ISSN 2616-5775 Vol. 7, Issue 5: 79-86, DOI: 10.25236/AJCIS.2024.070510

A Method for Generating Confusing Positions to Resist Long-term Observation Attacks

Qixin Zhan[1,a,*]

[1]School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, 411201, China
[a]1719713004@qq.com
[*]Corresponding author

Abstract: *With the upgrading of mobile smart devices and rapid innovation of positioning technology, the widespread application of Location Based Services (LBS) has been propelled. However, users face the threat of privacy leaks while using LBS, especially when their behaviors over a long period of time are collected and stored. The accumulated information can be exploited by adversaries, greatly increasing the risk of privacy leakage. This type of attack is known as long-term observation attack. How to better protect users' location privacy remains a challenging issue. In this paper, we propose a method for generating obfuscated locations to counter long-term observation attacks. Firstly, we determine the output set based on the user's quality of service requirements and custom sampling probability. Then, we introduce an improved quadtree structure to store user location information and select users with similar destinations to form an anonymous set. Finally, noise is added to prevent long-term observers from accurately obtaining the user's precise location. Experimental results demonstrate that our method can effectively protect users from long-term observation attacks while maintaining low time cost and high anonymity efficiency.*

Keywords: *Location privacy, Long-term observation attacks, Quadtree*

1. Introduction

LBS refers to obtaining the specific geographical location coordinates of users in the mobile Internet through positioning technology, and the location service provider (LSP) provides the user with corresponding information query, entertainment games and other related mobile Internet services [1]. In real-life scenarios, LBS is commonly used in applications such as maps and navigation, local searches, and location-based advertising. Examples include Facebook, Google Maps, among others. This demonstrates the widespread application of LBS. However, during the process of obtaining LBS services, users need to send their location information to untrusted LBS providers, posing the risk of privacy leakage [2]. These location information has high commercial value and research value. LBS providers may compare their query data with other datasets without user consent, potentially revealing information about users' religious beliefs and other vital personal details. Directly sending users' locations to LBS leaves their privacy unprotected.

To safeguard users' location privacy, several protection techniques are currently employed: Based on false location technology, this technology can hide the user's real location in false locations[3].Technologies utilizing anonymous zones, which protect users' location information by generalizing position data[4].Encryption-based methods, which apply cryptographic techniques to encrypt users' location information, ensuring that personal data is encrypted and protected during transmission[5].Differential privacy techniques, which protect data privacy by adding random noise to the data, preventing attackers from accessing it. Adjusting the privacy budget can achieve this noise, making it impossible for attackers to infer specific individual information[6].

Specifically, attackers can easily collect users' private information by leveraging the spatial and temporal correspondences in their trajectories through long-term observation. To address this, [7] investigated location inference attacks based on the probability distribution of historical location data. Confuse the user's location by generating a security hidden area (CR) and send the CR to the service provider. This approach's CR pruning technique establishes a balance between privacy and delay in LBS usage. [8] introduced the Long-term Statistical Attack (LSA) and proposed the MNAME method, where users store multiple usernames. When using LBS services, users select a name from the username set as their current username and send it to the LBS server. At the same time, the SNAME method is also proposed, that is, the anonymous server will change the query and the anonymous server will change each user name to the same user name and then send it to the LBS server. However, balancing user availability and privacy remains a significant challenge. To minimize the impact on availability and accuracy while preventing attackers from leveraging background knowledge of users' obfuscated locations and long-term data collection to infer their actual locations, [9]proposed a mechanism called Eclipse to resist long-term observation attacks. This mechanism combines geographic indistinguishability, k-anonymity, and expected inference errors to protect user location privacy. However, the scheme searches along the Hilbert curve to obtain the anonymous set specified by the scheme, which has the problem of low anonymity efficiency. Additionally, concentrating all operations on the client side can lead to issues such as long server response times and poor user experiences due to limited network bandwidth and cloud computing resources. Therefore, this paper proposes a method to generate obfuscated locations that resist long-term observation attacks. It uses an improved quadtree structure to store user location information and selects users with similar destination preferences to construct anonymous sets. Through a framework of "user end - edge end - cloud end," the user's actual location is processed through three endpoints: the user end, edge server, and LBS server, resulting in an obfuscated location. This approach ensures a prompt response to user location service demands while resisting long-term observation attacks.

