Scientific Research Publishing

Journal of Computer and Communications, 2024, 12, 53-70
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Trajectory Privacy Protection Method to Resist Long-Term Observation Attacks

Qixin Zhan

School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China
Email: 1719713004@qq.com

How to cite this paper: Zhan, Q.X. (2024) A Trajectory Privacy Protection Method to Resist Long-Term Observation Attacks. *Journal of Computer and Communications*, **12**, 53-70.
https://doi.org/10.4236/jcc.2024.125004

Received: April 23, 2024
Accepted: May 19, 2024
Published: May 22, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/

Open Access

Abstract

Users face the threat of trajectory privacy leakage when using location-based service applications, especially when their behavior is collected and stored for a long period of time. This accumulated information is exploited by opponents, greatly increasing the risk of trajectory privacy leakage. This attack method is called a long-term observation attack. On the premise of ensuring lower time overhead and higher cache contribution rate, the existing methods cannot utilize cache to answer subsequent queries while also resisting long-term observation attacks. So this article proposes a trajectory privacy protection method to resist long-term observation attacks. This method combines caching technology and improves the existing differential privacy mechanism, while incorporating randomization factors that are difficult for attackers to recognize after long-term observation to enhance privacy. Search for locations in the cache of both the mobile client and edge server that can replace the user's actual location. If there are replacement users in the cache, the query results can be obtained more quickly. Simultaneously obfuscating the spatiotemporal correlation of actual trajectories by generating confusion regions. If it does not exist, the obfuscated location generation method that resists long-term observation attacks is executed to generate the real anonymous area and send it to the service provider. The above steps can comprehensively protect the user's trajectory privacy. The experimental results show that this method can protect user trajectories from long-term observation attacks while ensuring low time overhead and a high cache contribution rate.

Keywords

Location Privacy, Long-Term Observation Attacks, K-Anonymity, Location Caching

1. Introduction

With the rapid development of mobile intelligent devices and positioning tech-

nology, location-based service applications (LBS) have surged [1]. At present, it can be said that people's lives cannot do without LBS. We often use LBS to query local weather and navigation, and so on [2]. Moreover, the implementation of LBS will depend on publicly available trajectory data [3]. However, when publishing trajectory data, it is possible to be attacked by attackers, leading to the leakage of user trajectory information. The leakage of trajectory information may lead to the exposure of more personal privacy information, so trajectory privacy has become one of the most important privacy concerns for people.

Typical trajectory privacy protection technologies mainly include: 1) Based on anonymous zone technology, this technology protects users' location information by generalizing location data [4]. 2) Based on encryption technology, this technology uses cryptographic techniques to encrypt user location information, ensuring that personal data is encrypted and protected during transmission [5]. 3) Based on differential privacy technology, this technology protects data privacy by adding random noise to the data, preventing attackers from accessing the data. Adjusting the privacy budget can achieve this noise, which can prevent attackers from inferring specific individual information [6]. Although the above track privacy technology can protect the privacy of users to a certain extent, it also has different disadvantages: 1) the size of the anonymous area based on anonymous area technology must be reasonably selected. If the anonymous area is too small, the attacker can determine the user's location through statistical analysis and inference; If the anonymous area is too large, the location accuracy of users will be significantly reduced, affecting the accuracy of location services; 2) The technology based on encryption needs to use encryption algorithms that are more complex and consume more computing resources to protect the security of data, which greatly increases the complexity and cost of the system; 3) The technology based on differential privacy needs to strictly control the privacy budget. Excessive noise addition will reduce the accuracy of location data and affect the quality of location service [7].

If the attacker observes the user for a long period of time, the risk that the user's trajectory will be reconstructed will increase. This attack is called a long-term observation attack in this paper. The existing trajectory privacy protection methods can't bring low time overhead to the user and better meet the user's quality of service requirements, but also resist the long-term observation attack. Therefore, this paper proposes a trajectory privacy protection method to resist the long-term observation attack. This method provides a novel solution for trajectory privacy protection, and provides good protection for users' trajectory privacy information.

The contributions of this paper mainly include the following aspects:

- This paper proposes a trajectory privacy protection method to resist long-term observation attacks. This method has the ability to resist long-term observation attacks while ensuring low time overhead and better meeting the quality of service requirements of users.
- In this paper, the cache technology is combined to facilitate users' next query,

and the existing differential privacy mechanism is improved. In the process of algorithm design, the randomization factor which is difficult to recognize by the attacker after long-term observation is added to improve privacy.

- In this paper, the various situations encountered by users are analyzed and discussed in detail. Using a confusion location generation algorithm to resist long-term observation attacks and combining confusion anonymous regions to confuse the spatiotemporal correlation of user actual trajectories.
- This paper compares the proposed scheme with the existing scheme on the real data set. The experimental results show that the overall effect of this scheme is better than the existing scheme.

The rest of this paper is organized as follows. Section 2 introduces the related work of this paper. Section 3 introduces the system model and the specific implementation of this method. Section 4 conducts experiments and analyzes the results. Section 5 summarizes this article.

2. Related Work

In recent years, scholars have proposed some solutions for the research direction of this article. Sun *et al.* [8] introduced Long Term Statistical Attack (LSA) and proposed the MNAME method to address this attack, which means that users store multiple usernames, and when using LBS services, users select a name from the set of usernames as the current username and send it to the LBS server. At the same time, the SNAME method is also proposed, which means that the anonymous server will change the query and have each username changed to the same username by the anonymous server before sending it to the LBS server.

Although the communication between users and LBS servers is encrypted, attackers can still link real users by reconstructing trajectories and establish connections between locations and users through semantic frequency differences. In response to this issue, Wang *et al.* [9] introduced comment based location-related attacks (RLCA) and semantic based long-term statistical attacks (SLSA) based on reference [8]. At the same time, an index is proposed for RLCA to measure the correlation between users and trajectories. Exchange reviews that resist RLCA by suppressing the number of positions on each reconstructed trajectory below a threshold. In order to resist SLSA, a metric is proposed to measure the location indistinguishability of long-term semantic frequency difference. This method can select reviews resisting SLSA to exchange by allowing two reviews whose indistinguishability is below the probability difference after the exchange to be exchanged. Protect trajectory privacy through the above two methods.

