Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3984099, 17 pages
https://doi.org/10.1155/2022/3984099

Research Article

LPPS-AGC: Location Privacy Protection Strategy Based on Alt-Geohash Coding in Location-Based Services

Zekun Zhang, Xiaoting Sun, Siyang Chen, and Yongquan Liang

College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

Correspondence should be addressed to Yongquan Liang; lyq@sdust.edu.cn

Received 19 May 2021; Revised 16 November 2021; Accepted 29 December 2021; Published 27 February 2022

Academic Editor: Carlo Giannelli

Copyright © 2022 Zekun Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Internet of Things (IoT) connects billions of physical devices around the world to the Internet to collect and share massive data. Location privacy leakage has received considerable attention in the field of security. In order to implement the *k*-anonymity location privacy protection mechanism, a previous work constructed an anonymous location set by retrieving historical request record database from the trusted third anonymous server, which advantageously protects user's location privacy. However, the performance of location-based services (LBSs) is weakened by the time overhead of continual retrieving database. Moreover, with the increasingly flourishing of the positioning technique, user location can be accurate to the user's altitude beyond the two-dimensional latitude and longitude coordinates. In this paper, we present a location privacy protection strategy in LBSs based on Alt-Geohash coding (LPP-AGC) method synthetically considering the altitude of user location and time overhead. Specifically, we give the Alt-Geohash coding algorithm (AGCA) to retrieve the historical request record database, which greatly reduces the time overhead and ensures the immediacy of LBSs. In addition, we propose the dummy location generating algorithm (DLGA) and location filtering algorithm (LFA) to provide users with autonomous *k*-anonymity location privacy protection. Extensive simulations are performed to verify the performance and security of the proposed strategy.

1. Introduction

With the prosperity of smart devices and communication technology, the Internet of Things (IoT) has become one of the most important areas of future technology. As a new technology paradigm, IoT is a large-scale interconnected network composed of various sensors, radiofrequency identification (RFID), global positioning system (GPS), and other interactive machines, equipment, vehicles, electrical appliances, and other things with communication and computing [1–3]. IoT is widely used in transportation and logistics, industrial manufacturing, healthcare, smart environment (home, office, and factory), smart city, etc. Meanwhile, location-based service (LBS) has shown a significant growth trend in recent years [4, 5]. LBS, along with the mobile networks, can determine the accurate geographical locations of the mobile users and provide convenient location services to users for navigation based on the dynamic geospatial database. As a result, the wide pervasiveness of LBSs significantly changes our lifestyle.

Although LBS plays a critical role in maintaining the diversity of public life, the potential leakage of privacy resulting from no standardized LBS market system may occur when users utilize their location information in exchange for corresponding services. Specifically, numerous users request the LBS service as well as tightly coupled with their identity, precise location, and query content to location service providers (LSPs). Unfortunately, deliberate LSPs may disclose these information to other malicious ones for their benefits [6, 7], resulting in certain things unaccomplished, disruption of fame, and physical damage. In view of this reason, the vulnerability of privacy may cause users to panic while enjoying LBSs. Nevertheless, since the public or industry users have an extensive demand for LBSs, the issue of imminently improving the security of location privacy has become a rigorous challenge [8–10].

Recently, many location privacy protection methods have been proposed, such as location k-anonymity [11–13], dummy location generation [14, 15], and encryption techniques [16–18].

The location k-anonymity technique [11] constructs an anonymous location set containing the real user and the remaining $k-1$ users' locations and sends the anonymous location set to LSPs by means of the trusted third anonymous server, which ensures the probability of the user's real location recognized is $1/k$. However, generating the remaining $k-1$ locations may bring about tremendous time overhead via retrieving huge location records. Moreover, as the number of location services requested by users increases, retrieving data in location records will become increasingly time-consuming, which will affect the anonymity protection effect and degrade the instant superiority of location services.

The dummy location generation technique [14] can satisfy the users' privacy protection requirement by generating dummy locations and usually does not rely on anonymous servers, reducing the communication overhead. Nevertheless, research has consistently shown that the generated dummy locations lack the rationality of spatial distribution, causing the anonymous protection to defeat [15].

The encryption technique [16] maps the user location and request content into the ciphertext space for calculating and searching, so that LSPs cannot obtain the decoded query content. However, the encryption technique has accentuated the issue of excessive communication overhead and complicated algorithm design.

The idea of the coding method is employed in the encryption technique, which is an effective solution to location privacy protection as well. The coding method can reduce the accuracy of the location by dimensionality reduction of the spatial location data. Especially, in the geospatial filed, Geohash coding [19–21] is often adopted for nearest neighbor search and location privacy protection.

Generally, Geohash coding converts the two-dimensional latitude and longitude of a location into a region represented by a one-dimensional string, which can obscure the precision of the location. Moreover, since the locations that have the same Geohash code are in the same region, Geohash coding can perform fast retrieval within this region as well. It is worth noting that as the positioning method becomes more and more accurate, the location of users in LBS includes not only latitude and longitude coordinates but also altitude.

It is encouraging that employing the user locations within the same altitude for location privacy protection will increase the difficulty for attackers to identify the real user and enhance the effect of anonymous protection. This finding will make a considerable contribution to the field of location privacy protection. For example, there are multiple users using LBS in a multistorey building. The dummy locations generated by the location privacy protection method combined with the altitude has the same longitudes and latitudes as the users' real locations, but the floors (altitudes) are different. Thus, this method can better protect the privacy of the users. To the best of our knowledge, considering synthetically the altitude of location to protect location privacy is largely under explored domain. For this motivation, in this paper, we propose the Alt-Geohash coding method, which is a novel coding method combining the altitude of the user location and Geohash coding.

The Alt-Geohash coding method converts the three-dimensional location of the user's latitude, longitude, and altitude into a one-dimensional string. Since user location still retains the characteristics through Alt-Geohash coding, the candidate anonymous location set can be constructed via selecting locations with encoded characters similar to real user. It can avoid time delay by means of string retrieval and provide users with exceedingly faster and safer location services.

To further provide users with autonomous personalized k-anonymity location privacy protection, we present dummy location generating algorithm (DLGA) and location filter algorithm (LFA), which allow the user to set the value of k according to their own preferences and needs. If the number of locations in the candidate anonymous location set is less than $k-1$, we use DLGA to divide the anonymous region into $k-1$ parts and randomly select in each subregion, meanwhile avoiding the unreasonable location. This not only ensures the uniform distribution of the generated dummy locations but also increases the difficulty for the attacker to identify the real user. If the number of locations in the candidate anonymous location set is greater than $k-1$, we adopt LFA to select $k-1$ locations. LFA divides user locations into four categories and then selects the location similar to the real user based on the corresponding ratio of every category. Thus, the constructed anonymous locations can be evenly distributed, and it is difficult for attackers to identify the user's real location.

Our main contributions are summarized as follows:

(1) To the best of our knowledge, we are the first to combine altitude of the user location into Geohash coding and propose Alt-Geohash coding method

(2) We present a location-based privacy protection strategy based on Alt-Geohash coding (LPPS-AGC) in LBS, which synthetically considers the altitude of the user location and time overhead

(3) Our strategy includes location generalization algorithm (LGA), Alt-Geohash coding algorithm (AGCA), reverse retrieval algorithm (RRA), dummy locations generating algorithm (DLGA), and location filtering algorithm (LFA) to effectively protect the user location privacy in LBSs

(4) Our strategy provides customizable k-anonymity model with users for protecting location privacy by virtue of the trusted third anonymous server

The rest of this paper is organized as follows. Section 2 begins by laying out the related work on location privacy protection and Geohash coding application. Section 3 is concerned with the preliminaries used for this study. The details of algorithms design and the security analysis are presented in Section 4. Then, the proposed algorithms are evaluated with extensive simulation experiment in Section 5. Finally, Section 6 depicts conclusions and future work.

2. Related Work

2.1. Spatial Cloaking Method. The spatial cloaking method [12, 22–26] signifies that it can decrease the accuracy of user's real location through enlarging the revealed spatial region. The most influential technique for spatial cloaking is location k-anonymity, which conceals the real user's location by other $k-1$ locations in the same spatial region.

In [22], Gruteser and Grunwald presented an adaptive interval cloaking algorithm to accomplish location k-anonymity via utilizing qaud-tree for spatial and temporal cloaking. However, this approach is unrealistic in practice as most users have varied privacy requirements under different contexts.

In [23], Chow et al. proposed a Casper model-based spatial cloaking method to achieve personalized privacy preserving. This method divides the space into H layers by a quad tree, and each layer maintains the information of the whole spatial structure to improve the performance of anonymity effectively. Unfortunately, since it needs the cooperation of trustworthy users, the failure to construct the anonymous region due to lack of sufficient dummy locations in a region with few users becomes a major defect.

In order to solve the location privacy problem in the region with few users, in [24], Ni et al. presented k-SDCA algorithm in a sparsely populated region to construct anonymous region. This algorithm achieves k-anonymity by selecting dummy users with high historical query probability, relatively uniform geographical distribution, and large difference in request contents by anonymous server, which had high security and better efficiency. However, since this method takes no account of the overhead of retrieving historical query records, it may spend a lot of time, affecting the service quality of LBSs.

