, respectively.

Definition 4 *(Region partition) Assume that Reg is a region and its radius is r. Based on historical experience, we set thresholds MinU and MinC which represent the minimum number of users and the minimum number of request content types of Reg, respectively. For any $u_i \in Reg$, the following region partitions hold:*

All the users in *Reg* who satisfied the condition whose distance from u_i is not greater than r are called *r-neighborhood* of u_i, denoted by $N_r(u_i)$, that is,

$$N_r(u_i) = \{u_j \in Reg | dist\left(u_i, u_j\right) \le r\}. \quad (3)$$

All the users in *Reg* who satisfied the condition whose request contents are different from u_i are called θ-*neighborhood* of u_i, denoted by $N_\theta(u_i)$, that is,

$$N_\theta(u_i) = \{u_j \in Reg | boolean\left(u_i, u_j\right) = False\}, \quad (4)$$

where *boolean()* is a boolean value, which represents whether the request contents of the users are same.

Definition 5 *(Densely/sparsely populated region) Assume there are at least MinU concurrent users initiating the request at the same time as u_i in the r-neighborhood of user u_i, that is, $|N_r(u_i)| \ge MinU$, and the θ-neighborhood of u_i contains at least MinC kinds of request contents, that is, $|N_\theta(u_i)| \ge MinC$, then we call Reg as a densely populated region; otherwise, the region is called a sparsely populated region.*

In Definition 5, *MinU* and *MinC* are all based on historical experience.

Definition 6 *(Historical query probability) Assume that Reg is divided into $n * n$ cells. Note that, according to the number of historical queries in the cell, each cell may have its own probability of being queried. Then, the historical query probability of cell i is the ratio of the number of cell i queried versus the total number of queries for all cells in Reg, denoted by p_i, that is,*

$$phq_i = \frac{QCell_i}{\sum_{j=1}^{n^2} QCell_j}, \quad \left(i = 1, \cdots, n^2\right), \quad (5)$$

where $QCell_i$ is the number of the ith cell queried and $\sum_{j=1}^{n^2} QCell_j$ is the total number of queries in the whole region. Obviously, we have $\sum_{i=1}^{n^2} phq_i = 1$.

Definition 7 *(Entropy [33]) Entropy is used to measure the uncertainty of a set, the bigger the value of the entropy the more uncertain of the set. Formally, it is defined as*

$$H(R) = -\sum_{i=1}^{n} p_i \log p_i, \quad (6)$$

where p_i represents the probability that the user is identified.

Many methods [33] use *entropy* to evaluate the anonymity of anonymous regions. In this paper, we adopt the same evaluation criteria as well.

4.2 Anonymous entropy method

In order to select users who are evenly distributed with real users, it is needed to consider the distance and the request content among users. Therefore, we propose the anonymous entropy method to construct the anonymous region.

Suppose that there are $2k$ users in *Reg*, $k-1$ users are selected randomly to form a user group U_i with real users. The process is repeated m times, and m user groups are formed, where m is defined by the user according to his privacy requirements.

4.2.1 Entropy of distance

On the one hand, the distance among users is considered. The goal of the anonymous entropy method on distance is as follows. If the sum of distances between $k-1$ users and the real user among m user groups are equal, the user group which is evenly distributed is selected. Otherwise, if the total distances are not equal, the user group with a larger distance is selected. In order to achieve the above goal, the *entropy* is used to select a user group on distance. Here, the weight of the neighbor user u_i in the nth user group is denoted as α_{ni}, that is,

$$\alpha_{ni} = \frac{dist\left(u_{real}, u_i\right)}{\sum_{j=1}^{2k} dist\left(u_{real}, u_j\right)} \quad (i = 1, \cdots, k-1). \quad (7)$$

Then, the *entropy* of the nth user group for distance is

$$H(n) = -\sum_{i=1}^{k-1} \alpha_{ni} \log \alpha_{ni}. \quad (8)$$

Figure 2 depicts a scenario that MSN users request LBS service, which is an illustration example of the neighbor user selection. Dots represent the users, where red dot A represents the real user and other seven black dots represent the neighbor users. Eight users with labels $\{A, B, \cdots, G\}$ are deployed in the region. Lines between the dots indicate the spatial neighbor relation between the users, where the digits marked on the lines indicate the distance. Suppose that there are three user groups $G_1 = \{A, B, C, F\}$, $G_2 = \{A, D, E, F\}$, and $G_3 = \{A, C, D, G\}$.

The total distance of users in G_1 to the real user A is less than those of G_2 and G_3. The total distance between G_2 and G_3 is equal. As shown in Fig. 2, G_2 is evenly distributed; however, G_3 is not uniformly distributed from the real user. According to formula (8), the *entropy* on distance of each group is obtained shown as $H(G_1) = 1.2174$, $H(G_2) = 1.3415$, and $H(G_3) = 1.3155$. It can be seen that G_2 has the largest *entropy* for distance; thus, it has the greatest degree of anonymity. According to the principle of anonymous entropy method, the user group G_2 is chosen here.

4.2.2 Content weights in user groups

On the other hand, the request content among users is considered. A weight is assigned to the request content in the user group based on the number of request types and the distribution in the user group, that is, the more request types in the user group and the more uniform the request type distribution, the greater the content weight of the user group. The weight of the request content is denoted by β_n, that is,

$$\beta_n = \frac{2boolean\left(u_i^c, u_j^c\right)}{k(k-1)}, \quad (i, j = 1, \cdots k, i \neq j), \quad (9)$$

where u_i^c and u_j^c represent the request contents of u_i and u_j, respectively.

As shown in Fig. 2, suppose a user group G_A contains five users. A is the real user and $\{B, C, D, E\}$ is the selected neighbors. Assume that there are three types of request contents in G_A, such as the requests of the adjacent hospital, hotel, and shopping mall. Then, we have two forms of request type distribution in user groups, $G_4 = \{\Diamond, \square, \Delta\Delta\Delta\}$, $G_5 = \{\Diamond\Diamond, \square\square, \Delta\}$, where the symbols $\Diamond$, $\square$, and Δ represent different types of request content from user, and each interval separated by a comma represents the same request type. For example, we can suppose that $G_4 = \{A, B, CDE\}$, $G_5 = \{AB, CD, E\}$. In G_4, user A may request the location of the adjacent hospital, user B may request the hotel, while users C, D, and E may request the shopping mall. Or user E may request the location of the adjacent hospital, user D may request the hotel, while users A, B, and C may all request the shopping mall. But they have the same request distribution. According to formula (9), the weight of each form of user group can be calculated as $\beta_{G_4} = 7/10$, $\beta_{G_5} = 8/10$.

If there are two types of request contents in G_A, the distribution of the request types in G_A is divided into two forms: $G_6 = \{\Diamond, \Delta\Delta\Delta\Delta\}$, $G_7 = \{\Diamond\Diamond, \Delta\Delta\Delta\}$. For example, suppose that $G_6 = \{A, BCDE\}$, $G_7 = \{AB, CDE\}$. According to formula (9), the weights are calculated to be $\beta_{G_6} = 4/10$, $\beta_{G_7} = 6/10$, respectively. According to

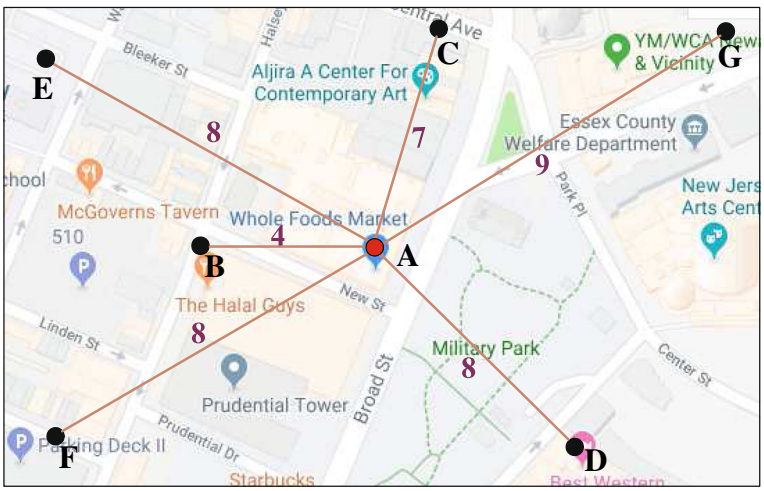


Fig. 2 Illustration of the neighbor user selection. It depicts a scenario that MSN users request LBS service. Dots represent the users, where red dot A represents the real user and other seven black dots represent the neighbor users. Eight users with labels $\{A, B, \cdots, G\}$ are deployed in the region. Lines between the dots indicate the spatial neighbor relation between the users, where the digits marked on the lines indicate the distance

the principle of anonymous entropy method, here the user group is selected who has the most number of request types and request types evenly distributed among users. In this example, the user group G_5 is selected who has three request types and even distribution.

