*electronics*

MDPI

*Article*

# Distributed K-Anonymous Location Privacy Protection Algorithm Based on Interest Points and User Social Behavior

**Ling Xing, Dexin Zhang, Honghai Wu, Huahong Ma and Xiaohui Zhang ***

School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China; xingling_my@haust.edu.cn (L.X.); 200320050350@stu.haust.edu.cn (D.Z.); honghai2018@haust.edu.cn (H.W.); mhh@haust.edu.cn (H.M.)

* Correspondence: 9906117@haust.edu.cn

**Abstract:** Location-based services have become an important part of our daily lives, and while users enjoy convenient Internet services, they also face the risk of privacy leakage. K-anonymity is a widely used method to protect location privacy, but most existing K-anonymity location privacy protection schemes use virtual locations to construct anonymity zones, which have the problem of being vulnerable to attackers through background knowledge, while the improved collaborative K-anonymity scheme does not sufficiently consider whether collaborating users share similar attributes. We propose a distributed K-anonymity location privacy-preserving algorithm based on interest points and user social behaviors to solve these problems in existing K-anonymity schemes. The method determines the similarity of users by their interest points and social behaviors and then selects users with high similarity to build an anonymous set of collaborative users. Finally, to ensure the relatively uniform distribution of collaborative users, a homogenization algorithm is used to make the anonymous location points as dispersed as possible. The experimental results showed that our algorithm can effectively resist background attacks, and the uniformly distributed anonymous location points can achieve higher-quality anonymous regions.

**Keywords:** location privacy protection; K anonymity; user similarity; location; points of interest; homogenization user collaboration



**Citation:** Xing, L.; Zhang, D.; Wu, H.; Ma, H.; Zhang, X. Distributed K-Anonymous Location Privacy Protection Algorithm Based on Interest Points and User Social Behavior. *Electronics* **2023**, *12*, 2446. https://doi.org/10.3390/electronics12112446

Academic Editors: Muath Obaidat and Kutub Thakur

Received: 15 April 2023
Revised: 23 May 2023
Accepted: 25 May 2023
Published: 29 May 2023



## 1. Introduction

Location-Based Service (LBS) [1,2] is a service developed based on location information provided by users, which provides users with services such as point of interest query and personalized information push with the support of a Location-based Services Platform (LSP). With the rapid development of 5G and IoT technology, terminal devices are becoming popular, LBS covers many application scenarios, and a series of privacy leakage problems come with this. When users request services from location service providers, they need to provide their location information, and the location service providers return services through the received location information, such as nearby hotels, restaurants, navigation routes, local weather conditions, and so on [3–7]. There is a hidden danger of privacy leakage, which is that if an attacker obtains the user's requested data, he/she can steal the user's true location, and the security of the user's personal information will also be threatened. Therefore, protecting user location privacy has also become a research topic that cannot be ignored.

There are two types of location privacy for users, namely spatial location privacy and spatio-temporal correlation location privacy. Among them, spatial location privacy is the location service request information initiated by the user at each point in time; spatio-temporal location privacy refers to the location service request information continuously initiated by users during a certain period of time. There are many protection methods for the two types of location privacy mentioned above, such as differential privacy [8,9],

the homomorphic encryption method [10], coordinate transformation [11], anonymous steganography, etc. K-anonymity [12] belongs to anonymous steganography, which constructs an anonymous area by combining the user's real location with K-1 virtual locations. When the requesting user sends an LBS request, the user replaces his/her real location with the anonymous area and submits it to the LSP, effectively protecting his/her personal location privacy [13,14]. It has the following advantages: (1) low computational overhead; (2) simple structure, independent of complex password systems; (3) accurate service requests ensure high-quality LBSs.

However, in the traditional K-anonymity approach, the virtual locations for collaboration are not controllable, and they may generate locations that are unlikely to exist in daily life, such as hilltops, lakes, rivers, etc. After these locations are submitted, LSPs can identify these unoccupied areas, which in turn leads to a reduction in the size of the anonymous area and a decrease in anonymity performance. To solve this problem, distributed K-anonymity based on user collaboration is proposed, in which users can autonomously seek the help of collaborating users around them and construct anonymity zones with the help of collaborating users' real locations. However, collaborating users are not associated with requesting users, and they are vulnerable to attackers using contextual inference attacks to exclude some of the collaborating users, resulting in degraded anonymity privacy protection performance. Although the collaborative user's location will not appear in the theoretical unoccupied area, the distribution of location points is still uncontrollable, and there is a high probability that the anonymous location points will be too concentrated or too scattered, which will make the constructed anonymous area smaller in scope. The K-anonymity scheme uses the entire anonymized area instead of the user's real location to request the query service, so we always want a relatively large anonymized area to protect the user's location privacy while ensuring the quality of service. To solve the above problems, this paper proposes a user-similarity-based privacy-preserving algorithm for K-anonymous locations. The main contributions of this paper are as follows:

- A comprehensive user similarity is constructed by user point of interest similarity and user social behavior similarity, so that the requesting user can find the undifferentiated collaborating user when the collaborating user constructs an anonymous zone. This approach makes it impossible for an attacker to infer the location of the real user through background knowledge.
- The real location of the collaborating user is used to form the anonymity set, avoiding the problem of uncontrollable virtual locations. The requesting user can construct the anonymity zone independently of the collaborating user without the help of a centralized anonymizer.
- By homogenizing the collaborative user set, the homogeneity of the anonymous location points within the anonymous region is improved while the range of the anonymous region is guaranteed to be constant, resulting in a higher quality of the anonymous region.

The rest of this paper is organized as follows. Section 2 describes the related work. Section 3 presents the definitions related to location privacy protection schemes. Section 4 presents the proposed Se-CUA approach. Section 5 discusses the security of the proposed scheme and presents the simulations and results, and Section 6 concludes the paper.

## 2. Related Works

The anonymity-based location privacy protection methods are divided into central-server-based and distributed location-based anonymity protection methods. In a central-server-based location privacy protection scheme, all user anonymization operations are centralized in a central anonymous server between the user and the location service provider. However, this location privacy protection scheme needs to place the burden of privacy protection on a trusted central anonymizer, and the central server becomes a performance and security bottleneck, which will directly cause the leakage of the user's location privacy if the central anonymizer is not trusted or breached by attackers. With the improvement of

the performance of various terminal devices and the advancement of computer processing power, a distributed terminal-based user collaborative location privacy protection scheme has emerged in order to prevent the problem of a single point breakthrough that exists in the central anonymous server. Table 1 provides a detailed comparison of these schemes.

**Table 1.** Comparison of existing studies.

| | **Virtual Location Solutions** | | | | **Real Location Solutions** | | | |
|---|---|---|---|---|---|---|---|---|
| | **[15]** | **[16,17]** | **[18]** | **[19]** | **[20]** | **[21]** | **[22]** | **[23]** |
| Privacy protection degree | Low | Medium | High | High | High | Medium | High | High |
| Calculated overhead | Low | Medium | Low | Low | Medium | Low | Low | Medium |
| Communication overhead | Low | Low | Medium | High | High | Low | Medium | High |
| Service quality | Medium | High | Medium | Medium | High | Medium | High | High |

### *2.1. Virtual Collaboration Location-Based Solutions*

Gruteser et al. [15] first used K-anonymity for location privacy protection, but they assumed that all users have the exact location privacy protection requirement K, which cannot meet the personalized needs of individual users. In addition, the solution is less efficient as it processes each user individually.

Niu et al. [16] proposed a virtual location selection algorithm that selects anonymity sets based on the query probability of each location and filters the regions with higher privacy levels together to form anonymity sets by entropy filtering. At the same time, geometric methods are used to make the location anonymity units as far away as possible to make the constructed anonymity regions larger. Subsequently, to improve the query cache hit rate, Niu et al. [17] proposed a cache-based virtual location selection algorithm. In this scheme, users cache past service data and prioritize the use of local search to obtain service data from partner users when LBSs are needed. Historical query probability is introduced to improve the cache hit rate and to further improve the degree of location privacy protection by maximizing query privacy.