In [25], Sun et al. proposed a location label-based (LLB) algorithm for privacy preservation in the scenario where the locations of k users are nearby, similar or identical. In addition, the request aggregation protocol and the pseudoidentity exchange protocol were presented to reduce the response time of LBS. However, LLB takes more uncontrolled factors and is difficult to realize.

In [26], Ruchika and Udai presented an interesting scheme VIC-PRO (vicinity protection) to fortify the location privacy of the client alongside vicinity protection by concealing location coordinates using geometrical transformations in LBS.

In [12], Simon and Alouini proposed a k-anonymity location privacy algorithm KABC based on clustering to eliminate outliers and establish the anonymous group in the anonymous model. This algorithm balances the conflict between the privacy preserving security and query quality of the location service. However, the time overhead is not optimized due to the iterative process of the clustering algorithm.

2.2. Dummy Location Generation Technique. The dummy location generation (DLG) technique [27–33] which does not rely on the trusted third anonymous server is one of the ubiquitous research effort to protect location privacy. For each request associated with a user's specific location, DLG generates multiple dummy locations close to the real one and submits them to the anonymous server as the service request. Thus, LSP cannot distinguish user's real location.

In [27], Kido et al. defined the dummy location generation approach where the user adopts a random walking model to generate dummy locations. However, in [28], Lu et al. pointed out that, when the generated dummy locations had high density distribution, the protection degree of the user's location privacy could be reduced. To solve this weakness, they split the circle/grid region into equal-sized subregions and distributed all the locations on different radiuses/vertices. Unfortunately, this method ignores the background information possessed by the adversary. If the adversary exploits the public knowledge such as the urban map, some dummy locations generated by this method could be identified easily because they may be in a river or on a mountain.

SpaceTwist is a classical privacy preserving method without the trusted third anonymous server, which employs an anchor to replace the user's real location [30]. In addition, users can gain the accurate result according to their location information. Although this algorithm realizes location privacy protection, the degree of privacy preserving is not high.

In [29], Niu et al. presented two rules to generate dummy locations: (1) the query probability of each dummy should be similar to that of the user's real location and (2) under the premise that the query probability of the locations remained unchanged. However, these locations should be spread as far as possible. Moreover, since the procedure of the algorithms runs on mobile clients, the amount of computing is relatively large. Thus, they have high requirements on computing power and storage space of mobile clients.

In [31], Hara et al. proposed a location privacy preservation method that can be applicable to a real environment, considering the traceability of the dummy locations to simulate the mobility of real user. This method can anonymize the users' locations even against continuous observation.

2.3. Geohash Coding Application. Geohash coding is proverbially applied in many fields with the advantages of rapid retrieval. In [19], Liu et al. proposed the Geohash to build a distributed spatial index for the distributed memory in geographical information system (GIS) fields. This method promotes the reading and writing performance of spatial data and provides a solid technical foundation for high-performance geographical computation.

In [34], Li et al. presented a location-aware wireless body area network (WBAN) data monitoring system on the not only structured query language (NoSQL) database system based on Geohash spatial index, which can efficiently process location-based queries for medical signal monitoring. This novel data analysis method advantageously promotes the development of medical analysis services.

In [35], Guo et al. proposed an adaptive Hilbert-Geohash meshing and coding method called AHG in the geographic meshing system, which could represent both the location and the approximate size of the coded object directly by the meshing hierarchy and the corresponding coding length. AHG shows favorable stability and scalability besides its capability in accelerating spatial query. The method is now applied

successfully to several spatial query tools in a high-performance geographic information system called HiGIS.

The Geohash coding technique [19] can encode the geographic coordinates and perform fast spatial neighbor search, avoiding the time overhead of complex space calculation and speeding up the retrieval. In addition, by mapping the user's precise location into the spatial region by encoding, the user's location privacy information can be effectively protected, which is very practical. Therefore, in this paper, we improve the Geohash coding technique to reduce time overhead and provide secure location services.

Geographical indistinguishability means that the user's real location and approximate location are indistinguishable within a specified range, which is an extension of traditional differential privacy. In order to protect user's privacy when dealing with LBS, Chatzikokolakis et al. [36] proposed two mechanisms to achieve geographic indistinguishability and extended them to location tracking, while providing a restriction due to the correlation between points sequence leads to the degradation of privacy.

If there is no privacy protection in the geographic location, users may be unwilling to share spectrum with secondary users. Therefore, Dong et al. [37] proposed the LpriDSS method to realize the geographical indistinguishability of the primary user while the first spectrum manager selects the spectrum sharing auxiliary user. User's privacy leakage is a common but fatal problem. Taking into account the randomness of task arrival and the complexity of task allocation, Liu et al. [38] proposed an online control mechanism to maximize the profit of the system platform while ensuring the stability of the system, providing personalized location privacy protection.

Existing location privacy protection schemes have their own advantages. However, most methods ignore the time overhead of retrieving historical data, which causes the speed tardiness of providing location services and affects the experience of users for LBS. In this paper, combining the convenience of the Geohash coding, our LPPS-AGC can be capable of encoding the three-dimensional location of user. Through coding, it can quickly retrieve and single out the candidate locations with the same code. Meanwhile, we execute corresponding process for locations in the candidate anonymous location set according to the k value, providing an anonymous location set that satisfies user's privacy requirements and achieving customized anonymous protection. As a result, our LPPS-AGC can not only speed up the anonymous processing time but also provide personalized privacy protection with great application prospects.

3. Preliminaries

In this section, we describe the related definitions and Alt-Geohash coding method.

3.1. Related Definitions. First, we declare the concept of location k-anonymity.

Definition 1 (location k -anonymity, see [22]). Location k -anonymity, formulated by Gruteser and Grunwald, employs user's location requesting service indistinguishable from other $k-1$ locations to guarantee the probability of identification successfully becomes $k-1$.

Here, k describes the desired anonymity degree, which is defined by the user. Larger k implies higher degree of anonymity.

In this paper, the location query information sent by the user is defined as follows.

Definition 2 (location query information (LQ_U)). Location query information is in the form of $\mathrm{LQ}_U(U_{\mathrm{id}}, U_{\mathrm{loc}}, T, \mathrm{QC}, k)$, which describes the information sent by the user to request information about LBS, where

(i) U_{id} is the user identity

(ii) U_{loc} is the real location of the user. It is a triplet (lat, lng, alt), where lat, lng, and alt represent the longitude, latitude, and altitude of the user's location, respectively

(iii) T is the current query time

(iv) QC is the content of the query

(v) k is the privacy demand

In our strategy, the anonymous server constructs an anonymous location set AS to replace user's real location U_{loc} and sends it to LSPs, which protects the location query information LQ_U. Anonymous location set AS is defined as follows.

Definition 3 (anonymous location set (AS)). Anonymous location set AS is defined as a set of k locations containing user's real location and the remaining $k-1$ locations generated by an anonymous server, which is in the form of $AS(U_{\mathrm{loc}}, U_{\mathrm{loc}}^1, U_{\mathrm{loc}}^2, \ldots, U_{\mathrm{loc}}^{k-1})$ $(k>1)$, where

(i) U_{loc} is the real location of the user

(ii) U_{loc}^i $(i=1,\cdots,k-1)$ represents the ith location generated by an anonymous server

It is worth noting that LQ_U becomes the anonymous request $\mathrm{AQ}_U(U_{\mathrm{id}}, \mathrm{AS}, T, \mathrm{QC})$ through the construction of AS by an anonymous server.

In order to protect the location privacy information, we transform the latitude and longitude of the user's location to the interval region, which is defined as follows.

Definition 4 (interval region (IR_U)). Extending the latitude and longitude coordinates of location to the region represented by a range of intervals is called the interval region, denoted by IR_U $([\mathrm{lat}_i, \mathrm{lat}_j], [\ln \mathrm{g}_i, \ln \mathrm{g}_j])$, where

(i) $[\mathrm{lat}_i, \mathrm{lat}_j]$ $(\mathrm{lat}_i < \mathrm{lat}_j)$ is the latitude interval

(ii) $[\ln g_i, \ln g_j]$ $(\ln g_i < \ln g_j)$ is the longitude interval

Figure 1 illustrates an example of interval region. As can be seen from the figure that the latitude and longitude coordinates of user U_A marked by the green five-pointed star are (lat, lng). Now, we extend lat and lng in latitude and longitude, respectively, which are indicated by white arrows in the figure. That is to say, lat expands to $[lat_1, lat_2]$, lng expands to $[\ln g_1, \ln g_2]$. In this way, $lat \in [lat_1, lat_2]$, $lng \in [\ln g_1, \ln g_2]$. Then, the interval region IR_U corresponding to the user U_A is $([lat_1, lat_2], [\ln g_1, \ln g_2])$, which is marked with a black border.

After extending the precise location to the interval region, Geohash coding is applied to retrieve the historical query locations in the region. Geohash coding is defined as follows.

Definition 5 (Geohash coding [20]). Geohash coding is the process of converting two-dimensional latitude and longitude coordinates into one-dimensional strings. The one-dimensional string is called Geohash code, representing a rectangular region.

In this way, user's location privacy can be protected sufficiently. In Geohash encoding, the Base32 encoding method [39] is applied. Base32 encoding converts the 5-bit binary number into a decimal number and then encodes the 32 letters using 0 to 9, *b* to *z* (removing *a*, *i*, *l*, *o*), thus reducing the length of the code.

