

# Optimizational Scheme for Constructing Anonymous Spatial Region under Continuous LBS Queries

Wei Wang

Science and Technology on Communication Information  
Security Control Laboratory, No.36 Research Institute of  
CETC, Jiaying, China, 314033  
E-mail: wwzwh@163.com

Wen-Hong Zhao

Nanhu College, Jiaying University, Jiaying, China,  
314001  
E-mail: wwlofty@gmail.com

Cheng Zhong, Xing-Hua Li

School of Cyber Engineering, Xidian University, Xi'an China, 70771

**Abstract**-K-anonymity technology is one of the important technologies of location privacy protection based on LBS(Location-based service, LBS), it protects users' location privacy by constructing ASR(anonymous spatial region, ASR). Existing construction schemes for ASR based on continuous LBS queries giving rise to the problem of generating a large ASR, which increases query latency and server load, thus affect the QoS(quality of service, QoS) of users. Considering the shortages of existing ASR construction schemes for continuous LBS queries and the influence of aggregation property and moving directions of historical users in ASR, a new ASR construction scheme is proposed in this paper by selecting regions with high frequency of users and introducing the idea of ASR segmentation. The experimental results show that this scheme can further reduce the ASR and improve the QoS under the premise of satisfying users' location privacy demands.

**Keywords**-continuous LBS queries; user aggregation; ASR segmentation; QoS

## I. INTRODUCTION

LBS is a kind of value-added service that provides corresponding services for users according to the position released by users and is supported by location service provider. In LBS, The most commonly used technology is K-anonymity technology, it is first proposed by Sweeney<sup>[1]</sup> in privacy protection issues with publicly available information. After that it was introduced into location privacy protection based on LBS by Gruteser<sup>[2]</sup> et al. Usually, LBS queries are divided into two categories depending on the frequency of users' queries. One is snapshot queries: the user sends only a single query in a certain period of time. The other is continuous queries, that is, the user sends several queries continuously in a certain period of time to obtain the same or different services. In continuous LBS queries, when the user's location privacy demand  $k$  or the number of continuous LBS queries increases, the size of the ASR will increase rapidly to obtain as many common users as possible

because of users' variable moving directions in the ASR, which resulting in the decline of QoS.

To solve the problems in continuous LBS queries, we propose an ASR construction scheme in this paper, which can effectively reduce the ASR and improve the QoS of users by giving priority to the regions with high frequency of historical users and elimination of redundant ASR based on ASR segmentation. The main contributions of this paper are as follows:

- Analyze the construction scheme for current ASR, point out that there is a disadvantage that the ASR is oversized in current scheme and the simulation results show that the ASR is redundant in existing schemes.
- Through analyzing the aggregation property and moving directions of historical users, we design an algorithm which can form candidate ASR based on the selection of regions with high frequency of historical users and an ASR segmentation algorithm based on grid division to form a new ASR construction scheme.
- By contrasting the experiment results, ASR constructed by the scheme in this paper is smaller than that constructed by existing schemes under the premise of meeting users' location privacy demands, and it grows slowly with the increase of users' privacy demand  $k$  and the number of continuous LBS queries. So our scheme has a good stability and effectively improves the QoS.

The rest of this paper is organized as follows: Section II reviews the existing location privacy protection schemes for continuous LBS queries. In Section III, we introduce the detailed privacy protection scheme for continuous LBS queries based on k-anonymity. We give the superiority and validity analysis of our scheme in Section IV. Finally, Section V summarizes our work.

## II. RELATED WORK

This section briefly reviews the existing k-anonymity schemes for location privacy protection based on continuous LBS queries.