4.2.3 Combination of distance and content metric

Taking into account the distance between users in the user group and the distribution of the request content jointly, the anonymous entropy for the nth user group is defined as the sum of the *entropy* in terms of distance and the difference on request content in the nth user group, denoted by

$$HA(R) = -\sum_{i=1}^{k-1} \alpha_{ni} \log \alpha_{ni} + \beta_n. \tag{10}$$

Now, the user group with the largest anonymous entropy is selected in m user groups as the anonymous region, represented as

$$U_{\max} = arg \max_{n\in(1,\cdots,m)} \{HA_n\}. \tag{11}$$

The above is the anonymous entropy method we proposed. In the following, we will illustrate our motivation for calculating the anonymous entropy based on the difference in distance and the request content.

4.2.4 Illustration of the selection of distance and content metric

Figure 3 shows three real scenarios that users send requests to the LBS server. In each subgraph, the left part represents the user's anonymous region and the right part shows the diversity of content requested by the user respectively. The lines connecting the two parts represent the relationship between the user request and the corresponding LBS service.

Fortunately, as the figure shows that Fig. 3a represents a perfect scenario where the location privacy of the user is protected because both the area of the anonymous region and the difference of the request content are all larger. Unfortunately, on the one hand, Fig. 3b shows that the user's content privacy information can be inferred according to the user's LBS request, namely, content privacy inferred. On the other hand, Fig. 3c shows that the user's real location can be inferred based on the user's LBS request, namely, location privacy inferred.

Note that Fig. 3b and c indicate two imperfect scenarios. In Fig. 3b, since the request contents are the same with different users, there exists risk of privacy leakage. For example, if the content requested by the four users are all nearby hospital, the probability of the users being sick is the greatest, and the attacker may use this information to propose a targeted strategy to attack those users. In Fig. 3c, although the request content is different, unfortunately, they are issued by the same location (i.e., multiple users are clustered at the same location). Thus, the area of the anonymous region is too small so the user's real location could be easily inferred by the attacker.

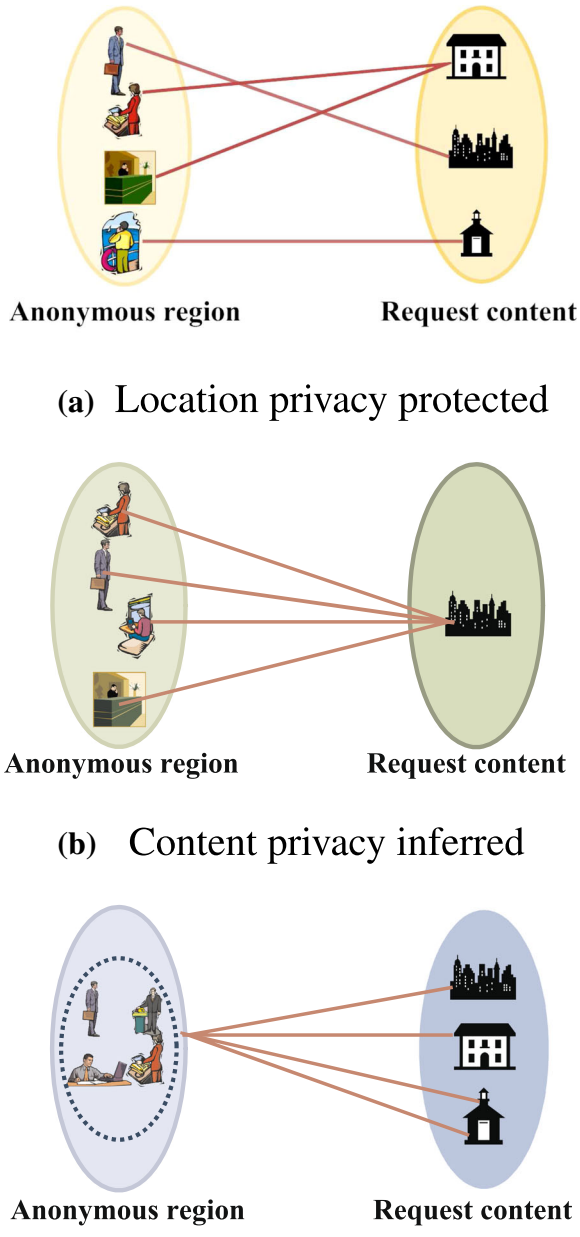


Fig. 3 LBS request type It shows three real scenarios that users send requests to the LBS server. In each subgraph, the left part represents the user's anonymous region and the right part shows the diversity of content requested by the user. The lines connecting the two parts represent the relationship between the user request and the corresponding LBS service. **a** Location privacy protected. **b** Content privacy inferred. **c** Location privacy inferred

In a word, we propose the anonymous entropy method comprehensively considering the distance between the request users and the difference of the request contents, which can effectively protect the privacy of the users.

4.3 kd-tree algorithm

kd-tree is a kind of balanced binary tree that divides data points in k-dimensional space, which is mainly applied to the search of key data in multi-dimensional space [24, 25]. Essentially, it is a spatial partitioning tree. In this paper, we use the nodes of kd-tree to store the users' locations. The operation of kd-tree is divided into two phases including kd-tree construction and kd-tree search [24].

4.3.1 Phase 1: kd-tree construction

The detailed steps of constructing a kd-tree are as follows:

(1) Construct the *root* node, which represents the region containing all the users, denoted by U.

(2) Select x axis as the split axis and the median of U's x coordinates as the segmentation point. Divide the corresponding region of the *root* node into two sub-regions, which is achieved by the line that is through segmentation point and perpendicular to the split axis. Thus,

- The *left child* node of the kd-tree corresponds to the sub-region whose x coordinates are smaller than the segmentation point.
- The *right child* node corresponds to the sub-region whose coordinates are greater than the segmentation point.
- The *root* node stores the points that fall on the split axis.

(3) For each node whose *depth* is j, calculate $d = j \ mod \ 2$. If $d = 0$, select y axis as the split axis. Otherwise, x axis is selected as the split axis.

The median of the split axis corresponding to the region of the node is used as the segmentation point. Thus, the corresponding region of the node is divided into two sub-regions.

(4) The process is iterated until there is no location data in these two sub-regions. The construction of kd-tree is completed.

4.3.2 Phase 2: kd-tree search

In a kd-tree, the neighbor search is achieve by maintaining a queue of $2k$ nodes. The detailed steps of searching a kd-tree are as follows:

(1) Find the *leaf* nodes containing the real user.

- Recursively visit the kd-tree downwards from the *root* node.
- If the coordinates of the real user on the current split axis are smaller than those of the segmentation points, then move to the *left child* node to search real user; otherwise, move to the *right child* node until the *child* node is a *leaf* node.
- Now, this *leaf* node is treated as the current nearest point and is put into the queue.

(2) Recursively roll back upwards the search path and perform the following operations on each node.

- If a location point saved by this node that is closer to the real user than the current nearest point is found, the current location point is taken as the current nearest point and placed in the queue. If the queue is full, the head of the queue is dequeued.
- Draw a circle around the center of the requesting user with the distance from the requesting user to the nearest point.
- Check whether the region corresponding to another *child* node intersects the circle. If it intersects, there may exist points closer to the request user in the another region. Then, move to another region and recursively search for the nearest neighbor. If they do not intersect, go back upwards.

(3) When going back to the *root* node, the search ends if it does not intersect another sub-region.

From the above description, it can be seen that the ineffective nearest neighbor search can be greatly reduced after the construction of kd-tree. Since many locations do not intersect with the circle, there is no need to calculate the distance at all, which saves a lot of computing time.

