International Journal of
Geo-Information

MDPI

Article

Efficient Location Privacy-Preserving *k*-Anonymity Method Based on the Credible Chain

Hui Wang [1,2], Haiping Huang [1,2,3,4,*], Yuxiang Qin [1,2], Yunqi Wang [1,2] and Min Wu [1,2,3]

[1] School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; hughwangmail@yeah.net (H.W.); 18705192253@163.com (Y.Q.); wangyunqi773@163.com (Y.W.); wumin@njupt.edu.cn (M.W.)

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

[3] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

[4] Institute of Computer Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

* Correspondence: hhp@njupt.edu.cn

Academic Editors: Chi-Hua Chen, Kuen-Rong Lo and Wolfgang Kainz
Received: 9 December 2016; Accepted: 30 May 2017; Published: 1 June 2017

Abstract: Currently, although prevalent location privacy methods based on *k*-anonymizing spatial regions (K-ASRs) can achieve privacy protection by sacrificing the quality of service (QoS), users cannot obtain accurate query results. To address this problem, it proposes a new location privacy-preserving *k*-anonymity method based on the credible chain with two major features. First, the optimal *k* value for the current user is determined according to the user's environment and social attributes. Second, rather than forming an anonymizing spatial region (ASR), the trusted third party (TTP) generates a fake trajectory that contains *k* location nodes based on properties of the credible chain. In addition, location-based services (LBS) queries are conducted based on the trajectory, and privacy level is evaluated by instancing θ privacy. Simulation results and experimental analysis demonstrate the effectiveness and availability of the proposed method. Compared with methods based on ASR, the proposed method guarantees 100% QoS.

Keywords: *k*-anonymity; location-based services; location privacy; the credible chain

1. Introduction

As one of the most important forms of digital information, geographical location data play a critical role in various applications (e.g., smart cities, social networks and intelligent navigation) via big data processing, mobile communications and sensing technologies. Consequently, location-based services (LBS) have become some of the most prevalent tools used in all kinds of Internet of things' applications. Many location applications can be downloaded via the applications market through users' smart phones or tablet computers. With the help of these applications, users can easily obtain location query services and relevant points of interest (POI) returned by a location server. For example, users can query nearby hospitals, restaurants or gas stations.

Location data can disclose private personal information while offering convenience to users; as such, data not only include user location coordinates but also reveal other sensitive personal data such as users' habits, health conditions, and social affiliations [1]. The abuse of location information can considerably compromise user privacy. Several ways to address such issues of location privacy have been proposed over the past few years. Such methods can be divided into two categories: those based on the location privacy-preserving model with the trusted third party (TTP) and those based on the location privacy-preserving model without TTP. The privacy-preserving model without TTP exacerbates communication costs, delays and computational complexity levels and presents obvious problems of usability and stability. The location privacy-preserving model based on TTP is consequently more suitable in use in practical scenarios in combination with a trusted third party service [2]. The model adds a firewall between the user and LBS server and uses location perturbation and obfuscation to achieve privacy protection, and most commonly via *k-anonymity*. To achieve *k-anonymity*, TTP expands the queried location into a broader anonymizing spatial region (ASR) that covers several other users (e.g., other $k-1$ users) geographically. As a result, it is difficult for an untrusted LBS server to determine a user's real location from other $k-1$ dummy locations [3]. However, these approaches based on k-anonymity achieve high-level privacy protection while sacrificing service quality levels. While a broader ASR achieves greater user privacy protection, it occurs at the cost of lower service quality and higher communication and computation costs. Therefore, the trade-off between privacy protection, service quality and resource costs is a major concern with respect to location privacy protection technologies.

This paper proposes a location privacy-preserving method of k-anonymity based on the credible chain. The method should not affect the QoS of LBS, as it adopts a trajectory that protects location privacy rather than constructing an ASR. This paper adopts a classical LBS system architecture based on a central anonymity server (anonymizer) and determines the best k value for a user based on the user's environment and social attributes. To address contradictions between privacy protection levels, service quality and resource costs, it utilizes properties of the credible chain to predict the next state and constructs an illusive location trajectory that contains k locations.

The main contributions of this paper can be summarized as follows:

(1) It proposes a new location privacy-preserving method of k-anonymity based on the credible chain. Rather than forming a k-ASR similar to existing schemes, a TTP forms a fake trajectory that includes k-locations based on properties of the credible chain. It can achieve 100% service accuracy while protecting user location privacy.

(2) A feasible k value selection scheme is proposed as a way to reduce unnecessary communication overhead while guaranteeing user location privacy. The k value is not static and is calculated in terms of a user's current environment and social attributes.

(3) Finally, it conducts privacy metrics by instancing θ privacy to validate the effectiveness and accessibility of this method. Compared with existing schemes, the proposed method achieves superior service performance.

The remainder of the paper is organized as follows. Related works are summarized in Section 2. The system model is introduced in Section 3. Section 4 describes the location privacy-preserving method of k-anonymity based on the credible chain in detail. Section 5 presents privacy metrics that involve instancing θ privacy. Section 6 presents the experimental analysis and performance evaluation. Finally, Section 7 draws conclusions.

2. Related Works

Several ways to ensure location privacy have been proposed [3–17].

As noted above, these approaches can be divided into two categories according to system structures: those related to the location privacy-preserving model based on TTP, which can protect a user's personal information via concealment or confusion, and those based on the location privacy-preserving model without TTP, which can be divided further into collaborative and non-collaborative methods.

For the former, many solutions have been proposed such as methods based on anonymous boxes and data features, for which k-anonymity is now the most widely used tool. Sweeney et al. [3] developed a k-anonymity model as a privacy protection measure for ensuring personal data privacy. In 2003, k-anonymity was first applied to location privacy protection by Gruteser et al. [4]. In [4], location perturbation of k-anonymity method is performed via the quadtree-based algorithm, which adopts spatial and temporal cloaking. However, this approach presents two drawbacks: (1) first, it uses a static k value as a privacy parameter for all mobile users, which likely affects service quality levels for those users whose privacy needs can likely be satisfied using a smaller k value. Furthermore, this assumption is unrealistic, as mobile users tend to have alterable privacy protection needs under different conditions and on different subjects. (2) Second, it can easily generate an excess anonymous region, which not only increases computation costs but also affects the quality of services. To address static k value issues, Gedik et al. [5] proposed the CliqueCloak method, which adopts an individualization k-anonymity model to protect location privacy. However, it only supports small k values (5–10) given its high degree of computational complexity. In [6], Bottom-Up Grid Cloaking and Top-Down Grid Cloaking methods are proposed as ways to form anonymous regions by respectively merging or decomposing grid regions. Bottom-Up Grid Cloaking is used to manage location queries with fewer privacy requirements and Top-Down Grid Cloaking is used to manage location queries with greater privacy requirements. Jagwani et al. [7] proposed a k-anonymity method based on fuzzy spatiotemporal contexts. While this method involves determining the k value to prevent location disclosure and using current fuzzy spatiotemporal attributes to guarantee a more reasonable k value, it does not account for a user′s social attributes such as the correlation degree between identity and location and the associated number for others. To address excess anonymous regions issues, the Casper algorithm [8] was proposed by Mokbel as a way to form an ASR based on [4]. When a user′s number of current quadtree leaf nodes where the request sender is located is less than k, the area of the current leaf node is merged with that of its adjacent sibling node. When a user′s number of anonymous areas is still less than k, the area of the parent of the existing leaf node must be searched for. The Casper algorithm is superior to the algorithm shown in [4], as it reduces computation costs and allows users to more easily control privacy parameters. However, it still suffers from excess anonymous regions and unsatisfactory service quality levels. Yong et al. [9] present a location privacy-preserving k-anonymous method based on service similarities. The location service similarity is introduced to assist anonymity servers in looking for anonymous areas, which not only improves an individual′s need for high-quality information services to some extent (however, not 100%), but also reduces the computation and communication overhead. Niu et al. [10] propose the Dummy-Location Selection (DLS) and enhanced-DLS algorithms. The DLS algorithm carefully selects dummy locations based on the entropy metric, as side information can be exploited by adversaries. The enhanced-DLS algorithm ensures that selected dummy locations are spread out as much as possible, and it can expand the cloaking region while maintaining privacy levels similar to those of the DLS algorithm.