Parmar et al. [24] proposed a new privacy-preserving technique based on virtual generation. The proposed virtual generation technique is a circle-based technique. It generates virtual locations in a circular region and is effective against anonymous spatial region center attacks, map matching attacks, and location homogeneity attacks. Yang et al. [18] utilized the Stackelberg game framework to identify the optimal set of virtual locations by considering location semantics, physical dispersion, and query probability. This method involves the mutual optimization of the user and adversary objectives and strives to maintain the quality of service while safeguarding location privacy from single-point attacks.

Niu et al. [19] designed a personalized spatial camouflage scheme considering the location privacy and query privacy of users and combined it with user customization to allow users to autonomously choose the size of anonymous regions. A set of real regions is first identified as candidates using a virtual location determination algorithm. Then, the user is assigned an anonymous region that satisfies the expectation based on the user's personalized needs.

Zhang et al. [25] suggested a method for improving user privacy while avoiding the issue of discarding unneeded historical query results. This approach utilizes caching and spatial K-anonymity and involves multiple levels of caching to reduce the reliance on Location-Based Services (LBSs). They used a Markov model to predict a user's next query location based on movement patterns. Combining this anticipated location with the cache's contribution and data freshness can increase the cache hit rate while safeguarding the user's location privacy.

Zhu et al. [26] proposed a knowledge-driven location privacy protection scheme to meet the demand for customized location privacy protection based on users' personalized features. The solution proposes the UBPG algorithm to mine the base persona, model user familiarity and user curiosity, and generate a psychological portrait. Then, a location

migration matrix based on user profiles is constructed to migrate real locations to anonymous locations. Sei et al. [27] proposed a privacy-preserving data collection method that considers many missing values to address the problem that existing methods for collecting personal information do not take into account the possible presence of missing values while ensuring data privacy.

Liu et al. [28] proposed a method to select virtual locations in terms of temporal accessibility, directional similarity, and in/out degree for the problem that the set of adjacent locations submitted in consecutive requests always contains tight spatio-temporal correlations, leading to degraded privacy-preserving performance. The method filters out the virtual locations that are identifiable considering spatio-temporal correlation and retains the remaining location points, thus satisfying the user's location privacy protection needs.

### 2.2. Real Collaboration Location-Based Solutions

Wang et al. [29] proposed a new approach to constrain users' destructive behaviors while constructing an anonymous blockchain. They used a Multi-Attribute Decision-Making (MADM) algorithm to transform users' credit data into credit values and store them in the anonymous blockchain along with transaction information. In addition, they proposed a credit value reward and punishment mechanism that treats anonymous block building as a two-sided game between requesters and participants. In this game, the credit value reward and punishment mechanisms constrain destructive behaviors. After the simulation experiments, it was verified that the method can be applied in practical scenarios to effectively constrain users' destructive behaviors, quickly build anonymous zones, and reduce the probability of user location leakage problems.

Yu et al. [20] proposed a method called Privacy-Preserving Trajectory Similarity Computation (PTSC), which aims to solve the privacy leakage problem that may result from trajectory similarity calculation in trajectory outsourcing services. In this method, the trajectory service stores the trajectory owner's trajectory in encrypted form while receiving encrypted interest trajectories from the trajectory querier. It can securely compute the similarity between the encrypted interest trajectories and the stored trajectories based on the encryption. A secure computation protocol based on the longest common subsequence was also proposed, which utilizes a homomorphism-like encryption algorithm and a secure comparison protocol to achieve efficient computation of the longest common subsequence of encrypted trajectories.

Ji et al. [21] proposed an approach to guarantee location privacy using a cache-based method. This approach involves a collaborative mobile Peer-To-Peer (P2P) system where users store service data on their mobile devices. The goal is to reduce the possibility of third-party servers becoming a single failure point and minimize service requests to safeguard location privacy. Nevertheless, the approach must address the issue of protecting location privacy when users are in contact with LSPs, which is a typical case.

Liu et al. [22] proposed a blockchain-based distributed K-anonymous location privacy protection scheme for the problem of non-mutual trust among collaborating users. In this scheme, the credit values of collaborative users are stored in the blockchain, and successful participation in constructing an anonymous zone will be rewarded with corresponding incentives. At the same time, a breach of trust will be punished as a solution to the problem of the incomplete trustworthiness of collaborative users in distributed K-anonymity schemes. While satisfying the principle of K-anonymity protection, a combination of multiple private blockchains is used to decentralize users' transaction records. A reward mechanism was proposed to encourage users' participation.

Yang et al. [23] dealt with issues related to the behavior of collaborating users and their ability to influence services in distributed K-anonymity algorithms. They proposed a method for evaluating collaborating parties' reputations based on entity integrity, location information entropy, and service influence. Furthermore, they suggested constructing a trusted invisible region to safeguard the location privacy of requesters. Nodes within this region can select highly reputable nodes to participate in creating the anonymous domain.

In summary, the existing techniques for safeguarding location privacy through K-anonymity do not tackle the issue of whether the combined distribution of user locations satisfies the conditions necessary for creating anonymous regions. As a result, there is a possibility of minimizing the risk of virtual location tracking by enlisting users' help in forming anonymous zones. However, the lack of certainty in the locations of collaborating users leads to ambiguity in the size of the anonymous zone. Moreover, the personal data of collaborating users from different backgrounds vary, undermining anonymity's efficacy. Therefore, the current K-anonymity-based location privacy protection method falls short of fully protecting users' location privacy since the resulting anonymous region fails to meet the privacy needs of the requesting users.

## 3. Preparatory Knowledge

### *3.1. System Architecture*

In this paper, we used a distributed architecture of user collaboration, where the system consists of mutually collaborating users, communication base stations, GPS satellites, and LSPs, without needing a third-party centralized anonymizer. The schematic diagram of the system structure is shown in Figure 1. When the requesting user initiates a request query to the LSP, it first autonomously looks for similar users to collaborate in generating the anonymous area. After receiving the real location of the collaborating user, it sends it to the LSP with the requesting user's location and query request and waits for the return result from the LSP. The users involved in the collaboration are all real users, and we selected the users with high similarity in the network to participate in the collaboration, aiming to ensure that the participating users are maximized from the attackers. The similarity of users is measured by the user similarity metric.

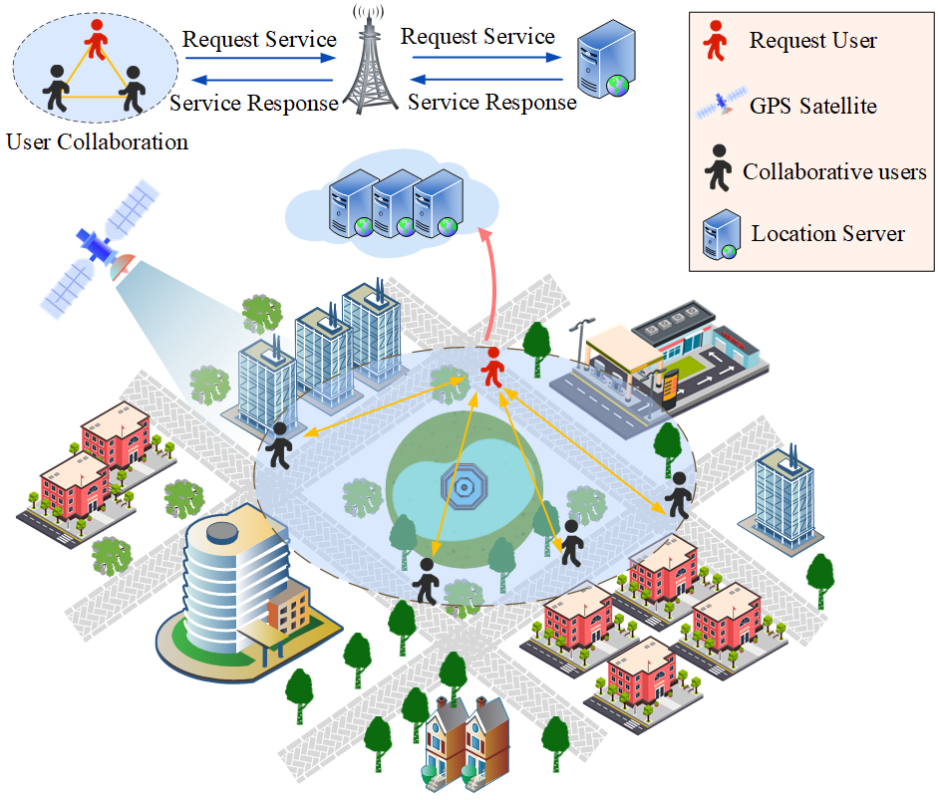


**Figure 1.** Schematic diagram of the system structure.

### *3.2. Attack Model*

The attacker's goal is to obtain sensitive information about a specific user, and we classified the adversaries into two categories: passive and active adversaries. A passive adversary is any entity that can listen and eavesdrop on wireless channels between entities

or can hack into users to obtain sensitive information about other users. Passive adversaries use eavesdropping attacks to obtain additional information about specific users. Active adversaries can compromise LBS servers and obtain information known to the server. In this work, we treated the LBS server directly as an active adversary that obtains global information and monitors the user's current query. In addition, active adversaries are able to access historical data and the current profiles of specific users and understand the location privacy protection mechanisms used in the system.