2. Proposed Method for Generating Confusing Positions

As shown in Figure 1, the system model of this article's scheme mainly consists of three layers: Mobile Client, Edge Server, and LBS Server. The introduction of each layer is as follows:

(1) Mobile Client: Users can make query requests to edge servers and adjust their QoS requirements to meet their personalized needs for different query points. Sample some elements from the candidate set C through a sampling method to form a possible output set, and the sampling probability b can be determined by the user.

(2) Edge Server: The edge server mainly processes data from the mobile client, and in this scheme, it is mainly responsible for generating quadtree and constructing anonymous set, and sending the two dense sets to the LBS server. In real life, it can be regulated by social and public information departments, such as the Public Security Bureau, the National Security Bureau, and other social roles, in order to improve the credibility of servers and also help track offenders for law enforcement work.

(3) LBS Server: The LBS server can provide accurate services to LBS users, and all service data is stored on the LBS server. When the LBS server receives a query request, it searches in the database and feeds back the query results to the user through the edge server. In this scheme, the main responsibility is to generate confusion positions and provide feedback on the results.

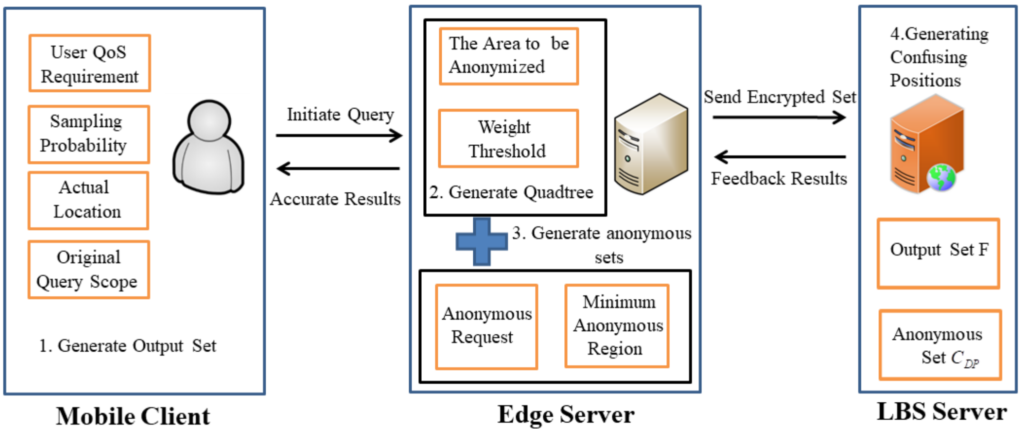

Figure 1: System Model Diagram

2.1. Identify Possible Location Output Set

Sometimes, the query results obtained by using the existing location privacy protection methods through obfuscating the real location do not meet the user's quality of service (QoS) requirements very well. At the same time, users have different needs in different scenarios. For example, in general, users have higher privacy requirements in hospitals than in shopping centers. In continuous LBS queries, If the QoS requirements are uniform across every query point rather than personalized, it will be impossible to provide users with better services tailored to their specific needs. QoS is related to the service data provided by LBS providers. When a user submits their actual location, they can obtain the required service data. However, if a user submits an obfuscated location generated through location privacy protection methods, they may receive two types of service data: data that satisfies their needs and data that is unrelated to their demands. Therefore, to meet the user's QoS requirements, the original query range r_0 is modified to the submitted query range r_s. As the submitted query range increases, both the data that satisfies the user's needs and the data unrelated to their demands will also increase. In summary, this article proposes a personalized QoS approach, which allows users to adjust their QoS requirements based on their needs. The specific calculation formula is as follows:

$$QoS(l',r_s)=\varphi\left[\frac{S(l',r_s)\cap S(l,r_o)}{S(l,r_o)}-\omega.\frac{S(l',r_s)-S(l,r_o)}{S(l',r_s)}\right] \tag{1}$$

$S(l',r_s)$ 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 represent 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.

