





Article

A Trajectory Privacy Preserving Scheme in the CANNQ Service for IoT

Lin Zhang ^{1,2,3,*} , Chao Jin ¹ , Hai-ping Huang ^{1,2,3} , Xiong Fu ^{1,2,3}  and Ru-chuan Wang ^{1,2,3}

¹ School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; 15195886318@163.com (C.J.); hhp@njupt.edu.cn (H.-p.H.); fux@njupt.edu.cn (X.F.); wangrc@njupt.edu.cn (R.-c.W.)

² Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China

³ Institute of Computer Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

* Correspondence: zhangl@njupt.edu.cn; Tel.: +86-25-8586-6354; Fax.: +86-25-8349-2152

Received: 28 February 2019; Accepted: 9 May 2019; Published: 12 May 2019



Abstract: Nowadays, anyone carrying a mobile device can enjoy the various location-based services provided by the Internet of Things (IoT). ‘Aggregate nearest neighbor query’ is a new type of location-based query which asks the question, ‘what is the best location for a given group of people to gather?’ There are numerous, promising applications for this type of query, but it needs to be done in a secure and private way. Therefore, a trajectory privacy-preserving scheme, based on a trusted anonymous server (TAS) is proposed. Specifically, in the snapshot queries, the TAS generates a group request that satisfies the spatial K-anonymity for the group of users—to prevent the location-based service provider (LSP) from an inference attack—and in continuous queries, the TAS determines whether the group request needs to be resent by detecting whether the users will leave their secure areas, so as to reduce the probability that the LSP reconstructs the users’ real trajectories. Furthermore, an aggregate nearest neighbor query algorithm based on strategy optimization, is adopted, to minimize the overhead of the LSP. The response speed of the results is improved by narrowing the search scope of the points of interest (POIs) and speeding up the prune of the non-nearest neighbors. The security analysis and simulation results demonstrated that our proposed scheme could protect the users’ location and trajectory privacy, and the response speed and communication overhead of the service, were superior to other peer algorithms, both in the snapshot and continuous queries.

Keywords: aggregate nearest neighbor query; trajectory privacy; spatial K-anonymity; secure areas; strategy optimization

1. Introduction

In recent years, with the popularization of mobile portable devices, advancement in spatial positioning technology and development of Internet of Things (IoT), location-based services (LBS) have successfully appeared in public view. Various personalized services, combined with location elements, such as location sharing [1,2], nearest neighbor query [3,4], friend discovery [5,6], etc., are popular among users. Among them, the k nearest neighbor (kNN) queries can find the k points of interest (POIs) nearest to the users, e.g., find three gas stations nearest to me. However, with the emergence of new scenarios, the kNN queries have been unable to meet more complex needs of users. For example, if several friends in different places are going to have dinner after work, how do you find a western restaurant nearest to them? If several users in different locations plan to carpool, how do you help the driver to plan the best route to pick up all passengers? Tan [7] defines these scenarios as ‘multiple object

convergence’, and the corresponding query mode is called ‘aggregate nearest neighbor (ANN) query’. In the ANN queries, multiple users (at least two) need to send query requests to the location-based service provider (LSP) by building query groups. The list of nearest neighbors returned by the LSP, depends on the specified aggregate functions, which includes ‘sum’, ‘max’, and ‘min’. Different aggregate functions cause differences in query results, where the ‘sum’ function makes sure that the total distance traveled by all persons is minimum, while the ‘max/min’ function aims to minimize the maximum/minimum traveling distance requested by any person. This paper focuses on the ‘sum’ function because it properly takes into account the distance traveled by everyone in the group.

However, a necessary condition for the widespread use of IoT is to protect the privacy of users, especially location privacy. As we all know, the LSP requires the inquirers to provide their location information to obtain the service. The more accurate the location that the user sends, the higher quality of the service the user receives, but the more privacy the user leaks [8]. Therefore, prevention of the LSP from obtaining the user’s location information during the process of obtaining the service is an urgent problem to be solved. For example, in academia, most [9–12] adopt an architecture, based on a trusted anonymous server (TAS). As shown in Figure 1, the TAS can help users to send query requests, which avoids direct communication between users and the LSP. In addition, some scholars have questioned that the TAS will become a performance bottleneck because all query requests need to pass through it, thus, increasing the response delay of the overall service. Therefore, balancing privacy protection and service response delay is another urgent problem to be solved.

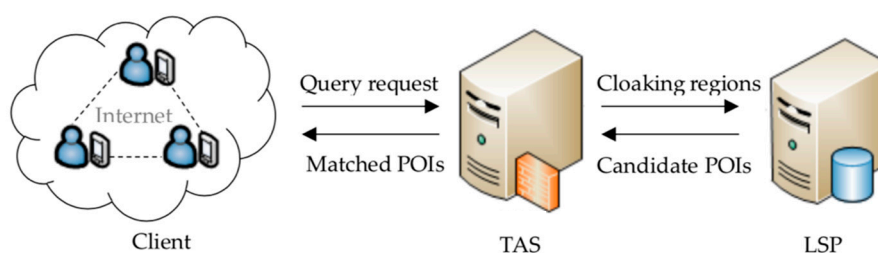


Figure 1. The architecture based on a trusted anonymous server.

For current surveys on location privacy protection in the IoT, there is not yet much research available on ‘the continuous aggregate nearest neighbor query service that protects the location and trajectory privacy of the group of users’ (PCANNQ). The main problems that exist are listed below.

First, in the snapshot queries, it is feasible to introduce the TAS into PCANNQ, to protect the user’s location privacy, but there is a little adjustment for protecting the location privacy of the group of users.

Second, in continuous queries, it is feasible to protect the user’s trajectory privacy by reducing the communication frequency between the user and the LSP, but there is a lack of a query scheme that protects the trajectory privacy of the group of users, while ensuring accurate query results.

Third, it is feasible to improve the response speed of the queries by improving the ANN query algorithm, but there is a lack of an efficient method for computing the aggregate nearest neighbors.

The rest of this paper is organized as follows. Section 2 presents the related work; Section 3 provides a general overview of our system model; Section 4 elaborates details of our proposed scheme and analyzes the security; Section 5 discusses the construction of circular secure areas; and Section 6 discusses the ANN query algorithm based on strategy optimization. Extensive simulation experiments were carried out and our findings have been reported in Section 7. Finally, Section 8 concludes the paper with some directions for future work.

2. Related Work

In this section, we reviewed certain available research on the main techniques in the IoT, including private nearest neighbor query schemes, and a new type of query—aggregate nearest neighbor query.

2.1. Private Nearest Neighbor Query Scheme

Private nearest neighbor queries include snapshot queries that protect location privacy [13–16] and continuous queries that protect trajectory privacy [17–20].

In snapshot queries, location privacy protection techniques mainly included location perturbation [13], spatial cloaking [14,15], and spatial transformation [16]. In Reference [13], the Space Twist technique sent a false location information to the LSP, to query the nearest neighbors about the fake location. The method was simple and did not require the intervention of any third party, but the communication efficiency was very low, because the number of single-inquiry communication was indefinite, which made it difficult to respond to continuous queries in a timely and stable manner. The Casper technique proposed in Reference [14] involved the location of an anonymous server and a private query processor. The method showed a high efficiency for the nearest neighbor queries on the cloaking regions. The basic idea of the spatial cloaking technique was to blur a user's exact location into a cloaked area that satisfied the user specified privacy requirements. Later, Chow [15] proposed a spatial cloaking algorithm for mobile P2P environments. In Reference [16], the Hilbert curve encryption technology was adopted to achieve privacy protection, but the query results were not completely accurate.

In continuous queries, trajectory privacy protection techniques mainly included reducing the number of data submissions [17,18] and submitting redundant queries [19,20]. Zhang [17] proposed a caching-based scheme to enhance user privacy which employed a two-level caching to cache result data. However, this scheme did not consider the user's neighbor cache and the anonymity, based on user mobility. To further enhance user privacy and increase the cache hit rate, Zhang [18] adopted a multi-level caching to cache users' results. In fact, this paper focused on using caching technology to cache the query results for users' future queries. When other users issued similar queries in the future, the results could be retrieved from the cache of the TAS without having to submit a request to the LSP, thereby, protecting the privacy of the users. Hwang [19] required the TAS issued redundant queries with K-anonymizing spatial regions (K-ASRs) and randomized the sequence of the query time, in order to protect the trajectory privacy. Inspired by Reference [19], Peng [20] proposed a collaborative trajectory privacy preserving (CTPP) scheme, which constructed the K-ASRs based on the gathered information obtained from multi-hop peers and the trajectory privacy was guaranteed by user collaboration. This method broke the correlations of the continuous LBS queries to obfuscate the user's actual trajectory.

2.2. Aggregate Nearest Neighbor Query

Aggregate nearest neighbor (ANN) query [21–27] is a variant of the kNN query used to help multiple users to find the optimal meeting place. Papadias [21] first proposed the concept of ANN query, and three proposed algorithms, including multiple query method (MQM), single point method (SPM), and minimum bounding method (MBM) suitable for Euclidean space, but the query efficiency was low due to the large search scope. At the same time, Yiu [22] applied the ANN query to the road networks. In order to optimize queries and improve query efficiency, Luo [23] proposed a Vp-ANN query algorithm, based on a 'two-points projection'. The algorithm, first, projects the query points to specific rows, analyzes their distribution, and then trims the search scope. The algorithm reduces the search space to some extent. The voronoi-based ANN query algorithm proposed by Sun [26] divides the query algorithm into the search phase and the prune phase. The search phase computes the nearest neighbors of each query point in some order, to obtain the candidate POIs, until all the query points find a common POI. The prune phase filters out the unqualified POIs, according to the prune strategy of a given aggregate function, until only one candidate POI remains as the final result. This algorithm is currently the most reasonable, but the search strategy and prune strategy are not optimal, and there is still room for improvement.

Additionally, the index structure of POIs is also an important factor that affects the query efficiency. Sun [23]'s use of R tree [28] to organize POIs has actually proved to be inefficient. Although there

have been more efficient indexing techniques to help queries, for example, the VOR tree [29], the AVR tree [30], etc., which satisfies a faster search speed, but the purpose of this paper is to improve the search and prune strategy, and highlight the superiority of the proposed strategy. Therefore, the R tree spatial index is still used to achieve comparability between the query algorithms.

2.3. Our Contributions

Location-based query services are the basic functions of the IOT, and the leakage of location and trajectory privacy is a widespread problem. This paper aims to design a secure and efficient PCANNQ scheme, to protect the location and trajectory privacy of multiple users, while ensuring the overall performance of the service. In other words, our scheme emphasizes the continuous and rapid response of this type of queries, while protecting the location and trajectory privacy of the group of users. Therefore, there is a fundamental difference from the above-related researches. Our contributions in this paper can be summarized as follows.