Due to the spatiotemporal correlation of trajectories, attackers can easily use historical trajectories and background knowledge to predict the future location of target users through long-term observation. Qiu *et al.* [10] referred to this type of inference attack as trajectory prediction attack. And generate an indistinguishable perturbation position that is robust to predictive attacks. When the user's true position is submitted to an untrusted server, the perturbation posi-

tion can be used instead. Zhao *et al.* [11] proposed a lightweight WiFi localization privacy algorithm and implemented a WiFi localization system, using an indoor localization algorithm based on WiFi fingerprints and combined with the proposed location privacy protection method. This algorithm can resist spatiotemporal attacks without relying on trusted third parties. But WiFi positioning is very susceptible to interference, which in turn reduces accuracy. To address this issue, Tang *et al.* [12] predicted the user's next action and generated all initial and future false positions, ensuring that attackers could not reconstruct the user's true trajectory through a series of false positions. However, this scheme cannot protect the user's location privacy when querying outside of the predicted location. Song *et al.* [13] obfuscated attackers by submitting anonymous regions generated by combining fake locations and user real locations to LBS service providers. In this scheme, the distance between the user's real location and the center of the constructed anonymous area remains unchanged. If the attacker collects anonymous areas through long-term observation attacks, there is a risk of leakage of the user's true location. In response to this issue, Siddiqie *et al.* [14] used Gaussian probability distribution to add randomization at this distance to prevent attackers from conducting long-term observation attacks.

However, these solutions cannot utilize the response of LBS servers, increasing the time cost for users to perform the same query. To solve this problem, Zhang *et al.* [15] proposed a cache based double K anonymity (CDKA) location privacy protection scheme in edge computing environment. This scheme can cache such responses to answer subsequent queries. Zhu *et al.* [16] proposed a Location Privacy Protection Method (LPPM) based on a Variable Order Markov Model, which achieves scrambling of query publication timing through spatial anonymity, location prediction, and the addition of cache, providing protection for the privacy of user trajectories. Huang *et al.* [17] proposed a method called TPPCD, which utilizes cache and virtual location to protect trajectories, an active cache update mechanism based on data popularity, and a passive cache update mechanism based on virtual location. However, such methods are not doing very well in resisting long-term observation attacks, and comprehensively protecting user trajectory privacy remains a major challenge.

In summary, although scholars have proposed many trajectory privacy protection methods, there is still no trajectory privacy protection method that can enable users to efficiently perform location queries while resisting long-term observation attacks. Therefore, this article combines caching technology and current mainstream trajectory privacy protection technologies to propose a trajectory privacy protection method to resist long-term observation attacks.

3. System Models and Methods

3.1. System Models

As shown in **Figure 1**, this article adopts a "Mobile Client-Edge Server-LBS Server" framework and proposes a trajectory privacy protection method to resist

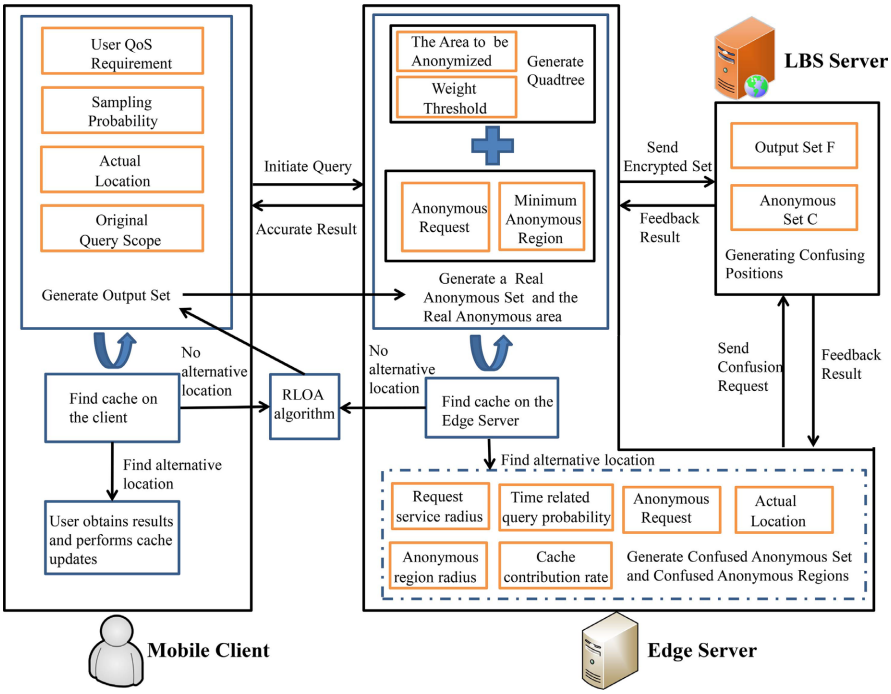

Figure 1. System model diagram.

long-term observation attacks, which consists of the following four algorithms: 1) Choose Replacement Location Algorithm, CRLA; 2) Confusion Location Generation Algorithm to Resist Long Term Observation Attacks, RLOA; 3) Building Confused Anonymous Regions Algorithm, BCAR; 4) Update Cache Algorithm, UCA.

The scheme of this paper is as follows:

1) Users obtain query results from the Mobile Client

When users use LBS services, they obtain their true location through GPS, and then search the user's cache to see if there is a location that can replace their true location. The query content obtained by replacing the location can meet the user's query needs just like the query content obtained by using the real location. If such a replacement position is found, the user can obtain the query results from the user end and then execute the cache update algorithm.

2) Users obtain query results from the Edge Server

When the client cache cannot find a replacement location that meets the requirements, the user needs to submit a query to the edge server to find whether there is a location that can replace the real location on the edge server. If such a replacement location is found, the edge server first sends the query results obtained from the replacement location to the user for use, and the user can execute the cache update algorithm to obtain the query results. Then the edge server, in order to prevent track privacy disclosure, will construct a confusion anonymous set to the LBS server, and then will receive the results from the LBS server, and the edge server can execute the cache update algorithm.