Combining with the altitude of a location, in this paper, we present the Alt-Geohash code to indicate user's location.

Definition 6 (Alt-Geohash code). Alt-Geohash code is a one-dimensional string that is converted from location U_{loc} (lat, lng, alt). It contains user's Geohash code and the code corresponding to the altitude of user's location.

By the lights of reference [19], after encoding the location coordinates, we search for similar locations by reverse retrieval.

Definition 7 (reverse retrieval). Reverse retrieval describes the process of seeking historical query records from the trusted third anonymous server based on the Alt-Geohash code of a user and finding out the corresponding location of the same code.

Figure 2 gives an example of reverse retrieval. As the figure depicted, suppose that the Alt-Geohash code of the user's location is "wtw3v4." Now, we reversely retrieve the historical request record database in an anonymous server and then select the corresponding first and last location codes marked in red, which are the same as the real user's Alt-Geohash code. Thus, U_{loc} (lat, lng, alt) of these two records are (31.281991, 121.510708, 492) and (31.248284, 121.53277, 492), respectively.

3.2. Alt-Geohash Coding Method. Geohash encoding, which is often used to retrieve geospatial data, mainly employs

Figure 1: An example of interval region.

the latitude and longitude of the encoded location to quickly match in order to find similar locations in geographic space. Considering the altitude of location, we propose Alt-Geohash coding which can convert the user's spatial three-dimensional location into a one-dimensional string. Specifically, the Alt-Geohash encoding process takes the following five steps:

Step 1: divide the earth latitude range (-90, 90) into two intervals (-90, 0), (0, 90). Here, negative number represents south latitude, while positive number represents north latitude. If the target latitude is in the previous interval, the code is 0; otherwise, the code is 1. Divide the interval into two intervals and code in the same way until the accuracy meets the requirements, attaining the latitude code.

Step 2: divide the earth longitude range (-180, 180) into two intervals (-180, 0) and (0, 180) and encode the target longitude via the same method as Step 1, obtaining the longitude code. Here, negative number represents west longitude, while positive number represents east longitude.

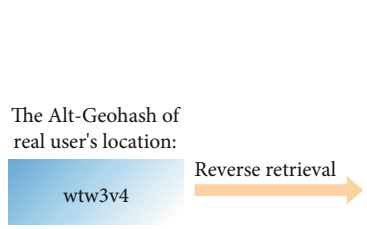
Step 3: merge the 0-1 codes of longitude and latitude codes from left to right to attain the merged 0-1 code, where the odd bits are latitude codes, the even bits are longitude codes, and the even numbers are started from 0.

Step 4: due to Base32 encoding method is employed, the merged 0-1 code obtained in Step 3 converts the 5-bit binary number into a decimal number from left to right. Utilize Base32 code in Figure 3 [39] to obtain the latitude and longitude codes.

Step 5: divide the target altitude by 100 and then connect to the codes obtained in Step 4, attaining the final Alt-Geohash code.

The following example illustrates the Alt-Geohash encoding process.

Suppose that the location of user U_{loc} is (39.92324, 116.3906, 492). The latitude range (-90, 90) is divided into two intervals (-90, 0) and (0, 90). Since the latitude 39.92324



lat	lng	alt	Alt-Geohash
31.281991	121.510708	492	wtw3v4
31.277176	121.458678	492	wtw3g4
31.253718	121.418146	492	wtw3f4
31.210489	121.410025	510	wtw3d5
31.190473	121.452569	507	wtw375
31.248284	121.53277	492	wtw3v4

Figure 2: An example of reverse retrieval.

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Base 32	0	1	2	3	4	5	6	7	8	9	b	c	d	e	f	g
Decimal	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Base 32	h	j	k	m	n	p	q	r	s	t	u	v	w	x	y	z

Figure 3: Base32 code corresponding to the decimal value.

belongs to (0, 90), the code is taken as 1. Then, we divide (0, 90) into two intervals of (0, 45) and (45, 90), and 39.92324 belongs to (0, 45), so the code is 0. By analogy, get the latitude code as 1011 1000 1100 0111 1001 by Step 1. Similarly, the encoding of longitude 116.3906 is 1101 0010 1100 0100 0100 by Step 2. Next, the longitude and latitude codes are combined to obtain the merged 0-1 code 11100 11101 00100 01111 00000 01101 01011 00001 by Step 3. Then, Base32 encoding is performed to obtain the encoding of (39.92324, 116.3906) as "wx4g0ec1" by Step 4. Thus, for the altitude $\lfloor 492/100 \rfloor = 4$, the user's Alt-Geohash code is "wx4g0ec14" by Step 5.

4. Algorithm Design

In this section, the LPPS-AGC system model and algorithm design are elaborated.

4.1. System Model. We adopt a centralized system architecture [40, 41], which contains the location request users, the trusted third anonymous server, and LSPs. Among them, the trusted third anonymous server conceals the user's location into AS to request LBS, achieving location privacy protection. Then, the anonymous server filters the returned responses to reduce the communication overhead. The system architecture of LPPS-AGC is shown in Figure 4. As depicted in the figure, the LBS request process is as follows:

(1) Users who need the LBSs transmit LQ_U (U_{id}, U_{loc}, T, QC, k) through the mobile terminals to the trusted third anonymous server

(2) Anonymous server constructs an AS (U_{loc}, U_{loc}^1, U_{loc}^2, …, U_{loc}^{k-1}) through LPPS-AGC

(3) The anonymous server transmits AQ_U (U_{id}, AS, T, QC) to LSPs

(4) LSPs query the location database to return the query results according to the transmitted request information by the anonymous server

(5) The trusted third anonymous server filters the returned query results

(6) The filtered results are returned to the users, finishing LBSs

The trusted anonymous server [30] plays an important role in protecting the location privacy of user. Specifically, LPPS-AGC includes the following 5 steps:

Step 1 (location generalization): receiving LQ_U launched by users, the anonymous server generalizes the latitude and longitude of locations into an IR_U.

Step 2 (Alt-Geohash coding): the generalized location is encoded to facilitate the retrieval of neighbor locations by executing Alt-Geohash coding.

Step 3 (reverse retrieval): perform a reverse retrieval based on the code generated in Step 2 to find the same code of locations and construct the candidate anonymous location set LS.

After that, we compare the number (that is, num value, as shown in Figure 4) of locations in LS with the user-defined k value. If num < k-1, execute Step 4. If num > k – 1, execute Step 5. If num = k – 1, take LS generated in Step 3 as AS to generate AQ_U and transmit it to LSPs.

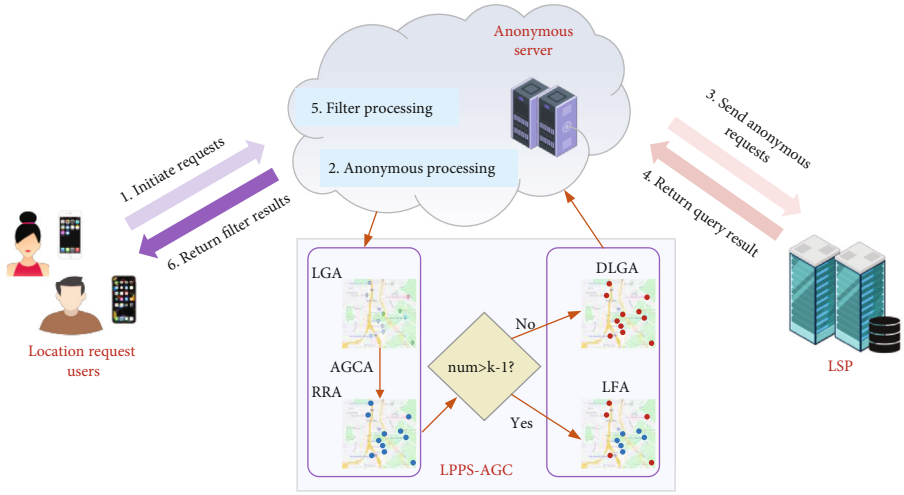


Figure 4: The system architecture of LPPS-AGC.

Step 4 (dummy location generating): IR_U is evenly divided into $k-1$ subregions. A dummy location is randomly generated in each subregion and an AS is formed.

Step 5 (location filtering): multiple locations in LS are classified based on user's real location, and the number of locations in each category is selected according to the proportion of each category in the total numbers of locations in LS. Thus, AS which satisfies user's privacy requirement is constructed.

The detailed algorithms implementation of LPPS-AGC is described in the following.

4.2. Location Generalization Algorithm. The location generalization algorithm (LGA) extends the latitude and longitude of location into the interval region defined in this paper. Moreover, the purpose is to obscure the location of user and reduce the precision of latitude and longitude coordinates. The detailed pseudocode of LGA is presented in Algorithm 1.

The algorithm generates four random numbers, each of which is randomly assigned to add/subtract with the user's longitude/latitude, ensuring that the latitude and longitude are only subjected to one addition and one subtraction. According to the principle that the small number is in front and the large number is in the latter, the longitude interval and the latitude interval are formed, respectively, obtaining the final interval region.