- (1) We propose a new PCANNQ scheme based on the TAS. The TAS helps the group of users to send group requests that satisfy spatial K-anonymity to the LSP. It prevents the LSP from directly obtaining the real location of the group of users, thus, protecting the location privacy of the group of users in snapshot queries.
- (2) We propose a circular secure areas construction method suitable for continuous queries. On one hand, the method ensures that the optimal aggregate nearest neighbor remains unchanged when the users move within the areas, on the other hand, the secure areas can conceal the real locations of the users, thus, reducing the probability of the LSP reconstructing the users' trajectories.
- (3) We propose an efficient aggregate nearest neighbor query algorithm. The algorithm includes a search phase and a prune phase. The search phase is designed to limit the minimum search scope of POIs, based on the aggregate subgroup, and the prune phase is designed to filter the non-nearest neighbors quickly, thus, improving the response speed of the query algorithm.
- (4) We experimentally compared the performance of our scheme with peer algorithms, in terms of security, average processing overhead, query efficiency, etc. The experimental results showed that our scheme could protect the location and trajectory privacy of the group of users, while ensuring the overall performance of the service, which achieved the expectation of the paper.

3. System Model and Definition

This section first analyzes the privacy leaks in the location-based query services. Then, the main entities involved in the system are introduced. Finally, we defined continuous aggregate nearest neighbor queries.

3.1. Problem Statement

We consider the following scenario. Alice, Bob, and Carl plan to have dinner after work and query some western restaurants nearest to them. Due to the rush hour, the road conditions of each user are unknown, and traffic jams might occur. Therefore, users need the LSP to update more appropriate places in real time, during their movement, so that they can meet more quickly.

There are two concerns—first, in snapshot LBS queries, users need to send their current locations to the untrusted LSP, to obtain the service; second, in the continuous LBS queries, users need to send their current locations to the untrusted LSP, within a fixed time interval. This increases the risk of exposure of users' location and trajectory privacy. Figure 2a shows the service usage of a user in the LBS queries. The LSP can construct the user's mobile trajectory, by collecting the user's location sequence, and then infer the user's private information, such as interests, habits, and religions.

For the first issue, the spatial K-anonymity method can be used, which generates a cloaking-region-satisfied K-anonymity, every time the user issues a query. The region contains k-1 dummy locations and a real location, which are indistinguishable. However, for the second issue,

there is a risk of failure. Take Figure 2b as an example, the method generates cloaking-regions-satisfied 5-anonymity for a user, at each timestamp, in continuous queries. On the one hand, there is a spatial correlation between locations. By connecting the cloaking regions, the spatial characteristics of the original trajectory, such as the moving path and direction, can still be inferred. On the other hand, there is a temporal correlation between the locations. The LSP can reconstruct the trajectory of the user with great probability, according to the sequence of the query issuing time. We can see that in this scenario, although each snapshot query is protected, based on the K-anonymity principle, the trajectory of the user can still be easily disclosed.

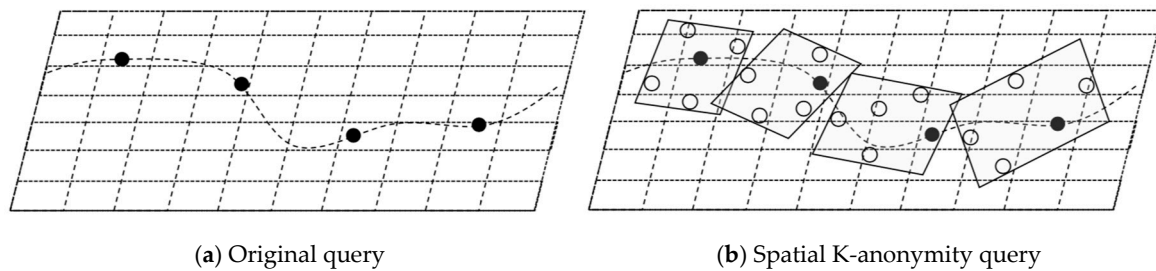


Figure 2. Original query and K-anonymity query of single user.

3.2. System Model

To address the issues mentioned above, we propose a PCANNQ scheme for the IOT. The architecture is shown in Figure 1 and details will be explained in Section 4. It consists of three main entities: Clients, TAS, and LSP. Their main functions can be described as follows:

- (1) **Client.** Each mobile client carries a mobile device with functions, such as information processing, data storage, and global positioning (e.g., GPS) to quickly connect to the IOT. The users participating in the LBS queries trust each other, and there is no collusion attack between any users and the LSP.
- (2) **TAS.** The TAS is trusted. Its main functions include concealing users' identities, generating cloaking regions satisfying the spatial K-anonymity for users, and filtering out the query results belonging to the real user, in the query results. In the continuous queries, TAS can also generate secure areas, according to whether the query result changes, which dynamically conceal the users' moving paths.
- (3) **LSP.** The LSP is semi-trusted. On one hand, the LSP stores map resources and location-related POIs, such as restaurants, hospitals, and tourist attractions, to provide users with timely location-based query services. On the other hand, users do not know whether the LSP-stored trajectory data generated in the query process, and whether the private information contained in the trajectories is abused or not, is also unknown.

Therefore, the LSP is considered as an adversary in this paper. For the location information collected in the snapshot queries, the LSP attempts to determine the location related to the user's true identity, and then infers other private information, based on the location. For example, Alice often queries nearby drivers at around 17:00. It could be inferred that the location might be her workplace.

For the trajectory data collected in continuous queries, the LSP could obtain more information about the user by analyzing the trajectory data. For example, Alice goes to work at 08:00, gets off work at 17:00, and often goes to Starbucks at noon. Therefore, the goal of the LSP is to capture as many of the users' continuous locations as possible, and then reconstruct the users' complete trajectories.

3.3. Continuous Aggregate Nearest Neighbor Query

Continuous aggregate nearest neighbor queries (CANNQ) refer to a group of users querying the latest aggregate nearest neighbors, during their movement, which is usually represented by a quadruple called CANNQ = $\langle Q, P, T, NN \rangle$.

$Q = \{q_1, q_2, \dots, q_n\}$ represents the group of users that issued the queries where $q_i = \langle Id_i, x_i, y_i, t_i \rangle$. Id_i , x_i and y_i represent q_i 's identity, latitude, and longitude, respectively.

$P = \{p_1, p_2, \dots, p_m\}$ represents the collection of POIs where $p_i = \langle x_i, y_i, type_i \rangle$. x_i , y_i and $type_i$ represent p_i 's latitude, longitude, and type, respectively.

$T = \{t_1, t_2, \dots, t_o\}$ represents the time sequence of the group of users from issuing the query to ending the query where t_1 and t_o represent start and end times, respectively.

$NN_{t \in T} = \{p \mid \text{dist}_{agg}(p_i, Q) \leq \text{dist}_{agg}(p_j, Q), i, j \in [1, n], i \neq j\}$ represents a list of the nearest neighbors of the group of users at time t , where the POIs are sorted in ascending order, by $\text{dist}_{agg}(p, Q)$.

$\text{dist}_{agg}(p, Q) = \sum_{k \in [1, n]} \text{dist}(p, q_k)$ represents the total distance from p to all users in Q .

Figure 3 shows an example of CANNQ. By the definition of CANNQ, $Q = \{q_1, q_2, q_3\}$, $P = \{p_1, p_2\}$, $T = \{t_1, t_2\}$. At time t_1 , the list of nearest neighbors $NN_{t_1} = \{p_2, p_1\}$ where p_2 is the aggregate nearest neighbor, and at time t_2 , the list of nearest neighbors $NN_{t_2} = \{p_1, p_2\}$, where p_1 is the aggregate nearest neighbor.

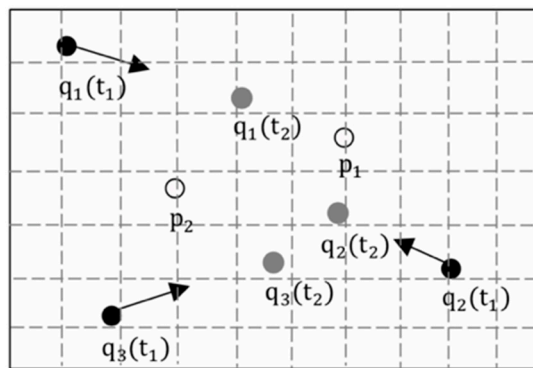


Figure 3. Example of a continuous aggregate nearest neighbor queries (CANNQ).

4. Our PCANNQ Scheme

In this section, first, we introduced the functions of entities in our system. Then, we elaborated the work process of the PCANNQ scheme. Finally, we analyzed the privacy security of the PCANNQ scheme. The important notations used in the PCANNQ scheme are listed in Table 1 for reference.

Table 1. Summary of notations.

| Notation | Description | Notation | Description |
|---------------------------|--|-------------|---------------------------------------|
| P | set of POIs | P_G | global public nearest neighbor |
| Q | set of users | P_L | local public nearest neighbor |
| S_q | nearest neighbor set of user q | S_{prun} | prune set |
| S_Q | nearest neighbor set of group Q | P_{ann}^k | the k th aggregate nearest neighbor |
| $\text{dist}(p, q)$ | distance between p and q | p^* | dominated POI |
| $\text{dist}_{agg}(p, Q)$ | aggregate distance between p and Q | R | set of secure areas |
| p_i^j | the j th nearest neighbor of q_i | R^* | set of maximum secure areas |

4.1. System Architecture

To enhance the privacy of users, we proposed a new PCANNQ scheme. Figure 4 shows the system architecture and some details. The TAS is the core component of this architecture and is responsible for privacy protection. The main functions include—group request anonymization, secure areas construction, and false results filtering. The LSP is a location service provider. The main functions include—POIs data loading and aggregate nearest neighbor query.



Figure 4. Details of our system architecture.

Although adding the TAS between users and the LSP will inevitably add an extra privacy protection overhead, it is reasonable, because (1) the cost of privacy protection accounts for a small proportion of the total service cost. (2) Users do not need to maintain real-time communication with the LSP, because the query results at some moments can be directly obtained from the cache of the TAS, so the efficiency of the overall service is improved, and users leak fewer locations to the LSP. (3) The LSP only needs to maintain the POIs data and the query algorithm itself, and then throw the results to the TAS, so the function is simpler.

4.2. Process of PCANNQ

The work process of PCANNQ is shown as follows.