5 Anonymous entropy-based location privacy protection scheme

In this section, we illustrate the system model of our scheme and elaborate the details of the algorithm design as well as security analysis.

5.1 System model

The architecture of our proposed system model consists of the following three components: (1) MSN users, (2) anonymous servers, and (3) LBS providers, as shown in Fig. 4. Our system model is inspired by the classical central server architecture. However, our system has more superiorities on anonymous servers.

1) MSN users: MSN user sends a LBS request $LBSQ$ to an anonymous server through mobile terminals, formally, $LBSQ = \{id, (x, y), c, k, A\}$, where id and (x, y) are the identity and location coordinates of the user, respectively; c represents the content of the request services; k represents the privacy degree set by the users, namely, the number of users in the anonymous region; and A denotes the minimum area of the anonymous region.

2) Anonymous servers: Anonymous server mainly consists of two modules, that is *anonymous processing module* and *candidate result filter module*.

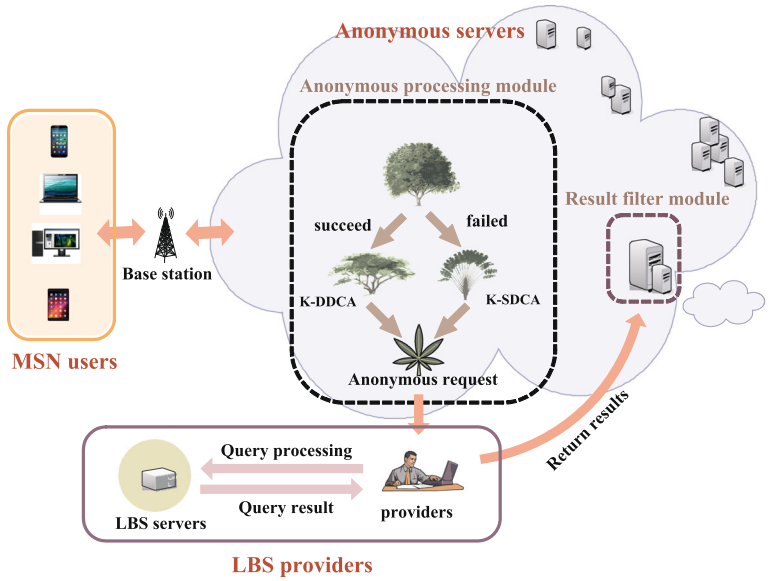


Fig. 4 Architecture of the proposed system model. It consists of the following three components: (1) MSN user, (2) anonymous servers, and (3) LBS providers. The system model is inspired by the classical central server architecture, and it has more superiorities on anonymous servers

The anonymous processing module mainly deals with the anonymity of MSN users, which is the core of our scheme. It involves in two algorithms K-DDCA and K-SDCA, which will be elaborated whereafter.

After receiving the LBS requests sent by users, the anonymous server anonymously processes the location privacy, identity, request content, and other information according to the requirements of the users. The procedure is as follows:

1) Construct the kd-tree based on the region where the requesting user is located.
2) If the kd-tree is constructed successfully, the nearest neighbor users are searched on the kd-tree, and the anonymity is processed according to the K-DDCA algorithm.
3) If the kd-tree fails to be constructed or the K-DDCA algorithm fails to process anonymously, the K-SDCA algorithm is executed anonymously, and then send the anonymous request to the *LBS provider*.

The candidate result filter module filters the result set returned by the LBS provider according to the real location of the requesting user and finally sends the accurate result to the *MSN user*.

3) LBS providers: According to the received LBS requests, LBS provider executes calculation to find out the candidate result set required by the MSN user and returns it to the anonymous server.

5.2 Algorithm design

In this paper, we adopt the central server architecture as our system architecture, which can reduce the requirements of the high computing power and storage space of the mobile terminals. Our scheme is divided into two processing alternatives including both K-DDCA algorithm in the densely populated region and K-SDCA algorithm in the sparsely populated region.

5.2.1 K-DDCA

In the densely populated region, we propose the K-DDCA algorithm to construct anonymous regions. K-DDCA mainly includes the following procedures.

It is notable that we use kd-tree to store user data. After the successful construction of kd-tree, we find the nearest $2k$ users of real users according to the kd-tree search algorithm [24]. We fully consider the location relationship among the neighbor users to ensure that no redundant regions are generated and time complexity is reduced. Here, the neighbor search is realized by maintaining a queue containing $2k$ nodes.

After finishing the search of the kd-tree, if $2k$ neighbors are found, then m user groups are formed according to the requirements of the real users. Each user group includes real users and the randomly selected $k-1$ users among $2k$ neighbor users.

According to our anonymous entropy method, the uncertainty of the user groups is evaluated. Then, the user group with the largest entropy is selected.

If the queue is not full after finishing the search of kd-tree or the diversity of request content of the final user group is less than the threshold value *MinC*, the real user is in a sparsely populated region. Then, the K-SDCA algorithm is employed to generate dummy locations.

The pseudo code of K-DDCA is elaborated in Algorithm 1.

Algorithm 1 : K-DDCA

Input: real user u_{real}, anonymity k, number of execution rounds m, threshold of content diversity *MinC*.
Output: anonymous region.

Initialize queue q and set its length to $2k$;
Use *kd-tree algorithm* to search for the nearest users to u_{real};
if the number of nearest users exceeds $2k$ **then**
Set the number of nearest users to $2k$;
end if
Store the nearest users in q;
if $|q| == 2k$ **then**
for $i = 1$ **to** m **do**
Select $k-1$ users randomly in $2k$ nearest neighbors and form a user group U with u_{real};
$HA_i \leftarrow \sum_{j=1}^{k-1} \alpha_{ij} \log \alpha_{ij} + \beta_i$;
end for
$U_{max} = arg \max_{n \in (1,\cdots,m)} \{HA_n\}$;
Calculate N_θ (u_{real}) according to formula (4)
if $|N_\theta$ $(u_{real})| \geq MinC$ **then**
Return U_{max};
else
u_{real} is located in sparsely populated regions, call *K- SDCA* in Algorithm 2 to generate dummy locations;
end if
else
u_{real} is located in sparsely populated regions, call *K-SDCA* in Algorithm 2 to generate dummy locations.
end if

Theorem 1 *The time complexity of the K-DDCA algorithm is O(logN), where N represents the total number of users in the region to be anonymized.*

Proof The K-DDCA algorithm is mainly divided into two phases. Firstly, K-DDCA needs to find the nearest $2k$ users from the real user based on the kd-tree algorithm [24]. Since the average computational complexity of the kd-tree search algorithm is $O(logN)$, the average time complexity of searching for neighbors is also $O(logN)$.

Then, K-DDCA selects $k-1$ neighbors among $2k$ users to achieve k-anonymity together with real users. We need to execute m rounds. In each round, K-DDCA forms a user group and calculates the anonymous entropy of the user group. Afterward, K-DDCA selects a user group with the largest entropy among them. Thus, the time complexity of this process is $O(m)$.

Due to $N \gg m$, the total time complexity of the K-DDCA algorithm is $O(logN + m) = O(logN)$. □

5.2.2 K-SDCA

Motivation of K-SDCA In a sparsely populated region, we utilize the K-SDCA algorithm to generate the dummy location. As the assumption before, the region where the real user is located is divided into $n * n$ cells. *Anonymous server* reads all the historical query probabilities of the locations near the real user in the whole region. Here, we adopt the method of calculating historical query probabilities proposed in [33]. The following is the construction process of anonymous region for the K-SDCA algorithm:

1) Firstly, considering the locations with similar query probability can be aggregated together to facilitate the search of targets, we sort all the historical query probabilities. After that, we can conveniently select the $3k$ cells with the similar historical query probability to the real user. The reason for selecting $3k$ cells is to construct an anonymous region with the appropriate number of candidates.
2) Secondly, the anonymous server constructs a smaller set of $2k$ candidate dummy locations from the $3k$ candidates using the heuristic method in order to exclude some dummy locations gathered together.
3) Thirdly, the anonymous server employs the anonymous entropy method to effectively and securely select $k-1$ users among $2k$ users and further achieve k-anonymity together with the real users.