3) Users obtain query results from the LBS Server

If the user cannot find a replacement location on the client and the edge server, it will take a confused location generation algorithm to resist the long-term observation attack. By generating a real anonymous set different from the confused anonymous set, it is sent to the LBS server, and then the LBS server returns the results to the edge server after processing, and the edge server executes the cache update algorithm. The edge server then sends the results to the mobile client, who can execute the cache update algorithm.

3.2. Choose Replacement Location

User A obtains its real location through GPS. When the user initiates a query request at d_r, if there is a user B near the user, its real location is very close to User A, and the content queried by User A just meets the needs of User A, then the real location of User B will become the replacement location of User A. The effective service rate of each location in the cache for user requests is the cache effective service rate. The higher the cache effective service rate, the better the query requirements obtained by replacing the location can meet user requirements.

As shown in **Figure 2**, Δd denotes the position offset between the user's actual position and the substituted position and R denotes the request service radius. Circular area A represents the range of requests made by the user at its exact location, and the alternative location is area *B*. The number of service requests in area A is n_A and area B is n_B, The request range overlap rate is defined as follows:

$$\Delta n = \frac{n_{A \cap B}}{n_A}. \tag{1}$$

The cached data mentioned in this article has a lifespan, so it is necessary to update these data in real-time. The higher the probability of data being queried, the higher the priority for updating. Freshness can be used to evaluate the priority of updating these data, which is defined as follows:

$$f = \sqrt{1 - \frac{\tau^2}{T^2}}, \quad \tau \le T \tag{2}$$

where T represents the lifetime and τ represents the time the data has been stored, the average data freshness of the anonymous area is defined as follows:

$$\overline{f} = \frac{1}{k} \sum_{i=1}^{k} f_i. \tag{3}$$

Use Δs to indicate the similarity of the requested content, $\Delta s = 0$ to indicate that the query content of the user's replacement location is inconsistent with the query content obtained by the user using the real location, and $\Delta s = 1$ to indicate that the content is consistent. In order to ensure that the selected alternative location can effectively implement user requests, it is necessary to select the location with a high cache service ratio as the alternative location as far as possible. The cache effective service rate is defined as follows:

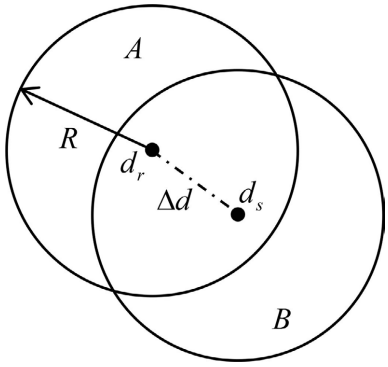

Figure 2. Cache effective service rate.

$$\alpha = \frac{\Delta s \cdot \Delta n \cdot f}{1 + \Delta d}. \tag{4}$$

The replacement location selection algorithm can perform the operation of finding the replacement location on the client and the edge server. For Δd, the acceptable distance for users is defined as $d_{\max}\left(0 \le d_{\max} < 2R\right)$. If there are special circumstances, such as users having strict query requirements, it can be reduced, and vice versa, it can be increased. We use $\alpha' = 20\%$ as the threshold to measure the existence of alternative positions, which is obtained through historical query data statistics and calculations. When $\Delta d = 0.4$ km, $\Delta n = 60\%$ and $\Delta s = 1$, $a \approx 21.43$, the query content obtained from the replacement position found in the cache can well meet the user's query requirements. So when a is not less than the threshold, this position can be used as a replacement for the user's actual position. If there are special circumstances, such as strict user requirements and the desire to obtain more accurate query content, a can be appropriately increased at this time.

3.3. RLOA Algorithm

This section mainly introduces the RLOA algorithm, which aims to generate an output set and a real anonymous set through a series of operations when users initiate query requests at each real location on the real trajectory, and perform differential obfuscation to generate obfuscation positions. This obfuscation location has the characteristics of meeting the quality of service (QoS) requirements of users and cannot be recognized by long-term observation attackers [18]. Firstly, in order to meet the QoS requirements of users, the algorithm designs a QoS calculation formula. In order to meet the user requirements as much as possible, we change the original query range r_0 to the submitted query range r_s. The larger the scope of the query submitted, the more data that meets the user's needs and has nothing to do with the user's needs. Therefore, the formula is as follows:

$$\mathrm{QoS}\left(l', r_s\right) = \varphi\left[\frac{S\left(l', r_s\right) \cap S\left(l, r_o\right)}{S\left(l, r_o\right)} - \omega \cdot \frac{S\left(l', r_s\right) - S\left(l, r_o\right)}{S\left(l', r_s\right)}\right]. \tag{5}$$

where $S(l', r_s)$ denotes the area covering all the obtained service data, $S(l, r_o)$ is the area covering all the service data the user needs. ω represents the proportion of redundant data when calculating QoS. The QoS formula can be modified based on specific scenarios, primarily determined by the coefficient represented by φ. This coefficient φ is set by the user and can be adjusted according to their needs. Using this method can make the formula of Qos more diversified, and to some extent, it can prevent attackers from carrying out long-term observation attacks. In order to further resist this attack, the algorithm generates a candidate set C that meets the user's QoS requirements through formula (5), and uses a sampling method to select some confusion positions from this candidate set. The sampling probability can be determined by the user themselves. Through the above steps, this article will form a position output set on the mobile client.

The second step of the algorithm is to determine an anonymous set. First, the region is divided according to the relative weight threshold μ, and when the weight in the region exceeds μ, it is divided. The first layer is represented as the root of the quadtree, and then the region is divided recursively until the anonymous weight in the leaf node region does not exceed μ. After the region is divided, the leaf node represents the smallest divided region. This paper uses this improved quadtree to store location information. Edge server locates the user location to the surrounding area and finds all users who satisfy location similarity of query destinations (LS-QD) [19] to construct an anonymous set C. The destination coordinates that the user A needs to query are (x_i, y_i) and B are (x_j, y_j) , The LS-QD calculation formula for users A and B is as follows:

$$L(u_i, u_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{6}$$

If the anonymity set does not meet the requirement of k-anonymity, this scheme will continue to partition the quadtree based on weights, and then continue to calculate the location similarity of the destination, adding users who meet the requirements to the anonymity set. And so on, until the final anonymous set can meet the k-anonymity requirement. This article also proposes adding random noise to the constructed real anonymous set. This method can improve the ability to resist long-term observation attacks.