In this work, the main attacks adopted by the adversaries were conspiracy attacks and inference attacks. Conspiracy attack means that some adversaries do not have the required strategy to determine the real location information of the target user; however, the concatenation of their private user datasets can satisfy the corresponding strategy, and they will join together to try to obtain the real location of the target user. An inference attack is an attack in which an adversary infers sensitive information about a user's location privacy by combining non-sensitive private data obtained by various means.

As an active adversary, LBS servers can not only launch inference attacks by obtaining historical data and current queries of specific users, but also join other attackers to launch conspiracy attacks on target users.

### *3.3. Collaborative User Similarity*

User similarity is an important indicator of how similar the attributes and behaviors of two users are. In this paper, user similarity consists of two parts: user interest point similarity determined by the users' interest points, which consists of the semantic similarity and distance similarity of interest points. The greater the interest point similarity, the more similar the users' attributes and interests seem to be. The other part is the similarity of users' social behavior determined by information about their interactions with each other in the social network, which is measured by the number of senders and recipients who are similar to the target user; the higher the number, the more similar the interaction between users is. The higher the similarity of users, the more suitable they are to participate in the construction of anonymous regions as collaborative users.

#### 3.3.1. Semantic Similarity of Interest Points

A Point Of Interest (POI) is a certain location point with semantic labels, and a meaningless location point is not a point of interest. For example, if a user requests navigation to home from a restaurant, the restaurant is a point of interest, the coordinates of the location are the location information, and the route home is the query content. In a considerable number of scenarios, the POI and query content have a semantic overlap. Location point similarity is the degree of similarity between the points of interest frequented by two users; the higher the degree of similarity, the less likely it is that the two users can be identified by background knowledge.

Interest point similarity consists of semantic similarity and positional similarity. Semantic similarity is the degree of semantic similarity between two points of interest, and we can obtain the semantic similarity by comparing the semantics of the information contained in two points of interest. The points of interest contain information such as the name of the point of interest, the type of point of interest, the text description address of the point of interest, the coordinates of the point of interest, etc. The types of points of interest are shown in Table 2. Thus, the points of interest can be expressed as $P$ = {name, typetext, coordinate}.... Since the name of a point of interest in quite a few cases does not give the user a clear idea of the services offered by the point of interest, we also need to know the functional classification of the point of interest and a textual description of the content of the services offered by the point of interest.

The function of an Apple-Authorized Franchise is described as mobile phone/sales; an Apple-Authorized Service Provider is described as mobile phone/repair; Starbucks Coffee is described as catering/coffee. Apple-owned stores and Starbucks Coffee are two points of interest with entirely dissimilar semantics, with completely different names and

functional descriptions. Apple-Authorized Franchise stores and Apple-Authorized Service Providers are two points of interest with medium semantic similarity, with the same name and different function descriptions. Multiple Apple-Authorized Franchise stores constitute the interest points with high semantic similarity, whose names and function descriptions are the same; only the locations are different.

**Table 2.** Types of points of interest contained in the dataset.

| Point Type | Point Type |
|---|---|
| Entrance/Exit | Finance |
| Real Estate | Hotel |
| Enterprises | Beauty |
| Shopping | Attractions |
| Transportation | Gourmet |
| Education | Car Service |
| Life Services | Media |
| Leisure | Medical |
| Sports | Government |

In this paper, we used the Levenshtein edit distance [30] algorithm to calculate the semantic similarity of location points. The edit distance is the minimum number of single-character edit operations needed to transform one string, $L1$, into another string, $L2$. There are three single-character editing operations, each with a weight of 1: insert, delete, and substitution. They are defined as follows:

$$Edit_{a,b}(i,j) = \max(i,j) \quad \min(i,j) = 0 \tag{1}$$

$$Edit_{a,b}(i,j) = \min \begin{cases} Edit_{a,b}(i-1,j)+1 & \\ Edit_{a,b}(i,j-1)+1 & \min(i,j) \neq 0 \\ Edit_{a,b}(i-1,j-1)+1 & (a_i \neq b_j) \end{cases} \tag{2}$$

$Edit_{a,b}(i,j)$ refers to the distance between the first $i$ characters in string $a$ and the first $j$ characters in string $b$. $(i,j)$ can be considered as the length of string $a$ and string $b$. Therefore, the final edit distance is $Edit_{a,b}(|a|, |b|)$, when $i = |a|$, $j = |b|$.

When $min(i,j) = 0$ corresponds to the first $i$ characters in $a$ and the first $j$ characters in $b$, at this point, $max(i,j)$ has an $a$ value of 0, so the distance between them is $max(i,j)$, which is the largest of $i$,$j$.

When $min(i,j) = 0$ corresponds to the first $i$ characters in $a$ and the first $j$ characters in $b$, at this point, $i$, $j$ have a value of 0, so the distance between them is $max(i,j)$, which is the largest of $i$, $j$.

When $min(i,j) = 0$, $Edit_{a,b}(i,j)$ is the minimum of the following three values:

1. $Edit_{a,b}(i-1,j)+1$ means delete $a_i$;
2. $Edit_{a,b}(i,j-1)+1$ means insert $b_j$;
3. $Edit_{a,b}(i-1,j-1)+1$ $(a_i \neq b_j)$ means replace $b_j$.

After finding the edit distance, the semantic similarity between the two semantic strings is as follows:

$$Sim\_sem(A,B) = \frac{Max(l_A,l_B) - Edit(num_A,num_B)}{Max(l_A,l_B)} \tag{3}$$

where $l_A$ and $l_B$ are the string lengths of the two interest point semantics and $Edit(num_A, num_B)$ is the minimum number of edit operations required to convert from string $A$ to string $B$. In this paper, we used the edit distance algorithm to calculate the semantic similarity of location points. The larger the $Sim\_sem(A,B)$ value, the greater the semantic string similarity of the two words. The $A$ maximum value of 1 indicates that the two

semantic strings are identical; the $A$ minimum value of 0 indicates that the two semantic strings are completely different.

3.3.2. Distance Similarity of Interest Points

There are often cases of "same name and different location" among users' interest points, and the semantic similarity of interest points alone is not enough to describe the similarity of users' interests. Therefore, this paper added the distance of interest points to describe the similarity of interest points comprehensively, and the distance similarity reflects the distance relationship of interest points with the same or similar names. The distance similarity is calculated using the Euclidean distance.

Let the coordinates of the points of interest of user $A$ be $x_1, y_1$ and the coordinates of the points of interest of user $B$ be $(x_2, y_2)$; let the spatial distance between point of interest $A$ and point of interest $B$ be $dis(A, B)$.

The minimum value of $dis(A, B)$ is 0, indicating the same latitude and longitude, and the maximum value $D$, where $D$ is the identification threshold, which indicates the maximum area of this point of interest. When $dis(A, B)$ is greater than $D$, it means that they are not the same point of interest. According to the threshold value $D$, it is obtained that

$$Sim\_dis(A, B) = \begin{cases} 1 & 0 \leq dis(A, B) \leq D \\ 0 & dis(A, B) > D \end{cases} \tag{4}$$

In the above equation, $A$ and $B$ denote two points of interest; $dis(A, B)$ is the Euclidean distance between two points of interest. When $0 \leq dis(A, B) \leq D$, $Sim\_dis(A, B) = 1$, this means that the two points of interest $A$ and $B$ are the same point of interest; when $dis(A, B) > D$, $Sim\_dis(A, B) = 0$, this means that the two points of interest $A$ and $B$ are not the same point of interest.