Procedure of K-SDCA To clarify the procedure of K-SDCA, let u_{real} denote the cell where the real user is currently located. We employ the heuristic method to choose $(u_1, \cdots, u_{2k-1})$ in turn through $2k-1$ rounds, namely, u_1 is selected in the first round, u_2 is selected in the second round, and so on. In each round, each remaining candidate is endowed with a weight and the dummy user of this round is selected with a probability proportional to its weight. Let x denote the number of the remaining candidates in each round, and w_i denote the weight of u_i. The procedure of K-SDCA is as follows.

First, we calculate the distances among $3k$ users and u_{real}, sort them in reverse order, and then store them in list $\widehat{U}$.

In the odd rounds, each remaining candidate u_i in list $\widehat{U}$ is endowed a weight, which is the ratio of the distance between u_i and u_{real} to total distances in $\widehat{U}$, denoted by w_{oi}, that is

$$w_{oi} = \frac{dist\,(u_i, u_{real})}{\sum_{u_j \in \widehat{U}} dist\,(u_j, u_{real})}. \tag{12}$$

In the even rounds, a straight line $line_{real,p}$ is generated through u_{real} and the point selected in the previous round. Then, each remaining candidate u_i in the list $\widehat{U}$ is endowed a weight based on its distance from $line_{real,p}$, denoted by w_{ei}, that is

$$w_{ei} = \frac{d_{ul}\,(u_i, line_{real,p})}{\sum_{u_i \in \widehat{U}} d_{ul}\,(u_i, line_{real,p})}. \tag{13}$$

In each round, $u_i\,(i = 1, \cdots, x)$ is selected as a dummy user with probability

$$\frac{w_i}{\sum_{j=1}^{x} w_j}. \tag{14}$$

The selected user u_i is added to the selected user group U and then removed from list $\widehat{U}$.

The detailed pseudo code of K-SDCA is specified in Algorithm 2.

Theorem 2 *The time complexity of the K-SDCA algorithm is $O\,(NlogN)$, where N represents the total number of cells.*

Proof The K-SDCA algorithm is mainly divided into three phases. Firstly, K-SDCA sorts all the historical query probabilities via Shell sort and selects $3k$ cells with the similar historical query probabilities to real user. Since the average time complexity of Shell sort is $O\,(NlogN)$, the time complexity of this phase is $O\,(NlogN)$.

Secondly, K-SDCA selects $2k$ evenly distributed cells from the $3k$ cells through $2k$ rounds. The time complexity of selecting $2k$ dummy locations is $O(k)$, where k represents the number of users in the final anonymous region.

Thirdly, K-SDCA selects the user group with the maximum *entropy* among the m user groups to achieve k-anonymity after executing m rounds. Thus, the time complexity of this phase is $O(m)$.

Due to $N > k > m$, the total time complexity of the K-SDCA algorithm is $O(NlogN+k+m) = O\,(NlogN)$. □

Theorem 3 *K-DDCA and K-SDCA can achieve k-anonymity.*

Proof According to the definition of k-anonymity (Definition 1) in Section 4, obviously, the meaning of k-anonymity is to form a cloaking region containing k users for each query user. In this way, the real user will become indistinguishable from other $k-1$ users.

We apply the K-DDCA algorithm in a densely populated region. Specifically, K-DDCA forms m user groups, where each user group includes the real user and the randomly selected $k-1$ users among $2k$ neighbor query users. Then, the user group with the largest entropy is selected by utilizing the anonymous entropy method. According to the principle of anonymous entropy, this user group has the maximum uncertainty. Therefore, the k-anonymity can be achieved in the densely populated region.

Due to the lack of enough neighbor query users in a sparsely populated region, our K-SDCA algorithm selects

Algorithm 2 : K-SDCA

Input: number of execution rounds m, anonymity k, real user u_{real}, historical query probability of each cell p_l,
Output: anonymous region.

1: Initialize set U;
2: Initialize list $\widehat{U}$ and set its length to $3k$;
3: Read $p_l, l \in (1, \cdots, n^2)$ and sort all p_l;
4: Select the $3k$ cells with the similar query probability to u_{real};
5: Calculate the distances between $3k$ users and u_{real}, and store them in $\widehat{U}$;
6: Add u_{real} to U, $U \leftarrow U \cup u_{real}$;
7: **for** $i = 1$ **to** $2k$ **do**
8: **if** $i \bmod 2! = 0$ **then**
9: Assign each remaining candidate u_j in $\widehat{U}$ a weight according to formula (12);
10: Select $u_j \in \widehat{U}$ with probability $\frac{w_j}{\sum_{z=1}^{x} w_z}$;
11: $U \leftarrow U \cup u_j$;
12: Remove u_j from $\widehat{U}$;
13: **else**
14: Form a straight line between u_{real} and the point selected in the previous odd round;
15: Assign each remaining candidate u_j in $\widehat{U}$ a weight according to formula (13);
16: Select $u_j \in \widehat{U}$ with probability $\frac{w_j}{\sum_{z=1}^{x} w_z}$;
17: $U \leftarrow U \cup u_j$;
18: Remove u_j from $\widehat{U}$;
19: **end if**
20: **end for**
21: **for** $i = 1$ **to** m **do**
22: Select $k-1$ users randomly in U and form user group with u_{real};
23: $HA_i \leftarrow -\sum_{l=1}^{k-1} \alpha_{il} \log \alpha_{il} + \beta_i$;
24: **end for**
25: $U_{max} = arg \max_{n \in (1, \cdots, m)} \{HA_n\}$;
26: Return U_{max}.

neighbor users and builds user groups by means of historical records in the anonymous server. In addition, each user group contains a real user and $k-1$ historical query users. According to the anonymous entropy method, the user group which has the largest uncertainty among user groups is chosen. Thus, our designed K-SDCA algorithm can achieve k-anonymity in the sparsely populated region. □

Example of K-SDCA The illustration of the dummy location (user) selection is shown in Fig. 5, which depicts a scenario that users send requests to the LBS server in the sparsely populated region. The solid red dot A represents the real user, the solid black dots represent the selected dummy locations (users) by the K-SDCA algorithm, and the numbers indicate the order in which they are selected. The hollow dots represent the unselected dummy locations. In addition, the solid lines between two locations represent the connection between dummy locations selected in the odd round, and the dotted lines are vertical lines representing the distances from the location point to the straight line.

When real user A requests an LBS service, since there are no adequate users that send LBS requests simultaneously in a sparsely populated region, the anonymous servers have to utilize the historical locations where previous users sent the LBS requests to construct the anonymous region.

Anonymous servers first locate $3k$ locations according to similar query probability with A among these historical records and then store the selected locations to $\widehat{U}$.

Thus, in the first round, the number of remaining candidates $x = 3k$ and the weight of each candidate u_j in $\widehat{U}$ is

$$w_{oj} = \frac{dist\left(u_j, A\right)}{\sum_{u_i \in \widehat{U}} dist\left(u_i, A\right)}$$

according to formula (12). In Fig. 5, we can see that the cells far away from the real location A have the higher probability of being selected.

Since it may be easier to deduce the actual location in some cases, here we do not directly select the cell that is the farthest from A. The dummy locations are selected according to formula (14). When the dummy user E is selected, it is queued into U and removed from $\widehat{U}$, then a line $line_{A,E}$ between A and E is formed.

In the second round, the number of remaining candidates $x = 3k-1$, and according to the distance between u_j and $line_{A,E}$, the weight of each remaining candidate u_j in $\widehat{U}$ is

$$w_{ej} = \frac{d_{ul}\left(u_j, line_{A,E}\right)}{\sum_{u_i \in \widehat{U}} d_{ul}\left(u_i, line_{A,E}\right)}.$$

Then, we calculate the probability of u_j being selected by formula (14). Suppose the dummy user D is selected here. We add it to U and remove it from $\widehat{U}$.

Repeat the above process until $2k$ dummy locations $\{A, B, C, D, E, F\}$ in Fig. 5 are found. Subsequently, we use the anonymous entropy method to construct an anonymous region.