Location privacy-preserving model without TTP consists of an LBS server and several mobile users, and mobile users form a fake location or k-ASR through a cooperative or independent way to meet the anonymous area requirement to achieve location privacy protection. Chow et al. [11] proposed a peer-to-peer spatial cloaking algorithm for anonymous location-based services. The main premise of this algorithm is that before requesting any location-based service, a mobile user must form a group based on his/her neighbor users. A user of the group is then selected to send a service request to the LBS server. The algorithm proposed in [11] has since been improved by Chow [12], who increased the system′s availability by allowing a user to use his/her historical neighbor data and corresponding anonymity levels to achieve location privacy when enough time is available. Zakhary et al. [13] proposed an HSLPO (Social-aware Location-Privacy in Opportunistic mobile social networks) algorithm that can identify a users′ social network and use it to obfuscate service requests and to hide the original sender′s location. The HSLPO algorithm can achieve higher levels of location privacy and service quality than other algorithms in terms of success ratios. However, these methods cannot address inherent defects of the location privacy-preserving model without TTP (i.e., due to high communication or computation costs).

In addition, some new methods combined with anonymous chains were proposed [14–17]. Historical trajectories are used to form anonymous location chains in [14,15] in order to achieve privacy protection. Many existing chain structures have also been introduced into privacy protection, Markov chain is one of the examples. Kang et al. achieved the user′s ID secure authentication with

Markov chain [16]. Montazeri et al. proved that Markov chain, as the result of stochastic process, can be used to simulate the users′ location trajectory, and, meanwhile, it achieves perfect location privacy [17]. However, how to combine the Markov chain with the real world has not been explained in [17]. Different from these methods, this paper introduces the users′ states and social attributes into an anonymization model and form a fake trajectory in real areas.

Most of the existing methods are devoted to protecting the location privacy with ASR, and some others (e.g., the proposal in this paper) try to achieve the win-win situation between the quality of service and the privacy protection. In this paper, it protects the privacy with a group of fake points and makes them up into a fake trajectory. With the help of anonymity algorithms, these fake points are creditable and not easily distinguishable from real points by the attackers. With this method, it can hide the user's true location information and keep the 100% quality of service at the same time.

3. Systems Model

When building a safe LBS model, three factors must be considered:

(1) The quality of service (QoS).
(2) When information (part of the database) disclosure occurs and LBS data are leaked, leaked information can be controlled as little as possible.
(3) When location-based services are taken over by an attacker, the attacker can be misled with false data.
(4) The LBS will adopt the timestamp from the received request messages to provide location services.

When the LBS is contacted without a TTP and information is sent directly, the accuracy of a given service can be ensured. However, this method is less secure when subjected to attacks. When the *k*-anonymous method is used, the real user's location is merely a point in an ASR, and the redundant area inevitably results in inaccuracy and QoS degradation. However, when an attacker takes over an entire LBS, he can likely navigate the ambiguous areas of real location by using a fuzzy user trajectory.

It is true that there are still some disadvantages in the privacy-preserving model with TTP. For example, all of the protections are in vain if the TTP is thoroughly taken over by the attacker. However, taking computation cost and convenience of network management into account, it is believed that the privacy-preserving model with TTP is still the better choice in most real situations compared with that without TTP whose computation completely relies on individuals' devices. Consequently, an assumption needs to be considered in this paper: TTP is trusted and secure. The purpose of this paper is mainly to protect the users' information from the potential security threats in LBS.

One strong solution involves combining a real location with fake locations via a TTP and sending this location data to an LBS. Once a reply from the LBS is received, the TTP can return the correct answer to the user. This can be carried out to ensure QoS and privacy at the same time. However, an experienced attacker can exclude fake points via logical plausibility analysis. For example, one cannot travel around a city in 10 min or stand in the middle of a lake. After applying such exclusions, user location data are probably exposed to the attacker. Therefore, the key to the success of this method involves generating 'trusted points' and forming a 'trusted trajectory'.

To apply this idea, it generates 'trusted points' and transforms a user's location into a fake trajectory rather than a dummy region. As is shown in Figure 1, a *true* node is a user′s true location node, and the *fake*1, *fake*2, *fake*3, . . . *fake*$(k-1)$ nodes are selected from a cloud server to form a fake trajectory. A cloud server is a server from a TTP that stores previous requests or realistic points and that can ensure that fake points are located in viable areas rather than in locations that a request cannot cover. A trajectory is a sequence of moving object location data sorted by time. Hence, the anonymity server must change the timing of nodes to allow the trajectory to confuse and distract attackers. In Figure 1, the *fake*1 node starts the trajectory followed by the *fake*2 node via the *true* node and finally the *fake*$(k-1)$ node. The resulting user position accuracy is superior to that of an ASR, and a user can achieve higher QoS while ensuring his/her location privacy.

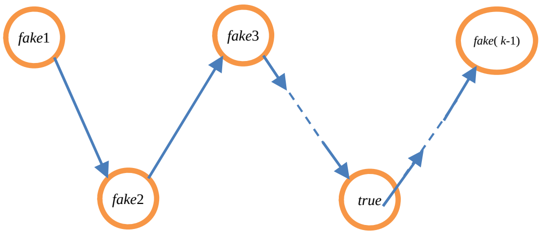


Figure 1. Diagrammatic sketch of the trajectory.

Based on Figure 1, it adopts the system structure shown in Figure 2. When a user sends a service request, the client determines the k value that meets the user's anonymous needs according to the user's environment and social attributes, and then the client sends this k value to the anonymity server. The anonymity server then obtains $k-1$ fake nodes by communicating with the cloud server, and $k-1$ fake nodes and the true node are constructed into a fake trajectory. Finally, the LBS server carries out inquiry processing for nodes of the fake trajectory in order.

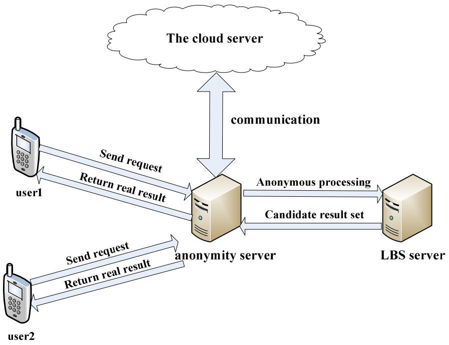


Figure 2. System structure.

4. *k*-Anonymity Method Based on the Credible Chain

Based on the above system model, it proposes a k-anonymous location privacy protection method based on the credible chain.

4.1. Preliminary Knowledge

Definition 1. *(Request message Q) the requested message Q can be expressed as a six tuple:*

$$Q = \{id, loc, t, qry, k, s\},$$

where id is the identity information of the user who sends the request; loc is the user's location, which can be directly obtained from a Global Position System (GPS) or using other positioning devices; t denotes the time at which the user sends the request; qry is the content that the user wants to request; k is the anonymous parameter of the location privacy level, which can be determined from the system; and s is the anonymous region where the user located. For example, if the user is in Shanghai, the history points, which will be chosen in following sections, can only be in the same city.