3.3.3. Similarity of Social Behavior of Collaborating Users

User social behavior similarity is an important part of user similarity, reflecting users' interest preferences, behavior habits, emotional orientation, and other important attributes, and the most-easily identified behaviors among users mostly exist in social networks. The user relationship in social networks is divided into the one-to-one relationship and one-to-many relationship: the one-to-one relationship indicates the interaction between two independent users, which is manifested as mutual message sending; the one-to-many relationship represents the information interaction between a single user and multiple users, which is reflected in the adsorption effect of influential users on other users in social networks, such as the followers of Weibo and Twitter.

If user $U$ sends a message to user $V$, at the same time, user $V$ receives a reply from user $U$. Then, a positive and successful interaction between user $U$ and user $V$ can be considered, indicating that a one-to-one relationship between user $U$ and user $V$ has arisen. If user $U_1$ and user $U_2$ follow user $V$, it means that a one-to-many relationship has been developed between user $V$ and users $U_1$ and $U_2$. Therefore, user behavior similarity can be defined as follows: when both senders $U_1$ and $U_2$ send messages to recipients $V_1$ and $V_2$, $U1$ and $U_2$ are similar senders. When both $V_1$ and $V_2$ receive messages from $U_1$ and $U_2$, $V_1$ and $V_2$ are identical recipients. When both $U_1$ and $U_2$ follow $V$, $U_1$ and $U_2$ are identical followers. The similarity of user behavior can be mathematically defined as follows:

Similar followers: $U_1 \overset{F}{\sim} U_2 : \exists V(U_1 \Rightarrow V \bigcap U_2 \Rightarrow V)$;

Similar speakers: $U_1 \overset{S}{\sim} U_2 : \exists V(U_1 \rightarrow V \bigcap U_2 \rightarrow V)$;

Similar recipients: $U_1 \overset{R}{\sim} U_2 : \exists V(V \rightarrow U_1 \bigcap V \rightarrow U_2)$.

$U_1 \Rightarrow V$ indicates that user $U_1$ follows user $V$; $U_1 \rightarrow V$ indicates that user $U_1$ sent a message to user $V$; $V \rightarrow U_1$ indicates that user $U_1$ received a message from user $V$. Then, the similarity of the social behavior of users $U$ and V can be expressed as:

$$Sim\_act(U, V) = Sim_F + Sim_S + Sim_R \tag{5}$$

where $Sim_F$ is the similarity of following, which indicates the number of users who follow the same user $V$ by both users $U_1$ and $U_2$. $Sim_S$ is the similarity of sending, which indicates the number of users who send messages to the same user $V$ by both users $U_1$ and $U_2$. $Sim_R$ is the similarity of receiving, which indicates the number of users who receive replies from user $V$ by both users $U_1$ and $U_2$. $Sim\_act(U, V)$ indicates the similarity of user behavior that combines the three behaviors of users in the social network degree.

By calculating the above user attributes, we can quantify the degree of similarity between collaborating users into specific values. With these values, we can further determine which users are more suitable to participate as collaborative users in the construction of anonymous regions.

## 4. K-Anonymous Location Privacy-Protection Algorithm Based on User Similarity

### *4.1. Collaborative User Similarity Calculation Algorithm*

The calculation of user similarity based on a single attribute has limitations, such as the occurrence of interest points with the same name distributed in different locations or interest points with different names in the same location. Therefore, it is necessary to combine user attributes to improve the accuracy of user similarity calculation. In addition, different attributes have different impacts on users, and the corresponding weights should be assigned based on the degree of impact of different attributes on users.

Set the data weight coefficients for different attributes of each user as $\varepsilon = \varepsilon_1, \varepsilon_2, \varepsilon_3$, then the relative importance of user attribute $i$ and attribute $j$ can be expressed as $\varepsilon_{ij}$. Assume the approximation of the ratio of the weight coefficient of attribute $i$ to that of attribute $j$ is $r_{ij} \approx \varepsilon_i/\varepsilon_j$. If there are $n$ user attributes, then $(n(n-1))/2$-times of comparison are required, and the matrix $R$ is obtained by comparing $n$ user attributes in pairs:

$$\begin{aligned} R &= \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & r_{nn} \end{pmatrix} \\ &\approx \begin{pmatrix} \varepsilon_1/\varepsilon_1 & \varepsilon_1/\varepsilon_2 & \dots & \varepsilon_1/\varepsilon_n \\ \varepsilon_2/\varepsilon_1 & \varepsilon_2/\varepsilon_2 & \dots & \varepsilon_2/\varepsilon_n \\ \dots & \dots & \dots & \dots \\ \varepsilon_n/\varepsilon_1 & \varepsilon_n/\varepsilon_2 & \dots & \varepsilon_n/\varepsilon_n \end{pmatrix} \end{aligned} \tag{6}$$

Based on the matrix $R$, we can use the Lagrange multiplier method to calculate the weight coefficients $\varepsilon$ for each indicator.

The Lagrange multiplier:

$$L(x, \lambda) = f(x) + \sum_{k=1}^{l} \lambda_k h_k(x) \tag{7}$$

where $\lambda_k$ is the Lagrange multiplier, means a coefficient that is determined for each constraint.

From the matrix $R$, we can obtain:

$$\left\{ \begin{array}{l} r_{ij} = 1/r_{ji} \\ r_{ij} = r_{ik} \cdot r_{kj} \ (\forall i, j, k \in J) \\ r_{ii} = 1 \end{array} \right\} \tag{8}$$

The relative importance of indicator $i$ and indicator $j$ can be expressed as $\sum_{i=1}^{n} r_{ij}$:

$$\sum_{i=1}^{n} r_{ij} = \frac{\sum_{i=1}^{n} \varepsilon_i}{\varepsilon_j} \tag{9}$$

When $\sum_{i=1}^{n} w_i = 1$, we have:

$$\varepsilon_j = \frac{1}{\sum_{i=1}^{n} r_{ij}} \tag{10}$$

According to the least-squares method, it is obtained that:

$$\begin{gathered} \sum_{i=1}^{n} \varepsilon_i = 1 \\ \min\left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} \left( r_{ij}\varepsilon_j - \varepsilon_i \right)^2 \right\} \\ \varepsilon_i > 0, i = 1, 2, \ldots, n \end{gathered} \tag{11}$$

According to Equation (8), we can obtain:

$$L = \sum_{i=1}^{n} \sum_{j=1}^{n} \left( r_{ij}\varepsilon_j - \varepsilon_i \right)^2 + 2\lambda \left( \sum_{i=1}^{n} \varepsilon_i - 1 \right) \tag{12}$$

$$\sum_{i=1}^{n} (r_{il}\varepsilon_l - \varepsilon_i) a_{il} - \sum_{j=1}^{n} \left( r_{lj}\varepsilon_j - w_l \right) + \lambda = 0 \tag{13}$$

where $l = (0, 1, 2, \ldots, n)$. The weight factor $\varepsilon = \varepsilon_1, \varepsilon_2, \varepsilon_3$ can be derived from Equation (14) and $\sum_{i=1}^{n} \varepsilon_i = 1$.

User similarity is composed of user interest point distance similarity, user interest point semantic similarity, and user social behavior similarity, which can be expressed as:

$$\begin{aligned} Sim_{user(U,V)} = \quad & \varepsilon_1 \times Sim\_act(U, V) \\ & + \varepsilon_2 \times Sim\_sem(U, V) \\ & + \varepsilon_3 \times Sim\_dis(U, V) \end{aligned} \tag{14}$$

where $Sim\_act(U, V)$ is the user behavior similarity of user $U$ and user $V$, $Sim\_sem(U, V)$ is the semantic similarity of the interest points of user $U$ and user $V$, $Sim\_dis(U, V)$ is the distance similarity of interest points of user $U$ and user $V$, and $Sim_{user(U,V)}$ is the combined similarity of user $U$ and user $V$. Moreover, $\varepsilon_1$ and $\varepsilon_2$ are the weights of the above attributes, and $\varepsilon_1 + \varepsilon_2 + \varepsilon_3 = 1$.