The third step of the algorithm uses an improved exponential mechanism to achieve differential privacy [20]. The Euclidean distance $d_{euc}(l, l')$ between the confused position l' and the actual position l is used to evaluate the advantages and disadvantages of the output confused position. The scoring function is defined as follows:

$$q(l, l') = -d_{euc}(l, l'). \tag{7}$$

The sensitivity of the rating function is:

$$\Delta q = \max_{l' \in F; l_i, l_j \in C_{DP}} \left\| \left[-d_{euc}(l_i, l') + d_{euc}(l_j, l') \right] \right\|_1. \tag{8}$$

where l' denotes one of the possible positions in the output set, l_i, l_j denotes

the user's position points on the real anonymous set. The algorithm selects and outputs the probability of confusion position directly proportional to $\exp\left(\frac{\varepsilon q(l,l')}{2\Delta q}\right)$ from output set.

In order to resist long-term observation attacks, the RLOA algorithm adopts a method of user-defined probability coefficients for differential obfuscation, and adds a function at the user end to remind each user to input the correct randomization coefficients to protect their personal privacy. When users input these self-defined coefficients, they can continue to query. So the index mechanism adopted in this article is as follows:

$$A_E\left(z|l\right)=\lambda_1\cdot A_E\left(z|l_1\right)+\lambda_2\cdot A_E\left(z|l_2\right)+\cdots+\lambda_k\cdot A_E\left(z|l_k\right). \tag{9}$$

where $l_i\left(i\in[1,k]\right)$ denotes the user location point of anonymous set, $A_E\left(z|l\right)$ denotes using an exponential mechanism at position l. $\lambda_i\left(i\in[1,k]\right)$ denotes user-defined coefficient of exponential mechanism. Through the above steps, the proposed algorithm randomly selects a location from the real anonymous set to generate confusing location. This confusing location cannot be inferred by the attacker while meeting the QoS requirements of users.

3.4. Building Confused Anonymous Regions

As shown in **Figure 3**, RLOA algorithm is used to generate 1000 confusion locations. By analyzing the distribution of all confused positions, it can be determined that the actual position of the user at a certain point in time is one of the three small circle center positions. The big circle is the anonymous area generated anonymously by *k*. In the scenario where the attacker has experienced a long-term observation attack for a long time, when the time passes from t_1 to t_3, the user's location passes through A, Band C. It can be seen from the figure that the experiment k is set to 3, that is, except the big circle center is the real location of the user, the other two small circle centers are confused locations that are difficult for the attacker to distinguish under the long-term observation attack.

However, because users submit requests in chronological order when using continuous LBS, attackers use this to infer the user's approximate trajectory. Then, attackers can collect the user's location information through long-term observation attacks. After a long period of time, attackers can obtain the user's true trajectory. As shown in **Figure 4**, the real anonymous area is represented by a solid circle, while the confused anonymous area is represented by a dashed circle. There are replacement positions near points A and B that meet the user's requirements, so the BCAR algorithm confuses time t_1,t_2,t_3 with t_2',t_1',t_3' by constructing a confusing anonymous region to confuse the spatiotemporal correlation of the user's actual trajectory. If a replacement position that meets the requirements cannot be found at point C, the real anonymous area will be generated using the RLOA algorithm.

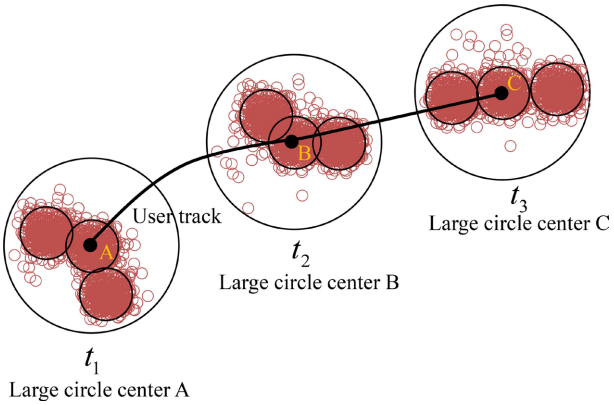

Figure 3. Unconstructed confused anonymous regions.

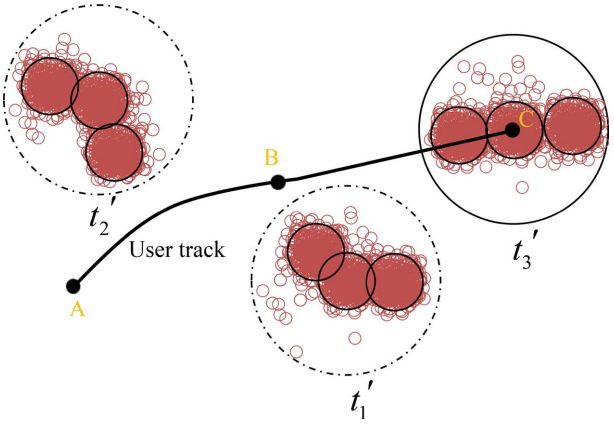

Figure 4. Constructing confused anonymous regions.