- Step1. Users Q send requests to the TAS to build a query group. $req_{q \in Q} = (q, P.type) = (Id_q, \langle x_q, y_q \rangle, P.type)$ where $Id_q, \langle x_q, y_q \rangle$ and $P.type$ represents the identity of user q , the location of user q and the type of POIs, respectively.
- Step2. After receiving the request req_q , the TAS determines whether it is the first time for the user group to issue the query, if so, turn to Step 3–7, and if not, turn to Step 8.
- Step3. The TAS constructs the cloaking regions for every user in the group, which covers their true locations to satisfy spatial k -anonymity, and then converts these cloaking regions into k group requests sent to the LSP. $req_{i \in [1, k]} = \{Id_{g_i}, \langle x_1^i, y_1^i \rangle, \langle x_2^i, y_2^i \rangle, \dots, \langle x_n^i, y_n^i \rangle, P_i.type\}$ where Id_g represents the identity of the group which replaces Id_q . As shown in Figure 5, the TAS represents all users' Id as a group Id , and five indistinguishable group requests are generated in this anonymous process.
- Step4. The TAS maintains a group-user mapping table locally, which records the corresponding relationship between the group Id and user Id . When the LSP returns the results, the table can help the TAS to quickly filter out the false results. As shown in Figure 5, the number of users $|Q| = 3$, the degree of anonymity $k = 5$. A record with a value of 1 in the 'State' field (e.g., the third record) represents the real group of users, and the other items are k -anonymized false records.
- Step5. The LSP performs an ANNQ, based on strategy optimization (see Section 6 for details) for all requests, generates a lists of nearest neighbors and then returns the results to the TAS. $res_{i \in [1, k]} = \{Id_{g_i}, NN\}$ where $NN = \{p_{ann}^1, p_{ann}^2, \dots, p_{ann}^m\}$.
- Step6. The TAS filters the results based on the group-user mapping table and returns the true neighbor lists to the true group of users.
- Step7. The TAS calculates the radius by $dist(p_{ann}^1, Q)_{max}$ and $dist(p_{ann}^2, Q)_{max}$, and constructs circular secure areas for the group of users (see Section 5 for details).
- Step8. The TAS monitors the group of users in real-time to see whether the users exceed their secure areas. If someone exceeds, the user group is prompted to update the query results. If the request needs to be resubmitted, turn to Step 3. If not, continue to monitor until the group of users end the query.

| Q | u_1 | u_2 | u_3 | | | | | |
|------------------------------|---|---|---|--------------------------|------------------------------|------------------------------|------------------------------|-------|
| (Id, Location) | (Id ₁ , < x ₁ , y ₁ >) | (Id ₂ , < x ₂ , y ₂ >) | (Id ₃ , < x ₃ , y ₃ >) | | | | | |
| | | | | group-user mapping table | | | | |
| group requests anonymization | | | | | | | | |
| K=5 | u_1 | u_2 | u_3 | Group-User | u_1 | u_2 | u_3 | State |
| Id _{g1} | < x ₁ ¹ , y ₁ ¹ > | < x ₂ ¹ , y ₂ ¹ > | < x ₃ ¹ , y ₃ ¹ > | Id _{g1} | Id ₁ ¹ | Id ₂ ¹ | Id ₃ ¹ | 0 |
| Id _{g2} | < x ₁ ² , y ₁ ² > | < x ₁ ² , y ₁ ² > | < x ₁ ² , y ₁ ² > | Id _{g2} | Id ₁ ² | Id ₂ ² | Id ₃ ² | 0 |
| Id _{g3} | < x ₁ ³ , y ₁ ³ > | < x ₁ ³ , y ₁ ³ > | < x ₁ ³ , y ₁ ³ > | Id _{g3} | Id ₁ ³ | Id ₂ ³ | Id ₃ ³ | 1 |
| Id _{g4} | < x ₁ ⁴ , y ₁ ⁴ > | < x ₁ ⁴ , y ₁ ⁴ > | < x ₁ ⁴ , y ₁ ⁴ > | Id _{g4} | Id ₁ ⁴ | Id ₂ ⁴ | Id ₃ ⁴ | 0 |
| Id _{g5} | < x ₁ ⁵ , y ₁ ⁵ > | < x ₁ ⁵ , y ₁ ⁵ > | < x ₁ ⁵ , y ₁ ⁵ > | Id _{g5} | Id ₁ ⁵ | Id ₂ ⁵ | Id ₃ ⁵ | 0 |
| CR | [x ₁ ² , y ₁ ²] | [x ₁ ³ , y ₁ ³] | [x ₁ ⁴ , y ₁ ⁴] | | | | | |

Figure 5. Anonymous process.

4.3. Security Analysis

First, we analyzed the location security in the snapshot queries and the ability of our scheme to resist the inference attack of the LSP. As shown in Figure 6a, the degree of anonymity $K = 5$. At each timestamp, the TAS randomly generated 4 dummy locations to conceal the real location. Since the LSP had background knowledge of the query probability of each location, the LSP could easily filter out some dummy locations (e.g., mountains, lakes) when the query probabilities of the randomly generated dummy locations were different, so the spatial K-anonymity method could not resist the inference attack of the LSP. Therefore, an entropy-based spatial K-anonymity method was adopted, where entropy represented the uncertainty of determining the true location from the candidate set, which was used to measure the anonymity of the method. Specifically, the entropy could be represented by H , as shown in Formula (1).

$$H = - \sum_{i=1}^k p_i \cdot \log_2 p_i \tag{1}$$

The maximum entropy $H_{\max} = \log_2 k$ was reached when the k locations had the same probability $1/k$. Therefore, our method selected the locations with the same query probability as the dummy locations, to resist the inference attack of the LSP.

Then, we analyzed the trajectory security in continuous queries and the ability of our scheme to resist the spatial-temporal correlation attacks of the LSP. As mentioned in Figure 2b, the cloaking regions in Figure 6a showed a spatial-temporal correlation. When the LSP detected that all locations of a certain user were in these regions, it could infer that the user had a high probability of being the user who issued the query, and then reconstructed the user’s trajectory. Figure 6b shows the location submission in the PCANNQ scheme, where the black dots (e.g., t_1, t_4) were submitted to the LSP and the gray dots (t_2, t_3) were uncommitted because they were concealed in the secure area.

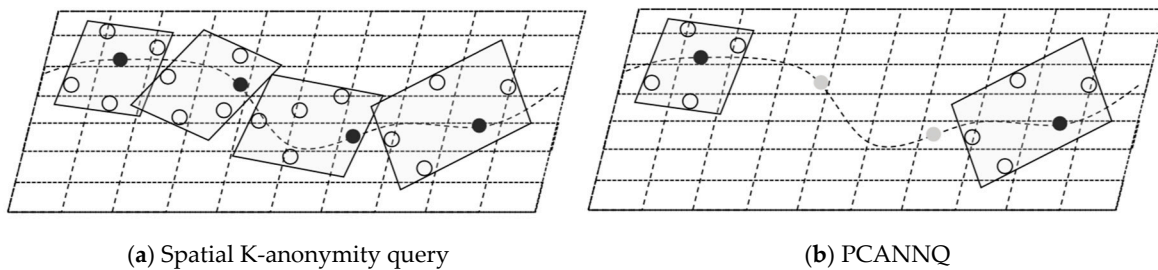


Figure 6. K-anonymity query and PCANNQ.

The principle of the PCANNQ scheme is that the TAS deconstructs continuous queries into irrelevant snapshot queries, and the secure areas can conceal part of the users’ trajectories, so that the cloaking regions generated in each query lose the spatial-temporal correlation. Therefore, even if the

LSP monitors that a user is completely in these discrete cloaking regions, it cannot reconstruct their trajectory. We use η to measure the security of trajectory privacy, as shown in Formula (2).

$$\eta = \frac{\text{\#actual queries}}{\text{\#total queries}} \quad (2)$$

Given the total number of queries, fewer actual queries means less private information is exposed to the LSP and a lower probability that the LSP reconstructs the user's trajectory.

Theoretically, our PCANNQ scheme can effectively resist the inference attack and the spatial-temporal correlation attack of the LSP.

5. Circular Secure Areas

In this section, we first introduced the motivation and the function of secure areas. Then, we elaborated on the verification and construction method of secure areas. Finally, we designed a PCANNQ based on the circular secure areas.

5.1. Motivation

Conventional continuous queries were usually converted to snapshot queries at each timestamp. However, it was found that not every time the location of the user changed did it lead to a change of the optimal nearest neighbor. The reason was that the user had a secure area with respect to the current optimal nearest neighbor, that is to say, this area was dominated by the optimal nearest neighbor. When the user moved in the area, the optimal nearest neighbor was always the same. When the user left the area, the query result was changed to other nearest neighbors.

Similarly, in a PCANNQ, users in the group also had their own secure areas, with respect to the current optimal aggregate nearest neighbor. When users moved in their secure areas, the optimal aggregate nearest neighbor was always the same, so the group did not have to continue to send query requests to the LSP, which meant that the users reduced the exposure of their trajectories. Therefore, the purpose of this section was to calculate the secure areas for the group of users.

Definition 1. Secure Areas. Given a group of users $Q = \left\{ q_i \mid i = 1, \dots, n \right\}$ and a group of areas $R = \left\{ R_i \mid i = 1, \dots, n \right\}$, if $\forall \langle q_1, q_2, \dots, q_n \rangle \in R_1 \times R_2 \times \dots \times R_n$ which guarantees that the optimal aggregate nearest neighbor p^* is invariable, then R are called secure areas.

As shown in Figure 7, we assumed that the dotted line was the moving boundary of each user. When users moved in their own areas, their optimal aggregate nearest neighbor was always p_1 . When the boundary was crossed, the aggregate nearest neighbor changed to other points, such as p_2 .

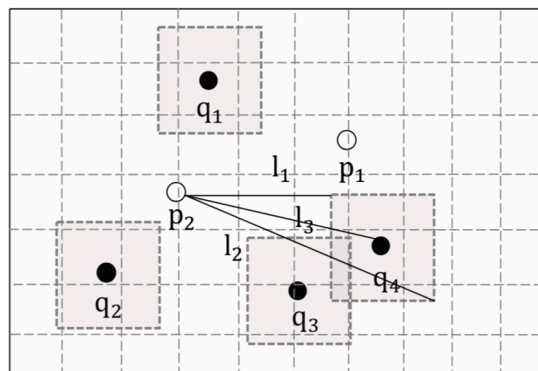


Figure 7. Secure areas and dominated distances.

Definition 2. Maximum Secure Areas. Given a group of users $Q = \left\{ q_i \middle| i = 1 \dots n \right\}$ and a group of areas $R^* = \left\{ R_i^* \middle| i = 1 \dots n \right\}$, if $\forall R' = \left\{ R_i' \middle| i = 1 \dots n \right\}$, $R' \neq R^*$, satisfy $R_i' \subseteq R_i^*$, then R^* are called maximum secure areas.