For location security threat in continuous LBS queries, Chow and Mokbel[3] et al first proposed and implemented a scheme with ASR constructed by using space anonymity to resist sampling attack and intersection attack. However, all the queries need to adopt the common user set used to form the initial ASR. The ASR generated by this scheme will expand rapidly as the number of continuous LBS queries increases. Tan[4] proposed a method for ASR segmentation, he segments the rectangular ASR which is oversized and single into a number of discrete small rectangular ASRs, the total number of users in these small ASRs should be not less than k, the sum area of these small ASRs should be not less than the minimum size of the ASR required by users and at least one small ASR should contain the user u who sent the continuous LBS queries. To solve the problem that ASR is oversized, Xu[5] et al first proposed a method based on users' historical footprint information, they choose k-1 of the nearest historical traces from the user who send continuous LBS queries to construct ASR. However, the scheme does not consider users' moving directions, leading to a huge amount of calculations and low efficiency. At the same time, Xu[6] et al proposed a feeling-based privacy model, which allowing users to express their privacy requirements by specifying a public region. However, neither of the two methods can meet users' customized privacy demands. Wang[7] et al proposed L2P2 scheme, they gradually expand anonymous regions of each position from the initial position where the continuous LBS queries sent out until all the requested privacy demands are met. Li[8] et al point out that when constructing ASR with other historical users' footprints, it is likely to cause large ASRs and poor QoS. In regions with few continuous LBS queries, the footprints are sparse, they reduce ASR by identifying and constraining these queries. However, this method suppresses users' queries and it still can not effectively improve the QoS of users.

## III. CONSTRUCTION OF ANONYMOUS SPATIAL REGION BASED ON CONTINUOUS LBS QUERIES

There are problems in existing privacy protection schemes for continuous LBS queries that when users' privacy demand k or the number of continuous LBS queries increases, the size of the ASR grows too fast, which increases query latency and server load and seriously lowers the QoS of users. With the problems above, considering the aggregation property and moving directions of historical users, we propose a scheme which giving priority to the regions with high frequency of historical users and

eliminating redundant ASR by ASR segmentation, it further ensures users' QoS under the premise of satisfying users' privacy demands. In this paper, we employ center server[9] structure which is commonly employed in k-anonymous algorithms to store users' historical information and deal with the anonymity processes. Meanwhile, we adopt grid division[10] structure as our data structure. The basic framework is shown in Fig. 1.

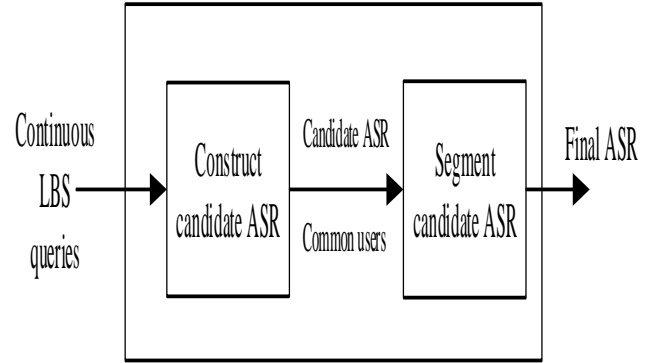


Figure 1. Construction framework of ASR based on continuous LBS queries.

### A. Construction of Candidate ASR

By analyzing users' historical footprint information and combining with the real scene, Liu[11] points out that the distribution of users has aggregation property. Associated with the data structure in this paper, historical users appear frequently in some grids while they appear infrequently in other grids. Therefore, when constructing candidate ASR, we preferred regions with high frequency of users so that more historical users can be chosen with the same size of ASR and it is easier to obtain the common user set under the condition of satisfying users' privacy demands k if the ASR is small.