When constructing anonymous regions using obfuscated locations, the likelihood of the obfuscated location being a future query location is directly proportional to the time related query probability of this obfuscated location. Therefore, the higher the time related query probability of the obfuscated location, the higher its contribution rate to the cache of the user or edge server [21]. The definition of cache contribution rate is as follows:

$$\delta = q \cdot g, \text{ where } g = 0 \text{ or } 1. \tag{10}$$

In the formula, q represents the probability of time related queries. When $g = 1$, it indicates that the location has not been cached yet. Conversely, it indicates that it has been cached. Therefore, the contribution of k obfuscated positions in the anonymous region to the cache is defined as cache contribution rate as follows:

$$G = \left(\sum_{i=1}^{k} \delta_i\right) \cdot \left(1 - \overline{f}\right). \tag{11}$$

The specific steps of the BCAR algorithm are as follows:

1) Firstly, construct a circular area with the user's real location as the center and the radius as the request service radius, and select a time related query probability high in this area as the set of obfuscated locations to ensure that

more useful data for the user can be cached.

2) Next, the edge server exports m ($m \le C_N^{2K}$) sets $list_i\left(i=1,2,\cdots,m\right)$ from N locations with high probability of time related queries. The number of confusing locations in these sets is uniformly 2*k*. The cache contribution rate of each set is calculated and compared, and the one with the highest value is selected.

3) Then, the edge server continues to export m' sets $list_i\left(i=1,2,\cdots,m'\right)$ from the set obtained in the second step, where the number of obfuscated positions in these sets is uniformly k, and the product of the distances between position pairs is $\prod_{j\neq l} Dis\left(d_j,d_l\right)$. Calculate the product of each set for comparison, and select the largest anonymous set to ensure better obfuscation of trajectories, preventing attackers from reconstructing trajectories and causing trajectory privacy leakage.

4) Finally, by obtaining *k* scattered confusion positions through the third step, a confusion anonymous region is generated. And add random noise to the anonymous set obtained in the third step to generate a confusing anonymous set. The noise is $\Upsilon_1 \cdot \text{Laplacenoise}\left(\varepsilon\right)+\cdots+\Upsilon_K \cdot \text{Laplacenoise}\left(\varepsilon\right)$. Υ denotes customizes probability coefficients for users and adds a reminder on the user end that each user must enter the correct randomization coefficients to protect their personal privacy. Users can only continue querying when they enter these customized coefficients. This random coefficient will vary according to the personalized needs of users, and this randomness greatly confuses long-term observations of attackers reconstructing user trajectories.

3.5. Update Cache

The client and edge server cache the query results of users, some of which are used more and some are used less. We need to update this data, update some of the less useful data, and use the indicators of data freshness f and location utilization u_i to evaluate whether it is updated. The specific calculation of data freshness is shown in formula (2), and the definition of location utilization is as follows:

$$u_i = \frac{\text{The number of times cache location i has been used}}{\text{Cache the number of times all locations have been used}}. \tag{12}$$

The data with lower freshness and higher location utilization needs to be updated first. The data with lower freshness is about to expire, and the more users send query requests at this location, the higher the location utilization. Therefore, these data are often quickly eliminated [22]. To improve cache contribution, it is necessary to focus on updating these data. The cache update indicators corresponding to the location are defined as follows:

$$\varphi = u_i \cdot \left(1-f\right). \tag{13}$$

When users and edge servers receive the results and result sets of LBS requests, it is necessary to perform cache updates to facilitate the next query. Users can still perform cache updates without communicating with LBS service providers and edge servers. At this time, the edge server can update cache by in-

itiating bogus requests. This process can also serve to confuse the user's actual trajectory. Its algorithm approach is similar to the construction of obfuscated anonymous area algorithm (BCAR algorithm), and the anonymous set generated by this algorithm can be named cache anonymous set. Just like the operation after generating the obfuscated anonymous set, send the cached anonymous set to the LBS service provider, and the edge server will receive new query results. After sending the results to the user end, the edge server will execute its own cache update algorithm, and the mobile client will also cache the query results from the edge server. The following is the specific process of cache update:

1) Firstly, calculate and select N locations with higher cache update metrics φ as candidate update locations.

2) Next, derive $m'\left(m' \le C_{2K}^{K}\right)$ sets $list_i\left(i=1,2\cdots,m'\right)$ from N candidate positions, and calculate the area formed by k positions in each set.

3) Then select the anonymous sets that are smaller than the actual anonymous area and place them in a set. Find and select the set with the largest area.

4) Like the BCAR algorithm, in order to resist long-term observation attacks, random noise $\xi_1 \cdot \text{Laplacenoise}(\varepsilon)+\cdots+\xi_K \cdot \text{Laplacenoise}(\varepsilon)$ is added to the anonymity set obtained in the third step to obtain the cached anonymity set, where ξ is the randomization probability coefficient set by the edge server based on the user's historical query situation. This randomness is also used to confuse long-term observation attackers in obtaining accurate user information. Finally, the final cached anonymity region c is constructed based on the distribution of k positions in the anonymity set.

4. Experiment

4.1. Experiment Environment

The experiment is implemented on Windows 10 operating system using python programming language and running hardware environment: 2.6 GHz Intel (R) Core (TM) i7-6700HQ CPU, 16GB RAM. This chapter's experiment used two datasets, namely the Gowalla dataset and the Geolife dataset [23]. The experiments are compared with CDKA [15], LPPM [16] and TPPCD [17]. The reason is that these schemes adopt similar technologies as this article, highlighting the advantages of the proposed schemes by comparing similar schemes.

4.2. Experiment Result

The solution proposed in this article is to satisfy user query requests through caching, while generating obfuscated anonymous regions that can interfere with attackers rebuilding user trajectories. Therefore, the degree of obfuscation can be defined by the following formula:

$$\rho = \frac{\text{The number of false anonymous regions}}{\text{The number of anonymous regions}}. \tag{14}$$

As shown in **Figure 5** and **Figure 6**, we can draw three conclusions. The first one is not difficult to observe that as the service radius R increases, the degree of

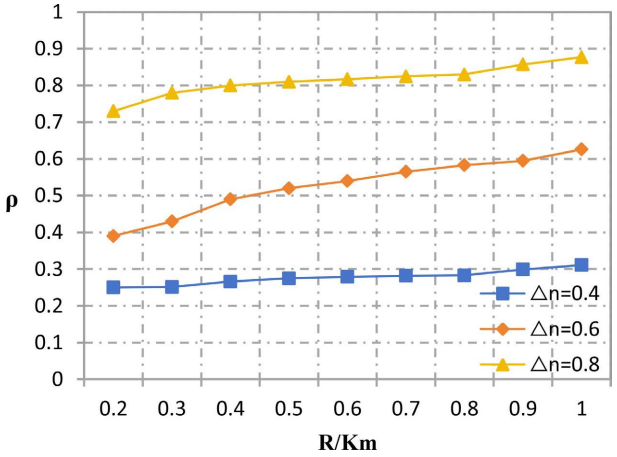

Figure 5. Confusion level analysis (Gowalla).