Since the query probability is similar, it is not easy to filter out the locations for an attacker. Furthermore, $2k$ evenly distributed locations are selected from the $3k$ candidates in order to prevent the selected locations from aggregating together which would expose the real users' location. Finally, k users are selected from $2k$ users to form

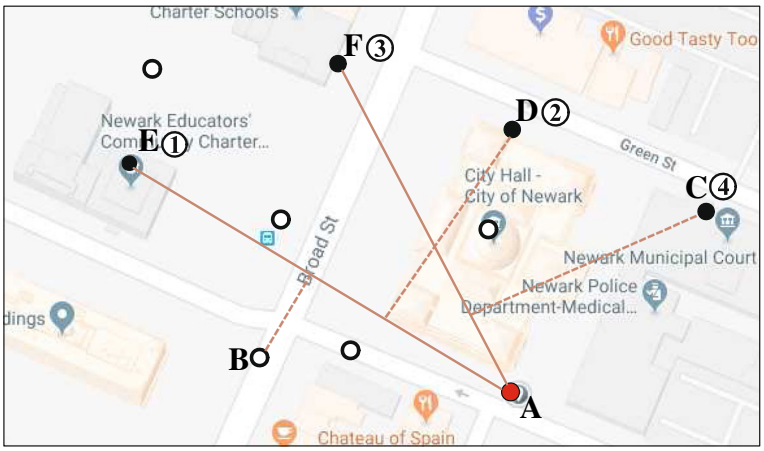


Fig. 5 Illustration of dummy location (user) selection. It depicts a scenario that users send requests to the LBS server in a sparsely populated region. The solid red dot A represents the real user, the solid black dots represent the selected dummy locations (users) by the K-SDCA algorithm, and the numbers indicate the order in which they are selected. The hollow dots represent the unselected dummy locations. The solid lines between two locations represent the connection between dummy locations selected in the odd round, and the dotted lines are vertical lines representing the distances from the location point to the straight line

m user groups. For each user group, one cell is the real location and the other $k-1$ cells are dummy locations. In addition, we need to calculate the anonymous entropy of each user group to select the user group with maximum anonymous entropy.

5.3 Security analysis

In some scenarios, an attacker may collude with some users to obtain additional information about other users. Furthermore, an attacker may also collude with a LBS service provider to utilize the information obtained by the LBS service provider to infer other sensitive information of legitimate users for profit. Fortunately, our scheme can resist the collusion attacks of adversaries and show stronger security. If the probability of successful deducing the real user's location does not increase with the size of colluding group, the algorithm is *colluding-attack resistant*.

We give the security analysis of our scheme in the following.

Theorem 4 *K-DDCA and K-SDCA are colluding-attack resistant.*

Proof Our scheme can protect the user's location privacy by the interference of other neighbors when the real user is located in densely populated regions. In sparsely populated regions, our scheme also can protect the user's location privacy by generating dummy locations. Now, we prove our algorithm is colluding-attack resistant.

1) In densely populated regions, when an attacker colludes with user u_A, the gained information may include the history query probability, the current query, and the historical query.

On the one hand, $2k$ locations are selected to randomly form m user groups which have the largest anonymous entropy. On the other hand, the distance to the real users and the request content of those is taken into account. Therefore, other users in the same group can not know the real user's location and only know their user group that possesses the largest anonymous entropy, which signifies that the user can only randomly deduce the location of the requesting user, and the probability of successful deducing is $1/k$.

Afterwards, the attacker also colludes with the user u_B. Because there is no connection between u_A and u_B, the probability that the attacker deduces the real user's position is still $1/k$. Therefore, collusion attacks can be resisted in densely populated regions.

2) In sparsely populated regions, the attacker intercepts user u_A to obtain the related information such as the historical search probability. Firstly, the cells which have the similar query probability with the real user are selected. Then, the cells that evenly distributed around the real user are chosen. Finally, it selects the user group with the highest entropy according to the anonymous entropy method. Since the information obtained by the attacker cannot help speculate the actual location of the real user, the successful deducing possibility is $1/k$.

Afterwards, the attacker also colludes with user u_B. As there is no connection between u_A and u_B, the probability for the adversary inferring the real user's location is still $1/k$, which means that the probability for obtaining the real user privacy cannot be increased with the size of the colluding group; it can resist collusion attack in sparsely populated regions.

In extreme circumstance, the LBS server may be act as an adversary. At the moment, the LBS service provider owns both the history queries and current queries which include the user's identifier, the mix of real and dummy locations, and the request content.

In the K-DDCA algorithm, $2k$ neighbor query users are first selected, and then the anonymous entropy method is used to select $k-1$ query users to construct an anonymous region based on the requested content and distance. The LBS service provider cannot speculate the real location of the query user based on the information already available.

In the K-SDCA algorithm, $3k$ locations with the similar query probability to the real location are first selected, and then the user group is stochastically constructed according to the distance relationship among $3k$ locations. On this basis, the anonymous entropy method is utilized to select the anonymous region to ensure the uncertainty. Therefore, even if the LBS service provider has possessed global information, it cannot infer the real location of the user. □

6 Results and discussion

In this section, we evaluate the performance of our scheme via extensive experiments. We give the detailed experimental results and discussion.

6.1 Experimental settings

The experimental environment is 64 bit Windows 7 system with Intel (R) Core i7-8700k CPU @ 3.70 GHz and the RAM of 32G. The programming language is Python on PyCharm.

We use OPNET [28, 29] to generate the simulation data which can better describe human behavior patterns, construct complex network topologies, and simulate the process of sending/receiving of message.

Assume that we take a 4 km × 4 km region as our simulation region which is divided into 40 × 40 cells where 20 points of interests (POIs) are randomly and uniformly generated. In the densely populated region, the K-DDCA algorithm is compared with the quad-tree algorithm [30] and the Casper algorithm [31]. However, in the sparsely populated region, the quad-tree algorithm and the Casper

algorithm cannot complete the construction of the anonymous region. We will compare the K-SDCA algorithm with the following four algorithms, that is the random dummy selection algorithm, DLS algorithm, enhanced-DLS algorithm [33], and MOS algorithm [34].

6.2 Experimental results

6.2.1 *Comparison of anonymous processes between the Casper algorithm and our algorithm*

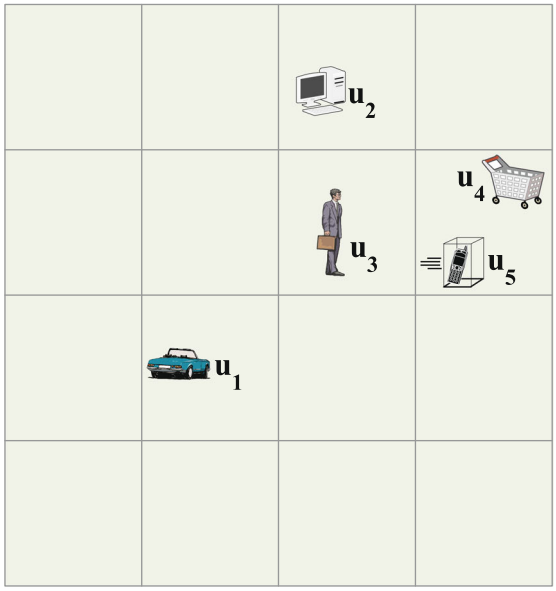
In Fig. 6, we simulate the anonymization process of the Casper algorithm and K-DDCA algorithm. Figure 6a shows the Casper anonymous process, and Fig. 6b indicates the K-DDCA algorithm anonymous process. The shaded area represents the anonymous region. Assume that there is a LBS request *req* and $k = 2$.

Figure 6a shows that u_1 sends a LBS request. Since the adjacent quadrant does not meet the requirement of k, it expands to the parent quadrant until it meets the requirement of k. In order to satisfy k, it should extend to the entire region. If an attacker finds that users u_2 to u_5 are in the anonymous region of u_1 and their respective anonymous regions do not contain u_1, it is possible to speculate that the request is issued by u_1 when an attacker finds that the anonymous region contains the entire space, which is a risk of privacy leaks. Fortunately, there is no danger of privacy disclosure in Fig. 6b which is an anonymous region formed by our algorithm.