Since we adopt grid structure in this paper, let  $\sum u$  represent the number of users in the grid and  $\sum u$  directly reflects the aggregation property of the grid. The larger  $\sum u$  is, the more aggregated users are. Conversely, The fewer  $\sum u$  is, The sparser the distribution of users is. When a user makes a continuous LBS queries with  $Q = \{Q_1, Q_2, \dots, Q_n\}$ ,  $Q_i = \langle u, t_i, l_i(x, y), k_i \rangle$ , the present task is divided into two parts:

- Construct the grids with  $[Q_i.x - Q_i.qx, Q_i.y - Q_i.qy] \times [Q_i.x + Q_i.qx, Q_i.y + Q_i.qy]$  by associating the data of historical users with grids according to  $Q_i$ , the location of the user who send LBS query at this time. Using the number of historical users in the grid as their frequency, and

the  $N$  grids with the highest frequency of historical users are selected as the candidate ASR  $A_c^i = \{A_{c_1}^i, A_{c_2}^i, \dots, A_{c_N}^i\}$ .

- We use  $U_c^i = U_{c_1}^i \cup U_{c_2}^i \cup \dots \cup U_{c_N}^i$  to represent the historical users in  $A_c^i$  and  $U_{c_j}^i = \{u_i | u_i \in A_{c_j}^i\}$ . According to  $U_c^i$  in each candidate ASR  $A_c^i$ , we can obtain a common user set  $U_c = U_c^1 \cap U_c^2 \cap \dots \cap U_c^n (U_c \geq k_i)$ . The algorithm in this section is described as below.

---

**Algorithm 1: Construct candidate ASR**

---

**Input:** A sequence of continuous LBS queries

$Q = \{Q_1, Q_2, \dots, Q_n\}$ ;

**Output:** Candidate ASR  $A_c = \{A_c^1, A_c^2, \dots, A_c^n\}$ ;

Common user set  $U_c$ ;

1. for each  $Q_i \in Q$  do  
     $[Q_i.qx - Q_i.qx, Q_i.qy - Q_i.qy] \times [Q_i.x + Q_i.qx, Q_i.x + Q_i.qy]$  ;
  2.  $G_i \leftarrow Q_i.qx, Q_i.qy + Q_i.qy$  ;
  3. Sort( $G_i.g, |U|$ );
  4. while  $j < N$  do
  5.    $A_c^i \leftarrow \text{SelectHighF}(G_i.g_j)$ ;
  6.    $U_c^i \leftarrow A_c^i$ ;
  7. end while
  8. end for
  9.  $A_c = \{A_c^1, A_c^2, \dots, A_c^n\}$ ;
  10.  $U_c = U_c^1 \cap U_c^2 \cap \dots \cap U_c^n (U_c \geq k_i)$ ;
  11. return  $A_c, U_c$ ;
- 

**B. Eliminate Redundant Candidate ASR**

The construction phase of the candidate ASR does not consider the relationship between the moving directions of historical users and the user who send the query. Historical users with different moving directions in current ASR will not arrive in the ASR where the user send his or her next LBS query, which resulting in the redundancy of ASR. These redundant parts of the ASR make no contributions (they contains no common users) but seriously affect the QoS of users. Therefore, under the condition of satisfying the privacy demand  $k$  of the users and  $A_{min}$  (we use  $A_{min}$  to represent the minimum area of ASR), the ASR can be further narrowed and the QoS of users can be further improved by ASR segmentation and the elimination of redundant ASR through the idea of grid division.

The grid structure itself is the segmentation of ASR, and the set consisting the grids where the users in the common user set locate are the effective parts of candidate ASR. The remainder has no common users because of users' inconsistent moving directions. Therefore, for candidate ASR  $A_c^i$  and common user set  $U_c$  formed by query  $Q_i$  in continuous LBS queries, select all grids containing common users according to the statistics of the distribution of all the

users in  $U_c$  located in grid  $G_i$ , and the region consisting of these grids is the final ASR  $A$ , notice that  $A \geq A_{min}$ . The specific process is described in Algorithm 2.