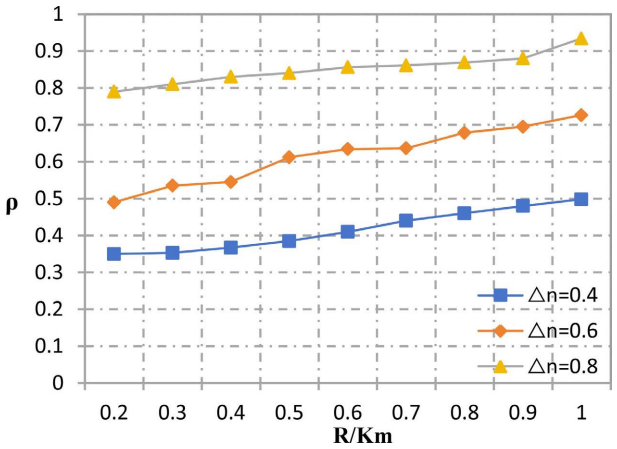

Figure 6. Confusion level analysis (Geolife).

confusion also increases. The reason for this result is that as the service radius increases, more cached data is used for user queries. At this time, more confusion anonymous regions will be constructed. According to formula (14), it can be inferred that the degree of confusion will increase. The second one is to conduct experiments with settings of 0.4, 0.6, and 0.8. From the graph, we can see that as the service radius remains unchanged, the degree of confusion will also increase. The third point is that when Δn is taken as 0.8 and R is taken as 0.5, the degree of confusion in our scheme can reach 0.8 or above, indicating that our scheme can effectively confuse the actual trajectory of users and achieve the effect of protecting the trajectory.

As shown in Figure 7 and Figure 8, the degree of confusion between our scheme and the other two schemes was compared on two datasets. The degree of confusion of all algorithms increased with the increase of k value, which is consistent with the analysis results in Figure 5 and Figure 6. LPPM uses predictive models to cache more possible information in advance to cope with time chaos. The effect is better than TPPCD and CDKA. Our scheme takes into account the time related query probability when generating false regions. We choose

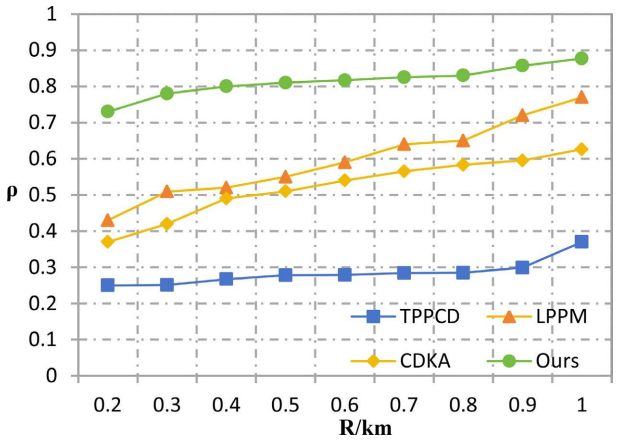

Figure 7. Comparison of confusion level (Gowalla).

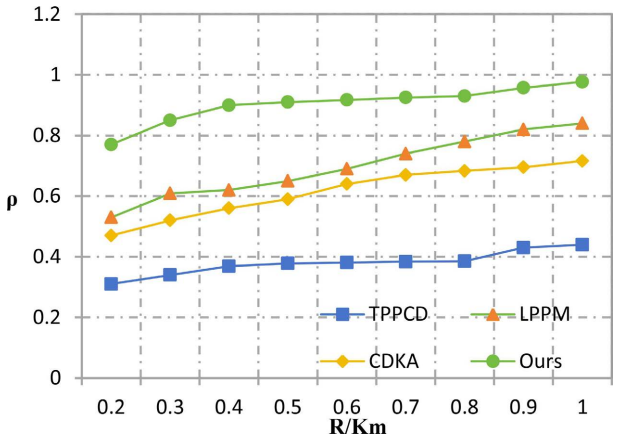

Figure 8. Comparison of confusion level (Geolife).

locations that are more dispersed and have a higher cache contribution rate to construct a confusing anonymous set. This method can better confuse trajectories and improve user query efficiency. Experimental results show that it is superior to other algorithms.

$$\theta = \angle\left(\text{Current request location}, \text{Last request location}\right) \tag{15}$$

Formula (15) can be used to describe the changes in trajectories. In this paper, 60 points were randomly selected from the trajectories generated on two datasets, and their trajectory changes were presented in a figure as shown in **Figure 9**. From the figure, we can conclude that the user's real trajectory follows much more regular patterns than the trajectory received by LBS. The acceptance of the trajectory by LBS proves that the proposed scheme has the ability to interfere with attackers in reconstructing trajectories.

As shown in **Figure 10**, three trajectories were generated through experiments, and ten points were randomly selected for each trajectory. The request range overlap rate of the ten points on the three trajectories was calculated and plotted in a figure format. From the figure, we can see that the range overlap rate of the random points on the three trajectories is all higher than 0.6. This situation

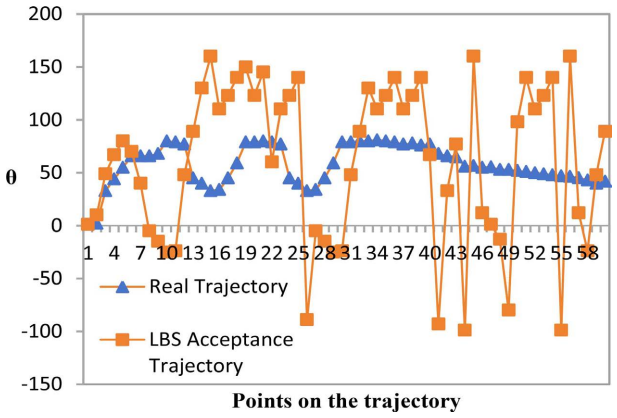

Figure 9. Shifting angle.