Of course, the larger the secure areas, the higher the privacy security, but it also brings greater computing overhead, so this study aimed to construct more efficient secure areas.

After this, we considered under what conditions the constructed areas formed the secure areas (see details in Section 5.1) and how could the secure areas be maximized (see details in Section 5.2).

5.2. Verification of Secure Areas

If $R = \left\{ R_i \middle| i = 1 \dots n \right\}$ are secure areas, then the conditions of definition 1 should be satisfied, but there are infinite number of combinations of user locations in these areas. It is not feasible to test all instances one by one, but we can find the critical value from the first aggregate nearest neighbor to the second aggregate nearest neighbor. The verification method is as follows.

Definition 3. Dominated Distance. Given a group of users $Q = \left\{ q_i \middle| i = 1 \dots n \right\}$ and $p \in P$, the dominated distance of p is $dist_{dom}(p, Q)$, as show in Formula (3).

$$dist_{dom}(p, Q) = \max_{q \in Q} (dist(p, q)) \quad (3)$$

Definition 4. Max/minimum Dominated Distance. Given $p \in P$ and area set $R = \left\{ R_i \middle| i = 1 \dots n \right\}$, the maximum dominated distance of p is $dist_{dom_up}(p, R)$, as shown in Formula (4).

$$dist_{dom_up}(p, R) = \max_{R_i \in R} (dist(p, R_i)_{max}) \quad (4)$$

the minimum dominated distance of p is $dist_{dom_down}(p, R)$, as shown in Formula (5).

$$dist_{dom_down}(p, R) = \max_{R_i \in R} (dist(p, R_i)_{min}) \quad (5)$$

As shown in Figure 7, we suppose the grey areas are the secure areas. Take p_2 as an example, the dominated distance of p_2 is $dist_{dom}(p_2, Q) = l_3$, the maximum dominated distance of p_2 is $dist_{dom_up}(p, R) = l_2$, the minimum dominated distance of p_2 is $dist_{dom_down}(p, R) = l_1$.

If the set of secure areas $R = \left\{ R_i \middle| i = 1 \dots n \right\}$ belongs to the current optimal aggregate nearest neighbor, that is to say, when the users are in their respective secure areas, p^* will not be replaced by any other $p \in P - \{p^*\}$, then the dominated distance of p^* should be guaranteed to be less than or equal to that of p . The Theorem 1 is shown below.

Theorem 1. Given a group of users $Q = \left\{ q_i \middle| i = 1 \dots n \right\}$ and a set of secure areas $R = \left\{ R_i \middle| i = 1 \dots n \right\}$, if $\forall p \in P - \{p^*\}$, satisfy

$$dist_{dom_up}(p^*, R) \leq dist_{dom_down}(p, R) \quad (6)$$

then $dist_{dom}(p^*, Q) \leq dist_{dom}(p, Q)$.

Proof.

1. Let $Q_{\text{test}} = \left\{ q_i \mid \begin{matrix} n \\ i = 1 \end{matrix} \right\} \in R$.
2. By definition 4, $\text{dist}_{\text{dom}}(p^*, Q_{\text{test}}) \leq \text{dist}_{\text{dom_up}}(p^*, R)$, $\text{dist}_{\text{dom_down}}(p^*, R) \leq \text{dist}_{\text{dom}}(p, Q_{\text{test}})$.
3. $\therefore \text{dist}_{\text{dom_up}}(p^*, R) \leq \text{dist}_{\text{dom_down}}(p, R)$.
4. $\therefore \text{dist}_{\text{dom}}(p^*, Q) \leq \text{dist}_{\text{dom}}(p, Q)$.

□

5.3. Construction of Secure Areas

Considering that the secure area should be constructed as conveniently as possible, this study approximated the secure area as a circle, as shown in Figure 8. At the same time, the radius of the circular area should be as large as possible, in order to reduce the communication frequency between the user and the LSP. Theorem 2 is used to calculate the maximum radius of a circular secure area, so that the areas remain valid for the current optimal aggregate nearest neighbor.

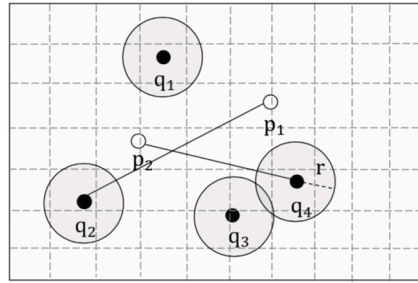


Figure 8. Circular secure areas.

Theorem 2. Given a group of users $Q = \left\{ q_i \mid \begin{matrix} n \\ i = 1 \end{matrix} \right\}$ and a set of secure areas $R = \left\{ R_i \mid \begin{matrix} n \\ i = 1 \end{matrix} \right\}$ where $R_i = C(q_i, r_{\text{max}})$ represents a circle with center q_i and radius r_{max} . If the current optimal aggregate nearest neighbor $p_{\text{ann}}^1 = p^*$, then

$$r_{\text{max}} = \frac{\min_{p \in P - \{p^*\}} (\text{dist}(p, Q)_{\text{max}}) - \text{dist}(p^*, Q)_{\text{max}}}{2} \quad (7)$$

Proof.

1. Let $R_i = C(q_i, r)$.
2. By definition 7, $\text{dist}_{\text{dom_up}}(p^*, R) = \max_{R_i \in R} (\text{dist}(p^*, R_i)_{\text{max}}) = \max_{q_i \in Q} (\text{dist}(p^*, q_i) + r)$,
and $\text{dist}_{\text{dom_down}}(p, R) = \max_{R_i \in R} (\text{dist}(p, R_i)_{\text{min}}) = \max_{q_i \in Q} (\text{dist}(p, q_i) - r)$.
3. By condition (6), $\forall p \in P - \{p^*\}$, satisfy $\text{dist}_{\text{dom_up}}(p^*, R) \leq \text{dist}_{\text{dom_down}}(p, R)$.
4. $\therefore \max_{q_i \in Q} (\text{dist}(p^*, q_i) + r) \leq \max_{q_i \in Q} (\text{dist}(p, q_i) - r)$.
5. $\therefore r \leq \frac{\max_{q_i \in Q} (\text{dist}(p^*, q_i)) - \max_{q_i \in Q} (\text{dist}(p, q_i))}{2}$.
6. $\therefore \forall p \in P - \{p^*\}, r \leq \frac{\text{dist}(p, Q)_{\text{max}} - \text{dist}(p^*, Q)_{\text{max}}}{2}$.
7. \therefore take the minimum of $\text{dist}(p, Q)_{\text{max}}$.
8. $\therefore r_{\text{max}} = \frac{\min_{p \in P - \{p^*\}} (\text{dist}(p, Q)_{\text{max}}) - \text{dist}(p^*, Q)_{\text{max}}}{2}$.

□

In fact, p is the second aggregate nearest neighbor p_{ann}^2 of the group of users. The purpose of constructing the secure area is to reduce the frequency of communication between the user and the LSP, thereby, reducing the probability that the LBS server reconstructs the user's trajectory, which is suitable for continuous queries.

Therefore, a circular secure areas construction algorithm is shown in Algorithm 1.

Algorithm 1: Circular secure areas construction

Input: P, Q

Output: R

```

1.  while  $Q_{\text{current}}$  not in  $\times_{i=1}^n R_i$  do
2.   $p^* \leftarrow \text{SOANN}(p, Q_{\text{current}})$  // Algorithm 4
3.   $p \leftarrow \text{SOANN}(P - \{p^*\}, Q_{\text{current}})$  // Algorithm 4
4.  compute  $r_{\text{max}} \leftarrow \frac{\min_{p \in P - \{p^*\}} (\text{dist}(p, Q_{\text{max}}) - \text{dist}(p^*, Q_{\text{max}}))}{2}$  // Formula (7)
5.  for each  $q_i \in Q_{\text{current}}$  do
6.   $R_i \leftarrow C(q_i, r_{\text{max}})$ 
7.  end for
8.  end while

```

In Algorithm 1, the first line limits the condition to enter the loop body, which includes two cases: (1) The group of users issue the query for the first time, that is, the secure areas have not been constructed; (2) a user in the group is about to leave the secure areas, that is, the optimal aggregate nearest neighbor changes. Lines 2–3 call Algorithm 4 (see details in Section 6) to find the optimal aggregate nearest neighbor and the second aggregate nearest neighbor. Line 4 calculates the maximum radius by Formula (7). Lines 5–7 construct circular secure areas for the group of users and monitors the users' movement in real time. Once a user leaves his secure area, it needs to reacquire the user's new location and send a new group request to the LSP.

6. ANNQ Based on Strategy Optimization

In this section, we first introduced the motivation of the section. Then, we elaborated on the search and prune phase of the ANNQ. Finally, we proposed the ANNQ, based on strategy optimization.

6.1. Motivation

Referring to the system architecture shown in Figure 4, we could derive the response delay (rd) of a complete query service, as shown in Formula (8).

$$\begin{aligned} \text{rd} = & \text{cd between user and TAS} \times 2 + \text{anonymous processing delay} \\ & + \text{cd between TAS and LSP} \times 2 + \text{ANNQ delay} \end{aligned} \quad (8)$$

Among them, cd represents communication delay, which is almost negligible, the anonymous processing delay on the TAS includes anonymous protection and filtering, and the ANNQ delay includes the traversal of the POIs index tree and filtering out the optimal location point that satisfies the user. When 'ANNQ delay \gg other delay', the ANNQ delay becomes the main reason that affects the total service response delay. Therefore, the purpose of this section is to optimize the query algorithm.

We divide the aggregate nearest neighbor query algorithm into the search phase and the prune phase, in which the search phase determines the search scope and the prune phase determines the aggregate nearest neighbor. Furthermore, we made two improvements—(1) optimization of the search strategy and further narrowing of the search scope; and (2) optimization of the prune strategy and acceleration of the prune speed.

6.2. Search Phase

The purpose of the search phase is to limit the search scope of POIs, and ensure that the aggregate nearest neighbor must be within this scope, so Theorem 3 is introduced, as shown below.

Theorem 3. Given a group of users $Q = \left\{ q_i \mid i = 1 \dots n \right\}$, each user's nearest neighbor set $S_i = \left\{ p_i^l \mid l = 1 \dots m \right\}$, $\forall j < k, j, k \in [1, m]$ satisfies $\text{dist}(p_i^j, q_i) \leq \text{dist}(p_i^k, q_i)$. Then, extending the nearest neighbor set of users in a group with any search strategy, as long as $S_1 \cap S_2 \cap \dots \cap S_n = \{p_G\}$, where p_G is the first point that all users once searched, $p_{\text{ann}} \in S_1 \cup S_2 \cup \dots \cup S_n$. The process of proof is shown below.