---

**Algorithm 2: Eliminate redundant candidate ASR**

---

**Input:**  $A_c = \{A_c^1, A_c^2, \dots, A_c^n\}$ ; Final common user set  $U_c$ ; Lower bound of privacy demand  $A_{min}$ ;

**Output:** Final ASR  $A = \{A_1, A_2, \dots, A_n\}$ ;

1. for each  $A_c^i \in A_c$  do
  2.    $A_c^i \leftarrow U_c$ ;
  3.    $A_i = \sum \text{SelectHavaUser}(G_i.g_j)$ ;
  4.   if  $A_i < A_{min}$  do
  5.      $A_i = A_i + \Delta A$ ;
  6.   end if
  7. end for
  8. return  $A$ ;
- 

**IV. SIMULATION EXPERIMENT**

We design different simulation experiments and select the scheme of INFOCOM in 2012 as an example to illustrate the superiority and effectiveness of our scheme proposed in this paper from the two aspects of the common user set in the constructed ASR and the size of ASR.

All algorithms are implemented in Java programming language, and the experiments are performed in a PC with a Intel Core i7, 1.7GHz processor, 8GB RAM and Windows 7 operation system. The experimental data in this paper is provided by the GeoLife[12] project of MSRA, the data collected the GPS trajectory of 182 volunteers during about 5 years, including walking, cycling, driving and other transportation modes. We choose about 2000 users' historical trajectory records within the 5th Ring of Beijing (about 20km \* 20km) as the experimental data for our work. As shown in Table I.

TABLE I. THE COMPOSITION OF THE GEOLIFE DATASET

Transportation mode	Distance (km)	Duration (hour)
Walk	10,123	5,460
Bike	6,495	2,410
Bus	20,281	1,507
Car & taxi	32,866	2,384
Train	36,253	745
Airplane	24,789	40
Other	9,493	404
Total	140,304	12,953

**A. Analysis of Superiority of the Algorithm**

In order to prove that users' aggregation property has a very important influence on ASR construction and the size of ASR, we designed a group of experiments to prove the superiority when we take users' aggregation property into consideration and explain the disadvantages of the existing construction schemes for ASR. First, the number of common users in the ASR who send continuous LBS queries for 10 times is counted in L2P2 scheme, after that, according to the frequency of users in the grids which forming the ASR, we

select the top 5%, 10%, 15%, 20%, 25%, 30% and 35% of the regions with the highest frequency of users to calculate the ratio of common users among all of the users. As shown in Fig. 2. It can be seen that with the size of the regions which have a high frequency of common users get larger, the ratio of common users gets increased. When the proportion of the regions with high frequency of common users accounts for 15%, the proportion of the number of common users has reached more than 80%. That is, 15% of the ASR contains 80% of the common users.

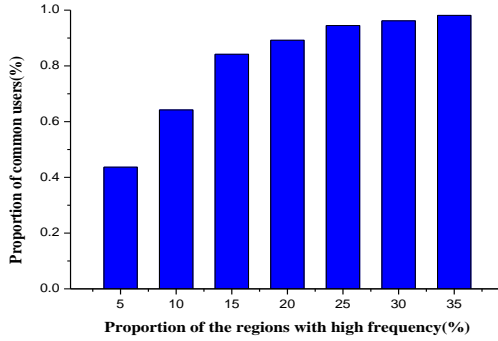


Figure 2. The proportion of the number of common users in the regions with high frequency of common users.

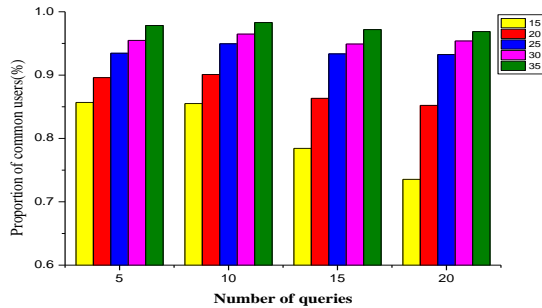


Figure 3. The proportion of common users in the high frequency regions with different number of continuous LBS queries.