By using this formula, the obfuscated locations obtained can meet the user's QoS requirements. However, as users generate more obfuscated locations using location privacy protection methods at the same location, long term observation allows attackers to count these confusion locations and regard the center of the circular area formed by all confusion locations as the user's real location. Therefore, the solution proposed in this article is to first generate a candidate set that satisfies the user's QoS requirements, and then select some obfuscated locations from this candidate set using a sampling method. The sampling probability can be determined by the user themselves. Through the above steps, a location output set F will be formed on the client side in this article.

2.2. Determine Aonymous Set

This scheme divides position data by introducing a relative weight threshold μ. Each node in a quadtree is represented as $e(value,p)$. Then its parent node is represented as $p(Value,sub)$. The $value$ is the weight of nodes and $e.value$ represent the number of location points in the area where this node is located. The weight calculation method is represented as:

$$w[e]=\frac{e.value}{p.value} \tag{2}$$

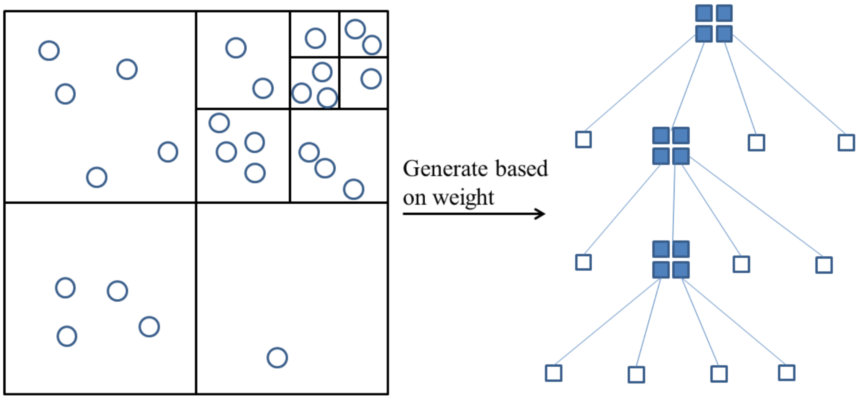

Figure 2: Weighted Region Division

Firstly, preprocess the geospatial area for constructing anonymous regions, divide the regions based on the relative weight threshold μ, and divide them when the weight in the region exceeds μ. The first layer is represented as the root of the quadtree, and then recursively divide the region until the anonymous weight in the leaf node area does not exceed μ. After dividing the area, the leaf nodes represent the smallest partition area. As shown in Figure 2, the quadtree T obtained by partitioning with μ=1/6.

Step 1: Calculate the weight of each node in the quadtree according to formula (2)

Step 2: Calculate the weight of each region relative to the region of interest based on the following formula:

$$W[e] = \times w_{ij}[e]W[p], 1 \le j \le 4 \tag{3}$$

where $w_{ij}[e]$ denotes the weight of the desired area.

Step 3: According to formula (4), compare the calculation result of the second step with μ, and then divide the region into a quadtree based on the prescribed relative weights until the entire region cannot be further divided.

$$D = \begin{cases} True, W[e] < \mu \\ False, W[e] < \mu \end{cases} \tag{4}$$

D denotes whether to divide the region. When D is true, it indicates continued partitioning, when D is false, it indicates stopped partitioning, and μ represents the threshold for relative weight.

After obtaining a potential set of location outputs using this solution, it is sent to the edge server along with the user's location information, including their current position and query content. Upon receiving this information, edge server locates the user location to the surrounding area and finds all users who satisfy location similarity of query destinations (LS-QD) 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 follows:

$$L(u_i, u_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{5}$$

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.

To prevent user information from being collected by attackers through long-term observation, in order to obtain true location of user. This article proposes adding random noise to the constructed anonymous set C, and the specific formula is as follows:

$$C_{DP} = C + (\psi_1 \cdot Laplacenoise(\varepsilon) + \psi_2 \cdot Laplacenoise(\varepsilon) + ... + \psi_K \cdot Laplacenoise(\varepsilon)) \tag{6}$$

Where $\psi_i (i \in [1, k])$ denotes the probability coefficient of the Laplace mechanism being used by k users. The coefficient can be determined by the user, and this multi randomization makes it difficult for long-term observation attackers to obtain the true information of the user. After adding random noise, obtain the target anonymous set, and then send the anonymous set and output set to the LBS service provider for further processing.