Proof. Use proof by contradiction.

1. Let $S_Q = S_1 \cup S_2 \cup \dots \cup S_n$. Suppose $p_{\text{ann}} = p_G$, but $p_{\text{ann}} \notin S_Q$.
2. $\therefore p_G \notin S_Q$.
3. $\therefore p_G \notin S_{i \in [1, n]}$.
4. Combined with Theorem 3, $\text{dist}(p_G, q_i) \geq \text{dist}(p, q_i), p \in S_Q$.
5. $\therefore \text{dist}_{\text{agg}}(p_G, Q) = \sum_{i \in [1, n]} \text{dist}(p_G, q_i), \text{dist}_{\text{agg}}(p, Q) = \sum_{i \in [1, n], j \in [1, m]} \text{dist}(p_i^j, q_i)$.
6. $\therefore \text{dist}_{\text{agg}}(p_G, Q) > \text{dist}_{\text{agg}}(p, Q)$. It's conflicted with above assumption.
7. $\therefore p_{\text{ann}} \in S_1 \cup S_2 \cup \dots \cup S_n$.

□

It has been indicated that, irrespective of whether the search strategy is good or bad, it can only determine the size of S_Q and cannot change the result of aggregate nearest neighbor query. From Theorem 3, the end condition of the computing search scope S_Q is to find a global public nearest neighbor p_G . Therefore, the goal of this section is to obtain the p_G , as soon as possible, which is related to the specific search strategy. For example, Table 2 records the nearest neighbor sets of each user in the group, when p_G appears. In this case, $p_G = p_2$, and the search scope $S_Q = \{p_1, p_2, p_3, p_4\}$.

Table 2. Example of the nearest neighbor sets at the end of the search phase.

| Q | S | D/km |
|-------|---------------------------|---------------------------|
| u_1 | $S_1 = \{p_1, p_3, p_2\}$ | $D_1 = \{1.1, 2.1, 3.5\}$ |
| u_2 | $S_2 = \{p_2, p_4\}$ | $D_2 = \{1.0, 2.3\}$ |
| u_3 | $S_3 = \{p_2\}$ | $D_3 = \{3.0\}$ |

From the vertical view of the table, we considered the extension order of the user's nearest neighbor set. From the horizontal view of the table, we could set the conditions that p was added to the user's nearest neighbor set.

This study argued that the distribution of the group of users affects the results of the aggregate nearest neighbor queries. When the distribution of the group of users was not uniform, that is, there was a relative aggregate subgroup, the p_{ann} tended to be POI, near the aggregate subgroup. The reason was that when a POI was far from the aggregate subgroup, the distance from the POI to the aggregate subgroup would increase, and the distance from the POI to a few objects would be reduced, and the total aggregate distance would still increase. Therefore, the definition of an aggregate subgroup is as follows.

Definition 5. Aggregate Subgroup. Given a group of users $Q = \left\{ q_i \mid i = 1 \dots n \right\}$, each user's nearest neighbor set $S_i = \left\{ p_l^i \mid l = 1 \dots m \right\}$, $\exists p_L \in S_i$, satisfy

$$\left| \left\{ S_i \mid p_L \in S_i \right\} \right| \geq \frac{|Q|}{e} \tag{9}$$

then $Q = \left\{ q_i \mid \left| \left\{ S_i \mid p_L \in S_i \right\} \right| \geq \frac{|Q|}{e} \right\}$ constitutes an aggregate subgroup. $\left| \left\{ S_i \mid p_L \in S_i \right\} \right|$ represents the size of the aggregate subgroup. p_L is the local aggregate nearest neighbor of the aggregate subgroup, and e is a regulator used to control the size of the aggregate subgroup.

As shown in Figure 9, $\left| \left\{ S_1, S_2, S_3 \right\} \right| = \left| \left\{ p_2, p_2, p_2 \right\} \right| = 3 > 5/2$, so q_1, q_2 , and q_3 constitute an aggregate subgroup, and the local public POI is $p_L = p_2$. At this point, the remaining q_4 and q_5 needed to be extended preferentially because extending the nearest neighbor set of the users outside the aggregate subgroup could search for the POIs around the aggregate subgroup, as soon as possible, thus, accelerating the appearance of the global public POI.

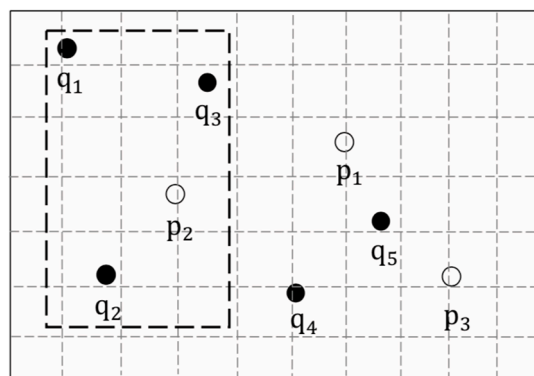


Figure 9. Aggregate subgroup.

In addition, we also considered the direction when searching the user's nearest neighbors. Since the gathering place of the group tended to be near their centroid, so the POIs towards the centroid should be preferentially searched, compared to the POIs in the opposite direction of the centroid.

Definition 6. Towards the Centroid. Given a group of users $Q = \left\{ q_i \mid i = 1 \dots n \right\}$, the centroid of group

$q_c = \left(\frac{\sum_{i=1}^{|Q|} x_i}{|Q|}, \frac{\sum_{i=1}^{|Q|} y_i}{|Q|} \right)$. Suppose the nearest neighbor set of q_{cur} is extended, and the current searched nearest neighbor is p . If

$$\vec{q}_{cur} q_c \cdot \vec{q}_{cur} p > 0 \tag{10}$$

then p is in the centroid direction of q_c , else, p is in the inverse centroid direction of q_c .

Each user in the group follows the principle of 'Towards the Centroid', when extending their nearest neighbor sets, which makes it faster to find the global public POI. As shown in Figure 10, q_c is the centroid of q_1 to q_5 , p_3 is the nearest neighbor of q_5 , but $\vec{q}_5 q_c \cdot \vec{q}_5 p_3 < 0$, which means that p_3 is in the inverse centroid of q_5 , so p_3 would not be added to q_5 's nearest neighbor set.

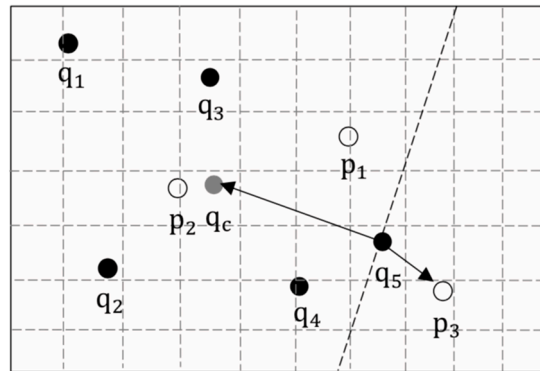


Figure 10. Towards the centroid.

Combining the above two optimizations, the pseudocode of the search phase is shown in Algorithm 2. The main idea is, first, calculate the centroid of the group, and then specify that all users search for the nearest neighbors, based on the principle of ‘Towards the Centroid’, then search for the nearest neighbor p_i^1 of each user, add to the nearest neighbor set S_i , and calculate $\text{dist}(p_i^1, q_i)$; second, extend the nearest neighbor set, based on the current shortest distance, until the aggregate subgroup and local public nearest neighbor are found; finally expand their nearest neighbor sets, based on the subgroup, until the global public nearest neighbor appears. The search scope is limited to the union of all users’ nearest neighbor sets.

Algorithm 2: Search global public nearest neighbor

Input: P, Q

Output: $S, S_Q, P_G, \text{dist}_{\text{agg}}$

1. **for** each $q_i \in Q$ **do**
 2. $S_i \leftarrow \emptyset$
 3. search q_i 's first NN p_i^1
 4. $S_i \leftarrow S_i \cup \{p_i^1\}, d_i \leftarrow \text{dist}(p_i^1, q_i)$
 5. **end for**
 6. compute $pd \leftarrow \left| \{S_i \mid p_L \in S_i\} \right|$
 7. **while** $pd < \frac{|Q|}{e}$ **do**
 8. select q_{cur} who hasmin(d_{cur})
 9. search q_{cur} 's next NN p
 10. **if** $\vec{q}_{\text{cur}} \cdot \vec{q}_c \cdot \vec{q}_{\text{cur}} p > 0$ **then**
 11. $S_{\text{cur}} \leftarrow S_{\text{cur}} \cup \{p\}, d_i \leftarrow \text{dist}(p, q_{\text{cur}})$
 12. compute $pd \leftarrow \left| \{S_i \mid p_L \in S_i\} \right|$
 13. **end while**
 14. compute Outgroup $\leftarrow \{q_i \mid p_L \notin S_i\}$
 15. **while** $\bigcap_{i=1}^n S_i \neq \emptyset$ **do**
 16. **for** each $q_i \in \text{Outgroup}$ **do**
 17. select q_{cur} who hasmin(d_{cur})
 18. search q_{cur} 's next NN p
 19. **if** $\vec{q}_{\text{cur}} \cdot \vec{q}_c \cdot \vec{q}_{\text{cur}} p > 0$ **then**
 20. $S_{\text{cur}} \leftarrow S_{\text{cur}} \cup \{p\}, d_i \leftarrow \text{dist}(p, q_{\text{cur}})$
 21. **end for**
 22. **end while**
 23. **return** $S, S_Q, P_G = \bigcap_{i=1}^n S_i, \text{dist}_{\text{agg}}(P_G, Q)$
-

In Algorithm 2, lines 1~5 initialize the relevant variables, and lines 6~22 are the core part of the algorithm. Lines 6~13 analyzes the positional relationship of the user in the group, to determine the aggregate group. Lines 14~22 determine the search scope based on Theorem 3, ensuring that the aggregate nearest neighbor must be in it, and that the search scope is the smallest. Line 23 returns the intermediate values, such as the nearest neighbor sets and the global public POI, which is used as the input parameters for the prune phase.

6.3. Prune Phase

After obtaining the search scope S_Q , this section aimed to quickly prune the non-nearest neighbors. We consider a situation—when a user q_i 's nearest neighbor set has already contained all POIs in S_Q , we should not extend his nearest neighbor set any more, because if we continue to extend the user, the next nearest neighbor must not be in S_Q . Then, the prune strategy cannot prune any POI in S_Q , which is equivalent to an invalid extension.