The similarity values of the participating collaborating users in the network can be obtained according to the user similarity calculation formula. The users with relatively higher similarity can be obtained by ranking the similarity values from the largest to the smallest. The anonymous region constructed by the participation of these users can effectively improve the location privacy protection performance of the anonymous area. In order to avoid the disclosure of user privacy data by the server, the algorithm is executed by the client, as shown in Algorithm 1:

**Algorithm 1:** User similarity calculation algorithm.

**Input:** user's POI name, POI coordinates

**Output:** user's similarity set $\eta$:

1. Calculate semantic similarity $Sim\_sem(A, B)$;
2. Calculate the spatial distance of the POI location $dis(A, B)$;
3. If $dis(A, B)$>D, then $Sim\_dis = 0$;
4. If $0 \leq dis(A, B) \leq D$, then $Sim\_dis = 1$;
5. From $Sim\_act(U, V) = Sim_F + Sim_S + Sim_R$, find the behavioral similarity $Sim\_act(U, V)$;
6. Superimpose the weights to find user similarity $Sim_{user(A,B)}$;
7. Arrange $Sim_{user(A,B)}$ from largest to smallest, and select $2 * K$ maxima to obtain the user similarity set $\eta$.

### *4.2. Collaborative User Selection Algorithm*

The collaborative user anonymity set obtained by the similarity algorithm in the previous section can effectively avoid the problem of attackers filtering false locations through background knowledge, but this does not yield an ideal anonymity set for the

location distribution. Expanding the size of the anonymization area can lead to better anonymization, but too large an anonymization area can affect the quality of service. Therefore, we filter the anonymized set once to obtain an anonymized region with the near-uniform distribution of location points while ensuring that the anonymized part does not change as much as possible.

Algorithm 1 is used to obtain the user similarity set, in which all users forming set have high user similarity. However, according to the calculation method of users' interest points, it is known that the higher the similarity, the higher the likelihood that the similarity of their demonstrated interest points is also high. These users may be located in close proximity to each other when constructing collaborative anonymous regions, resulting in regional aggregation. Uneven anonymity zones are likely to allow attackers to infer the area where the user's real location is located, resulting in a smaller anonymity zone and exposure of the user's real location, which is something we need to strongly avoid. Therefore, in order to make the selected collaborative user locations as dispersed as possible, the locations need to be homogenized.

The location point homogenization process is shown in Figure 2. According to Algorithm 1, the user similarity set $\eta$ is obtained, which contains $2 * K$ collaborative users with the highest similarity. However, the $2 * K$ collective users may be unevenly distributed, which causes the quality of the anonymous region to decrease. Thus, the privacy protection performance and the ORB algorithm were used in this paper to homogenize the locations of collaborative users.

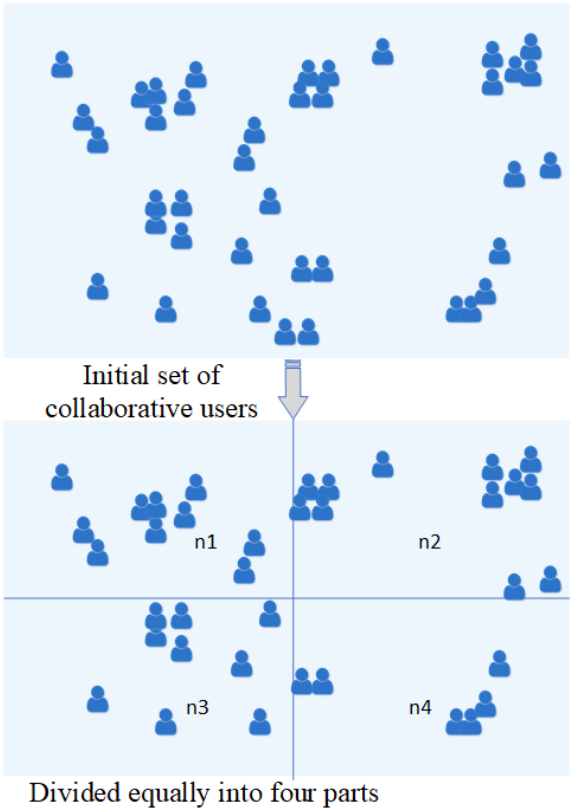


**Figure 2.** Position homogenization process.

As shown in Figure 2, we considered the set of $2 * K$ collaborative users with real locations as points on a square region, which is the root node with value 1 and can be denoted as $node = 1$. Then, the square region composed of $2 * K$ collaborative users is quadratically divided; each region is viewed as a node, and the location point of the collaborative user is an element in the node; at this time, $node = 4$. Continue to quadratically divide the new four nodes; at this time, if the number of elements in the new node obtained

is 0, the node is deleted. If the number of elements in the node is 1, the node is retained and no longer divided. If the number of elements in the node is greater than 1, then continue the quadruple split operation; after n splits, the number of nodes $node = 4n, n \in N^+$. At this time $node \geq K - 1$, stop node splitting. $K$ is the location anonymous parameter, that is the number of anonymous locations that the user expects to include in the location anonymous set.

The obtained *nodes* all have at least 1 element inside, that is they contain at least 1 collaborating user's location point. The location point near the center in each node is selected and put into the collaborative user set $\xi$, and there are $K - 1$ elements in the collaborative user set, that is $K - 1$ similar collaborative users with real locations; the results are shown in Figure 3.

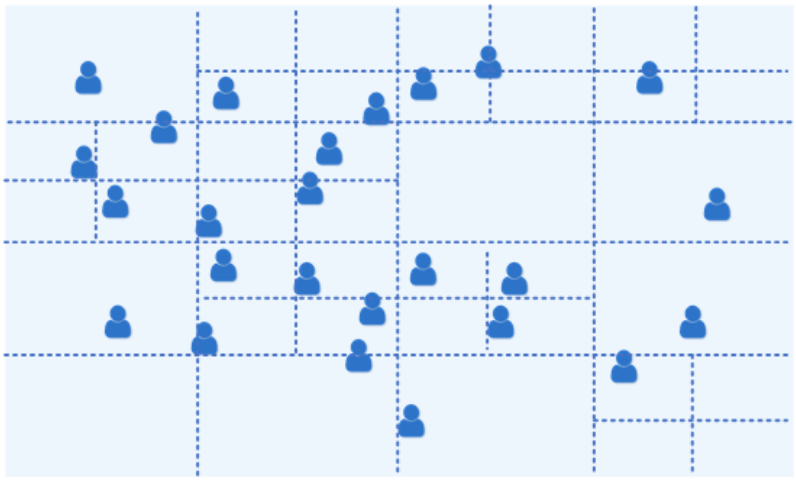

**Figure 3.** Position point homogenization results.

Finally, the obtained $K - 1$ collaborating users' locations and the requesting users' locations are jointly constructed as anonymous regions and sent to the service provider.

In order to test the uniformity effect of position points, we used the Hopkins statistic to judge whether the data are evenly distributed in space. First, $n$ points were found randomly from the location points obtained by the algorithm, denoted as $X_1, X_2, ..., X_n$. For each of these points $X_i$, find the nearest point in the set of locations and calculate the distance between them to obtain the distance vector $x_1, x_2, ..., x_n$. Then, $n$ points are randomly generated from the range of possible values of the anonymous set, denoted as $Y_1, Y_2, ..., Y_n$. For each randomly generated point $Y_n$, find a sample point nearest to it and calculate the distance between them to obtain $y_1, y_2, ..., y_n$. Then, the Hopkins statistic $H$ of the cooperative user set can be expressed as:

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i} \tag{15}$$

As shown in Figure 3, the location points selected by the algorithm are all points near the center of the square region in which they are located, and the randomly generated points within the square region are more likely to be near the center of the region. Therefore, $\sum_{i=1}^{n} x_i \approx \sum_{i=1}^{n} y_i$. The Hopkins statistic $H$ is close to 0.5, indicating that the location points selected by the algorithm are closer to a uniform distribution.