For the setting of the random number range, the following two points need to be explained:

(1) The reason why the range of generating random numbers is set from 0 to 0.025 is that the one degree of latitude of geographic location corresponds to 111 kilometers. Thus, 0.025 latitudes correspond to 2.5 kilometers. We can obtain that the maximum length of the longitude interval and the latitude interval is 0.05. Therefore, the region indicated is approximately: $(0.05 \div 0.025) * 2.5\,\text{km} * (0.05 \div 0.025) * 2.5\,\text{km} = 25\,\text{km}^2$, which is enough to ensure a wide anonymous protection region

(2) The purpose of randomly generating four random numbers is to prevent the attacker from predicting the user's precise location. If merely one random number is generated, the longitude and latitude fluctuate forward and backward, respectively. According to the symmetry of the interval, the amplitude of the fluctuation can be easily calculated. Thus, the user's latitude and longitude coordinates are also exposed. Similarly, if two random numbers are generated, the latitude and longitude can also be predicted by attackers. Therefore, four random numbers are randomly generated here, which can protect the location of the user triumphantly

Obviously, the time complexity of LGA is $O(1)$, which is simple and easy to implement.

4.3. Alt-Geohash Coding Algorithm. After the interval region is determined by LGA, the Alt-Geohash code is generated by the Alt-Geohash coding algorithm (AGCA) for location retrieval.

Concretely, AGCA initializes two variables geohash and code, respectively. geohash stores the Geohash code of the generated interval region, while code preserves the intermediate generated 0-1 code temporarily.

Then, AGCA takes the median of the earth latitude range $[-90, 90]$ and longitude range $[-180,180]$, respectively. The longitude interval is processed as follows:

(i) If all the numbers in the longitude interval are less than or equal to the median, code adds 0 and updates the upper limit of the longitude range of the earth to the median

(ii) If all the numbers in the longitude interval are greater than the median, code adds 1 and updates

Input: user's location U_{loc} (lat, lng, alt).
Output: interval region IR_U ([lat_1, lat_2], [ln g_1, ln g_2]).
1: Generate four random numbers from 0 to 0.025: r_1, r_2, r_3, and r_4, which are used to extend latitude and longitude coordinates
2: $\text{lat}_1 \longleftarrow \text{lat} - r_3$
3: $\text{lat}_2 \longleftarrow \text{lat} + r_1$
4: $\text{lat}_3 \longleftarrow \text{lng} - r_4$
5: $\text{lat}_4 \longleftarrow \text{lng} + r_2$
6: **return**IR_U ([lat_1, lat_2], [ln g_1, ln g_2])

Algorithm 1: Location generalization algorithm (LGA)

the lower limit of the longitude range of the earth to the median

(iii) If there exist partial numbers in the longitude interval which are greater or less than the median, the encoding ends

Encoding the latitude interval is the same as the longitude interval encoding process. Iteratively repeating the coding process, we can get the 0-1 encoding of the interval region.

Next, we encode the height of the user's location, divide it by 100, and round it up. The resulting value is added to geohash to get the final Alt-Geohash. Note that the division 100 here is determined by the characteristics of the Geolife dataset [42] we employed, and the purpose is to obtain the encoding of 0-9. In addition, the length l of geohash is calculated for the next stage location retrieval to protect user's location privacy.

The detailed pseudocode of AGCA is presented in Algorithm 2.

To illustrate the process of of Alt-Geohash coding, we give an example in the following.

Suppose the real location is U_{loc} (121.51071, 31.281991, 397). The interval region obtained in the location generalization stage is ([121.51050, 121.51105], [31.281801, 31.282003]). According to AGCA, we iterate the latitude interval processing via the dichotomy for 0,1 coding.

Specifically, according to AGCA, the earth latitude range $[-90, 90]$ is divided into two intervals $[-90, 0]$ and $[0, 90]$, and the latitude interval ($[31.281801, 31.282003]$) falls in the latter half $[0, 90]$, coded as 1. Then, we divide $[0, 90]$ into $[0, 45]$ and $[45, 90]$, and the latitude interval falls in the first half $[0, 45]$, coded as 0. By analogy, the dichotomy is used to divide the earth's latitude range for 0-1 coding, and the iteration is stopped until the partition interval cannot completely cover the interval region.

Table 1 describes the process of latitude interval coding of the example. As table shown, in 19th round, [31.281801 , 31.282003] cannot be completely covered by the divided interval 1 [31.281507788, 31.28219458] and the divided interval 2 [31.281507788, 31.281851184]. The longitude interval adopts the same method. After the 0-1 encoding is completed, the Base32 code is used for further encoding, obtaining the geohash code of this interval region: "wtw3vn9." Finally, the altitude is divided by 100, that is, 397%100 = 3, 3 is obtained. Therefore, the Alt-Geohash code is "wtw3vn93."

Theorem 8. *The time complexity of AGCA is $O(n)$, where n is the number of encoding.*

Proof. The time for encoding the interval region is spent on the division of the interval, that is, step 3 to step 25 in AGCA. Steps 6 to 23 deal with the latitude and longitude encoding of the location. In the worst case, when the accuracy of the divided interval is smaller than the range of the interval region, the encoding ends. That is, the number of iterations n satisfies the following two conditions:

$$\frac{(90 - (-90))}{2n} < l_1, \quad \frac{(180 - (-180))}{2n} < l_2 \tag{1}$$

where l_1 and l_2 are the length of the longitude interval and the latitude interval, respectively. Simplifying it, we obtain $n > \max\{90/l_1, 180/l_2\}$. Step 26 handles the altitude of the location, and its time overhead is $O(1)$. Therefore, the time complexity of AGCA is $O(n) = O(\max\{90/l_1, 180/l_2\})$.□ □

4.4. Reverse Retrieval Algorithm. After the interval region of the user is encoded by AGCA, the history database of the third party anonymous server is retrieved based on the code. Specifically, for each location in the history request record, the reverse retrieval algorithm (RRA) calculates the Alt-Geohash code according to AGCA and the Geohash code length l, denoted as AG_i. Then, if AG_i is the same as the interval region code Alt-Geohash, add this location to LS, and num accumulates 1. Otherwise, the next location in historical request record location data set LDS is performed. Finally, the algorithm returns the candidate anonymous location set LS and the location number num.

The pseudocode of RRA is elaborated in Algorithm 3.

Obviously, the time complexity of RRA is $O(n)$.

Since the number of locations num in LS retrieved by RRA does not necessarily meet the privacy requirements of the user, we propose Algorithm 4 and Algorithm 5 in the following to provide personalized location privacy protection for the user. If the output of RRA num = $k - 1$, the real user is added to LS as AS, LPPS-AGC anonymous processing ends, and AS is sent to the LSP. If num < $k - 1$ then execute Algorithm 4 dummy location generating algorithm (DLGA). If num > $k - 1$, Algorithm 5 location filtering algorithm (LFA) is executed.

Input: interval region $IR_U([lat_1, lat_2], [\ln g_1, \ln g_2])$, altitude of user's location alt.
Output: code of interval region Alt-Geohash, length of Geohash code *l*.
1: lat_range = [−90, 90], lng_range = [−180,180]
//lat_range is the range of latitude, lng_range is the range of longitude
2: geohash = ∅, code = ∅
3: **while**IR_U! = ∅**do**
4: lat_mid =sum(lat_range)/2
5: lng_mid =sum(lat_range)/2
6: **while** TRUE **do**
7: **if**ln g_1 < =lng_mid and ln g_2 < =lng_mid**then**//longitude
8: code ⟵ code ∪ {0}
9: lng_range[1] = lng_mid
10: **else if**ln g_1 > lng_mid and ln g_2 >lng_mid**then**
11: code ⟵ code ∪ {1}
12: lng_range[0] = lng_mid
13: **else break**
14: **end if**
15: **if**lat_1 < =lat_mid and lat_2 < =lat_mid**then**//latitude
16: *c*ode ⟵ code ∪ {0}
17: lat_range[1] = lat_mid
18: **else if**lat_1 >lat_mid and lat_2 >lat_mid**then**
19: code ⟵ code ∪ {1}
20: lat_range[0] = lat_mid
21: **else break**
22: **end if**
23: **end while**
24: **end while**
25: According to the Base32 code shown in, code will be encoded as geohash
26: $a = alt\%100$//altitude
27: Alt-Geohash ⟵ geohash ∪ {*a*}
28: $l \longleftarrow len(geohash)$//len(*geohash*) represents the length of geohash code
29: **return** Alt-Geohash, *l*

Algorithm 2: Alt-Geohash coding algorithm (AGCA)

Table 1: Example of latitude interval coding.

	Range	Divided Interval 1	Divided Interval 2	Coding
1	[−90, 90]	[−90, 0]	[0, 90]	1
2	[0, 90]	[0, 45]	[45, 90]	0
3	[0, 45]	[0,22.5]	[22.5,45]	1
4	[22.5,45]	[22.5,33.75]	[33.75,45]	0
	...	...	...	...
19	[31.28150,7788,31.28219458]	[31.28150,7788,31.281851184]	[31.28185,1184,31.28219458]	Termination

4.5. Dummy Location Generating Algorithm. When the number of location num in LS is less than the user's privacy requirement k value, we employ DLGA to generate $k-1$ locations for location privacy protection. The core of this algorithm is to uniformly divide the interval region into $k-1$ subregions and generate a dummy location in each subregion to protect user's real location. It is worth noting that DLGA can generate dummy locations according to user's preference, that is, the k value set by the user himself, which can provide users with personalized privacy protection services and avoid singularization of the services provided.