Definition 2. *(The credible chain) Let $\{X(t), t \geq 0\}$ denote a discrete time process taking values in state space $I = \{0,1, 2, \ldots\}$. For $0 \leq t_1 < t_2 < \ldots < t_{n+1}$ and $i_1, i_2, \ldots, i_{n+1} \in I$, $P\{X(t_{n+1}) = i_{n+1} \mid X(t_1) = i_1, X(t_2) = i_2, \ldots, X(t_n) = i_n\} = P\{X(t_{n+1}) = i_{n+1} \mid X(t_n) = i_n\}$. Note that $\{X(t), t \geq 0\}$ is defined as the credible chain.*

When the locations where the user is positioned at the present time t_n and at all past times are known, the location where the user is located in the future t_{n+1} is only related to t_n. In addition, $P\{X(t_{n+1}) = i_{n+1} \mid X(t_n)$

= $i_n\} \neq 0$ denotes that from the location where the user is currently located at time t_n, the user can arrive at the location where he/she will be positioned at time $t_{n+1.}$

There are two advantages to adopting the credible chain:

(1) The unit of a credible chain is only related to the units preceding it. This ensures the uniformity of the entire trajectory and renders it more 'trusted'.

(2) Due to the non-aftereffect properties of the credible chain, it is impossible for an attacker to identify previous points according to leaked data.

Definition 3. *(Trajectory based on the credible chain T) The trajectory is generated by a TTP and includes at least k request messages (containing a user request message), the location of q_i and the next reachable location of q_{i+1} (i = 0,1, . . . ,k − 1) satisfy the inequality $P\{X(t_{i+1}) = q_{i+1} .loc \mid X(t_i) = q_i .loc\} \neq 0$.*

Definition 4. *(Query function R) R (loc) is a function that queries POI according to loc, and it can obtain a sorted set of top-m POIs. Sorting rules can be customized by an LBS (e.g., distance, popularity, rate and quality of service). The Euclidean Distance between the user and POI is adopted as the sorting rule.*

Definition 5. *(Candidate result set W) W denotes the set of all query results searched by the LBS based on location nodes of the trajectory provided by the anonymity server.*

4.2. k Value Selection

k is an important value in this model that represents the anonymous parameter of the location privacy level and that can be calculated through the system. It also denotes the number of points in the fake trajectory.

To further improve location privacy outcomes, the optimal user *k* value must be determined. There are a lot of factors that may influence the disclosure of the users' privacy information. Sometimes, the users' personal requirements need to be considered. According to users' location privacy requirements, investigation and analysis [18], in this paper, the four most prevalent factors are chosen which can be directly analyzed with the TTP's database or set by the users without the help of other data sources or technical tools. It is worth stressing that these factors are not essential to all anonymous scenarios, which can be substituted or added in terms of actual demands and situations, and they should not affect the effectiveness and availability of the algorithm.

(1) Density of the anonymous area

The density of the anonymous area (crowdedness) has a strong effect on location privacy. While individuals do not wish to expose themselves to less crowded areas, they may feel relatively safe in crowded areas. The less the density of the anonymous area is, the more important the location information is. In this paper, anonymous area density levels can be classified into four categories: sparse, moderately crowded, crowded, and extremely crowded. Such classifications are not fixed and can be altered according to realistic conditions. For example, the number of levels can be three, five, or greater. Level classifications do not affect the validity of the algorithm.

(2) Time interval of one day

Different users have different location privacy needs for different time intervals based on distinguished social attributes. For example, at night, individuals who work during the day usually have greater location privacy needs than those who work at night, and they thus require larger *k* values. Time intervals for a single day can be classified into 4 levels: night, morning, afternoon and evening.

(3) Correlation between identity and location

Users often have different location privacy needs even when they are located in the same area because, in certain environments, their locations are closely related to their identities. When relationships between user identity and location are stronger, smaller *k* values are required. For example, when a teacher queries an LBS on a campus, the value of *k* should be smaller than that used when the teacher is located in a bar. For example, correlation levels can be classified into four levels: irrelevant, low, moderate, and high.

(4) Associations with others

In forming social networks, many users determine their respective social circles. Associations with others denote the extent of a user's social circle. The larger the number of associations becomes, the stronger privacy-preserving demands become. For example, movie stars always keep their locations private, as their fans would bloat their number of associations, thus requiring a larger k value. Here, the number of associations can be divided into four levels: few, some, many and numerous.

The above four factors are considered to strongly influence user location privacy. Linguistic variables are used to describe the influence levels of the four factors. The linguistic variable is typically a fuzzy value (e.g., "few" or "many") rather than an accurate value (0, 1, 2, 3, etc.), thus allowing users to make decisions more intuitively. The number of levels for factors can be adjusted according to the user experience and actual demands of applications. The users have chances to choose their own levels of refinement. Actually, it does not affect the validity of the algorithm. These factors cannot be directly applied due to their varying scales. To eliminate the influence from the scales of different factors, the weight of each factor must be scientifically measured. The value of k will be calculated in turn based on these weights.

When calculating weights, it is necessary to obtain a large number of surveyed samples regarding the factors on a user's location privacy. In order to reflect different users' social attributes, these factors are quantified into specific values X_{ij} (the value of the ith factor of the jth sample). In this case, X_{ij} can be 0, 1, 2, 3, . . . , $n-1$, which denote the number of levels of each factor. Then, average the value of each factor via Equation (1):

$$\overline{X}_i = \frac{\sum_{j=1}^{N} X_{ij}}{N} (i = 1, 2, 3, 4) \tag{1}$$

where N represents the total amount of surveyed samples selected from the database.

The weight of each factor can be standardized via Equation (2):

$$W_i = \frac{\overline{X}_i}{\overline{X}_1 + \overline{X}_2 + \overline{X}_3 + \overline{X}_4} (i = 1, 2, 3, 4) \tag{2}$$

Thus, the weights can be between 0 and 1.

It can calculate a synthetic factor $\sum_{i=1}^{4}(W_i * U_i)$whose value lies in the range [0, $n-1$] due to different emphases from four factors, where U_i denotes the value of the ith factor of the current user. Based on the synthetic factor value, it is important to find the most optimal k value for each user between K_{max} and K_{min}. The anonymous value k (i.e., the number of location nodes in the credible chain) is calculated according to Equation (3):

$$k = \left\lceil \frac{k_{max} - k_{min}}{n} * \left(\sum_{i=1}^{4} (W_i * U_i) - 1 \right) \right\rceil + k_{min}. \tag{3}$$

The lower and upper bound values of anonymous levels are expressed as k_{max} and k_{min}, respectively, and both of them can be set according to the specific situation by the anonymity server. The results in Equation (3) should round upwards to the nearest integer because k is an integer.

In Equation (3), the value of k is directly proportional to the values of the user's attributes. The more sensitive user data is the higher privacy level user needs (the value of k is bigger). However, the specific relationship between k and the factor values is various and alternative. A questionnaire has been conducted to investigate the user's privacy requirements and the corresponding results have been taken into linear regression to prove the feasibility to use linear dependency.

4.3. Anonymous Processing

Anonymous processing is a critical step of the location privacy-preserving model, where a user's location is involved in a fake trajectory based on the credible chain generated via an anonymity server. In this paper, two anonymous parameters (*k* and *s*) can be set according to a user's demands and background knowledge to satisfy his/her personalized location privacy-preserving needs. The procedure described above can be explained via Algorithm 1.

Algorithm 1 Make a fake trajectory

```
Input: The user's request $q$
Output: array P
$q_{1:k-1} \leftarrow k-1$ messages selected by anonymity server
$p_{1:k} \leftarrow$ RANDOMSHUFFLE ($q$,$q_1$, ... ,$q_{k-1}$) // The details of this step are shown in Algorithm 2
// The following steps are just outlines, the details of the following part are shown in Algorithm 3
for $i$ = 2, 3 ... $k$ do
    $T \leftarrow$ NECESSARYTRAVELTIME ($p_{i-1}, p_i$)
    If $T \leq t_{p_i} - t_{p_{i-1}}$
$t_{p_i} \leftarrow t_{p_i} + T + RandomDelay$
End If
End for
Return P
```

The following two algorithms will show the details of some parts of Algorithm 1.

In Algorithm 2, $k-1$ messages will be selected from the server. Then, the true request message will be mixed into them according to the request time.

