International Journal of Smart Home
Vol. 10, No. 6 (2016), pp. 137-154
http://dx.doi.org/10.14257/ijsh.2016.10.6.15

A Novel Dummy-Based KNN Query Anonymization Method in Mobile Services

Huan Zhao [1], Jiaolong Wan [1] and Zuo Chen [2]

[1] *School of Information Science and Engineering, Hunan University, Changsha, China*
[2] *School of Information Science and Engineering, Hunan University,Changsha, China*
[1]*{ hzhao, hnu_long}@ hnu.edu.cn,* [2] *chenzuo@iie.ac.cn*

Abstract

Due to the advances of mobile devices with GPS (Global Positioning System), a user's privacy threat is increased in location based services (LBSs).So, various Location Privacy-Preserving Mechanisms (LPPMs) have been proposed in the literature to address the privacy risks derived from the exposure of user locations through the use of LBSs. However, these methods obfuscate the locations disclosed to the LBS provider using a variety of strategies, most of which come at a cost of resource consumption. Therefore, we propose a privacy-protected KNN query anonymization method based on Bayesian estimation for Location-based services. Unlike previous dummy-based approaches, in our method, the request to the LBS server doesn't contain the genuine user location, so we can't calculate whether meet the threshold condition of two location directly, but must to decision making by transition probability. In addition, our method just requires the server returns the results the client needs. Further, we propose an effective search algorithm to improve the server processing. So it can reduce bandwidth usages and efficiently support K-nearest neighbor queries without revealing the private information of the query issuer. An empirical study shows that our proposal is effective in terms of offering location privacy, and efficient in terms of computation and communication costs.

Keyword*: Location privacy protection, Location-based services(LBSs),K-nearest neighbor query, Bayesian estimation*

1. lntroduction

The widespread use of smart mobile devices with continuous connection to the Internet has fostered the development of a variety of successful location-based services (LBSs).Even though LBSs can be very useful, these benefits come at a cost of users' privacy. For instance, user's political beliefs can be inferred by specific information such as duration of stay in a specific building occupied by political parties. Therefore, in order to provide safe and reliable location-based services, user's privacy should be guaranteed.

Numerous researches have been studied to solve the privacy issues arising from the use of LBS. One of current research elaborates on dummy-based anonymization[1-6]. A dummy-based anonymization tries to blur a user's exact location by generating a cloaking region which includes a query issuer and k-1 other dummy locations. One of the recent cloaking-based approach is 2PASS (2-Phase Asynchronous Search)[7]. 2PASS is based on a notion of voronoi cells and each cell contains one object that is the nearest neighbor of any point in its cell. The user can fix the cloaked area to the voronoi cell of the nearest neighbor object, if he/she knows the voronoi cells in advance. In 2PASS, a K-NN query consists of two steps. First, the user requests the voronoi cell information corresponding to the query. Secondly, it selects objects to request. However, 2PASS suffers from

ISSN: 1975-4094 IJSH

bandwidth waste and privacy attack.

To solve the problems, the paper propose a privacy-protected K-NN query processing algorithm based on Bayesian estimation without the true location for location-based services. In the proposed method, the request to LBS server doesn't contain the real user location, that is, clients send the whole dummy locations to the server. Due to the lack of true location, server can't directly calculate two location relation, and have to use other methods. Here, the paper suggest the Bayesian estimation model. Combined without the true location, this approach has many benefits. First, it follow client/server model and don't need the trusted third party. Secondly, it can get the better cloaking and safety, because the request the server received doesn't contain the real user location, just dummies. Thirdly, the paper propose a telescopic search method combined with the feature of the whole dummy, which can greatly improve the processing efficiency of the server. Finally, the proposed method just requires the server returns the results the client needs, which can save a lot of bandwidth. Empirical evaluation shows that our proposals are effective in terms of offering location privacy, are efficient in terms of computation and communication costs, and are flexible in terms of balancing between privacy requirements.

The rest of this paper is organized as follows: the next section reviews related cloaking-based methods. Section 3 describes the proposed method and implementation in detail; and Section 4 explains the experiment. Finally, we conclude in Section 5.

2. Related Work

Recently, considerable research interest has focused on preventing identity inference in LBSs. The main concern is to allow the mobile user to request services without compromising his/her privacy. We can categorize them into four approaches that (a) spatial cloaking, (b) obfuscation, (c) dummy-based anonymization, and (d) encryption.

Spatial cloaking ensures a user's location is mixed with at least k candidates. It collects k users' locations and sends the minimum region including those k users to the server as a query instead of the user's exact location. However, these methods need to pool users' locations, and they thus assume a trusted third-party server to mediate interactions between the users and the LBS server[8-12], or use peer-to-peer collaboration between mobile users[13]. In addition, it is difficult in practice to deploy a completely safe third-party server. Further, these methods fail to anonymize a user's location if there are insufficient numbers of users around him/her.

Obfuscation replaces a user's location with a near-by intersection or building to obscure his/her real location[14, 15]. However, if there are no appropriate targets around the user, the substitute location is far from that of the user, which degrades the quality of the LBS.

The third approach generates dummies and sends their locations with the actual user's location to the server[2, 3, 5, 6], as was described in Figure. 1. It outlines an example where a user is issuing a query asking for near-by hotels within 1km; this approach sends the user's location with the locations of dummies. The LBS provider then returns lists of hotels that are close to each of the locations in the query (the user's and dummies' locations). The user can choose a hotel from the list by ordering results based on the distances from his/her location.