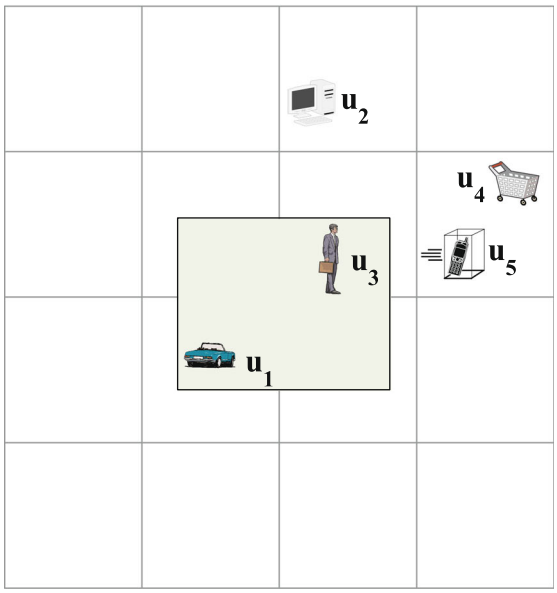
6.2.2 *Relationship between the area of the anonymous regions and k value*

Figure 7 indicates the evaluation results in details where we can clearly see that the area of the anonymous region formed by the quad-tree algorithm, Casper algorithm, and K-DDCA algorithm gradually increases with the raise of the k value. Assume that the number of people in the region is 1000 here. With the increase of k, the area of anonymous regions becomes increasingly large. However, since the quad-tree algorithm and Casper algorithm use the quad-tree model for storage, they do not take full account of the location relationships of the neighboring users. In particular, when it extends to a high level, each expansion leads to a larger area increase, generating redundant space. Therefore, the size of the anonymous region is too large which means that the anonymous users are denser accordingly, so that the real user's location could be easily identified by the attacker.

It can be seen from the figure that the area of anonymous regions of K-DDCA becomes stable to 2 km^2 when K=16. The reason is that the K-DDCA algorithm takes full account of the location relationship between neighboring users and then selects evenly distributed users among neighbors. Therefore, it can guarantee that the area of the anonymous region formed will be moderate, and the area of the anonymous region can be set according to the



(a) Request scenario of user u_1 utilizing Casper algorithm



(b) Request scenario of user u_1 employing K-DDCA algorithm

Fig. 6 Comparison of anonymous processes between Casper algorithm and K-DDCA algorithm. It depicts the anonymization process of the Casper algorithm and the K-DDCA algorithm. Subgraph **a** shows the Casper anonymous process, and subgraph **b** indicates the K-DDCA algorithm anonymous process. The shaded area represents the anonymous region. Assume that there is a LBS request *req* and $k = 2$

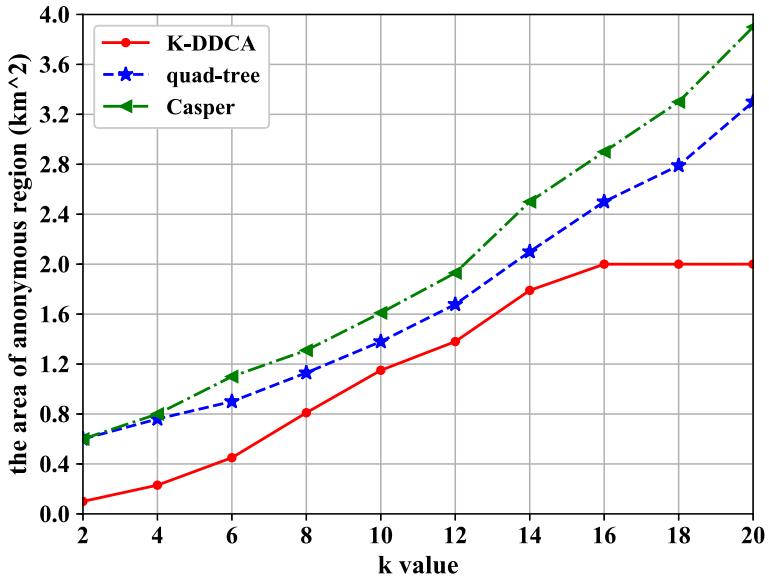


Fig. 7 Relationship between the area of anonymous region and k value. It shows that as the k value increases, the area of the anonymous region constructed by the K-DDCA algorithm is appropriate compared to the other two algorithms

user's needs. If the area does not meet the requirements of the user's k value, then K-SDCA is used to generate the dummy location, and the quad-tree algorithm and Casper algorithm can only achieve the user's requirements by expanding the area of the anonymous region.

6.2.3 *Relationship between the number of users and the area of anonymous region*

Figure 8 shows the relationship between the area of the anonymous region and the number of users under the condition of $k = 10$, $k = 15$ from the Casper algorithm and K-DDCA algorithm. It can be seen from the curves that the area of the anonymous region gradually decreases with the increase of user number. After the user density reaches 900, it tends to be stable.

According to the K-DDCA algorithm, the user can determine the area of the anonymous region (set to 4 km^2). When the number of users is less than k at the beginning, the K-SDCA algorithm generates dummy locations and construct an anonymous region. Therefore,

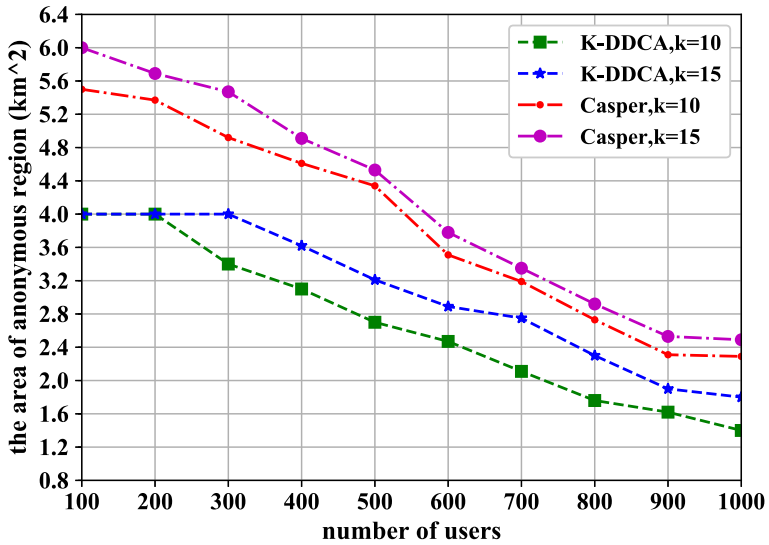


Fig. 8 Relationship between the number of users and the area of the anonymous region. It shows the relationship between the area of the anonymous region and the number of users under the condition of $k = 10$, $k = 15$ from the Casper algorithm and K-DDCA algorithm

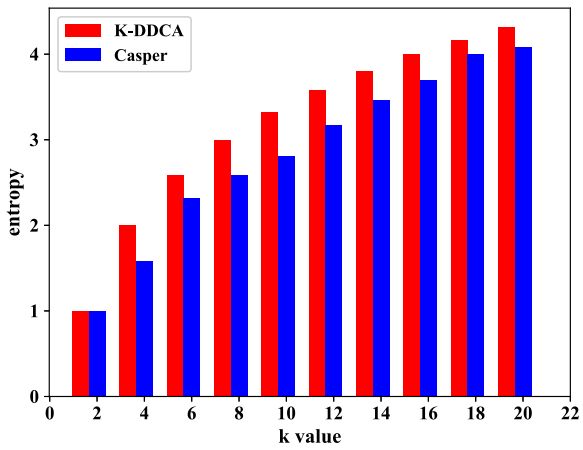


Fig. 9 Comparison of anonymity between the K-DDCA algorithm and the Casper algorithm under different *k* values. It indicates the privacy level in terms of entropy of different schemes, and the anonymity of the K-DDCA algorithm and Casper algorithm gradually increases with the raise of the *k* value

when the number of users is large enough, the area of generating anonymous region tends to be stable. However, in order to achieve the *k* value specified by the user, the Casper algorithm will always enlarge the area of the anonymous region. If the number of users in the entire area is less than *k*, the Casper algorithm will not complete the construction of the anonymous region.