Algorithm 2 Initial anonymous processing algorithm

```
Input: The anonymity server selects $k-1$ messages from the cloud server according to $s$ generated by the user and then places $k-1$ messages and the user's message $q_0$ into the array Q (namely Q = {$q_0$, $q_1$, $q_2$, ... , $q_{k-1}$}).
Output: array P
$date = q_0.t$ // Assign $q_0.t$ (the time of the user's request message) to the variable $date$.
$i = 1$
$\Delta T$ = max ($q_0$-$q_i$), $i \in [1, k-1]$
while $i \leq k-1$ do // Update the $k-1$ message selected from the cloud server in turn.
    $q_i.id = q_0.id$
    $q_i.k = q_0.k$
    $q_i.s = q_0.s$
    if $qi.t \notin (date -$ random($\Delta T$), $date$) then
        $qi.t = date$ + random ($\Delta T$)
    end if // Substitute $date$ + random($\Delta T$) for $qi.t$ while $qi.t$ does not fall within the range of ($date -$ random($\Delta T$), $date$). The function of random($\Delta T$) is used to generate a random number in the range of (0, $\Delta T$), and the random number is retained to one decimal place.
    $i = i + 1$
end while
P = Sort ({Q $-$ $q_i$}) // Place these messages from the array Q into the other array P after sorting according to the value of $t$.
Return P
```

In the process of selection, the region s can be divided into k sub-regions with the same size. For each sub-region where the current user is not located, it picks up the most inactive (i.e., the least frequently used recently) historical request message from the sub-region. If there is no inactive message in this sub-region, then one from the nearest sub-region can be borrowed. Thus, it can finally select $k - 1$ fake messages as the input of Algorithm 2.

In this algorithm, messages that have been requested before $date$ − random (ΔT) will be put after the true request message and their request time will be reset as $date$ + random (ΔT). By this means, the true point can be mixed with those fake points. However, the request time of true point is the current time, and the attacker can recognize it easily. The fake points may not be accessible at the current speed within the current time interval, and they can be easily excluded by the attacker. Algorithm 3 is used to solve all of these problems.

Each point in this trajectory has a region that it can access with its current speed in the current time interval. The transition probability for each point in this accessible region is equal, while the transition probability out of this region is zero. In this paper, all of the fake points are historical points chosen from the cloud service. The transition probability of each point in the trajectory is $1/M$, M is the number of the historical points within the current point's accessible region (a round area with the radius of $S_{i,i+1}/v_{i,i+1}$). Any point of this trajectory is inaccessible if its last point's transition probability is 0. In this case, it needs to delay the request time of the current point to expand its last point's accessible region and make the current point accessible. The detailed steps are described as the following.

Algorithm 3 The credible chain algorithm

```
Input: array P = {$p_0, p_1, p_2, \ldots, p_{k-1}$}, $date$
Output: the user's trajectory T based on the credible chain
T = {$p_0$} // Initialization should be completed before the credible chain is formed, and $p_0$ is placed into the trajectory T.
if $p_0.t = date$
flag = 0 // When the message $p_0$ is exactly the user's message, flag is set to 0. This denotes that the user's message has been added to the credible chain.
else flag = 1
end if
$\Delta = 0$; $i = 0$ // $\Delta$ is the interval between the time of the user's request message after anonymity and the real time of the user's request message.
while $i \leq k - 2$ do
    if $\Delta t = S_{i,i+1}/v_{i,i+1} > (p_{i+1}.t - p_i.t)$ do // $p_i.loc$ cannot arrive at $p_{i+1}.loc$.
        if $p_{i+1}.t - \Delta = date$ && flag then // Judge whether $p_{i+1}$ is the user's message.
            $\Delta = p_i.t + \Delta t - date$
        end if
        $p_{i+1}.t = p_i.t + \Delta t$ // $p_{i+1}.t$ is updated to guarantee that $p_i.loc$ can arrive at $p_{i+1}.loc$.
    end if
        add ($p_{i+1}$) // $p_i.loc$ can arrive at $p_{i+1}.loc$, illustrating that $P\{X(t_{i+1}) = q_{i+1}.loc \mid X(t_i) = q_i .loc\} \neq 0$. Hence, $p_{i+1}$ should be added to the trajectory T.
        delete ($p_{i+1}$) // $p_{i+1}$ should be removed from the array P.
        when $p_{i+1}.t - \Delta = date$, then // Determine whether $p_{i+1}$ is the user's message.
            flag = 0; // if the user's message has been added to the credible chain, and flag should be set to 0.
    end if
    $i = i + 1$
end while
Return T
```

Algorithm 3 is executed after Algorithm 2, and the input for Algorithm 3 is obtained from the result of Algorithm 2. Initialization needs be carried out before the credible chain is formed. The critical step involves determining whether $\Delta t = S_{i,i+1}/v_{i,i+1} > (p_{i+1}.t - p_i.t)$ (where $S_{i,i+1}$ is the distance between $q_i.loc$ and $q_{i+1}.loc$ and $v_{i,i+1}$ is the maximum average speed at which the user arrives at $q_{i+1}.loc$ from $q_i.loc$. This can be fabricated based on actual conditions.). Algorithm 3 adopts a one-pass approach that can decrease the memory complexity while dealing with quantities of data. Furthermore, to avoid a time span that is too long, Algorithm 3 will check whether the node needs additional time (whether it is reachable with the current situation).

The request time of each point in the trajectory is related to all of its previous points. Figure 3 displays how to change the request time according to Algorithms 2 and 3 when the value of k is set as 5. At the phase of initialization, t_1, t_2, t_3 and t_4 are the request times of fake points while t_0 is the request time of true points. Algorithm 2 forms the trajectory by reorganizing the request time as $t_3{}'$, $t_4{}'$, $t_0{}'$, $t_1{}'$ and $t_2{}'$. If the point of $t_3{}'$ cannot arrive at that of $t_4{}'$ within the time interval between $t_3{}'$ and $t_4{}'$, then it delays $t_4{}'$ to $t_4{}''$. However, the time interval $t_0{}' - t_4{}''$ will be changed due to the delay. It needs to check whether it is accessible from the point of $t_4{}''$ to that of $t_0{}'$.

Therefore, if the request time of any point ahead of the true point is changed, the request time of the true point may be changed due to Algorithm 3, and it may not be the current time anymore. Even if one point's request time is still the current time, the attacker cannot ensure whether it is a coincidence (e.g., $t_4{}''$ = $t_0{}'$).

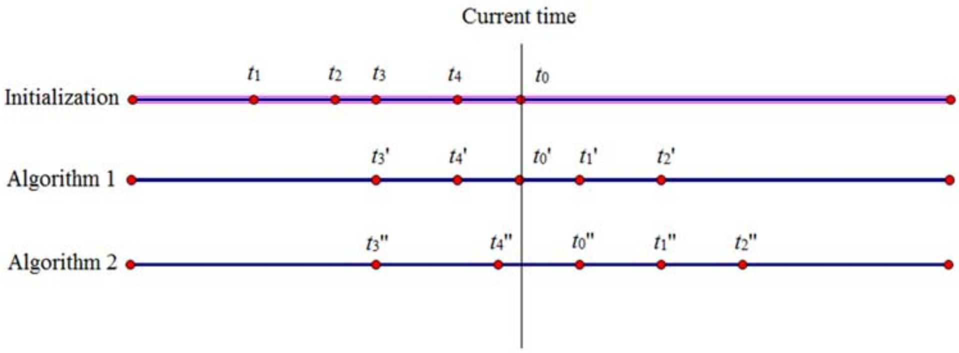


Figure 3. The paradigm of request time change according to Algorithms 2 and 3.

4.4. Inquiry Processing

The anonymity server sends a query request involved in trajectory T, and then the LBS server responds to the request. The LBS server identifies query results by traversing all message nodes in T and then adds the results to a candidate result set W in order. Finally, W is returned to the anonymity server. The final result of this process is $W = \bigcup_{i=0}^{k-1} R(p_i.loc)$.