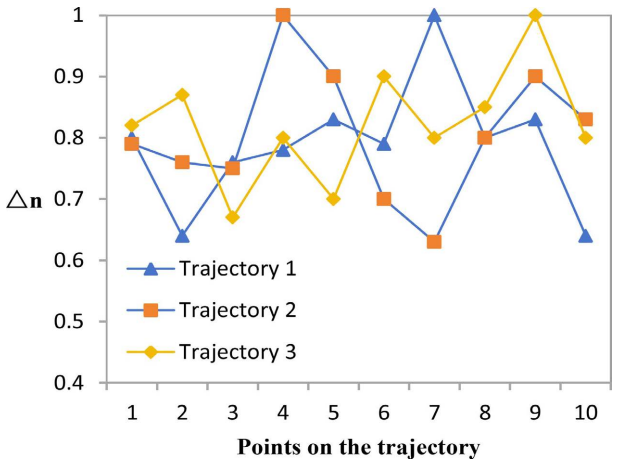

Figure 10. Request range coincidence rate.

indicates that this scheme can maintain service quality at a good level while also protecting user trajectory privacy.

As shown in Figure 11, we conducted experiments on the changes in cache contribution rates of the four schemes by changing the parameter *k*. The figure shows that as the value of *k* increases, the cache contribution rates of all four methods improve. It is not difficult to see that LPPM and our proposed scheme both have high cache contribution rates, which can efficiently answer future query requests from users.

Figure 12 shows the impact of k on execution time under different schemes. Among them, TPPCD has the highest time cost, mainly in the construction of information matrices and the collection of hotspot data, while LPPM uses a variable order Markov model to predict user motion trends, which requires time to predict. CDKA and the proposed solution in this article do not require too much time to wait for the previous step of operation, and adopt a "Mobile Client-Edge Server-LBS Server" framework. Even in the case of a large number of users operating simultaneously, these two solutions can respond quickly. The time cost of this article is slightly higher than CDAK, but the proposed solution is significantly better than CDAK in resisting long-term observation attacks.

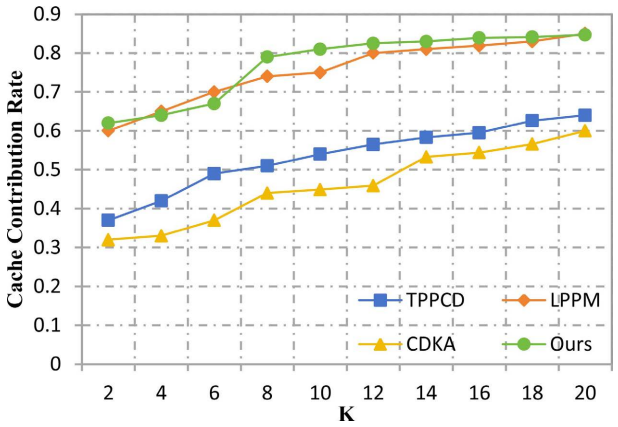

Figure 11. Comparison of cache contribution rate.

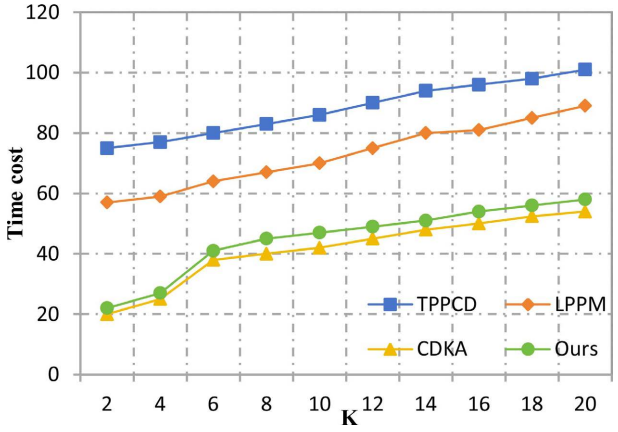

Figure 12. Comparison of time cost.

5. Conclusion

In response to the shortcomings of existing trajectory privacy protection schemes, this paper proposes a trajectory privacy protection method that resists long-term observation attacks. This method mainly consists of four algorithms: CRLA, RLOA, BCAR and UCA. It uses an improved differential privacy mechanism and quadtree, personalized service quality requirement settings, and randomization factors. At the same time, it combines real anonymous regions and obfuscated anonymous regions to obfuscate the temporal and spatial correlation of user actual trajectories, thereby comprehensively protecting user trajectory privacy. The experimental results show that this scheme can resist long-term observation attacks while ensuring a high cache contribution rate, low time overhead, and high availability.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Zhang, M., Li, X., Miao, Y., Luo, B., Ren, Y. and Ma, S. (2024) PEAK: Priva-

cy-Enhanced Incentive Mechanism for Distributed—Anonymity in LBS. *IEEE Transactions on Knowledge and Data Engineering*, **36**, 781-794. https://doi.org/10.1109/TKDE.2023.3295451

[2] Jiang, H., Li, J., Zhao, P., Zeng, F., Xiao, Z. and Iyengar, A. (2021) Location Privacy-Preserving Mechanisms in Location-Based Services: A Comprehensive Survey. *ACM Computing Surveys*, **54**, Article No. 4. https://doi.org/10.1145/3423165

[3] Zhang, J., Xu, L. and Tsai, P.W. (2020) Community Structure-Based Trilateral Stackelberg Game Model for Privacy Protection. *Applied Mathematical Modelling*, **86**, 20-35. https://doi.org/10.1016/j.apm.2020.04.025