6.2.4 Relationship between the privacy level and k value in densely populated regions

Figure 9 indicates the privacy level in terms of *entropy* of different schemes; we can see that the anonymity of the K-DDCA algorithm and Casper algorithm gradually increases with the raise of the *k* value. Obviously, the performance of the K-DDCA algorithm is better than that of the Casper algorithm. The anonymous region formed by the Casper model contains a large number of redundant regions which mean that there is no user there as shown in Fig. 6a. An attacker can exclude a number of users based on the background information, so the anonymity of the Casper algorithm cannot reach $1/k$.

6.2.5 Relationship between privacy level and k value among various algorithms in sparsely populated regions

In Fig. 10, we can see that the degree of anonymity gradually increases with the raise of *k*. Compared to other four algorithms, the dummy locations generated by K-SDCA has a better effect. In this figure, the random scheme performs worse than other schemes, since it just generates dummy locations randomly and without considering background information. Compared with K-SDCA, enhanced-DLS, and MOS algorithms, DLS algorithm does not work well, since it only considers the condition that the query probability of dummy locations is similar to that of the real user and does not consider the distance between users and the difference in request content.

On the basis of similar query probability, the enhanced-DLS and MOS algorithms generated dummy locations which are as far as possible from real users. The MOS algorithm is better than the enhanced-DLS algorithm, but it does not consider the diversity of the request content, so the anonymity is relatively good. The dummy locations chosen by the K-SDCA algorithm satisfy three constraint conditions: (1) the query probability is similar to that of

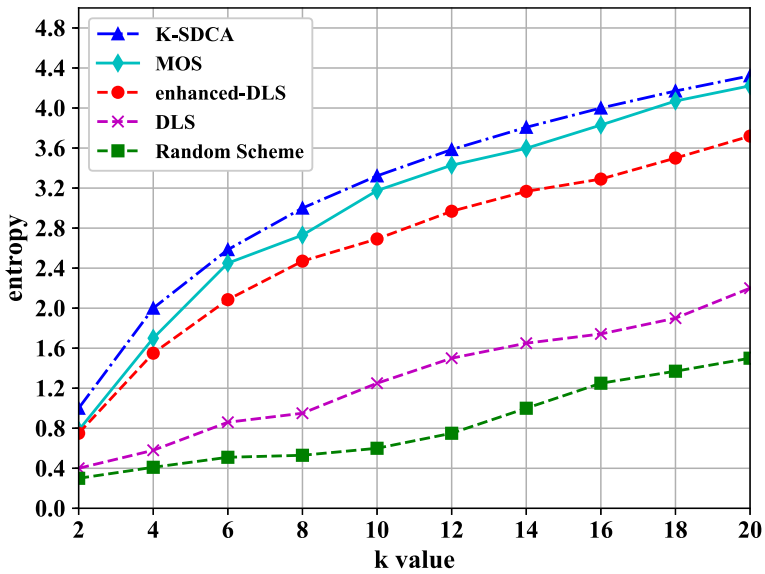


Fig. 10 Comparison of anonymity among different algorithms under different *k* values. It shows that the degree of anonymity gradually increases with the raise of *k*. Compared to other four algorithms, the dummy locations generated by K-SDCA has a better effect

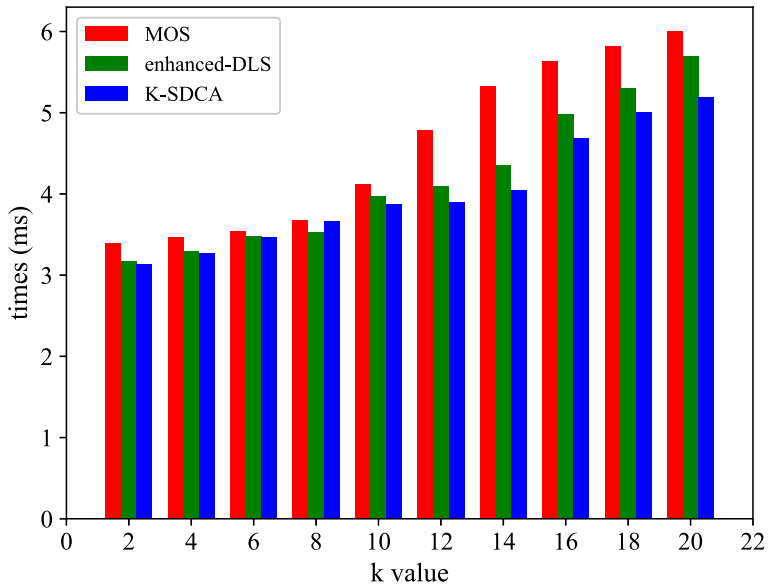


Fig. 11 Relationship between anonymous time and *k* value among different algorithms in sparsely populated regions. It indicates that the anonymous time of enhanced-DLS, K-SDCA, and MOS algorithms gradually increases with the raise of *k*. Under the conditions of the same *k* value, the K-SDCA algorithm is more efficient than the enhanced-DLS and MOS algorithms

the real user, (2) the geographical distribution is relatively uniform, and (3) the content of the request is different. Afterward, the K-SDCA algorithm selects the optimal user group utilizing anonymous entropy method, which increases the uncertainty.

6.2.6 Relationship between anonymous time and k value in sparsely populated regions

In Fig. 11, we can see that the anonymous time of enhanced-DLS, K-SDCA, and MOS algorithms gradually increases with the raise of *k*. Under the conditions of the same *k* value, the K-SDCA algorithm is more efficient than the enhanced-DLS and MOS algorithms. The reason is that the MOS algorithm needs to compute overlapping areas when forming anonymous regions and then expand the total anonymous area as much as possible.

The similarity between the enhanced-DLS algorithm and the K-SDCA algorithm is to select one dummy user per round. However, the difference between them is that it is necessary for the enhanced-DLS algorithm to calculate the distance between the candidate users and the selected users and then compare the product of the distances in each round. Fortunately, the K-SDCA algorithm does not need to calculate the product of the distances. It only needs to calculate the distance from the candidate set to the straight line in even rounds by formula (2). Therefore, the K-SDCA algorithm is more efficient than the enhanced-DLS algorithm.

7 Conclusions

This paper takes kd-tree as the storage structure and puts forward an anonymous entropy-based location privacy protection scheme in MSN. Our scheme consists of two algorithms K-DDCA in a densely populated region and K-SDCA in a sparsely populated region to tackle the problem of location privacy leakage. Specifically, an anonymous entropy method is proposed which takes into account the distance between users in the user group and the distribution of the request content jointly. According to the anonymous entropy method, we select the user group with the largest anonymous entropy to guarantee that the user group has the greatest uncertainty. In addition, our scheme effectively reduces the time complexity and provides users with high-quality services.

In the future, we will research on the location privacy protecting scheme in the sparsely populated region where there is a lack of historical records. Since LBS service providers have many limitations for the data usage, we will research how to evaluate these demand and challenge. Furthermore, we plan to cooperate with some LBS service providers to further validate the effectiveness of our scheme.

Abbreviations

α_{ni}: The weight of the user u_i on distance; β_n: The weight of the user group on request content; d_{ul}: The distance between user u_l and line; $H(R)$: The anonymity of a user group R; *LBSQ*: The LBS request; *MinU*: The minimum threshold for the number of user; *MinC*: The minimum threshold for the kind of request contents; $N_r\ (u_i)$: The r-neighborhood of u_i; $N_\theta\ (u_i)$: The θ-neighborhood of u_i; phq_i: The historical query probability of cell i; $dist\ (u_i, u_j)$:

The distance between user u_i and user u_j; U_{max}: The user group with the largest anonymous entropy; w_{ei}: The weight of candidate u_i in the even round; w_{oi}: The weight of candidate u_i in the odd round; (x_i, y_i): The location of user u_i

Acknowledgements
Not applicable.

Funding
This work is supported by National Key R & D Programs Project of China under Grant 2017YFC0804406, NSF of China under Grant 61672321, 61771289, 61832012 and 61373027, Training Program of the Major Research Plan of NSF of China under Grant 91746104, Project of Shandong Province Higher Educational Science and Technology Program under Grant J15LN19, Open Project of Tongji University Embedded System and Service Computing of Ministry of Education of China under Grant ESSCKF 2015-02.

Availability of data and materials
The datasets used in this paper is generated by OPNET generator, and the website is https://www.opnet.com/.