Therefore, we should give priority to the expansion of the user whose $|S_i|$ is minimum because the smaller the $|S_i|$, the nearer the q_i is to the aggregate nearest neighbor, the larger the prune set that can be constructed, and the more non-neighbor POIs that can be pruned. The pseudocode of the prune phase is shown in Algorithm 3.

Algorithm 3: Prune non-nearest neighbors

Input: $S, S_Q, P_G, \text{dist}_{\text{agg}}$

Output: p_{ann}

```

1.  while  $|S_Q| > 1$  do
2.    Search  $q_{\text{cur}}$  who hasmin( $|S_{\text{cur}}|$ )
3.    Compute  $q_{\text{cur}}$ 's next NN  $p$ 
4.     $S_{\text{cur}} \leftarrow S_{\text{cur}} \cup \{p\}$ 
5.     $\text{dist}_{\text{agg}}^p \leftarrow \sum_{i=1}^n \text{dist}(x_i, q_{\text{cur}}), x_i \leftarrow \begin{cases} p, & p \in S_i \\ p_i^1, & p \notin S_i \end{cases}$ 
6.    if  $\text{dist}_{\text{agg}}^p > \text{dist}_{\text{agg}}$  then
7.      for each  $q_i \in Q$  do
8.        if  $p \in S_i$  then
9.           $S_i^p \leftarrow \{o | \text{dist}(o, q_i) \geq \text{dist}(p, q_i)\}$ 
10.       else
11.          $S_i^p \leftarrow \emptyset$ 
12.       end for
13.      $S_Q \leftarrow S_Q - (\cup_{i=1}^n S_i^p)$ 
14.   else
15.     if  $p \in \cap_{i=1}^n S_i$  then
16.        $S_Q \leftarrow S_Q - \{p\}$ 
17.      $p_G \leftarrow p$ 
18.      $\text{dist}_{\text{agg}} \leftarrow \text{dist}(p_G, Q)$ 
19.   end if
20. end while
21. output  $p_{\text{ann}} = p_G$ 

```

In Algorithm 3, the first line determines whether there is any POI in the search scope S_Q . If so, it enters the loop body, else, it exits the loop. In the loop body, lines 2~5 extend the user q_i with the smallest nearest neighbor set $|S_i|$, by an efficient expansion mode. Lines 6~13 construct the prune set S_i^p

and prune from S_Q . Lines 14~20 update the current aggregate nearest neighbor. Finally, when $|S_Q| = 1$, the aggregate nearest neighbor is the output.

The So-ANN query algorithm in this study is composed of Algorithms 2 and 3, as shown in Algorithm 4.

Algorithm 4 ANNQ based on strategy optimization

Input: P, Q

Output: P_{ann}

1. SGPNN() //Algorithm 2
 2. PNN() //Algorithm 3
-

The algorithm calculates the aggregate nearest neighbor of a group, by extending the nearest neighbor set of each user, and the most frequent operation ‘each q ’s next NN p ’ is hardly time consuming. Fortunately, with the support of Spatial Oracle’s spatial query component, it can create a kNN index, based on the current location of each user, once the POIs dataset is loaded into an Oracle database. Therefore, the algorithm does not need to care about how to search the next nearest neighbor of the user, and only needs to record the subscript of the nearest neighbor that is currently added to the user’s nearest neighbor set. The time complexity of Algorithm 4 is $O(|Q|)$ and the spatial complexity is $O(|S| + |S_Q|)$.

7. Simulation Experiment and Analysis

In this section, the experimental environment and parameter settings are introduced. Then, the state of security that the PCANNQ scheme can achieve is analyzed, and compared to the processing time ratio on the TAS and the LSP. Finally, we compared the performance of the SOANN with peer algorithms.

7.1. Experimental Setup

Our experiments were implemented by the Java Development Kit (JDK)-1.7 and Eclipse Integrated Development Environment (IDE), running on two local machines with an Intel Core-i7 2.5 GHz, 8 GB RAM, and Microsoft Windows 7 OS, to simulate the TAS and the LSP, in our system architecture.

As shown in Table 3, we used the simulator Network-Based Generator of Moving Objects [31], to generate mobile nodes and simulate their movement on the real map of Dongcheng District, Beijing, which was extracted from the Open Street Map (<https://www.openstreetmap.org/>), as well as the corresponding POIs. The district covered an area of approximately 5 km×13 km, with about 165,326 POIs, including various POI types, such as shopping, dining, medical care, and scenic spots. We stored shopping POIs and medical POIs in the Oracle Spatial and organized them with the R tree, because the distributions of these two types of POIs were significantly different, which was conducive to our experimental comparison.

Table 3. Data sets.

| Type | Name | Amount |
|----------------|-----------------------------|---------|
| POI set | Dongcheng District, Beijing | 165,326 |
| Trajectory set | Thomas Brinkhoff | 120 |

We explored several factors in our experiments as summarized in Table 4, with default values shown in bold. In every set of experiments, we only changed one parameter, with the rest set as their defaults. All default parameter values have been included in this table unless otherwise noted.

Table 4. Parameters and their values.

| Description | Notation | Ranges | Default Values |
|------------------------------|----------|------------|----------------|
| Number of users | n | 3–10 | 5 |
| Number of POIs | m | 5 K–15 K | 10 K |
| Distribution of users | a | 1–5% | 3% |
| Moving speed | v | 5–100 km/h | 50 km/h |
| Distribution of POIs | d | 10–100% | 50% |
| Anonymity degree | k | 2–10 | 6 |
| Number of continuous queries | q | 10–100 | 50 |

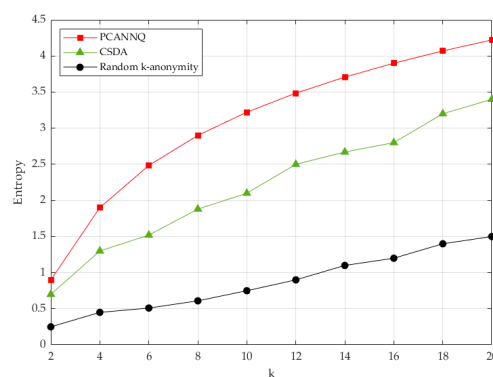
It is worth mentioning that the generation of the initial location of the group of users followed the following principles: $\forall Q = \left\{ q_i \mid \begin{matrix} n \\ i = 1 \end{matrix} \right\}$, satisfy that $\text{convex hull}(Q)$ covers the entire map. This is consistent with the scenario in the study that required users to converge from all directions. Additionally, the speed of users could be set to fast, medium, and slow, in Brinkhoff, respectively.

The experiment consisted of two parts. In the first part, we evaluated the security of our PCANNQ scheme. We selected the CSKA [18] scheme as the baseline algorithm, compared these two schemes in terms of the abilities to resist the LSP from inferential attack and spatial-temporal correlation attack, and further analyzed the advantage of our scheme. In the second part, we evaluated the performance of our PCANNQ scheme. We selected the VANN [26] and voronoi-based range spatial skyline algorithm (VRSSA) [7] as the baseline algorithms, compared these algorithms in terms of response delay, and further analyzed the key factors that affected the efficiency. All experiments were performed a 1000 times and the average was taken as the final comparison result.

7.2. Security Comparison

We focused on location security in the snapshot queries and the trajectory security in the continuous queries. The location security in the snapshot queries was guaranteed by entropy, which indicated the probability that the LSP determined the real location from the set candidate. A higher entropy meant a higher location security.

As shown in Figure 11, the spatial k-anonymity method used in our PCANNQ scheme could achieve the maximum entropy under different anonymity degrees, which was superior to the random k-anonymity method and the CSDA scheme. The reason was that we selected the locations which had the same query probability with the real location as the dummy locations, to prevent the LSP from inference attack, based on the background knowledge. Although the CSDA scheme did not take into account the background knowledge of the LSP, the CSDA scheme was still better than the random k-anonymity method because it did not generate dummy locations in impossible areas, such as lakes, mountains, etc.

**Figure 11.** Security of locations.

The trajectory security in continuous queries was guaranteed by η , which indicated the ratio of the actual number of query submissions to the total number of query submissions. Fewer the times that the queries were actually issued, the less private information was exposed to the LSP. In other words, we could improve the security of the trajectories by reducing the number of actual submitted queries.

Figure 12 shows the users' query submissions in continuous queries. In the CSKA scheme, (1) as the number of continuous queries increased, η gradually decreased and tended to be stable, since at the beginning, the hit rate was low, due to the small contents in the cache, but this situation was alleviated as the number of queries increased. (2) This method could reduce the submission of query requests, but η was always large in general (>0.8), because the query object was a group of users, not a single user, so the hit rate of similar queries in the TAS's cache was very low. (3) The mobile speed of users had almost no impact on η , because the caching technology had nothing to do with user speed.

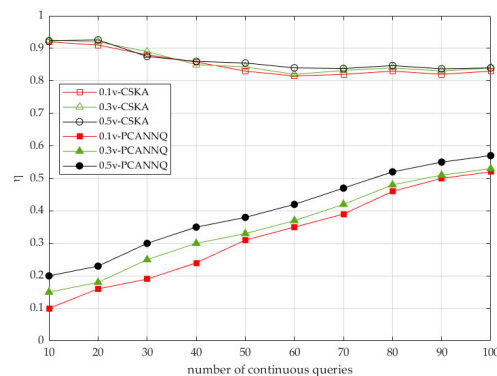


Figure 12. Security of trajectories.

In our PCANNQ scheme, (1) as the number of continuous queries increased, η increased. The reason was that, the more continuous the queries, the greater the moving distance of users, which led to a more frequent replacement of the optimal aggregate nearest neighbor. (2) Although the number of continuous queries had increased, η could still maintain a low value (<0.6) because the secure areas concealed the submission of partial query requests. (3) The user's moving speed affected η , and the faster the speed, the larger η . The reason was that, the faster the user moved, the more times the secure areas were exceeded, which led to more query requests that needed to be submitted.

To sum up, the advantage of our PCANNQ scheme was that, we actively constructed secure areas to conceal part of the users' trajectories, which not only overcame the problem of cold starts in the cache, but also greatly reduced the number of actual submitted queries. Therefore, our PCANNQ scheme could achieve a higher trajectory security. Then, we further analyzed the factors that affected the secure areas.