The specific process of the Collaborative User Selection Algorithm (Se-CUA) is described in Algorithm 2.

**Algorithm 2:** Collaborative anonymous user selection algorithm.

**Input:** user similarity set, user real location loc, location K, anonymous parameter $K$
**Output:** optimal collaborative anonymous set:

1. Collaborate the user's location set as the root node, and each location point is an element within that node;
2. Divide the root node into four equal parts, and check the elements in each child node;
3. If the element in node is 0, the child node is deleted;
4. If the element in node is 1, the child node is no longer split;
5. Continue to divide each child node into four equal parts until the number of nodes $m \geq k - 1$;
6. Select the nearest element of each child node to the center to obtain the collaborative user set $\xi$.

## 5. Privacy Analysis and Experimentation

This section tests and verifies the performance and privacy of the algorithm proposed in this paper, respectively, from the aspects of privacy analysis, performance analysis, and privacy verification. The types of attacks that the scheme can resist were proven by analysis, and the privacy of the scheme was proven by the degree of protection of the user's location privacy by privacy measurement.

The experiment was implemented using the python language on a windows11 platform, and the operating environment was based on the PyCharm integrated development environment. The point of interest dataset in the experiment was the geographical dataset of Luolong District, Luoyang City, containing 12,425 POIs. The user social behavior dataset refers to the likes, comments, reposts, and mutual following data among microblog users, including 120,000 microblog dynamic data.

### *5.1. Privacy Analysis*

**Definition 1.** *It is known that $\forall i \in O$ satisfies $P(\widetilde{loc}|loc_i) = \phi$, where the set $O$ is the set of collaboration, $\widetilde{loc}$ is the location information of the user inferred by the attacker through the received data, $loc_i$ is the location information received by the attacker through observation, the set $C$ is the set of location information observed by the attacker, and $P(\widetilde{loc}|loc_i)$ denotes the probability of the attacker inferring the location information through the collected location information.*

**Lemma 1.** *Se-CUA can resist complicity attacks.*

*An attacker may join other users or even location service providers to launch a conspiracy attack to obtain a user's real location. In this paper, the scheme, through the collaboration of $K - 1$ users with high similarity, can find K locations in the set of anonymous locations that are most similar to the requested user's location, thus achieving K-anonymity. The probability of inferring the true location of the requesting user is $1/K$ regardless of how many users the attacker unites.*

*The collaborating users in the algorithm only provide their real locations to participate in the collaboration and do not know the information of other collaborating and requesting users. Anonymous areas are constructed from real locations provided by users that do not contain personal information, and these locations are not confidential and can be easily accessed by anyone. The real location of the requesting user is always hidden in the K anonymous location information, so it is resistant to conspiracy attacks.*

**Definition 2.** *Given $loc_i \in C$, the set $C$ is the location information observed by the attacker, and $C = K$. The attacker inferred the locations as $\widetilde{\mathrm{loc}_a}$, $\widetilde{\mathrm{loc}_b}$. If $P\left(\widetilde{\mathrm{loc}_a} \mid \mathrm{loc}_i\right) = P\left(\widetilde{\mathrm{loc}_b} \mid \mathrm{loc}_i\right)$ is satisfied, this shows that the attacker observes two identical location units obtained because the probability is the same and indistinguishable; if $P\left(\widetilde{\mathrm{loc}_a} \mid \mathrm{loc}_i\right) \neq P\left(\widetilde{\mathrm{loc}_b} \mid \mathrm{loc}_i\right)$ is satisfied, this*

*shows that none of the location units observed by the attacker are identical, and the attacker cannot distinguish them. This indicates that the scheme can resist inference attacks.*

**Lemma 2.** *Se-CUA can resist inference attacks.*

*In location-based services, suppose the attacker is a location service provider who has the most background information about the user. In the Se-CUA algorithm, the anonymous location unit selected based on user similarity satisfies* $P\left(\widetilde{\mathrm{loc}_a} \mid \mathrm{loc}_i\right) \neq P\left(\widetilde{\mathrm{loc}_b} \mid \mathrm{loc}_i\right)$*, and the attacker cannot analyze the real location of the requesting user based on the difference in the background information of the location unit. Therefore, the attacker infers that the true location of the requesting user satisfies* $P\left(\widetilde{\mathrm{loc}_a} \mid \mathrm{loc}_i\right) \neq P\left(\widetilde{\mathrm{loc}_b} \mid \mathrm{loc}_i\right)$ *and is able to resist the inference attack.*

### 5.2. Performance Verification

When requesting users adopt this scheme to protect their location privacy, the selection of collaborating users is decided by the requesting users according to their own situation. Therefore, as the value of privacy-preserving requirement $K$ gradually increases, the average computational delay for its successful construction of the anonymity zone shows an increasing trend, while for collaborating users, who only need to contribute their location, so the required average computational delay does not correlate with the value of $K$. When the requesting user needs a higher privacy protection capability, the requesting user needs to receive more location information from the collaborating user, resulting in a larger communication overhead for the requesting user, while the collaborating user uses peer-to-peer communication, whose communication overhead does not increase with the value of $K$.

As can be seen from Figure 4, the average computational delay and average communication overhead of collaborating users do not change much with $K$ as the privacy protection requirement $K$ increases, and their computational delay and communication overhead are within the acceptable range. As the privacy requirement $K$ increases, the computational latency and communication overhead of the requesting user increase significantly. When the privacy requirement $K$ is 50, the average computational delay of the requesting user is 700.46 ms and the average communication overhead is 43.72 KB, indicating that this scheme has good available lines and can effectively generate anonymous zones for the requesting user to satisfy the requirements. Compared to the MADM algorithm in [17], which uses credit values to find cooperative users, both cooperative and requesting users must maintain the blockchain, which introduces an additional communication overhead. As can be seen in Figure 4b, both requesting and collaborating users in this paper's scheme spend less communication overhead in constructing anonymous regions. However, in terms of average computational latency, this scheme uses more stringent selection criteria for collaborating users, resulting in more computational time for requesting users, but not for collaborating users, and the experimental results are shown in Figure 4a. Therefore, compared to the MADM algorithm, the requesting user in the scheme of this paper requires more computational latency and the collaborating user requires less communication latency.

### 5.3. Privacy Verification

The privacy performance of the Select Collaborative User Algorithm (Se-CUA) was evaluated below in terms of location privacy leakage probability, anonymity entropy, and query accuracy, respectively. The En-2ps algorithm [22] uses real user locations as collaborative locations to construct anonymous sets and uses blockchain to constrain users' behaviors and select users with high trustworthiness as collaborative users. The pati-PPM algorithm [28] selects virtual locations from three aspects based on the existing virtual location algorithm to achieve better privacy protection. Both of the above schemes have excellent performance in their respective directions. In this paper, we compared the above two algorithms to verify the performance of this scheme.

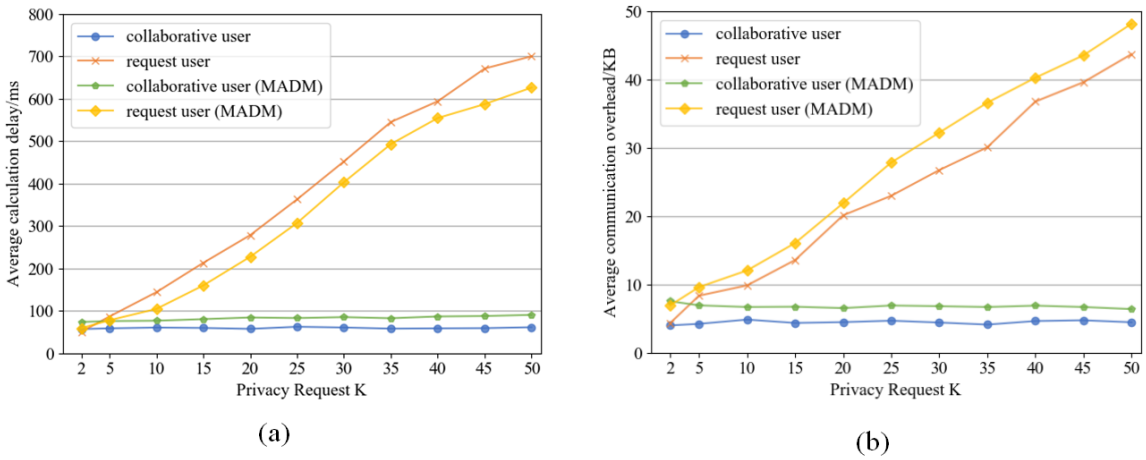


**Figure 4.** Average computational delay and communication overhead. (**a**) Average calculation delay. (**b**) Average communication overhead.