Then we make statistics on the proportion of the number of common users in 15%, 20%, 25%, 30%, and 35% of the regions with high frequency of common users in the ASR where the users sent 5, 10, 15, and 20 continuous LBS queries. As shown in Fig.3, We found that 20% of the regions with high frequency of common users contains 85% of common users and 35% of the regions contains more than 95% of common users. Thus it can be seen that the original scheme construct ASR without considering users' aggregation property which result in the existence of a large number of redundant areas, so there are disadvantages in the original scheme. However, the Algorithm 1 of this paper considers the problem.

#### B. Analysis of the Effectiveness of the Algorithm

The size of the ASR directly reflects the QoS obtained by users. It is an important criterion to evaluate the performance of the scheme. In order to further illustrate the effectiveness of our scheme, this section uses different size of ASR generated by different scheme under different privacy demand  $k$  and different number of continuous LBS queries to

describe the scheme in this paper can provide better QoS while ensuring the user's privacy demand.

Alg2 (L2P2) and Alg3 (L2P2) are ASR construction algorithms in L2P2 scheme. Alg1 (Ours) represents the algorithm 1 proposed in this paper. Alg1 + Alg2 (Ours) represents the combination of algorithm 1 and algorithm 2 proposed in this paper. The region with high frequency of common users in algorithm 1 should be selected from 20% to 35% through the experiment above.

##### 1) Influence of privacy demand $k$ on the size of ASR.

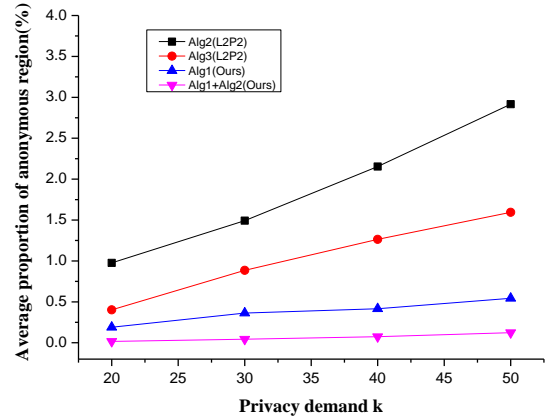


Figure 4. Influence of privacy demand  $k$  on the size of ASR.

As shown in Fig.4, We give the size of the ASR (the percentage of ASR in the total area of the selected map) in different conditions where the privacy demand  $k$  equals to 30,40,50 and 60. When  $k$  increases, the size of ASR increases. However, since the L2P2 scheme does not take into account the aggregation property of historical users, the size of the ASR generated by L2P2 is the largest and grows fastest. The algorithm 1 proposed in this paper can get more common users with the same size of ASR because of the consideration of users' aggregation property. In other words, the size of the ASR is relatively small in this scheme compared with L2P2. With the addition of algorithm 2, the redundant part of the ASR is further eliminated. Through the combination of algorithm 1 and algorithm 2 proposed in this paper, we can get a smaller ASR, and with the increase of privacy demand  $k$ , the size of ASR grows slow in our scheme so the scheme proposed in this paper is more stable.

##### 2) Influence of the number of continuous LBS queries on the size of ASR.

Considering the size of ASR is not only affected by privacy demand  $k$ , the number of continuous LBS queries is also one of the factors. Therefore, we have done experiments based on the change of the size of ASR under different number of continuous LBS queries. The experimental data of 8,12,16,20 continuous queries were selected. As shown in Fig.5, with the increase of the number of continuous LBS queries, the size of the ASR constructed by the two algorithms in L2P2 scheme is still much larger than the algorithms proposed in this paper. After the combination of algorithm 1 and algorithm 2 in this paper, we can still get a smaller ASR in the case of increasing the number of



continuous LBS queries and the size of ASR still grows slow, so our scheme is still stable.