Figure 13a analyzes the effect of the distribution of POIs on η . The experimental results showed that the sparser the distribution of POIs, the larger was the η . Combined with Section 5, we knew that the calculation of the radius of secure areas depended on the calculation of $\text{dist}(p_{\text{ann}}^1, Q)_{\text{max}}$ and $\text{dist}(p_{\text{ann}}^2, Q)_{\text{max}}$. The sparser the distribution of POIs, the larger was the radius and the area of the secure areas are, so η was smaller. Figure 13b also verifies that η was negatively correlated with the area of the secure area. Additionally, the shopping POIs itself were denser than the medical POIs, in terms of the distribution characteristics. The experimental data showed that shopping POIs needed more data submissions than the medical POIs. Therefore, there was a suggestion that if the POIs information of the query was sensitive to the user, we could increase the distribution of such POIs, so that the correlation between the trajectory and the POIs information could be reduced.

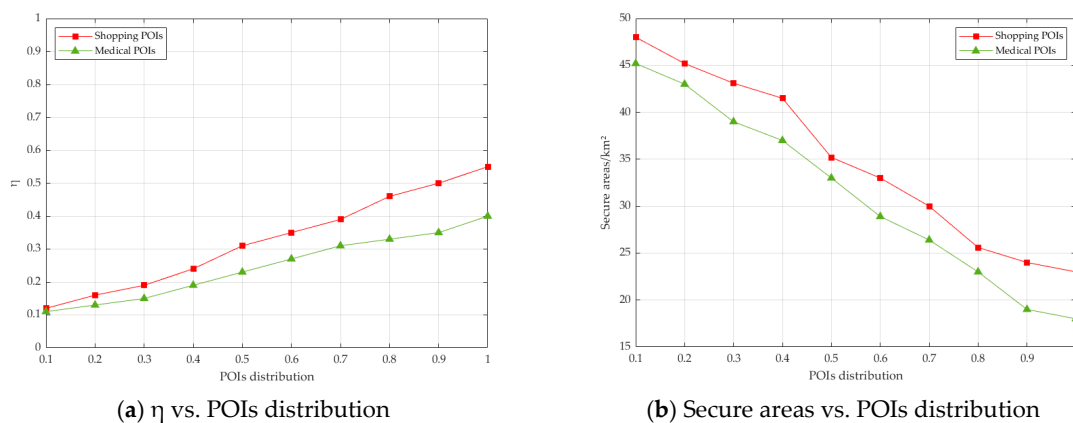


Figure 13. The effect of points of interest (POIs) distribution on η and Secure areas.

Finally, the experiment explored the applicable scope of our PCANNQ scheme. We set the number of users for n as 5, and the continuous query passed 1000 timestamps. In theory, the higher the moving speed, the higher the communication frequency, because the faster the speed, the faster the secure area will be exceeded.

As shown in Figure 14, as the speed increased, the communication frequency between the user and the LSP in our PCANNQ scheme showed an exponentially increasing trend. The user communicated with the TAS in real-time, so the communication frequency between the user and the TAS was constant. The difference between the two lines in the figure was the number of communications concealed by secure areas. Additionally, from 0.1 v to 0.6 v , a lower communication frequency could be guaranteed, and the speed of this level was equivalent to the speed of human walking and even the speed of a human riding. When the moving speed was greater than 0.6 v , the gap between our PCANNQ scheme and the CSKA scheme gradually decreased. Therefore, this algorithm was more applicable to the continuous query under the speed range between 0.1 v and 0.6 v .

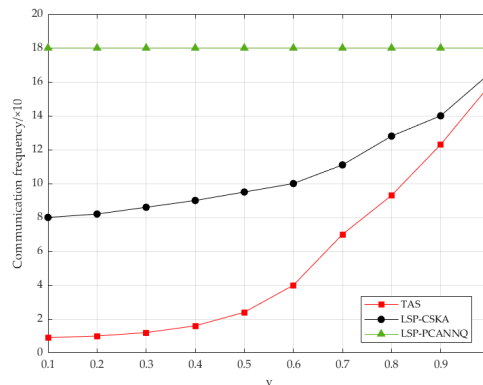


Figure 14. The effect of moving speed on communication frequency.

7.3. Efficiency Comparison

As show in Figure 15, we calculated the proportion of the processing time of the TAS and the LSP in the total query service. (1) The processing time of the TAS was much smaller than the processing time of the LSP, which was dominant in the total processing time. (2) With the increase of the number of continuous queries, the processing overhead of the LSP had a tendency to decline. Due to the secure areas, the users did not need to continuously send all requests to the LSP and some queries returned directly from the TAS, so the processing time of the TAS had increased slightly.

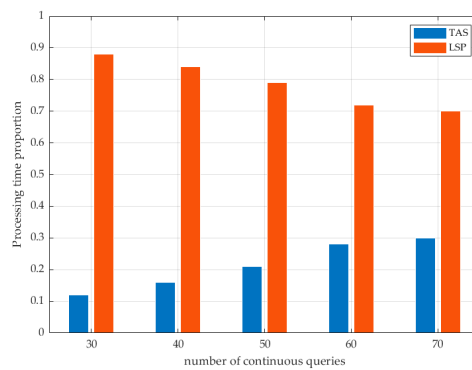


Figure 15. Service processing time proportion.

Therefore, we continued to optimize the query algorithm. The following experiment verified the performance of the SOANNQ proposed in this paper. We mainly explored the relationship between the query efficiency and the number of users and the degree of distribution between users.

In Figure 16a,b, the user distribution was first made constant ($a = 3\%$), and then the shopping POIs and the medical POIs were searched separately. The following points could be drawn from the figure. (1) The SOANN algorithm showed obvious advantages in the query response relay, which proved that this algorithm improved the VANN algorithm and had advantages over the VRSSA algorithm. (2) The SOANN algorithm showed linear growth, the VANN and VRSSA algorithms showed exponential growth. The larger was the n , the longer was the query response delay. (3) The SOANN and VANN algorithms were not sensitive to the type and distribution of POI. However, the TANN algorithm and the VRSSA algorithm were more sensitive to the type and distribution of POI, because the former did not need to retrieve all POI information, but the latter did need them.

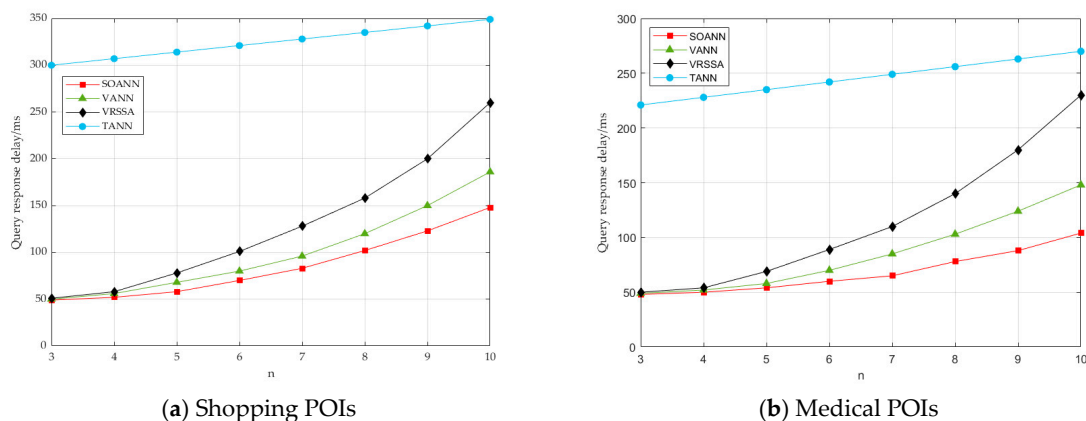


Figure 16. The effect of number of users on query response time.

In Figure 17a,b, the reason why the SOANN algorithm was superior to the VANN algorithm and the VRSSA algorithm is shown. This was because the key factor that influenced the query response time was whether the search scope could be minimized. It can be seen from the figure that the search scope constructed by the SOANN algorithm was the smallest, and with the increase of n , the change trend was the same, with a response delay of the query, which also proved the effectiveness of the searching strategy optimization proposed in this study.

Finally, we explored the influence of user distribution on the query response time. The user distribution was the ratio of the number of POIs and the total number of POIs included in the convex hull formed by the users in the group. The number of users in the group was $n = 5$.

Figure 18 shows that when the user distribution in the group was sparse, that is, the value of 'a' was large, the SOANN algorithm performed better because the SOANN algorithm analyzed the

distribution of users in the group and preferentially extended the outlier users. The VANN algorithm extended all users, one by one, in accordance with the principle of fairness, so the constructed search scope was also larger and the query response time was longer. The VRSSA algorithm needed to calculate all POIs contained in a convex hull, composed of the group of users, consequently, the cost became very expensive.

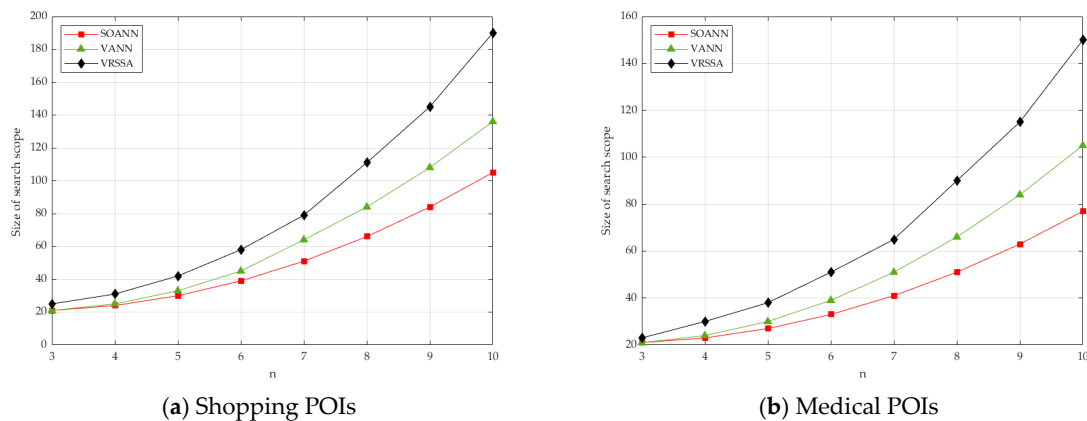


Figure 17. The effect of number of users on the searching set size.