In the above process, the LBS server traverses all message nodes in T, and W is the union of all message query results after T is received from the anonymity server. When the anonymity server receives W from the LBS server, TTP will find out the answer to the user's original request by selecting that the reply from W whose t before modification matches the user's request time. Finally, the user's real message node will be returned to the user.

5. Privacy Metrics

In this section, it proposes a privacy measure mechanism for evaluating the efficiency of privacy protection in the service system presented. For guaranteeing the availability and effectiveness of the proposed scheme, θ privacy instancing [19] will be adopted.

It uses a series of true user historical location data including position coordinates loc and time t, which have been collected by an attacker before the user's current location node is attacked. The attacker can thus probe all message nodes in the credible chain to determine their authenticity according to the latest user location data collected. This is defined as follows:

$$P\{\mathrm{U} \mid \mathrm{L}_t\} - P\{\mathrm{U}\} \leq \theta. \tag{4}$$

In Equation (4), θ denotes the degree of location privacy protection, and, meanwhile, it can be defined as the difference degree of the attack effect between an attacker with background knowledge and someone without background knowledge; $P\{\mathrm{U}\,|\,\mathrm{L}_t\}$ denotes the posterior probability that an attacker will infer the user's real location in the current credible chain on the premise that he collects these location data before the moment t; $P\{\mathrm{U}\}$ denotes the priori probability that an attacker will infer the user's real location in the current credible chain. Assume that the number of inaccessible nodes excluded by the attacker is α, that the total number of nodes is k, the value of $P\{\mathrm{U}\,|\,\mathrm{L}_t\}$ is $1/(k-\alpha)$ and the value of $P\{\mathrm{U}\}$ is $1/k$. Consequently, Equation (4) can also be substituted as Equation (5):

$$\frac{1}{k-\alpha} - \frac{1}{k} \leq \theta \tag{5}$$

Assume that the sequence of all location nodes in the credible chain is $p_1, p_2 \ldots p_i \ldots p_k$ and that the latest user location node information collected by the attacker is p_0. The method for calculating the value of θ is described as follows:

Step 1: Judge whether inequality $(p_i.loc - p_0.loc)/v_{0,i} \leq (p_i.t - p_0.t)$ is established. Inequality denotes whether from the location of p_0 the user can arrive at the location of p_i. In this case, proceed to step 2; otherwise, proceed to step 3.

Step 2: Inequality is established, denoting that node p_i may be the user's true node. The value of α is then put into Equation (5) to determine the value of θ.

Step 3: Inequality is not established, denoting that the node p_i is a dummy node. Then, α++ and i++. Proceed to step 1 to continue the calculation.

As is shown in Figure 4, it assumes that k is 4 and that a credible chain $\{p_1, p_2, p_3, p_4\}$ has been constructed by the anonymity server. An attacker collects a series of true user historical location data, and the latest user location data is p_0. First, it determines whether inequality $(p_i.loc - p_0.loc)/v_{0,i} \leq (p_i.t - p_0.t)$ is established according to the above steps in order to calculate the value of θ. As location p_0 cannot arrive at location p_1, p_1 is a dummy one and the value of α is added to 1. It then determines whether location p_0 can arrive at location p_2. As location p_0 cannot arrive at location p_2, the value of α is 2. Subsequently, it determines whether location p_0 can arrive at location p_3. As location p_0 can arrive at location p_3 and location p_3 can arrive at location p_4, p_3 and p_4 may be the true nodes of the user. Finally, it can derive the value of θ as $\frac{1}{4}$.

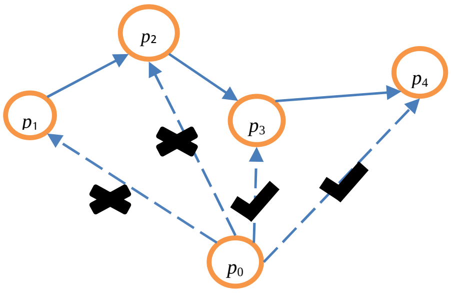


Figure 4. Calculating the value of θ.

6. Experimental Analysis

In this section, the performance of the proposed location privacy-preserving method will be evaluated from three aspects using MATLAB: the degree of anonymity, θ privacy and the quality of service (QoS).

6.1. Degree of Anonymity Analysis

In this paper, the degree of anonymity is determined based on the value of k. It uses a series of data to simulate four weights of influencing factors that can be set as $W_1 = 0.16$, $W_2 = 0.15$, $W_3 = 0.4$, and $W_4 = 0.29$. It samples some location requests from different identities and locations, which are partly shown but not limited in Table 1.

Table 1. The data on the user's environment.

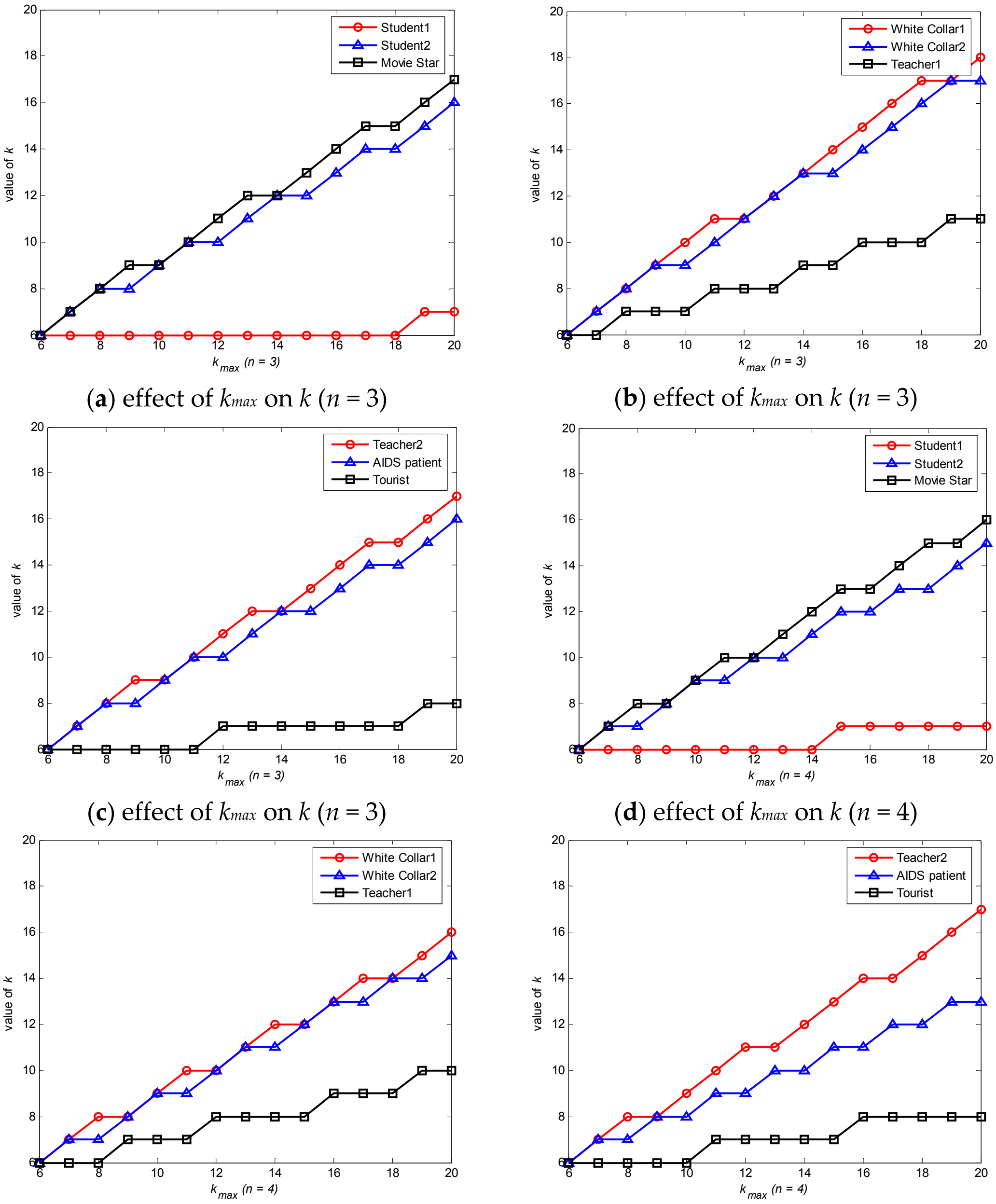
User	Location	Density	Time Interval	Correlation Degree	Associated Number
Student 1	canteen	crowded	morning	high	few
Student 2	hospital	crowded	afternoon	low	numerous
AIDS patient	hospital	crowded	morning	high	numerous
White collar 1	home	sparse	morning	high	numerous
White collar 2	road	sparse	night	irrelevant	numerous
Movie Star	market	extremely crowded	evening	low	numerous
Teacher 1	campus	moderately crowded	afternoon	high	many
Teacher 2	bar	crowded	night	irrelevant	many
Tourist	scenic area	crowded	morning	irrelevant	some