2.3. Differential Confusion and Feedback

Through the first two steps, we obtain a set of possible position output set C_{DP} and anonymous set F. In this paper, we adopt the idea of selecting confusing positions from the position output set. The exponential mechanism is another mechanism for implementing differential privacy. Given a dataset D, if the output set of a query function is R, then each value r in the set represents the output term of the output set. Using a scoring function $q(D, r) \rightarrow R$ to evaluate the quality of output items r. The sensitivity of this function is Δq. Assuming there is a location privacy protection mechanism A , taking the dataset D as input to this mechanism will output r, If the location privacy protection mechanism satisfies the following formula at this time:

$$A(D,q)=\{r\,|\Pr[r\in R]\propto\exp(\frac{\varepsilon q(D,r)}{2\Delta q})\} \tag{7}$$

This indicates that the location privacy protection mechanism satisfies differential privacy.From this, it can be seen that the mechanism is in line with the ideas of this article. Therefore, this article proposes an improved exponential mechanism to achieve differential privacy. Firstly, we need to set up a scoring function to evaluate the quality of the output confusion position. This article uses the Euclidean distance between the confusion position l' and the actual position l to evaluate.

A smaller Euclidean distance means that the confusion position output in this article is better. The scoring function in this article will be defined as follows:

$$q(l,l')=-d_{euc}(l,l') \tag{8}$$

Because the anonymous set represents the user's neighboring users, the sensitivity of the rating function is:

$$\begin{aligned}\Delta q&=\max_{l'\in F;l_i,l_j\in C_{DP}}\|[-d_{euc}(l_i,l')]-[-d_{\mathrm{euc}}(l_j,l')]\|_1\\&=\max_{l'\in F;l_i,l_j\in C_{DP}}\|[-d_{euc}(l_i,l')+d_{\mathrm{euc}}(l_j,l')]\|_1\end{aligned} \tag{9}$$

where l' denotes one of the possible positions in the output set, l_i,l_j denotes the user's position points on the anonymous set. According to the following triangular inequality:

$$|-d_{euc}(l_i,l')+d_{euc}(l_j,l')|\le d_{euc}(l,l')<L(C_{DP}) \tag{10}$$

where $L(C_{DP})$ denotes the maximum euclidean distance that may exist between two users inside the large square when constructing the anonymous area in step two, which is the diagonal length of the square. So we can get $\Delta q=L(C_{DP})$.

Given a user's correct location l, a set of possible location output set F and an anonymous set C, differential privacy is experimented through exponential mechanism A_L. The scheme selects and outputs the probability of confusion position l' directly proportional to $\exp(\frac{\varepsilon q(l,l')}{2\Delta q})$ from output set F.

In order to resist long-term observation attacks, our important goal is to maximize the prevention of long-term observation attackers from accurately obtaining the output set F, anonymous set C, and other related information generated through the method described in this article. This article adopts the method of user-defined probability coefficients for differential obfuscation, and adds a function on the mobile client to remind user to input the correct randomization coefficients to protect their personal privacy. When users input these customized coefficients, they can continue to query. So the index mechanism adopted in this article is as follows:

$$A_E(z\,|\,l)=\lambda_1\cdot A_E(z\,|\,l_1)+\lambda_2\cdot A_E(z\,|\,l_2)+\ldots+\lambda_k\cdot A_E(z\,|\,l_k) \tag{11}$$

where $l_i(i\in[1,k])$ denotes the user location point of anonymous set C, $A_E(z\,|\,l)$ denotes using an exponential mechanism at position l. $\lambda_i(i\in[1,k])$ denotes user-defined coefficient of exponential mechanism.

Through the above steps, the proposed solution randomly selects a location from the anonymous set to generate confusing location. Finally, the service provider processes the request and sends the desired results to the user through the edge server.

3. Experiment

3.1. Experiment Settings

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[10]. The experiments are compared with Eclipse [9] and ISTDP[11].The reason is that these schemes adopt similar technologies as this article, highlighting the advantages of the proposed schemes by comparing similar schemes.