When there is $K$ location information in the anonymization set, then the probability of location leakage is $p = 1/K$. The real location of the user is known to be $i$. After the K-anonymity algorithm to obtain the $K-1$ collaborative user location anonymity set, the anonymity entropy can be expressed as:

$$H_i = -\sum_{i=1}^{k} p_i \log_2 p_i \tag{16}$$

where $p_i$ denotes the probability that the user's true location $j$ is the collaborating user's location $i$. The larger the entropy value, the lower the probability that the user's true location is leaked and the better the privacy protection performance. When the probabilities are the same, the anonymity entropy reaches the maximum value, the uncertainty of the event is the highest, and the privacy protection is the best.

The larger the value of $K$ in the anonymity set, the better the quality of the anonymity region, the more request information the user sends to the LSP, and the smaller the probability that the user's true identity is leaked. The experiments compared the theoretical best value, Spati-PPM algorithm, and En-2ps algorithm as their privacy-preserving performance changes as the parameter $K$ increases. As shown in Figure 5a, when the $K$ value is the smallest, the Se-CUA algorithm does not have an advantage over the En-2ps algorithm due to the ineffectiveness of the virtual location homogenization algorithm in the Se-CUA algorithm. When $K = 6$, the two algorithms have close anonymity entropy. When $K > 6$, the anonymity entropy of the algorithm in this paper is higher than that of the En-2ps algorithm and keeps relatively high privacy-preserving performance afterward. This is due to the fact that, as the value of $K$ increases, the homogenization of virtual locations by the En-2ps algorithm becomes more and more effective, and the location points within its anonymization region are more dispersed, making the anonymization region larger in size and with stronger privacy protection performance.

The En-2ps algorithm also involves the protection of query privacy. The behavior of collaborative users is constrained by blockchain and smart contracts, which ensure the trustworthiness of collaborative users, and the location of their participation in constructing anonymous regions is the real location of collaborative users. From Figure 5b, it can be seen that the privacy leakage probability of the En-2ps algorithm is significantly smaller than that of the Se-CUA algorithm in this paper when $K < 10$. When $K > 10$, as the value of $K$ increases, the Se-CUA algorithm achieves a lower privacy leakage probability because the collaborative user is less likely to be identified using background knowledge. After all, the similarity of the collaborative user is considered, but the difference between the two privacy leakage probabilities is not significant. When $K > 30$, the Se-CUA algorithm has a privacy

leakage probability of less than 10 percent, which protects the location privacy of users more effectively. Figure 5c shows that the Se-CUA algorithm has a higher query accuracy than the other two algorithms, and this advantage becomes more obvious as the value of $K$ increases. This is because the Se-CUA algorithm does not let the anonymous area increase unrestrictedly while ensuring a relatively large anonymous area. The homogenization of collaborating users by the Se-CUA algorithm does not expand the area of the anonymous area, but improves the resistance of the anonymous area to attacks.

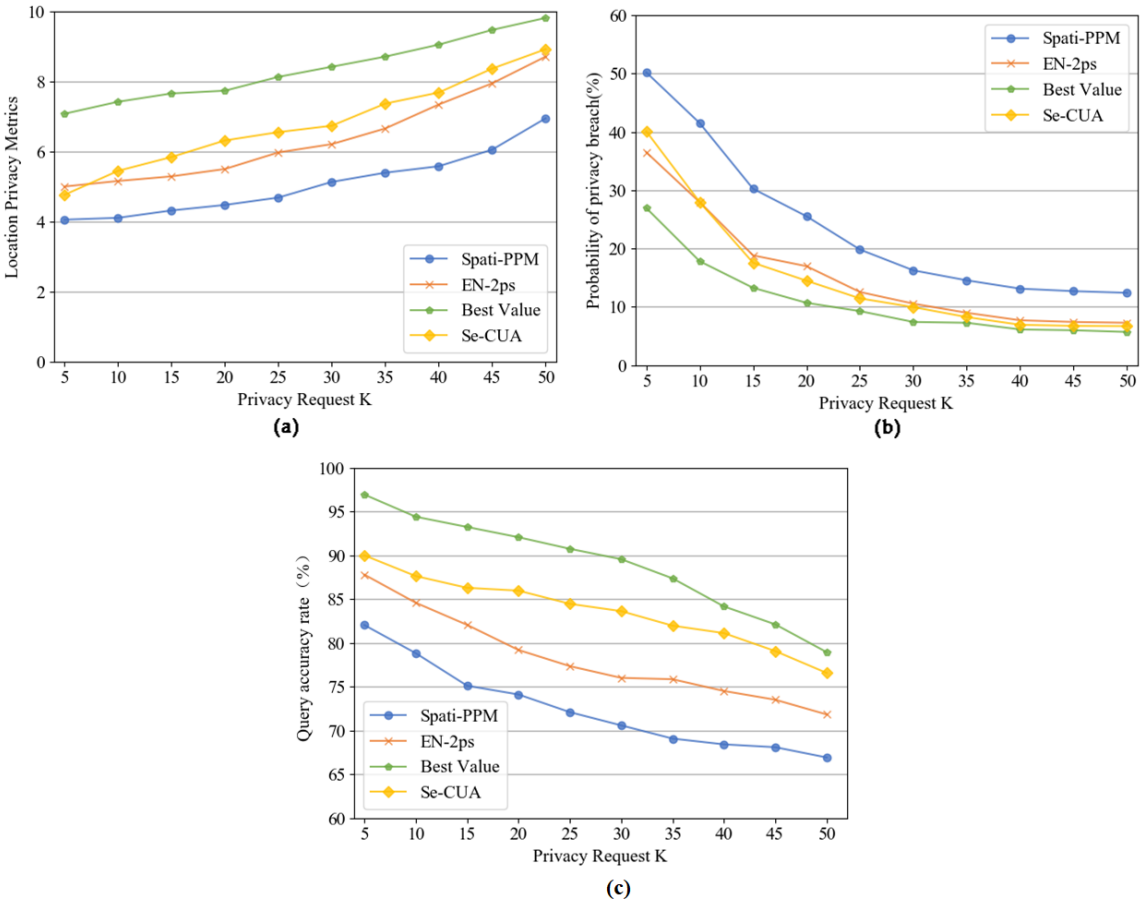


**Figure 5.** Location privacy and leakage probability. (**a**) The graph of the variation of position privacy with parameter $K$. (**b**) Graph of privacy leakage probability with parameter $K$. (**c**) Query accuracy plotted with parameter $K$.

In summary, the proposed algorithm in this paper, while ensuring a lower privacy leakage probability, effectively improves the anonymity entropy. When the privacy protection demand $K < 10$, because the number of collaborative users is small, the area of the anonymity region is small, and there is no advantage compared with other algorithms. When the privacy protection demand $K > 10$, with the increase of the number of collaborative users, collaborators with high user similarity are not easily detected, and the uniformly distributed anonymity region also has a high resistance to background knowledge attacks. The experimental results showed that the algorithm proposed in this paper can protect the location privacy of the requesting users more effectively when the privacy-preserving demand $K > 10$.

## 6. Conclusions

In location-based services, the virtual locations that constitute the anonymity set are easily recognized by attackers through background knowledge, which reduces the anonymity effect. We proposed a collaborative user-based location privacy protection

scheme that considers users' POI and behavioral similarity when selecting collective users to prevent attackers from achieving inference attacks based on background knowledge. In this paper, we first calculated the similarity of users and filtered the suitable collaborative users according to their similarity. To prevent the location of joint users from being too concentrated, we used the location point homogenization algorithm to make the collective users as evenly distributed as possible.