DLGA first calculates the lengths of the longitude interval and the latitude interval D_1 and D_2, respectively.

If k is an even number and $D_1 \geq D_2$ (that is, the length of longitude interval is longer), DLGA divides the longitude interval into $k-1$ equal parts and randomly selects a location in each subinterval. Among them, the longitude is within the range of the longitude subinterval, and the latitude is within the latitude interval, while the altitude is the same as the altitude of the user's location. If k is even and $D_1 < D_2$ (that is, the length of latitude interval is longer), DLGA divides the latitude interval into $k-1$ equal parts.

Input: code of interval region Alt-Geohash, length of Geohash code l, anonymity degree k, historical request record location data set $\mathrm{LDS}(U_{\mathrm{loc}}^1, U_{\mathrm{loc}}^2, \cdots, U_{\mathrm{loc}}^i, \cdots, U_{\mathrm{loc}}^n)$, ($U_{\mathrm{loc}}^i$ is the three-dimensional location, n is the number of locations).
Output: candidate anonymous location set LS, number of locations within the candidate anonymous location set num.
1: LS = ∅
2: num = 0
3: **for** i = 1 **to** n **do**
4: **Call** Algorithm 2 (AGCA), calculate the Alt-Geohash code of U_{loc}^i, denoted as AG_i; calculate the code length l
5: **if** AG_i == Alt-Geohash **then**
6: LS ⟵ LS ∪ $\{U_{\mathrm{loc}}^i\}$
7: num = num + 1
8: **else**
9: **continue**
10: **end if**
11: **end for**
12: **return** LS, num

Algorithm 3: Reverse retrieval algorithm (RRA)

Meanwhile, the longitude of the selected location is within the longitude interval, and the latitude is in the latitude subinterval, the altitude is the same as above.

If k is an odd number and $D_1 \geq D_2$, DLGA divides the longitude interval into $(k-1)/2$ equal parts. In order to ensure that $k-1$ locations are generated, the latitude interval is divided into 2 equal parts, and a location is randomly selected in each subinterval. Similarly, if k is an odd number and $D_1 < D_2$, the latitude interval is divided into $(k-1)/2$ equal parts. To ensure that $k-1$ locations are generated, the longitude interval is equally divided into two parts, and one location is randomly selected in each subregion.

Finally, the user's real location is added to AS as well, and the AS containing k locations is successfully constructed. The pseudocode of DLGA is presented in Algorithm 4.

Now, we give the time complexity analysis of RRA in the following.

Theorem 9. *The time complexity of DLGA is $O(k)$.*

Proof. In Algorithm 4, steps 1 to 2 are simple calculations, in which time overhead is $O(1) + O(1) + O(1) + O(1) + O(1) + O(1) = O(1)$, preparing for later calculations. Steps 3 to 42 are the core of the algorithm, where the interval region is divided according to the parity of the k value and the location is selected within the subregion. Steps 3 to 18 are actions performed when k is even. It is divided into two processes by the length comparison result between the longitude interval and the latitude interval. Since the longer length intervals are also evenly divided into $k-1$ subregions, while the shorter length interval is not divided, and $k-1$ dummy locations are generated iteratively, the time overhead is $O(k-1) + O(k-1) = O(k)$. Steps 19 to 41 are actions performed when k is an odd number. Similarly, there are two cases. The longer interval is divided into $(k-1)/2$, and the shorter length is divided into 2 parts. Then, the location points are selected, respectively. At this time, the time overhead is $O(2 * ((k-1)/2) + O(2 * ((k-1)/2)) = O(k)$. Therefore, the total time complexity of DLGA is $O(1) + O(k) + O(k) = O(k)$.□ □

4.6. Location Filtering Algorithm. The candidate anonymous location set LS (U_{loc}^1, U_{loc}^2,…, U_{loc}^i,…, $U_{\mathrm{loc}}^{\mathrm{num}}$) is generated by RRA algorithm and contains num locations, where U_{loc}^i consists of longitude $\ln g_i$, latitude lat_i, altitude alt_i. These locations are candidates for anonymous protection.

When num is greater than privacy requirement k value, we need to filter the locations in the interval region to select the locations that are most similar to real user to improve privacy. Therefore, we design location filtering algorithm (LFA) to implement it. Specifically, LFA divides the locations in LS into four categories A, B, C, and D according to the latitude and longitude values of the real user and then put them in four lists. After that, the number of locations (a, b, c, d) in each list is counted, respectively. Accordingly, the proportion of the locations number of every category in the total numbers of LS is a/num, b/num, c/num, and d/num, respectively.

Note that in order to meet the privacy requirements of user and implement k-anonymity protection, LFA selects $a(k-1)$/num, $b(k-1)$/num, $c(k-1)$/num, and $d(k-1)$/num locations in A, B, C, and D, respectively, which are added to AS. Finally, the real location of the user is added to AS as well to complete the anonymous processing. The detailed pseudocode of LFA is elaborated in Algorithm 5.

Theorem 10. *The time complexity of LFA is $O(num)$, where num is the number of locations in LS.*

Proof. In LFA, the main work is to divide the locations in the candidate anonymous location set into four categories according to the real location of user. Steps 1 to 12 compare the latitude and longitude of each location in LS with the latitude and longitude of the user's real location, which the time overhead is $O(\mathrm{num})$. Then, steps 13 to 14 randomly select the corresponding locations in the four categories of lists according to different proportions, which the time overhead is $O(k)$. Since num > k, thus, the time complexity of this algorithm is $O(\mathrm{num})$.□ □

Input: interval region $IR_U([lat_1, lat_2], [\ln g_1, \ln g_2])$, anonymity degree k, user's real location U_{loc} < lat, lng, alt > .
Output: anonymous location set AS.
1: AS = ∅, $d_1 = lat_1$, $d_2 = \ln g_1$
2: $D_1 = (lat_2 - lat_1)$, $D_2 = (\ln g_2 - \ln g_1)$
3: **if** k%2 == 0 and $D_1 >= D_2$ **then**
4: **for** $m = 0$ **to** k-1 **do**
5: Lat = random(d_1, $d_1 + (D_1/(k-1))$)
6: Lng = random($\ln g_1$, $\ln g_2$)
7: Alt = alt
8: AS ⟵ AS ∪ {<Lat, Lng, Alt > }
9: Lat = $d_1 + (D_1/(k-1))$
10: **end for**
11: **else if** k%2 == 0 and $D_1 < D_2$ **then**
12: **for** $m = 0$ **to** $k - 1$ **do**
13: Lat = random(lat_1, lat_2)
14: Lng = random(d_2, $d_2 + (D_2/(k-1))$)
15: Alt = alt
16: AS ⟵ AS ∪ {<Lat, Lng, Alt > }
17: $d_2 = d_2 + (D_2/(k-1))$
18: **end for**
19: **else if** k%2! = 0 and $D_1 >= D_2$ **then**
20: **for** $j = 0$ **to** 2 **do**
21: **for** $i = 0$ **to** $(k-1)/2$ **do**
22: Lat = random(d_1, $d_1 + (D_1/(k-1))$)
23: Lng = random(d_2, $d_2 + (D_2/2)$)
24: Alt = alt
25: AS ⟵ AS ∪ {<Lat, Lng, Alt > }
26: $d_1 = d_1 + (D_1/(k-1))$
27: **end for**
28: $d_2 = d_2 + (D_2/2)$
29: **end for**
30: **else**
31: **for** $j = 0$ **to** 2 **do**
32: **for** $i = 0$ **to** $(k-1)/2$ **do**
33: Lat = random(d_1, $d_1 + (D_1/2)$)
34: Lng = random(d_2, $d_2 + (D_2/(k-1))$)
35: Alt = alt
36: AS ⟵ AS ∪ {<Lat, Lng, Alt > }
37: $d_2 = d_2 + (D_2/(k-1))$
38: **end for**
39: $d_1 = d_1 + (D_1/2)$
40: **end for**
41: **end if**
42: **return** AS ⟵ AS ∪ {<Lat, Lng, Alt > }

Algorithm 4: Dummy location generating algorithm (DLGA)

4.7. Security Analysis. In some scenarios, a malicious attacker may collude with some users or LSPs to obtain the available private information about the legitimate user. Fortunately, our strategy can triumphantly resist some attacks of adversaries, such as map-matching attack [43], centre-of-ASR (anonymized spatial region) attack [43] and colluding attack [44], demonstrating superior security. In the following, we give the security analysis of our strategy.

Map-matching attack means that an attacker restricts the obfuscation region to certain locations where users can be located to find the real user by eliminating specific dummies based on map details [43]. Since our strategy adds height to the location information, there will be multiple users with different heights in the same location point. Even if a malicious attacker locates the user's location, that is, the user's longitude and latitude, through the positioning system, he cannot distinguish between real users and dummies. At the same time, our strategy also satisfies k-anonymity. Therefore, our strategy can resist map-matching attacks.