Authors' contributions
The main idea of this paper was proposed by LNN. The system model and algorithm were designed by JQZ and FLT. The architecture was given by JQZ. The simulations were conducted by FLT, YY, and LNN. The writing of the paper was completed by FLT, QHN, YY, and JQZ. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Author details
[1]College of Computer Science and Engineering and Shandong Province Key Laboratory of Wisdom Mine Information Technology, Shandong University of Science and Technology, Qingdao, Shandong, China. [2]College of Foreign Languages, Shandong Agricultural University, Taian, Shandong, China.

Received: 13 November 2018 Accepted: 19 March 2019
Published online: 08 April 2019

References

1. D. E. Kouicem, A. Bouabdallah, H. Lakhlef, Internet of things security: a top-down survey. Comput. Netw. **141**(4), 199–221 (2018)
2. Y. Huo, C. Hu, X. Qi, T. Jing, Lodpd: A location difference-based proximity detection protocol for fog computing. IEEE Internet Things J. **4**(5), 1117–1124 (2017)
3. J. Mao, Y. Chen, F. Shi, Y. Jia, Z. Liang, Toward exposing timing-based probing attacks in web applications. Sensors. **17**(3), 464 (2017)
4. Y. Liang, Z. Cai, J. Yu, Q. Han, Y. Li, Deep learning based inference of private information using embedded sensors in smart devices. IEEE Netw. Mag. **32**(4), 8–14 (2018)
5. Z. Cai, Z. He, X. Guan, Y. Li, Collective data-sanitization for preventing sensitive information inference attacks in social networks. IEEE Trans. Dependable Secure Comput. **15**(4), 577–590 (2017)
6. Y. Liang, Q. Han, Y. Li, et al., Location privacy leakage through sensory data. Secur. Commun. Netw. **2017**(11), 1–12 (2017)
7. L. Ni, J. Zhang, C. Jiang, et al., Resource allocation strategy in fog computing based on priced timed petri nets. IEEE Internet Things J. **4**(5), 1216–1228 (2017)
8. y. Li, M. L. Yiu, Route-saver: leveraging route apis for accurate and efficient query processing at location-based services. IEEE Trans. Knowl. Data Eng. **27**(1), 235–249 (2015)
9. M. Xin, M. Lu, W. Li, An adaptive collaboration evaluation model and its algorithm oriented to multi-domain location-based services. Expert Syst. Appl. **42**(5), 2798–2807 (2015)
10. R. Al-Dhubhani, J. M. Cazalas, An adaptive geo-indistinguishability mechanism for continuous LBS queries. Wirel. Netw. **24**(8), 3221–3239 (2018)
11. Y. Jing, L. Hu, W. S. Ku, C. Shahabi, Authentication of k nearest neighbor query on road networks. IEEE Trans. Knowl. Data Eng. **26**(6), 1494–1506 (2014)
12. V. Bindschaedler, R. Shokri, in *2016 IEEE Symposium on Security and Privacy (SP)*. Synthesizing plausible privacy-preserving location traces (IEEE, 2016), pp. 546–563
13. Y. Sun, M. Chen, L. Hu, Y. Qian, Hassan, ASA: Against statistical attacks for privacy-aware users in location based service. Futur. Gener. Comput. Syst. **70**, 48–58 (2017)
14. Z. Cai, X. Zheng, A private and efficient mechanism for data uploading in smart cyber-physical systems. IEEE Trans. Netw. Sci. Eng. (2018). https://doi.org/10.1109/TNSE.2018.2830307
15. X. Zheng, G. Luo, L. Tian, et al., Privacy-preserved community discovery in online social networks. Futur. Gener. Comput. Syst. (2018). https://doi.org/10.1016/j.future.2018.04.020
16. L. Ni, C. Li, X. Wang, H. Jiang, Yu J., DP-MCDBSCAN: Differential privacy preserving multi-core DBSCAN clustering for network user data. IEEE Access. **6**, 21053–21063 (2018)
17. T. Xu, Y. Cai, in *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems. Vol. 39*. Location anonymity in continuous location-based services (ACM, 2007), pp. 300–307
18. C. Li, B. Palanisamy, in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. Reversecloak: protecting multi-level location privacy over road networks (ACM, 2015), pp. 673–682
19. J. Shao, R. Lu, X. Lin, in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. FINE: a fine-grained privacy-preserving location-based service framework for mobile devices (IEEE, 2014), pp. 244–252
20. X. Li, M. Miao, H. Liu, J. Ma, K.-C. Li, An incentive mechanism for k-anonymity in LBA privacy protection based on credit mechanism. Soft Comput. **21**(14), 3907–3917 (2017)
21. B. Ying, D. Makrakis, in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. Protecting location privacy with clustering anonymization in vehicular networks (IEEE, 2014), pp. 305–310
22. J. Zhang, Y. Yuan, X. Wang, et al., RPAR: location privacy preserving via repartitioning anonymous region in mobile social network. Secur. Commun. Netw. **2018**, 1–10 (2018)
23. T. Hara, A. Suzuki, M. Iwata, et al., Dummy-based user location anonymization under real-world constraints. IEEE Access. **4**, 673–687 (2016)
24. H. Samet, The design and analysis of spatial data structures. Off. Sci. Tech. Inf. Tech. Rep. **50255**(4), 1211 (1990)
25. X. Wang, Y. Luo, Y. Jiang, W. Wu, Q. Yu, Probabilistic optimal projection partition kd-tree k-anonymity for data publishing privacy protection. Intell. Data Anal. **22**(6), 1415–1437 (2018)
26. S. Hayashida, D. Amagata, T. Hara, X. Xie, Dummy generation based on user-movement estimation for location privacy protection. IEEE Access. **6**, 22958–22969 (2018)
27. B. Niu, Q. Li, X. Zhu, et al., in *2015 IEEE conference on computer communications (INFOCOM)*. Enhancing privacy through caching in location-based services (IEEE, 2015), pp. 1017–1025
28. Opnet. https://www.opnet.com/. Accessed 25 Jul 2018
29. G. Sun, D. Liao, H. Li, H. Yu, V. I. Chang, L2P2: A location-label based approach for privacy preserving in LBS. Future Gener. Comput. Syst. **74**, 375–384 (2017)
30. M. Gruteser, D. Grunwald, in *Proceedings of the 1st international conference on Mobile systems, applications and services*. Anonymous usage of location-based services through spatial and temporal cloaking (ACM, 2003), pp. 31–42
31. O. Abul, F. Bonchi, M. Nanni, in *2008 IEEE 24th International Conference on Data Engineering (ICDE '08), IEEE Computer Society,*. Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases, (2008), pp. 376–385
32. M. F. Mokbel, C.-Y. Chow, W. G. Aref, in *Proceedings of the 32nd International Conference on Very Large Data Bases*. The new casper: Query processing for location services without compromising privacy (VLDB Endowment, 2006), pp. 763–774
33. B. Niu, Q. Li, X. Zhu, G. Cao, H. Li, in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. Achieving k-anonymity in privacy-aware location-based services (IEEE, 2014), pp. 754–762

34. D. Wu, Y. Zhang, Y. Liu, in *2017 IEEE Trustcom/BigDataSE/ICESS. IEEE*. Dummy Location Selection Scheme for K-Anonymity in Location Based Services (IEEE, 2017), pp. 441–448
35. D. Liao, X. Huang, V. Anand, et al., in *2016 IEEE International Conference on Communications (ICC)*. k-DLCA: an efficient approach for location privacy preservation in location-based services (IEEE, 2016), pp. 1–6
36. R. JiangM, R. Lu, K.-K. R. Choo, Achieving high performance and privacy-preserving query over encrypted multidimensional big metering data. Future Gener. Comput. Syst. **78**, 392–401 (2018)
37. Z. Xu, Z. Cai, J. Li, G. Hong, in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. Location-privacy-aware review publication mechanism for local business service systems (IEEE, 2017), pp. 1–9
38. Z. Xu, Z. Cai, J. Li, H. Gao, Data linkage in smart internet of things systems: a consideration from a privacy perspective. IEEE Commun. Mag. **56**(9), 55–61 (2018). https://doi.org/10.1109/MCOM.2018.1701245
39. L. Sweeney, k-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **10**(05), 557–570 (2002)