This article mainly uses long-term privacy measurement to measure the degree of privacy protection. By collecting confusing locations from historical queries, select the three most frequently occurring locations:

$$\arg\max_{l_1,l_2,l_3\in L} o_{l_1}+o_{l_2}+o_{l_3} \tag{12}$$

Where $l_1 \neq l_2 \neq l_3$ and o_l denotes frequency of location l to be considered as the obfuscated location. This experiment measures long-term privacy by calculating the average distance between the actual location and these selected locations, and then using the following formula:

$$\frac{\sum_{i=1}^{i=3} d_{euc}(l,l_i)}{3} \tag{13}$$

3.2. Experiment Result

As shown in Figure 3, the experimental results describe the change of privacy protection degree of this scheme when the privacy budget changes from 0.1 to 1.3. As the privacy budget increases, the degree of privacy protection will also decline. This is because the larger the privacy budget, the less noise will be generated, which will lead to the reduction of privacy.

As shown in Figure 4, as the value of k increases, the time cost also increases. This is because as k increases, it means higher privacy is required, which increases the number of anonymous users to ensure their privacy. Therefore, more time is needed to expand the anonymous area. The quadtree in this article is divided using threshold indicators to make the division of the quadtree more rational. Therefore, it is efficient than ISTDP when performing anonymous operation, as it requires recursive division of the sub line segment tree. Eclipse concentrates all operation on mobile Client, especially when large-scale users query at the same time. This solution will handle encryption and decryption operations on mobile devices, and also need searching for anonymous sets along the Hilbert curve, which consumes a lot of time. The scheme of this paper is to use the framework of "mobile client-edge server-LBS server" and the improved quadtree to store the location data of users, and lead into the new concept of user movement trend to form k-anonymity, so the time cost will be lower.

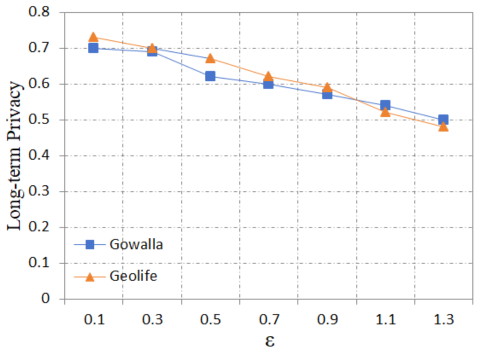

Figure 3: The Impact of ε *on Privacy*

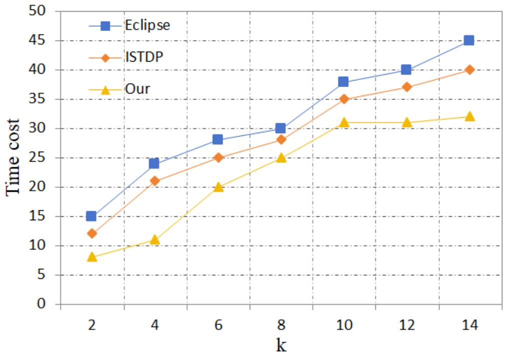

Figure 4: Comparison of Time Cost

As shown in Figure 5, the success rate of each scheme gradually decreases with the increase of k. This is because as the demand for anonymity among users expands, more anonymous users need to form anonymous set to ensure privacy. The proposed scheme in this article can provide an excellent anonymity success rate. Eclipse searches along the Hilbert curve to find an anonymous set. Due to the large number of anonymous users being searched for, the success rate of anonymity is relatively low. Both the ISTDP scheme and the anonymous extension method proposed in this paper are based on a tree structure, and these two schemes have different anonymous time cost, but the difference in anonymous success rates is not significant

As shown in Figure 6, the long-term privacy change of ISTDP is much smaller when the privacy budget changes from 0.1 to 1.3. This is because long-term observation of attackers will collect user location information over a long period of time, and finally infer the actual location of the user from the center of numerous confusing locations. ISTDP cannot protect the user's location privacy under long-term observation attacks. According to the experimental results shown in Figure 3, we can know that with the increase of privacy budget, the degree of privacy protection will also decline. User will not allocate excessively high privacy budgets when using this solution. In this case, the long-term privacy protection effect of the proposed solution in this article is slightly better than that of Eclipse, as this article introduces more personalized and diverse random factors, making it more difficult for attackers to infer the true location of users. In summary, this proposed solution can resist long-term observation attacks while maintaining low time overhead and high anonymity success rates.