ASR (anonymized spatial region) is the region used to hide the user's actual location information during the anonymization process. In case of dummy generation-based techniques, if the actual user's location is in the center of the region composed of dummy locations and the real user location, a centre-of-ASR attack [43] will occur. In our strategy, after adding the height to the user's location information, more dummy users' locations can be constructed in an anonymous region. Moreover, these dummy users' locations

Input: candidate anonymous location set LS $(U_{\text{loc}}^1, U_{\text{loc}}^2, \cdots, U_{\text{loc}}^i, \cdots, U_{\text{loc}}^{\text{num}})$ $(U_{\text{loc}}^i < \text{lat}_i, \ln g_i, \text{alt}_i >)$, location number of candidate anonymous location set num, anonymity degree k, user's location $U_{\text{loc}} < \text{lat}, \text{lng}, \text{alt} >$.
Output: anonymous location set AS.
$A = B = C = D = \varnothing$//initialize four empty lists to hold the four categories of locations.
For $i = 1$ **to** num **do**
  If $\text{lat}_i >$ lat and $\ln g_i >$ lng **then**
    $A \longleftarrow A \cup \{U_{\text{loc}}^i\}$
  Else if $\text{lat}_i <=$ lat and $\ln g_i >$ lng **then**
    $B \longleftarrow B \cup \{U_{\text{loc}}^i\}$
  Else if $\text{lat}_i >$ lat and $\ln g_i <=$ lng **then**
    $C \longleftarrow C \cup \{U_{\text{loc}}^i\}$
  Else
    $D \longleftarrow D \cup \{U_{\text{loc}}^i\}$
  End if
End for
Count the number of locations in A, B, C, and D, respectively, denoted as a, b, c, d
Randomly select $a(k-1)/\text{num}$, $b(k-1)/\text{num}$, $c(k-1)/\text{num}$, and $d(k-1)/\text{num}$ locations from the list A, B, C, and D, respectively
AS $\longleftarrow$ AS $\cup \{U_{\text{loc}}\}$
Return AS

Algorithm 5: Location filtering algorithm (LFA)

are more similar, which is more confusing. In this way, it is difficult for an attacker to attack real user. Hence, our strategy can resist centre-of-ASR attack.

Colluding attack means that the probability of successfully deducing the real location of user out of anonymous location information generated by certain strategy does not increase with the growth of the size of the colluding group [44]. Since our strategy employs random numbers in the interval region, there are random factors in DLGA and LFA. Note that the generation of anonymous location set is irregular. Thus, the attacker cannot infer the real location of the user based on the anonymous location set. The probability of inferring the real user is still $1/k$. At the same time, users are independent of each other and do not affect each other; therefore, this strategy can resist colluding attack by attackers and malicious users.

5. Simulations

In this section, we verify the performance of our proposed LPPS-AGC through simulation experiments. The experimental configuration settings and the analysis of the experimental results are described in detail.

5.1. Experimental Settings. The experimental environment is 64 bit Windows 7 system with Intel (R) Core i5 Processor (2.30 GHz) and 8 GB memory. The programming language is Python on PyCharm.

The location data is from the Geolife dataset [42], which collects the trajectory data for 182 users from April 2007 to August 2012. The dataset includes a series of time sequence locations, covering latitude, longitude, altitude, and so on. It contains 17,621 trajectories with a total distance of more than 1.2 million kilometers and a total time of more than 48,000 hours. These data record not only the location of user's home and work but also a wide range of outdoor activities such as shopping, traveling, hiking, and cycling.

Without loss of generality, we apply the first record of a user in the Geolife dataset as the real location, and the remaining records as historical records to construct anonymous location set in our experiment. The relevant parameters of attributes are shown in Table 2.

5.2. Comparison of Time Overhead in Processing Data. Most existing techniques apply distance calculation to construct candidate anonymous location set. Suppose two locations are $U_{\text{loc}}^i(\text{lat}_i, \ln g_i, \text{alt}_i)$ and $U_{\text{loc}}^j(\text{lat}_j, \ln g_j, \text{alt}_j)$, respectively. Thus, the universally accepted Manhattan distance M_{dist} [22, 45] and Euclidean distance E_{dist} [46] between U_{loc}^i and U_{loc}^j are measured by

$$M_{\text{dist}}\left(U_{\text{loc}}^i, U_{\text{loc}}^j\right) = \left|\text{lat}_i - \text{lat}_j\right| + \left|\ln g_i - \ln g_j\right| + \left|\text{alt}_i - \text{alt}_j\right|,$$
$$E_{\text{dist}}\left(U_{\text{loc}}^i, U_{\text{loc}}^j\right) = \sqrt{\left(\text{lat}_i - \text{lat}_j\right)^2 + \left(\ln g_i - \ln g_j\right)^2 + \left(\text{alt}_i - \text{alt}_j\right)^2}. \quad (2)$$

In our experiment, we execute the algorithms 100 times and average the output results for comparison. Then, we evaluate the efficiency of our Alt-Geohash encoding technique.

Figure 5 shows the comparison of the time in processing different data volumes of Alt-Geohash encoding technique, Manhattan distance, and Euclidean distance measures. The result reveals that as the amount of data increases, the time overhead of the three methods all increases. Obviously, our Alt-Geohash encoding technique has an excellent advantage in terms of time overhead.

We can see that once the amount of data reaches 900, the other two methods take more than 1 s, while Alt-Geohash

Table 2: Experimental parameter settings.

Parameter	Value
The precision of longitude	5-6 digits
The precision of latitude	5-6 digits
The length of code	6-8 bits
The number of request records	100-900

coding takes less than 0.2 s. It is because that Alt-Geohash coding simply turns three-dimensional location into one-dimensional strings without any redundant calculating. As for distance-based retrieval methods, distance calculating is indispensable, and the time taken is greatly grown with the increasing of the processed data volume. In particular, since Euclidean distance measure exists in multiplication operation, it immensely increases the time overhead. Therefore, Alt-Geohash coding retrieval is faster and more suitable for processing large volumes of data.

5.3. Comparison of Time Overhead in Generating Dummy Locations. Next, we consider the impact of the number of generated dummy locations on time overhead. We compare our dummy locations generating algorithm (DLGA), dummy-based user location anonymization algorithm (DBULA) [31], and random generating dummy locations strategy [47].

As shown in Figure 6, the time overhead of these three algorithms aggravates with the increasing of the number of dummy locations. Since the random strategy randomly selects locations within the specified region, its time overhead is the smallest. However, its anonymous protection effect is very inferior. Furthermore, since the randomly generated dummy locations do not consider the rationality of the locations, an attacker may filter out many locations, leading to the failure of anonymous protection. DBULA generates dummies around the user in a grid shape to satisfy the anonymous region requirement. The distance between two adjacent locations is L, which is calculated as follows:

$$L = \frac{\sqrt{S}}{\sqrt{k-1}}, \tag{3}$$

where S is the area of the anonymous region and k indicates the anonymity degree. The generating dummy location satisfies

$$\begin{aligned} \mathrm{lat_i} &= \mathrm{lat} + \left(\left\lfloor \frac{i}{\sqrt{k}} \right\rfloor - \left\lfloor \frac{\mathrm{UID}}{\sqrt{k}} \right\rfloor \right) \cdot L, \\ \ln \mathrm{g}_i &= \mathrm{lng} + \left(i \bmod \sqrt{k} - \mathrm{UID} \bmod \sqrt{k} \right) \cdot L, \end{aligned} \tag{4}$$

where $(\mathrm{lat}_i, \ln \mathrm{g}_i)$ is the ith dummy location to be generated, $(\mathrm{lat}, \mathrm{lng})$ is the location of the user, UID is the number of user location, and i is the index of the ith dummy location.

Because DBULA carries out aforementioned iterative calculations when generating dummy locations, its time overhead is longer than that of our DLGA. Recall that DLGA divides the interval region into $k-1$ parts and then randomly selects among them. In addition, in an actual scene, the position of the generated dummy may not be located on the road. In that case, our strategy can adjust the generation of dummies so that they are located at the nearest intersection closest to the road, which ensures the rationality of the dummy locations.

5.4. Comparison of Anonymous Processing Time on Different Algorithms. Now, we compare the anonymous processing time with the value of k for different anonymous algorithms, including Spacetwist [30], KABC [12], DLS [29], Casper [23], and our LPPS-AGC. Here, the historical request record in the simulation is 500.

As depicted in Figure 7, with the increase of k value, the time overhead of all algorithms or strategies increases. The only fortunate that the time overhead is minimal in our strategy.

Concretely, since SpaceTwist utilizes an anchor near the client to replace user's real location and every user obtains the accurate result according to his location information, it has the longest anonymous processing time without the trusted third anonymous server.

When Casper algorithm generates dummy locations, the generated anonymous region is too large, which will generate a large number of redundant locations. Thus, the time overhead is more excessive than the rest of the algorithms.

The anonymous processing time of KABC is similar to that of Casper. It is because that KABC applies a clustering algorithm and has a large time overhead. By virtue of historical request record database, DLS fully considers the background information. Based on the query probability, a round of $2k$ dummy locations selection is performed. Thus, the number of redundant locations shrinks extremely, and the algorithm time complexity of DLS is better than that of Casper algorithm.

Our strategy executes AGCA, which tremendously reduces the time overhead. In addition, DLGA and LFA in our strategy can quickly provide secure personalized location services. Therefore, our strategy has a huge advantage in terms of time overhead.