The algorithm proposed in this paper had a better privacy-preserving effect when $K > 10$, but when there were fewer collaborating users in the network, there were even fewer collaborating users with high similarity, and the algorithm had poor privacy-preserving performance. How to improve the privacy-preserving performance of the algorithm in the environment of fewer collaborating users and further reduce the communication overhead are the focus of the next research.

**Author Contributions:** Conceptualization, L.X. and D.Z.; methodology, L.X.; software, D.Z. and X.Z.; validation, H.W., H.M. and X.Z.; formal analysis, L.X.; investigation, H.W.; resources, L.X.; writing—original draft preparation, D.Z.; writing—review and editing, all authors; visualization, H.W.; supervision, H.M.; funding acquisition, L.X., H.W. and H.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is fully supported by the National Natural Science Foundation of China (62071170, 62171180, 62072158, 62272146), the Program for Innovative Research Team at the University of Henan Province (21IRTSTHN015), in part by the Key Science and the Research Program at the University of Henan Province (21A510001), the Henan Province Science Fund for Distinguished Young Scholars (222300420006), the Science and Technology Research Project of Henan Province under Grant 222102210001, and Leading Talent in Scientific and Technological Innovation in Zhongyuan (234200510018).

**Data Availability Statement:** The data used to support the findings of this study was downloaded from https://www.microsoft.com/en-us/download/details.aspx?id=52367 (accessed on 3 July 2022).

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

## References

1. Zhang, T.; Li, X.; Zhang, Q. Location privacy protection: A power allocation approach. *IEEE Trans. Commun.* **2018**, *67*, 748–761. [CrossRef]
2. Ni, W.; Gu, M.; Chen, X. Location privacy-preserving k nearest neighbor query under user's preference. *Knowl.-Based Syst.* **2016**, *103*, 19–27. [CrossRef]
3. Ghinita, G.; Nguyen, K.; Maruseac, M. A secure location-based alert system with tunable privacy-performance trade-off. *GeoInformatica* **2020**, *24*, 951–985. [CrossRef] [PubMed]
4. Li, W.; Li, C.; Geng, Y. APS: Attribute-aware privacy-preserving scheme in location-based services. *Inf. Sci.* **2020**, *527*, 460–476. [CrossRef]
5. Albelaihy, A.; Thayananthan, V. BL0K: A New Stage of Privacy-Preserving Scope for Location-Based Services. *Inf. Sci.* **2019**, *19*, 696. [CrossRef]
6. Luo, H.; Zhang, H.; Long, S.; Lin, Y. Enhancing frequent location privacy-preserving strategy based on geo-Indistinguishability. *Multimed. Tools Appl.* **2021**, *80*, 21823–21841. [CrossRef]
7. Zhao, P.; Huang, Y.; Gao, J.; Xing, L.; Wu, H.; Ma, H. Federated learning-based collaborative authentication protocol for shared data in social IoV. *IEEE Sens. J.* **2022**, *22*, 7385–7398. [CrossRef]
8. Zhao, X.; Pi, D.; Chen, J. Novel trajectory privacy-preserving method based on prefix tree using differential privacy. *Knowl.-Based Syst.* **2020**, *198*, 105940. [CrossRef]
9. Andrés, M.E.; Bordenabe, N.E.; Chatzikokolakis, K.; Palamidessi, C. Geo-indistinguishability: Differential privacy for location-based systems. In Proceedings of the 2013 ACM SIGSAC Conference on Computer, Berlin, Germany, 4–8 November 2013; pp. 901–914.
10. Farouk, F.; Alkady, Y.; Rizk, R. Efficient privacy-preserving scheme for location based services in VANET system. *IEEE Access* **2020**, *8*, 60101–60116. [CrossRef]
11. Li, M.; Salinas, S.; Thapa, A.; Li, P. n-CD: A geometric approach to preserving location privacy in location-based services. In Proceedings of the 2013 IEEE INFOCOM, Turin, Italy, 14–19 April 2013; pp. 3012–3020.
12. Andrew, J.; Eunice, R.J.; Karthikeyan, J. An anonymization-based privacy-preserving data collection protocol for digital health data. *Front. Public Health* **2023**, *11*. [CrossRef]

13. Jia, X.; Xing, L.; Gao, J.; Wu, H. A survey of location privacy preservation in social internet of vehicles. *IEEE Access* **2020**, *8*, 201966–201984. [CrossRef]
14. Wu, Z.; Wang, R.; Li, Q.; Lian, X.; Xu, G.; Chen, E.; Liu, X. A location privacy-preserving system based on query range cover-up or location-based services. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5244–5254. [CrossRef]
15. Gruteser, M.; Grunwald, D. Anonymous usage of location-based services through spatial and temporal cloaking. In Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, San Francisco, CA, USA, 5–8 May 2003; pp. 31–42.
16. Niu, B.; Li, Q.; Zhu, X.; Cao, G.; Li, H. Achieving k-anonymity in privacy-aware location-based services. In Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April 2014–2 May 2014; pp. 754–762.
17. Niu, B.; Zhu, X.; Li, W.; Li, H. Epcloak: An efficient and privacy-preserving spatial cloaking scheme for lbss. In Proceedings of the 2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems, Philadelphia, PA, USA, 28–30 October 2014; pp. 398–406.
18. Yang, D.; Ye, B.; Zhang, W.; Zhou, H.; Qian, X. KLPPS: A k-Anonymous Location Privacy Protection Scheme via Dummies and Stackelberg Game. *Secur. Commun. Netw.* **2021**, *2021*, 9635411. [CrossRef]
19. Niu, B.; Zhu, X.; Li, W.; Li, H.; Wang, Y.; Lu, Z. A personalized two-tier cloaking scheme for privacy-aware location-based services. In Proceedings of the 2015 International Conference on Computing, Networking and Communications (ICNC), Garden Grove, CA, USA, 16–19 February 2015; pp. 94–98.
20. Yu, H.; Zhang, H.; Yu, X. Trajectory similarity calculation method for privacy protection. *J. Commun.* **2022**, *43*, 1–13.
21. Ji, Y.; Zhang, J.; Ma, J.; Yang, C.; Yao, X. BMPLS: Blockchain-based multi-level privacy-preserving location sharing scheme for telecare medical information systems. *J. Med. Syst.* **2018**, *42*, 147. [CrossRef] [PubMed]
22. Liu, H.; Li, X.; Luo, B. Blockchain-based Distributed K Anonymous Location Privacy Protection Solution. *J. Comput. Sci.* **2019**, *42*, 19
23. Yang, M.; Ye, B.; Chen, Y. A trusted de-swinging k-anonymity scheme for location privacy protection. *J. Cloud Comput.* **2022**, *11*, 2. [CrossRef]
24. Parmar, D.; Rao, U.P. Privacy-preserving enhanced dummy-generation technique for location-based services. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7501. [CrossRef]
25. Zhang, S.; Li, X.; Tan, Z.; Peng, T.; Wang, G. A caching and spatial K-anonymity driven privacy enhancement scheme in continuous location-based services. *Future Gener. Comput. Syst.* **2019**, 40–50. [CrossRef]
26. Zhu, L.; Liu, X.; Jing, Z.; Yu, L.; Cai, Z.; Zhang, J. Knowledge-Driven Location Privacy Preserving Scheme for Location-Based Social Networks. *Electronics* **2023**, 70. [CrossRef]
27. Sei, Y.; Andrew, J.; Okumura, H.; Ohsuga, A. Privacy-preserving collaborative data collection and analysis with many missing values. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 2158–2173. [CrossRef]
28. Liu, H.; Li, X.; Li, H.; Ma, J.; Ma, X. Spatiotemporal correlation-aware dummy-based privacy protection scheme for location-based services. In Proceedings of the IEEE INFOCOM 2017—IEEE Conference on Computer Communications, Atlanta, GA, USA; 1–4 May 2017; pp. 1–9.
29. Wang, H.; Wang, C.; Shen, Z.; Liu, K.; Liu, P.; Lin, D. A MADM location privacy protection method based on blockchain. *IEEE Access* **2021**, *9*, 27802–27812. [CrossRef]
30. Levenshtein, V.I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Phys. Doklady* **1966**, *10*, 707–710.