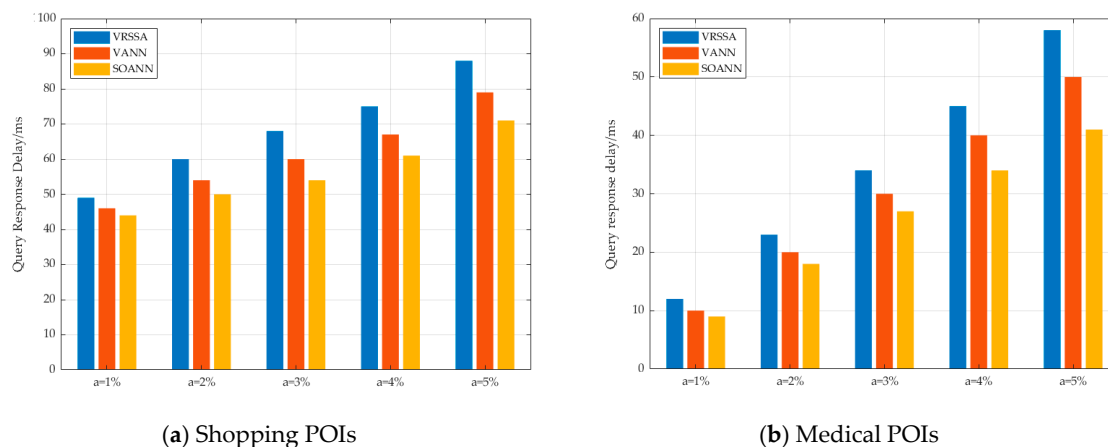


Figure 18. The effect of user distribution on query performance.

In summary, it could be concluded that our SOANN algorithm showed a good efficiency when the number of users was large and the distribution was dispersed.

8. Conclusions

In order to deal with the leakage of location and trajectory privacy faced by the IOT, when providing query services, we proposed a secure and efficient PCANNQ scheme to protect the location and trajectory privacy of the group of users. Specifically, the entropy based spatial K-anonymity method was used to prevent the LSP from inference attacks and protect the location privacy of the group of users. The circular secure areas were used to deconstruct the continuous queries into uncorrelated snapshot queries, and conceal part of the users' trajectories to prevent the LSP from spatial-temporal correlation attacks. Then, the aggregate nearest neighbor query algorithm, based on strategy optimization, was used to further reduce the overall delay of the service. Finally, the simulation results also verified that our proposed scheme could ensure a high privacy security, and that the response speed and communication overhead of the service were superior to other peer algorithms, both, in the snapshot and the continuous queries. In addition, although this article considered the

user's mobility, it did not consider that often the user would be moving along the road network. The next step could be considered to extend the Euclidean space to the road network space.

Author Contributions: L.Z. proposed the main schemes of PCANNQ, while C.J. designed the implementation process of the algorithm. H.-p.H. and X.F. conducted the simulation experiment and verified the theory. R.-c.W. served as advisor to the above authors and gave suggestions on simulations and performance evaluation. The manuscript was a combined effort from all five authors. All authors have read and approved the final manuscript.

Funding: The subject is sponsored by the National Natural Science Foundation of China (61872194, 61402241, 61572260, 61572261); the Jiangsu Natural Science Foundation for Excellent Young Scholar (BK20160089); Jiangsu Provincial Research Scheme of Natural Science for Higher Education Institution(17KJB520026); Research of Natural Science of NJUPT(NY217050); and Jiangsu Government Scholarship for Overseas Studies.

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

References

- Schlegel, R.; Chow, C.Y.; Huang, Q.; Wong, D.S. Privacy-Preserving Location Sharing Services for Social Networks. *IEEE Trans. Serv. Comput.* **2016**, *99*, 1. [[CrossRef](#)]
- Son, J.; Kim, D.; Bhuiyan, M.Z.; Tashakkori, R.; Seo, J.; Lee, D.H. Privacy Enhanced Location Sharing for Mobile Online Social Networks. *IEEE Trans. Sustain. Comput.* **2018**, *11*, 439–443. [[CrossRef](#)]
- Zhu, S.Z.; Huang, L.; Zhou, C.L.; Ma, Y. A Privacy-Preserving Method Based on POIs Distribution Using Cloaking Region for K Nearest Neighbor Query. *Acta Electron. Sin.* **2016**, *44*, 2423–2431.
- Pournajaf, L.; Tahmasebian, F.; Xiong, L. Privacy Preserving Reverse k-Nearest Neighbor Queries. In Proceedings of the 2018 19th IEEE International Conference on Mobile Data Management (MDM), Aalborg, Denmark, 26–28 June 2018; pp. 177–186.
- Wang, H.Y.; Cao, R.J.; Liu, Y.S.; Liu, W.Y.; Jin, S.F. Dynamic Friend Discovery Based on Privacy for Proximity-based Mobile Social Networking. *J. Chin. Comput. Syst.* **2016**, *37*, 104–109.
- Luo, E.T.; Wang, G.J.; Chen, S.H.; Pinal, K.B. Privacy Preserving Friend Discovery Cross Domain Scheme Using Re-encryption in Mobile Social Networks. *J. Commun.* **2017**, *38*, 81–93.
- Tan, R.; Si, W.; Sheng, J. A Privacy-Sensitive Approach for Group Convergence in Location-Based Services. In Proceedings of the IEEE International Conference on Cyberworlds, Chongqing, China, 28–30 September 2016; pp. 1–8.
- Sun, Y.M.; Chen, M.; Hu, L.; Qian, Y.F.; Mohammad, M.H. ASA: Against Statistical Attacks for Privacy-Aware Users in Location Based Service. *Future Gener. Comput. Syst.* **2017**, *70*, 48–58. [[CrossRef](#)]
- Gedik, B.; Liu, L. Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms. *IEEE Trans. Mob. Comput.* **2007**, *7*, 1–18. [[CrossRef](#)]
- Schlegel, R.; Chow, C.Y.; Huang, Q.; Wong, D.S. User-Defined Privacy Grid System for Continuous Location-Based Services. *IEEE Trans. Mob. Comput.* **2015**, *14*, 2158–2172. [[CrossRef](#)]
- Zhang, S.; Wang, G.; Liu, Q.; Abawajy, J.H. A trajectory Privacy-Preserving Scheme Based on Query Exchange in Mobile Social Networks. *Soft Comput.* **2018**, *22*, 6121–6133. [[CrossRef](#)]
- Ni, W.W.; Ma, Z.X.; Chen, X. Safe Region Scheme for Privacy-Preserving Continuous Nearest Neighbor Query on Road Networks. *Chin. J. Comput.* **2016**, *39*, 628–642.
- Yiu, M.L.; Jensen, C.S.; Huang, X.; Lu, H. SpaceTwist: Managing the Trade-Offs Among Location Privacy, Query Performance, and Query Accuracy in Mobile Services. In Proceedings of the International Conference on Data Engineering, Cancún, México, 7–12 April 2008; pp. 266–375.
- Chow, C.Y.; Mokbel, M.F.; Aref, W.G. Casper*: Query processing for location services without compromising privacy. *ACM Trans. Database Syst.* **2009**, *34*, 1–48. [[CrossRef](#)]
- Chow, C.Y.; Mokbel, M.F.; Liu, X. Spatial Cloaking for Anonymous Location-Based Services in Mobile Peer-to-Peer Environments. *Geoinformatica* **2011**, *15*, 351–380. [[CrossRef](#)]
- Khoshgozaran, A.; Shahabi, C. Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy. In Proceedings of the International Conference on Advances in Spatial and Temporal Databases, Boston, MA, USA, 16–18 July 2007; pp. 239–257.

17. Zhang, S.B.; Liu, Q.; Wang, G.J. A Caching-Based Privacy-Preserving Scheme for Continuous Location-Based Services. In Proceedings of the Security, Privacy and Anonymity in Computation, Communication and Storage, Zhangjiajie, China, 16–18 November 2016; pp. 73–82.
18. Zhang, S.B.; Li, X.; Tan, Z.Y.; Peng, T.; Wang, G.J. A Caching and Spatial K-Anonymity Driven Privacy Enhancement Scheme in Continuous Location-Based Services. *Future Gener. Comput. Syst.* **2019**, *94*, 40–50. [[CrossRef](#)]
19. Hwang, R.H.; Hsueh, Y.L.; Chung, H.W. A Novel Time-Obfuscated Algorithm for Trajectory Privacy Protection. *IEEE Trans. Serv. Comput.* **2014**, *7*, 126–139. [[CrossRef](#)]
20. Peng, T.; Liu, Q.; Meng, D.C.; Wang, G.J. Collaborative Trajectory Privacy Preserving Scheme in Location-Based Services. *Inf. Sci.* **2017**, *387*, 165–179. [[CrossRef](#)]
21. Papadias, D.; Tao, Y.; Mouratidis, K.; Hui, C.K. Aggregate Nearest Neighbor Queries in Spatial Databases. *ACM Trans. Database Syst.* **2005**, *30*, 529–576. [[CrossRef](#)]
22. Yiu, M.L.; Mamoulis, N.; Papadias, D. Aggregate nearest Neighbor Queries in Road Networks. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 820–833. [[CrossRef](#)]
23. Luo, Y.; Chen, H.; Furuse, K.; Ohbo, N. Efficient Methods in Finding Aggregate Nearest Neighbor by Projection-Based Filtering. In Proceedings of the International Conference on Computational Science & Its Applications, Kuala Lumpur, Malaysia, 26–29 August 2007; pp. 821–833.
24. Hashem, T.; Kulik, L.; Zhang, R. Privacy preserving group nearest neighbor queries. In Proceedings of the International Conference on Extending Database Technology, Lausanne, Switzerland, 22 March 2010; pp. 489–500.
25. Elmongui, H.G.; Mokbel, M.F.; Aref, W.G. Continuous aggregate nearest neighbor queries. *Geoinformatica* **2013**, *17*, 63–95. [[CrossRef](#)]
26. Sun, W.W.; Chen, C.N.; Zhu, L.; Gao, Y.J.; Jing, Y.N.; Li, Q. On Efficient Aggregate Nearest Neighbor Query Processing in Road Networks. *J. Comput. Sci. Technol.* **2015**, *30*, 781–798. [[CrossRef](#)]
27. Yao, B.; Chen, Z.; Gao, X.; Shang, S.; Ma, S.; Guo, M. Flexible Aggregate Nearest Neighbor Queries in Road Networks. In Proceedings of the IEEE 34th International Conference on Data Engineering (ICDE), Paris, France, 16–19 April 2018.
28. Guttman, A. R Trees: A Dynamic Index Structure for Spatial Searching. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Boston, MA, USA, 18–21 June 1984; pp. 47–57.
29. Sharifzadeh, M.; Shahabi, C. VoR-Tree: R-trees with Voronoi Diagrams for Efficient Processing of Spatial Nearest Neighbor Queries. In Proceedings of the 36th International Conference on Very Large Data Bases, Singapore, 13–17 September 2010.
30. Lin, Q.; Zhang, Y.; Zhang, W.; Lin, X. AVR-Tree: Speeding Up the NN and ANN Queries on Location Data. In *Database Systems for Advanced Applications*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 116–130.
31. Brinkhoff, T. A Framework for Generating Network-Based Moving Objects. *Geoinformatica* **2002**, *6*, 153–180. [[CrossRef](#)]