5.5. Comparison of Anonymous Protection Effect on Different Algorithms. We employ anonymous entropy to evaluate anonymous protection effectiveness, which indicates the degree of location anonymity protection [24], the formula is as follows:

$$H = -\sum_{i=1}^{n} p_i \log p_i, \tag{5}$$

where p_i is the probability that the real user is recognized and H is the anonymous entropy. The greater the value of the anonymous entropy, the better the effect of anonymous protection.

Figure 8 illustrates the comparison of the anonymous entropy on k value of different algorithms, including random strategy [47], Casper [23], DLS [29], enhanced-DLS [29], and our LPPS-AGC.

It can be seen from Figure 8 that as the value of k increases, the anonymous entropy of different algorithms also increases. The random strategy performs worse than other schemes,

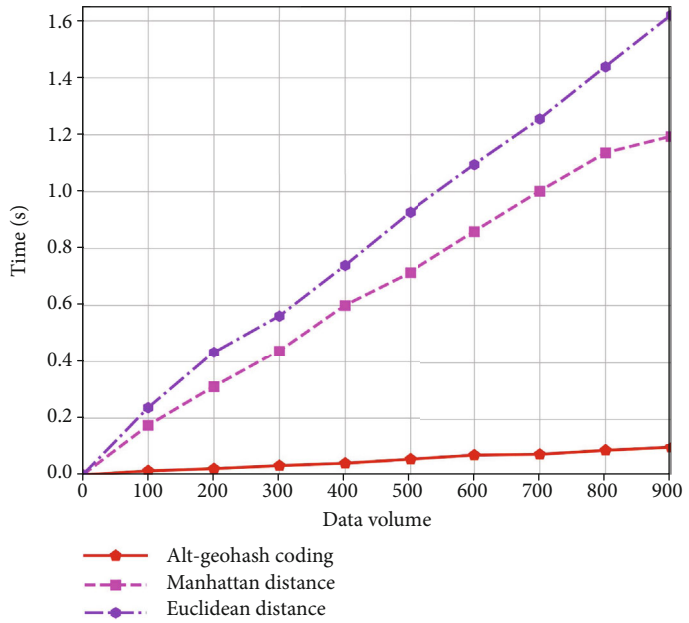


Figure 5: Comparison of time vs. processing data volume.

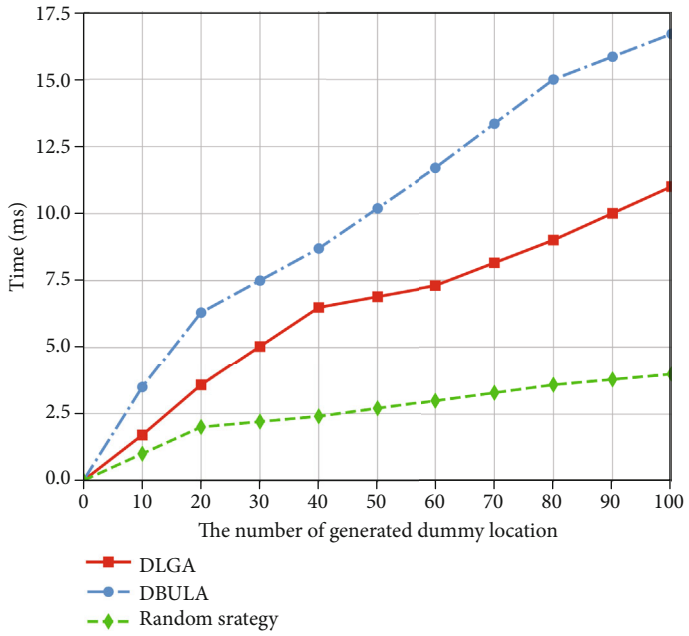


Figure 6: Comparison of time overhead vs. the number of generated dummy locations.

since it just generates dummy locations randomly without considering the background information.

The Casper algorithm has a better degree of anonymity protection than random strategy because it greatly deduces the probability that the real user can be recognized by means of the history request information of the third party anonymous server. However, since Casper generates many redundant locations, its anonymity effect is not very good as well.

In contrast, since the enhanced-DLS and DLS algorithms consider the background information and the generated dummy locations are based on similar probability of queries in history request record, both algorithms have higher anonymous entropy. Moreover, because enhanced-DLS further considers the degree of dispersion of the locations, it has higher anonymous entropy.

In our LPPS-AGC strategy, DLGA generates dummy locations uniformly distributed in the interval region. Meanwhile, fully considering the rationality of the generated locations, the user sets the value of k according to his preference such that the generated locations can be dynamically adjusted according to the actual road situation information. Moreover, LFA can further select locations similar to the real

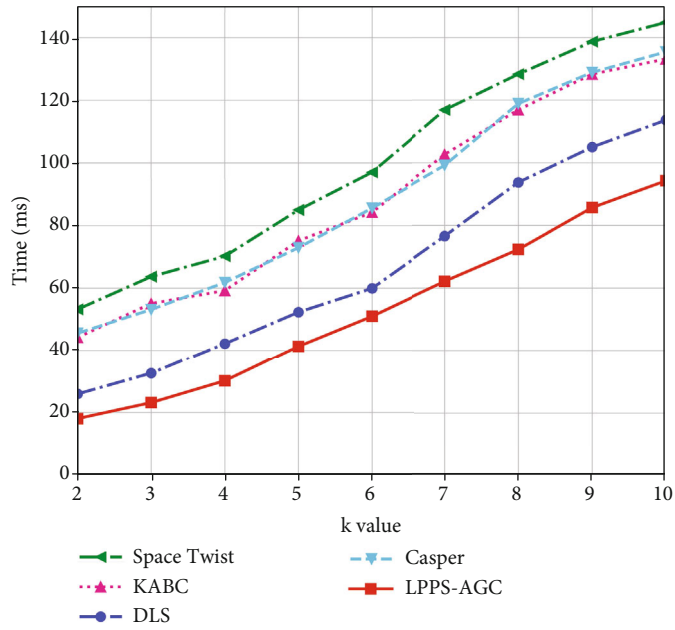


Figure 7: Comparison of anonymous processing time vs. k value.

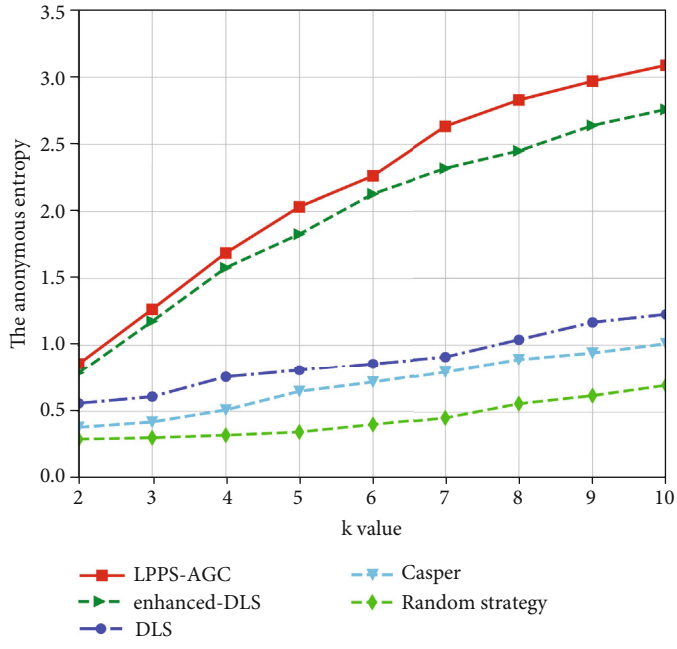


Figure 8: Comparison of anonymous entropy vs. k value.

user to improve the anonymous protection. Therefore, LPPS-AGC has more advantageous anonymous protection effect than that of other algorithms.

6. Conclusions and Future Work

The existing k-anonymity mechanism employs anonymous server to construct an anonymous region for users, protecting location privacy. However, most work rarely ponders the huge time overhead of retrieving the database. Moreover, with the boom of IoT and positioning technology, user's location has also evolved from the original two-dimensional to three-dimensional latitude and longitude and altitude. At present, few researches have considered the location protection by altitude.

In this paper, we propose a location privacy protection strategy based on Alt-Geohash coding (LPPS-AGC), which can effectively protect the three-dimensional coordinates. The strategy first generalizes user's latitude and longitude into the interval region and then retrieves the location of the same

code utilizing our Alt-Geohash coding algorithm (AGCA). After that, it provides user with personalized k-anonymous privacy protection service according to user's privacy requirements by dummy locations generating algorithm (DLGA) and location filter algorithm (LFA). In doing so, it not only considers the altitude of location but also speeds up the process of retrieval and reduces time overhead. Comprehensive analysis and simulation experiments show that our LPPS-AGC strategy has better superiority in performance and can achieve exceptional anonymous protection.

In the future, we will further investigate the location privacy protection scheme. Since mining a large amount of location information can analyze the social attributes of users, our future work will adequately consider the time dimension of users requesting location services and analyze user's behavior models. A predictable dynamic location privacy protection approach may be proposed, which can resist the uncertain insecure factors in advance.

Data Availability

The data used to support the findings of this study are available from the website: "https://www.microsoft.com/en-us/download/confirmation.aspx?id=52367".