[4] Jiang, J., Han, G., Wang, H. and Guizani, M. (2019) A Survey on Location Privacy Protection in Wireless Sensor Networks. *Journal of Network and Computer Applications*, **125**, 93-114. https://doi.org/10.1016/j.jnca.2018.10.008

[5] Wei, J., Lin, Y., Yao, X. and Zhang, J. (2019) Differential Privacy-Based Location Protection in Spatial Crowdsourcing. *IEEE Transactions on Services Computing*, **15**, 45-58. https://doi.org/10.1109/TSC.2019.2920643

[6] Huang, Y., Cai, Z. and Bourgeois, A.G. (2018) Search Locations Safely and Accurately: A Location Privacy Protection Algorithm with Accurate Service. *Journal of Network and Computer Applications*, **103**, 146-156. https://doi.org/10.1016/j.jnca.2017.12.002

[7] Jin, W., Xiao, M., Guo, L., Yang, L. and Li, M. (2021) ULPT: A User-Centric Location Privacy Trading Framework for Mobile Crowd Sensing. *IEEE Transactions on Mobile Computing*, **21**, 3789-3806. https://doi.org/10.1109/TMC.2021.3058181

[8] Sun, Y., Chen, M., Hu, L., Qian, Y. and Hassan, M.M. (2017) ASA: Against Statistical Attacks for Privacy-Aware Users in Location Based Service. *Future Generations Computer Systems*, **70**, 48-58. https://doi.org/10.1016/j.future.2016.06.017

[9] Wang, Y., Li, M., Xin, Y., Yang, G., Tang, Q., Zhu, H., *et al.* (2021) Exchanging registered Users' Submitting Reviews towards Trajectory Privacy Preservation for Review Services in Location-Based Social Networks. *PLOS ONE*, **16**, e0256892. https://doi.org/10.1371/journal.pone.0256892

[10] Qiu, S., Pi, D., Wang, Y. and Liu, Y. (2023) Novel Trajectory Privacy Protection Method against Prediction Attacks. *Expert Systems with Applications*, **213**, Article 118870. https://doi.org/10.1016/j.eswa.2022.118870

[11] Zhao, P., Liu, W., Zhang, G., Li, Z. and Wang, L. (2020) Preserving Privacy in Wifi Localization with Plausible Dummy Locations. *IEEE Transactions on Vehicular Technology*, **69**, 11909-11925. https://doi.org/10.1109/TVT.2020.3006363

[12] Tang, J., Zhu, H., Lu, R., Lin, X., Li, H. and Wang, F. (2021) DLP: Achieve Customizable Location Privacy with Deceptive Dummy Techniques in LBS Applications. *IEEE Internet of Things Journal*, **9**, 6969-6984. https://doi.org/10.1109/JIOT.2021.3115849

[13] Song, D., Song, M., Shakhov, V. and Park, K. (2021) Efficient Dummy Generation for Considering Obstacles and Protecting User Location. *Concurrency and Computation: Practice and Experience*, **33**, e5146. https://doi.org/10.1002/cpe.5146

[14] Siddiqie, S., Mondal, A. and Reddy, P.K. (2021) An Improved Dummy Generation Approach for Enhancing User Location Privacy. *Proceedings of the* 26*th International Conference on Database Systems for Advanced Applications*, Taipei, 11-14 April 2021, 487-495. https://doi.org/10.1007/978-3-030-73200-4_33

[15] Zhang, S., Hu, B., Liang, W., Li, K.C. and Gupta, B.B. (2023) A Caching-Based Dual K-Anonymous Location Privacy-Preserving Scheme for Edge Computing. *IEEE Internet of Things Journal*, **14**, 9768-9781. https://doi.org/10.1109/JIOT.2023.3235707

[16] Zhu, S., Lv, X. and Yu, L. (2021) Location Privacy Protection Method Based on Variable Order Markov Prediction Model. *Proceedings of the 4th International Conference on Computer Science and Software Engineering*, Singapore, 22-24 October 2021, 25-30. https://doi.org/10.1145/3494885.3494890

[17] Huang, Q., Xu, X., Chen, H. and Xie, L. (2022) A Vehicle Trajectory Privacy Preservation Method Based on Caching and Dummy Locations in the Internet of Vehicles. *Sensors*, **22**, Article 4423. https://doi.org/10.3390/s22124423

[18] Li, F., Yin, P., Chen, Y., Niu, B. and Li, H. (2020) Achieving Fine-Grained QoS for Privacy-Aware Users in LBSs. *IEEE Wireless Communications*, **27**, 31-37. https://doi.org/10.1109/MWC.001.1900469

[19] Zhang, L., Zhu, S., Li, F., Li, R., Meng, J. and Li W. (2020) A Trajectory-Privacy Protection Method Based on Location Similarity of Query Destinations in Continuous LBS Queries. *Wireless Algorithms, Systems, and Applications*: 15*th International Conference*, Qingdao, 13-15 September 2020, 704-715. https://doi.org/10.1007/978-3-030-59016-1_58

[20] Gao, Z., Huang, Y., Zheng, L., Lu, H., Wu, B. and Zhang, J. (2022) Protecting Location Privacy of Users Based on Trajectory Obfuscation in Mobile Crowdsensing. *IEEE Transactions on Industrial Informatics*, **18**, 6290-6299. https://doi.org/10.1109/TII.2022.3146281

[21] Gupta, A.K. and Shanker, U. (2020) OMCPR: Optimal Mobility Aware Cache Data Pre-Fetching and Replacement Policy Using Spatial K-Anonymity for LBS. *Wireless Personal Communications*, **114**, 949-973. https://doi.org/10.1007/s11277-020-07402-2

[22] Liu, Z., Miao, D., Li, R., Liu, Y. and Li, X. (2023) Cache-Based Privacy Protection Scheme for Continuous Location Query. *Entropy*, **25**, Article 1569. https://doi.org/10.3390/e25121569

[23] Wang, X. and Yang, W. (2020) Protection Method of Continuous Location Uploading Based on Local Differential Privacy. 2020 *International Conference on Networking and Network Applications*, Haikou, 10-13 December 2020, 157-161. https://doi.org/10.1109/NaNA51271.2020.00035