Users have different privacy needs due to their different identities and environments. To obtain a better k value, it analyzes the effects of k_{max} and n on the selection of k using the data shown in Table 1.

Assume that k_{min} is 5 for ease of analysis. According to Figure 5, it can conclude that the value of anonymity degree k should increase as the maximum of the anonymity degree k_{max} increases. This means that the location privacy-preserving method can be used to determine a reasonable k_{max} value to obtain a suitable k according to different anonymity needs while further enhancing the protection of real location data. The value of anonymity degree k should likely decrease or remain unchanged while influencing factor n (the number of levels) grows. When influencing factors cannot be classified specifically, privacy requirements cannot be comprehensively determined or analyzed, and more location messages are needed to complete the anonymity process. It can marginally reduce the number of location messages used in the anonymity process while each influencing factor is accurately classified and measured. It is beneficial to reduce time costs, improve efficiency and protect real location data when forming the credible chain.

Figure 6 displays the relationship between the anonymity degree k, the maximum anonymity degree k_{max} and influencing factors n tri-dimensionally.

As is shown in Figure 6, the value of k will gradually increase as k_{max} gradually increases or as n gradually decreases. This means that more location data must be used in the anonymity process, and, accordingly, communication costs should increase. Thus, the value of k must be set within a reasonable range to limit unnecessary communication overhead; meanwhile, fine-grained classifications of influencing factors and reasonable k_{max} values are needed.

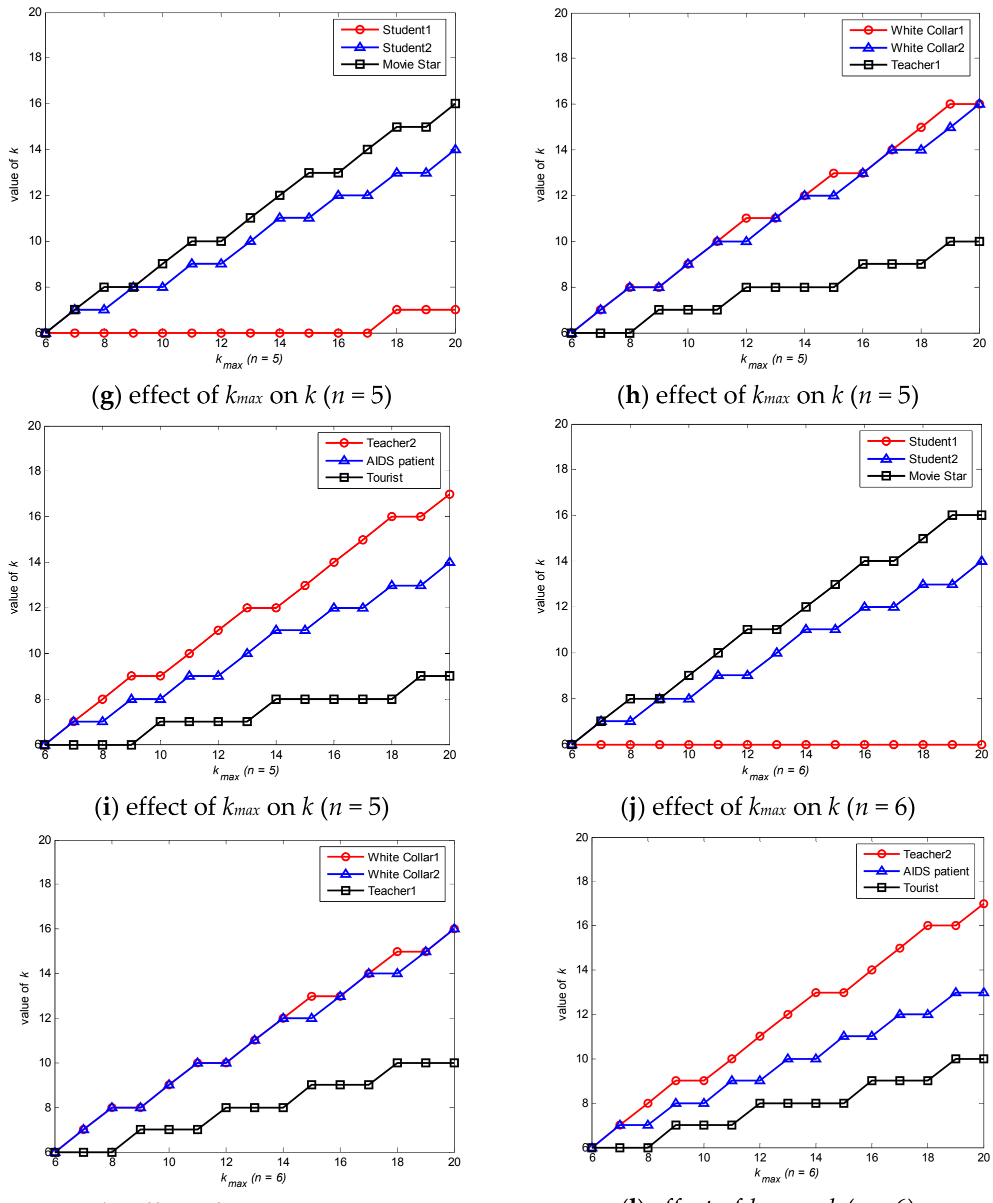


(**a**) effect of k_{max} on k (n = 3) (**b**) effect of k_{max} on k (n = 3)

(**c**) effect of k_{max} on k (n = 3) (**d**) effect of k_{max} on k (n = 4)

(**e**) effect of k_{max} on k (n = 4) (**f**) effect of k_{max} on k (n = 4)

Figure 5. *Cont.*



(**g**) effect of k_{max} on k (n = 5)

(**h**) effect of k_{max} on k (n = 5)

(**i**) effect of k_{max} on k (n = 5)

(**j**) effect of k_{max} on k (n = 6)

(**k**) effect of k_{max} on k (n = 6)

(**l**) effect of k_{max} on k (n = 6)

Figure 5. Effect of k_{max} and n on k (**a**–**l**).

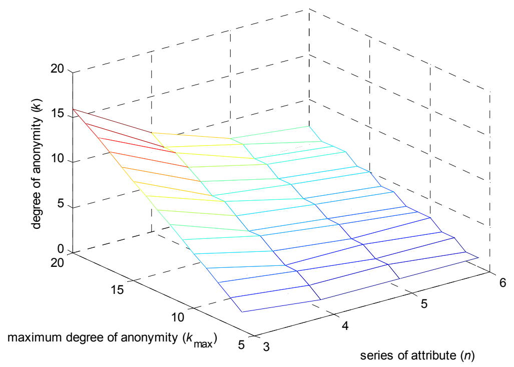


Figure 6. The relationship between k, k_{max} and n.