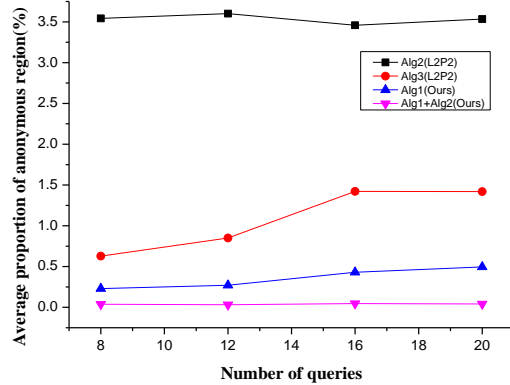


Figure 5. Influence of the number of continuous LBS queries on the size of ASR.

By comparing the size of the ASR constructed based on different privacy demand  $k$  and different number of continuous LBS queries, it can be shown that the scheme in this paper can get a smaller ASR in both above cases, and with the increase of independent variable, the size of ASR grows slowly with more stability so as to effectively reduce query latency and server load. It can make users get better QoS under the premise of meeting their privacy demands. So the scheme in this paper is more superior and effective.

## V. CONCLUSION

In this paper, we focus on the problem of location privacy protection in the scenarios of continuous LBS queries. Considering the shortcomings of existing schemes, we take the aggregation property and moving directions of historical users into account, and we propose a scheme that eliminates the redundancy of ASR by prioritizing areas with high frequency of historical users and the idea based on ASR segmentation. The experimental results show that the scheme proposed in this paper is more superior and effective than the existing schemes, it can further guarantee the QoS of users on the basis of satisfying the their privacy demands.

## REFERENCES

- [1] L. Sweeney . "Achieving  $k$ -anonymity privacy protection using generalization and suppression," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, pp. 571-588, 2002.
- [2] M. Gruteser, D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," Proceedings of the 1st international conference on Mobile systems, applications and services. ACM, pp. 31-42, 2003.
- [3] M. F. Mokbel, C. Y. Chow and W.G. Aref, "The new Casper: query processing for location services without compromising privacy," Proceedings of the 32nd international conference on Very large data bases. VLDB Endowment, pp. 763-774, 2006.
- [4] K.W. Tan, Y. Lin and K. Mouratidis, "Spatial cloaking revisited: Distinguishing information leakage from anonymity," International Symposium on Spatial and Temporal Databases, Springer Berlin Heidelberg, pp. 117-134, 2009.
- [5] T. Xu, Y. Cai, "Exploring historical location data for anonymity preservation in location-based services," INFOCOM 2008. The 27th Conference on Computer Communications. IEEE. 2008.
- [6] T. Xu, Y. Cai, "Feeling-based location privacy protection for location-based services," Proceedings of the 16th ACM conference on Computer and communications security. ACM, pp. 348-357, 2009.
- [7] Y. Wang, D. Xu and X. He, et al, "L2P2: Location-aware location privacy protection for location-based services," INFOCOM, 2012 Proceedings IEEE, pp. 1996-2004, 2012.
- [8] X. Li, L. Deng and S. Gao, et al, "A demand-aware location privacy protection scheme in continuous location-based services," International Conference on Connected Vehicles and Expo. IEEE, pp. 137-150, 2014.
- [9] A.Y. Zhou, B. Yang and C.Q. Jin, et al, "Location-Based Services: Architecture and Progress," Chinese Journal of computer, vol. 34, pp. 1155-1171, 2011.
- [10] Y.G. Zou, Y.H. Zhang, "Location--cloaking algorithm based on grid--divided space," Application Research of Computers, vol. 29, pp. 3059-3061, 2012.
- [11] H. Liu, T. Wang and M. Sun, et al, "Location privacy in sparse environment," Advanced Computer Control (ICACC), 2010 2nd International Conference on. IEEE, vol. 4, pp. 258-261, 2010.
- [12] Y. Zheng, Y. Chen and X. Xie, et al, "GeoLife2. 0: a location-based social networking service," 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware. IEEE, pp. 357-358, 2009.