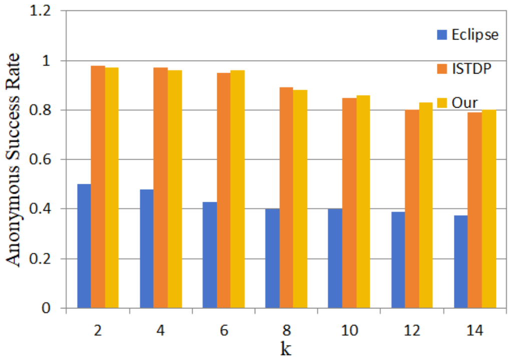

Figure 5: Comparison of Anonymous Success Rate

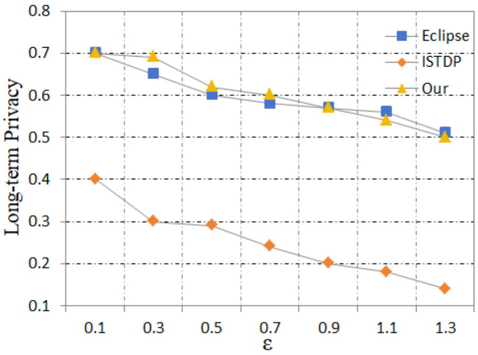

Figure 6: Comparison of Long-term Privacy

4. Conclusions

With the widespread development of location service applications, location privacy protection methods are still a hot topic of discussion among scholars. In this paper, we introduce the existing location privacy protection methods and propose a confusion location generation method that can resist long-term observation attacks, aiming to address the shortcomings of the current methods. This paper ensures a positive user experience by adopting personalized QoS settings. Additionally, it proposes an enhanced quadtree method for constructing an anonymous set. Based on existing location privacy protection techniques, it add randomization that is difficult for attackers to recognize, while maintaining the accuracy of query results, to safeguard users from long-term observation attacks. The experimental results show that this method can resist long-term observation attacks while ensuring low time cost and high anonymity success rate.

References

[1] Zhang M, Li X, Miao Y, et al. PEAK: Privacy-Enhanced Incentive Mechanism for Distributed K-Anonymity in LBS [J]. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(2): 781-794.

[2] Qiu C, Squicciarini A, Pang C, et al. Location privacy protection in vehicle-based spatial crowdsourcing via geo-indistinguishability[J]. IEEE Transactions on Mobile Computing, 2020, 21(7): 2436-2450.

[3] Tang J, Zhu H, Lu R, et al. DLP: Achieve customizable location privacy with deceptive dummy techniques in LBS applications[J]. IEEE Internet of Things Journal, 2021, 9(9): 6969-6984.

[4] Jiang J, Han G, Wang H, et al. A survey on location privacy protection in wireless sensor networks [J]. Journal of Network and Computer Applications, 2019, 125: 93-114.

[5] Wei J, Lin Y, Yao X, et al. Differential privacy-based location protection in spatial crowdsourcing [J]. IEEE Transactions on Services Computing, 2019, 15(1): 45-58.

[6] Huang Y, Cai Z, Bourgeois A G. Search locations safely and accurately: a location privacy protection algorithm with accurate service[J]. Journal of Network and Computer Applications, 2018,

103: 146-156.

[7] Shahid A R, Pissinou N, Iyengar, S S, et al. Delay-aware privacy-preserving location-based services under spatiotemporal constraints[J]. International Journal of Communication Systems , 2021, 34(1): 1-20.

[8] Sun Y, Chen M, Hu L, et al. Asa: against statistical attacks for privacy-aware users in location based service [J]. Future Generations Computer Systems, 2017, 70: 48-58.

[9] Niu B, Chen Y, Wang Z, et al. Eclipse: Preserving differential location privacy against long-term observation attacks [J]. IEEE Transactions on Mobile Computing, 2022, 21(1): 125-138.

[10] Wang X, Yangg W. Protection method of continuous location uploading based on local differential privacy[C]// 2020 International Conference on Networking and Network Applications (NaNA). IEEE Press, 2020: 157-161.

[11] Hu d, Liao Z. Differential Privacy of Location Privacy Protection Method for Irregular line segment Tree [J]. Journal of Chinese Computer Systems, 2020, 41(2): 333-337.