6.2. θ Privacy Analysis

It uses θ to measure the privacy level. The smaller θ is, the higher the user's location privacy level becomes. In Equation (5), α denotes the number of inaccessible nodes excluded by the attacker according to the user's previous locations. Different values of α in a credible chain denote that the attacker has different background knowledge.

According to Figure 7, as the value of α increases, the value of θ also increases. This shows that when the attacker has more background information, the user's degree of location privacy decreases and protection costs increase, as more fake nodes must be added to the chain. When the value of k increases gradually, the value of θ gradually tends toward 0, which denotes perfect privacy.

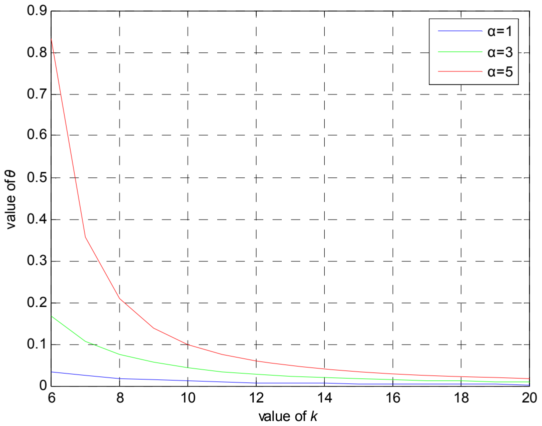


Figure 7. The relationship between k and θ under different background knowledge conditions.

Figure 8 denotes the relationship between k, the number of excluded fake nodes α, and the value of θ privacy.

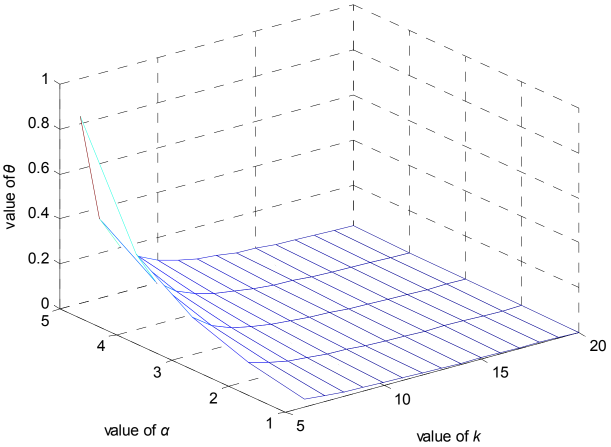


Figure 8. The relationship between k, α, and θ privacy.

According to Figure 8, when the number of excluded fake locations is constant, the larger the value of k is, the smaller the value of θ becomes and the better the degree of privacy protection becomes. When k is constant, the larger the number of excluded fake locations is, the larger the value of θ becomes and the worse the degree of protection becomes.

Assume that the user's current position in the trajectory is x, and the attacker has maximum attack capacities. For example, in Figure 9, A-O-B is a fake trajectory (k = 8 in this figure), and P-H-O is the user's true trajectory. Therefore, the attacker holds all of the request messages ahead of O in the true trajectory. It can only ensure that, in A-O-B, for any two adjacent nodes, the previous one to the next one is accessible. Thus, P-O-B is also accessible between any two nodes. It is not certain whether it is accessible from H to any point between A and O (for example, Q). The worst situation is that all points between A and O are excluded. In this case, $\alpha = 4$ ($\alpha = x - 1$).

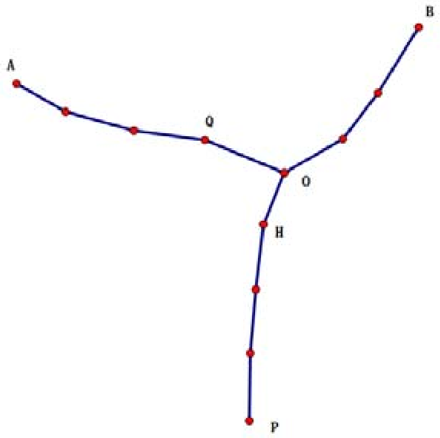


Figure 9. True trajectory P-H-O and fake trajectory A-O-B.

After these analyses, the expected values of α and θ can be figured out, respectively.

Under normal circumstances, the value of α can vary from 0 to $(x-1)$ with equal probability. Therefore, the following equation regarding α can be derived:

$$E(\alpha) = \sum_{i=0}^{x-1} \frac{i}{x} = \frac{x-1}{2} \tag{6}$$

It can also calculate the expected value of θ by listing all valid combinations of α and x.

The value of x can vary from 1 to k with equal probability. Under this premise, the value of α can vary from 0 to $(x-1)$ with equal probability. Therefore:

$$\begin{aligned} E(\theta) &= \frac{1}{k}\sum_{x=1}^{k}\left\{\frac{1}{x}\sum_{\alpha=0}^{x-1}[\theta(\alpha,k)]\right\} \\ &= \frac{1}{k}\sum_{x=1}^{k}\left\{\frac{1}{x}\sum_{\alpha=0}^{x-1}\left[\frac{1}{k-\alpha}-\frac{1}{k}\right]\right\} \\ &= \frac{1}{k}\sum_{x=1}^{k}\left\{\frac{1}{x}\left[H_k - H_{k-x} - \frac{x}{k}\right]\right\} \\ &= \frac{1}{k}\sum_{x=1}^{k}\left\{\frac{H_k - H_{k-x}}{x} - \frac{1}{k}\right\} \\ &= \frac{1}{k}\left\{\sum_{x=1}^{k}\left[\frac{H_k - H_{k-x}}{x}\right] - 1\right\} \end{aligned} \tag{7}$$

where H_n means the n-th harmonic number ($H_0 := 0$).

With this result, the relationship between θ and k can be depicted in Figure 10, where $k = 1$ means that TTP sends the true point to LBS, while $k = 2$ means that there are only one fake point and one true point in the fake trajectory. These two situations will be excluded in reality. Therefore, it can be concluded that the bigger k is, the smaller $E(\theta)$ and the higher the privacy-preserving level of the user data will be.

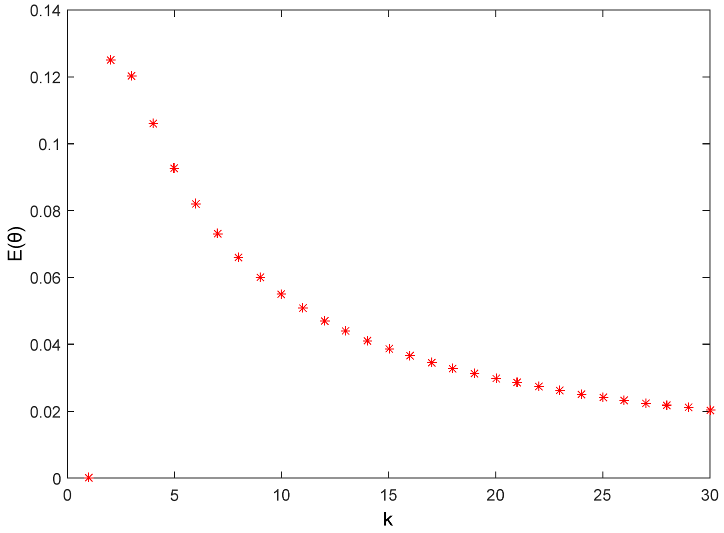


Figure 10. The relationship between k and $E(\theta)$.

6.3. Quality of Service (QoS) Analysis

The following representative anonymous methods are used for comparisons in this section: the quadtree-based [2], Casper [6], service similarity [7] and enhanced-DLS algorithms [8]. Simulation experiments are conducted under the same conditions to compare service accuracy. The service accuracy of an anonymity server can be denoted as $C = \frac{W_{true}}{W}$, denoting the ratio between the valid number of queries and the total value. An increase in C indicates that the accuracy of query results has improved. When $C = 1$, all feedback results are correct. Moreover, 100 random queries are simulated in each algorithm.