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Humanity and Social Science Fund of the Ministry of Education under Grant 20YJAZH078 and 20YJAZH127, Open Project of Tongji University Embedded System and Service Computing of Ministry of Education of China under Grant ESSCKF 2019-06 and ESSCKF 2019-08, and National Innovation and Entrepreneurship Training Program for College Students under Grant 201910424014.

References

[1] R. Li, C. Sturtivant, J. Yu, and X. Cheng, "A novel secure and efficient data aggregation scheme for IoT," *IEEE Internet of Things Journal.*, vol. 6, no. 2, pp. 1551–1560, 2019.

[2] L. Ni, J. Zhang, C. Jiang, C. Yan, and K. Yu, "Resource allocation strategy in fog computing based on priced timed Petri nets," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1216–1228, 2017.

[3] J. Pan, C. Lee, A. Sghaier, M. Zeghid, and J. Xie, "Novel systolization of subquadratic space complexity multipliers based on toeplitz matrix-vector product approach," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 27, no. 7, pp. 1614–1622, 2019.

[4] Z. Wu, R. Wang, and Q. Li, "A location privacy-preserving system based on query range cover-up or location-based services," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5244–5254, 2020.

[5] J. Zhang, X. Wang, Y. Yuan, and L. Ni, "RcDT: privacy preservation based on R-constrained dummy trajectory in mobile social networks," *IEEE Access*, vol. 7, pp. 90476–90486, 2019.

[6] Z. Sahnoune and A. Esma, "Deloc: a delegation-based privacy-preserving mechanism for location-based services," *International Journal of Mobile Communications*, vol. 19, no. 1, pp. 22–52, 2021.

[7] V. K. Yadav, S. Verma, and S. Venkatesan, "Efficient and secure location-based services scheme in VANET," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13567–13578, 2020.

[8] L. Ni, C. Li, X. Wang, H. Jiang, and J. Yu, "DP-MCDBSCAN: differential privacy preserving multi-core DBSCAN clustering for network user data," *IEEE Access*, vol. 6, pp. 21053–21063, 2018.

[9] X. He, R. Jin, and H. Dai, "Leveraging spatial diversity for privacy-aware location-based services in mobile networks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 6, pp. 1524–1534, 2018.

[10] G. Ruchika and P. Udai, "An exploration to location based service and its privacy preserving techniques: a survey," *Wireless Personal Communications*, vol. 96, pp. 1973–2007, 2017.

[11] P. Zhao, J. Li, F. Zeng, F. Xiao, C. Wang, and H. Jiang, "ILLIA: enabling -anonymity-based privacy preserving against location injection attacks in continuous LBS queries," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1033–1042, 2018.

[12] M. K. Simon and M. S. Alouini, "Digital communication over fading channels," *IEEE Access*, vol. 6, pp. 28328–28338, 2017.

[13] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[14] I. Natgunanathan, A. Mehmood, Y. Xiang, L. Gao, and S. Yu, "Location privacy protection in smart health care system," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3055–3069, 2019.

[15] L. Zhou, L. Yu, S. Du, H. Zhu, and C. Chen, "Achieving differentially private location privacy in edge-assistant connected vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4472–4481, 2019.

[16] R. Jiang, R. Lu, and K.-K. R. Choo, "Achieving high performance and privacy-preserving query over encrypted multidimensional big metering data," *Future Generation Computer Systems*, vol. 78, pp. 392–401, 2018.

[17] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, 2017.

[18] J. Zhang, H. Liu, and L. Ni, "A secure energy-saving communication and encrypted storage model based on RC4 for EHR," *IEEE Access*, vol. 8, pp. 38995–39012, 2020.

[19] J. Liu, H. Li, Y. Gao, H. Yu, and D. Jiang, "A Geohash-based index for spatial data management in distributed memory," in *2014 22nd International Conference on Geoinformatics*, pp. 1–4, 2014.

[20] *Wikipedia*, 2014, http://en.wikipedai.org/wiki/Geohash.

[21] G. M. Morton, "A computer oriented geodetic data base and a new technique in file sequencing," *Physics of Plasmas*, vol. 24, no. 7, pp. 159–173, 1966.

[22] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *in Proc. Int. Conf. on Mobile Syst., Applications and Services. 1em plus 0.5em minus 0.4em*, pp. 31–42, San Francisco, CA, USA, 2003.

[23] C. Y. Chow, M. F. Mokbel, and W. G. Aref, "Casper," *ACM Transactions on Database Systems*, vol. 34, no. 4, pp. 1–48, 2009.

[24] L. Ni, F. Tian, Q. Ni, Y. Yan, and J. Zhang, "An anonymous entropy-based location privacy protection scheme in mobile social networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 2019.

[25] G. Sun, D. Liao, H. Li, H. Yu, and V. Chang, "L2P2: a location-label based approach for privacy preserving in LBS," *Future Generation Computer Systems*, vol. 74, pp. 375–384, 2017.

[26] G. Ruchika and P. Udai, "VIC-PRO: vicinity protection by concealing location coordinates using geometrical transformations in location based services," *Wireless Personal Communications*, vol. 107, pp. 1041–1059, 2019.

[27] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *in Proc. Int. Conf. on Pervasive Services (ICPS)*, pp. 88–97, 2005.

[28] H. Lu, C. S. Jensen, and M. L. Yiu, "PAD: privacy-area aware, dummy-based location privacy in mobile services," in *in Proc. ACM Int. Workshop on Data Engineering for Wireless and Mobile Access*, pp. 16–23, 2008.

[29] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *in Proc. IEEE INFOCOM Conf. on Comput. Communications*, pp. 754–762, 2014.

[30] M. L. Yiu, C. S. Jensen, X. Huang, and H. Lu, "Spacetwist: managing the trade-offs among location privacy, query performance, and query accuracy in mobile services," in *in Proc. IEEE 24th Int. Conf. on Data Engineering*, pp. 366–375, 2008.

[31] T. Hara, A. Suzuki, M. Iwata, Y. Arase, and X. Xie, "Dummy-based user location anonymization under real-world constraints," *IEEE Access*, vol. 4, pp. 673–687, 2016.

[32] J. Zhang, Y. Yuan, X. Wang, and L. Ni, "RPAR: location privacy preserving via repartitioning anonymous region in mobile social network," *Security and Communication Networks*, vol. 2018, 10 pages, 2018.

[33] P. Dilay and P. Udai, "Dummy generation-based privacy preservation for location-based services," in *in Proc. 21st Inter. Conf. on Distributed Computing and Networking (ICDC)*, pp. 1–1, 2020.

[34] Y. Li, D. Kim, and B. S. Shin, "Geohashed spatial index method for a location-aware WBAN data monitoring system based on NoSQL," *Security and Communication Networks*, vol. 12, no. 2, 2016.

[35] N. Guo, W. Xiong, Y. Wu, L. Chen, and N. Jing, "A geographic meshing and coding method based on adaptive Hilbert-Geohash," *IEEE Access*, vol. 7, pp. 39815–39825, 2019.

[36] K. Chatzikokolakis, C. Palamidessi, and M. Stronati, "Geo-indistinguishability: a principled approach to location privacy," in *in International Conference on Distributed Computing Internet Technology*, pp. 49–72, 2015.

[37] X. Dong, T. Zhang, D. Liu, and G. X. Li, "Preserving geo-indistinguishability of the primary user in dynamic spectrum sharing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8881–8892, 2019.

[38] Y. Liu, T. Feng, M. G. Peng, and Z. B. Jiang, "COMP: online control mechanism for profit maximization in privacy-preserving crowdsensing," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1614–1628, 2020.

[39] S. Josefsson, *The Base16, Base32, and Base64 data encodings*, 2006.

[40] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, 2008.

[41] N. Guo, W. Xiong, Y. Wu, L. Chen, and N. Jing, "Exploring historical location data for anonymity preservation in location-based services," in *in Proc. IEEE INFOCOM 27th Conf. on Comput. Communications*, pp. 547–555, 2008.

[42] *Geolife*, 2019, https://www.microsoft.com/.

[43] P. Dilay and P. Udai, "Towards privacy-preserving dummy generation in location-based services," *Procedia Computer Science*, vol. 171, pp. 1323–1326, 2020.

[44] D. Liao, X. Huang, V. Anand, G. Sun, and H. Yu, "k: DLCA: an efficient approach for location privacy preservation in location-based services," in *in Proc. IEEE Int. Conf. on Communications (ICC)*, pp. 1–6, 2016.

[45] Z. Li, X. Zhong, J. Wei, and H. Shi, "The application of Manhattan tangent distance in outdoor fingerprint localization," in *in Proc. IEEE Global Communications Conf. (GLOBECOM)*, pp. 1–5, 2018.

[46] Y. Shitov, "Euclidean distance matrices and separations in communication complexity theory," *Discrete & Computational Geometry*, vol. 61, no. 3, pp. 653–660, 2019.

[47] R. Yarovoy, F. Bonchi, and L. V. Lakshmanan, "Anonymizing moving objects: how to hide a mob in a crowd?," in *in Proc. ACM 12th Inter. Conf. on Extending Database Technology (EDT): Advances in Database Technology*, pp. 72–83, 2009.