According to Figure 11, the proposed method always achieves a service accuracy level of 1 (or 100%), while values achieved by other anonymous methods decline as the value of anonymity degree k increases. This is because the proposed method forms a credible chain based on a user's real location and several fake locations, ensuring that the positional accuracy levels are never reduced. The other methods form ASRs based on k user locations, which decreases positional accuracy levels. From Figure 11, it can be concluded that the proposed method does not suffer from the service accuracy limitations of existing algorithms based on ASRs.

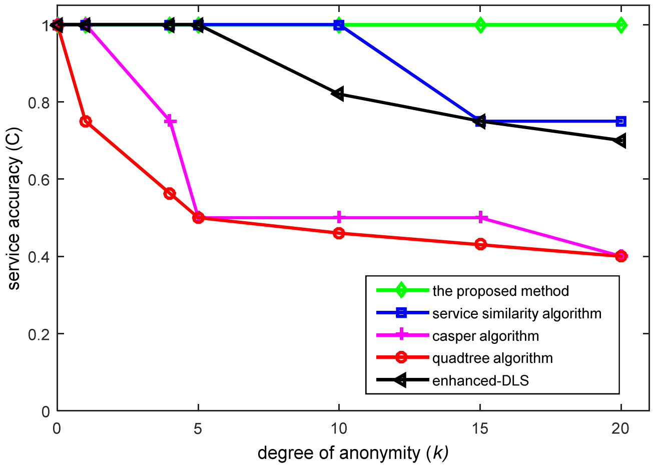


Figure 11. The relationship between k and service accuracy.

7. Conclusions

To address the issue that privacy levels are improved by sacrificing the quality of service in current location privacy-preserving mechanisms, it proposes a location privacy-preserving *k*-anonymity method based on the credible chain. The method involves utilizing properties of the credible chain and forming a fake trajectory of *k* location nodes by changing their timing. It also optimizes the value of *k* and renders it suited to current users' environments and social attributes, thus reducing communication overhead. Furthermore, privacy metrics is suggested by instancing θ privacy. The experimental analysis results show that the proposed method is more effective at addressing contradictions between service accuracy and location privacy. All of the parameters in this paper are extensible and can be changed according to actual requirements. However, the significance of this algorithm is protecting the database and preventing user data from being sold by the LBS providers. If the attacker takes over the LBS and launches a real-time and well-planned attack, the proposed method will degenerate. Future work will involve using big data techniques to analyze and process location data to further improve the effectiveness of location anonymization measures. It is also important to find a suitable method to deal with the real-time and well-planned attack and avoid algorithm degradation. At the same time, the security of third parties and more reasonable *k* value selection methods also need further investigation.

Acknowledgments: This work was supported by the National Natural Science Foundation of the People's Republic of China (Nos. 61672297, 61572260 and 61373138), the Key Research and Development Program of Jiangsu Province (Social Development Program, No. BE2016185), the Postdoctoral Foundation (Nos. 2015M570468 and 2016T90485), the Sixth Talent Peaks Project of Jiangsu Province (No. DZXX-017), the Jiangsu Natural Science Foundation for Excellent Young Scholars (No. BK20160089), and the Fund of the Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks (WSNLBZY201516).

Author Contributions: H.W., H.H. and Y.Q. conceived and designed the experiments; H.W. and Y.Q. performed the experiments; Y.W. and M.W. analyzed the data; H.W. and H.H. wrote the paper.

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

References

1. Lu, W.; Feng, M.X. Location privacy preservation in big data era: A survey. *J. Softw.* **2014**, *4*, 693–712.
2. Jia, J.; Zhang, F. Non-deterministic k-anonymity algorithm based untrusted third party for location privacy protection in LBS. *Int. J. Secur. Appl.* **2015**, *9*, 387–400. [CrossRef]
3. Sweeney, L. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2002**, *5*, 557–570. [CrossRef]
4. Gruteser, M.; Grunwal, 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 (MobiSys'03), San Francisco, CA, USA, 5–8 May 2003; pp. 163–168.
5. Gedik, B.; Liu, L. A customizable k-anonymity model for protecting location privacy. In Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05), Columbus, OH, USA, 6–10 June 2005; pp. 620–629.
6. Bamba, B.; Liu, L.; Pesti, P.; Wang, T. Supporting anonymous location queries in mobile environments with privacy grid. In Proceedings of the 17th International World Wide Web Conference (WWW'08), Beijing, China, 21–25 April 2008; pp. 237–246.
7. Jagwani, P.; Kaushik, S. K-anonymity based on fuzzy spatio-temporal context. In Proceedings of the 2014 IEEE 15th International Conference on Mobile Data Management (MDM'14), Brisbane, Australia, 14–18 July 2014; pp. 15–18.
8. Mokbel, M.F.; Chow, C.Y.; Aref, W.G. Casper: Query processing for location services without compromising privacy. *ACM Trans. Database Syst.* **2016**, *4*, 24–48.
9. Yong, Y.A.; Cheng, L.Y.; Feng, M.J.; Li, X. Location privacy-preserving method of k-anonymous based on service similarity. *J. Commun.* **2014**, *11*, 162–169.

10. Niu, B.; Li, Q.H.; Zhu, X.Y.; Cao, G.H.; Li, H. Achieving k-anonymity in privacy-aware location-based services. In Proceedings of the 33rd IEEE International Conference on Computer Communications (INFOCOM'14), Toronto, ON, Canada, 27 April–2 May 2014; pp. 754–762.
11. Chow, C.Y.; Mokbel, M.F.; Li, X. A peer-to-peer spatial cloaking algorithm for anonymous location-based services. In Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems (GIS' 06), Virginia, VA, USA, 10–11 November 2006; pp. 171–178.
12. Chow, C.Y.; Mokbel, M.F.; Li, X. Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. *Geoinformatica* **2011**, *2*, 351–380. [CrossRef]
13. Zakhary, S.; Radenkovic, M.; Benslimane, A. The quest for location-privacy in opportunistic mobile social networks. In Proceedings of the 2013 IEEE 9th International Conference on Wireless Communications and Mobile Computing (IWCMC'13), Cagliari, Italy, 1–5 July 2013; pp. 667–673.
14. Gheorghita, M.O.; Solanas, A.; Forne, J. Location privacy in chain-based protocols for location-based services. In Proceedings of the 2008 IEEE 3rd International Conference on Digital Telecommunications (ICDT'08), Bucharest, Romania, 29 June–5 July 2008; pp. 64–69.
15. Cao, L.; Sun, Y.; Xu, H. Historical trajectories based location privacy protection query. In Proceedings of the IEEE 11th International Conference on Ubiquitous Intelligence and Computing (ICUIC'14), Ayodya Resort, Bali, Indonesia, 9–12 December 2014; pp. 228–235.
16. Kang, D.; Jung, J.; Mun, J.; Lee, D.; Choi, Y. Efficient and robust user authentication scheme that achieve user anonymity with a Markov chain. *Secur. Commun. Netw.* **2016**, *11*, 1462–1476. [CrossRef]
17. Montazeri, Z.; Houmansadr, A.; Pishro-Nik, H. Achieving Perfect Location Privacy in Markov Models Using Anonymization. 2016. Available online: http://www-unix.ecs.umass.edu/~dgoeckel/zarrin_isita.pdf (accessed on 1 December 2016).
18. Wang, Y.Z.; Xie, L.; Zheng, B.H.; Lee, K.C.K. High utility k-anonymization for social network publishing. *Knowl. Inf. Syst.* **2016**, *3*, 697–725. [CrossRef]
19. Dai, J.Z.; Li, Z.L. A location authentication scheme based on proximity test of location tags. In Proceedings of the 2013 1st International Conference on Information and Network Security (ICINS'13), Beijing, China, 22–24 November 2013; pp. 1–6.