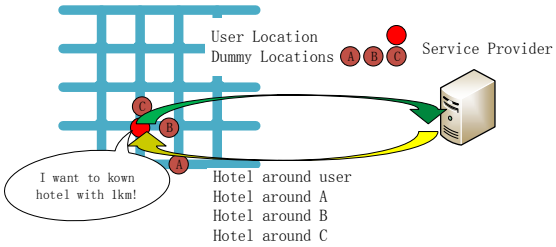


Figure 1. Example of Dummy-Based Approach

Finally, we note that other existing techniques include cryptographic protocol-based approaches [16-17]. Based on a specific transformation fully known only to the clients, the server processes user queries without the ability to decipher exact user locations. The drawback of such techniques is that the query results returned by the server do not offer correctness guarantees.

3. Proposed Method

The proposed method anonymizes a user's location with all the dummies based on Shokri et al.'s framework[18] in a real environment. This section first presents some assumptions and definitions about LBSs, next discusses how to achieve anonymization and query in a real environment, then describes the implementation in detail, finally analyzes the communication cost.

3.1 Definitions and Assumptions

3.1.1. Definitions: DEFINITION 1. (Position) Users' location submitted to the server is abstracted as Pos = (pos_1, pos_2 , …, pos_k), where the pos_i is dummy location around user's real location.

DEFINITION 2. (Query) A location-dependent query is abstracted as Q = (pos, r, d, k, Req), where parameter pos is the mobile user location, parameter r is the cloaking radius, parameter d is the query radius, parameter k is the number of required return and parameter Req denotes user-specified predicates. We call such a query Q the original query. With the location dummy approach, the original query is typically converted into a query Q′= (Pos, r, d, k, Req), where the Pos consists of k dummy locations, and Req is the original query predicate that applies to all k locations. We call query Q′ a location privacy query, since it hides the user location.

DEFINITION 3. (T) T is composed of a collection of triple (u, r, d), where constituents in the triple correspond, respectively, to:

I. user ID: a value can uniquely identify a user.
II. repeat: repeat is equal to the number of intersection, as showed Figure.4. For the two groups of user's position data, we compare their distance with k^2 times. If distance is less than d (d represents query distance), we make repeat plus one.
III. distance: here's distance is not two users' real distance, but based on statistics. This distance can be calculated in two ways: the first way will see the sum of k^2 group distance from two users as the distance, we call this as summation method; the second finds minimum circle center, which contains each user k group position data, and then calculate the center distance between the user and the target user as two users' distance. We call it as circle center method.

3.1.2. Assumptions: ASSUMPTION 1.For the two groups of users' position data, we compare their distance with k^2 times. If the distance is less than d (d represents query distance), we make repeat plus one, and assume the bigger the repeat, the closer to users

each other. This assumption is easy to prove, because if two users has closer position, the greater number generated in a hidden area of the dummy location falls within the scope of the query.

ASSUMPTION 2. For the same number of the repeat, the smaller user distance, the closer to the target user. Here distance means the statistical distance.

3.2. Privacy-Area Dummy Generation Algorithms

There are two kinds of way to generate dummy locations, at random and in control. One way is to generate the dummy locations totally at random. Besides, we are interested in the other approach for generating the dummies that aid in satisfying some conditions, too. The paper thus propose two privacy-area dummy generation algorithms. The one algorithm uses a ran

3.2.1. Random-Based Dummy Generation: The random dummy generation is easy. We just need to generate k points within a given cloaking region. The algorithm, called RandomDummy, is shown in Algorithm 1.The algorithm first initializes an empty set P (line 1).Then, it randomly generates k dummy locations, and enters all positions into a set P (lines 2-6). Finally, it returns the position set P. Point distribution looks like the example in Figure. 2.

Algorithm1 RandomDummy

Input:
user position **pos**, anonymity **k**, cloaking radius **r**
Output:
dummy location set **P**
1:**P**←ϕ
2:for i from 0 to **k**-1 do
3: $pos_{i,x}$←(random (1)*2-1)***r**+ $\mathbf{pos}_x$
4: $pos_{i,y}$←(random (1)*2-1)***r**+ $\mathbf{pos}_y$
5: append pos_i to **P**
6:end for
7: return **P**

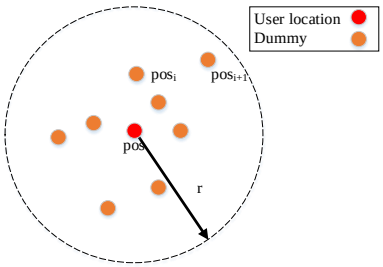


Figure 2. Distribution of Dummy based on Random

3.2.2. Circle-Divided Dummy Generation: To understand the idea behind the circle-divided dummy generation, consider the example in Figure. 3, where k = 7 and pos is the user location. All dummy locations are constrained by a circle centered at position pos with cloaking radius r. Every dummy location pos_i is determined by random r_i and θ. All positions are distributed in such a way that all θ_s are equivalent.

The algorithm, called Circle-Divided Dummy, is shown in Algorithm 2.The algorithm first initializes an empty set P (line 1).Next, it calculates the angle θ between any

consecutive position pair (line 2).Then, k dummy locations are generated as discussed already: their distances to the circle center pos are constrained by random r_i, while they are scattered evenly in terms of their angles with respect to pos (lines 3-8). Finally, having generated all dummies, dummy location set P is returned.

Algorithm2 Circle-Divided Dummy

Input:
user position **pos**, anonymity **k**, cloaking radius **r**
Output:
dummy location set **P**
1:**P**←ϕ
2:θ←2π/**k**
3:for i from 0 to **k**-1 do
4: r_i←random(1)***r**
5: $pos_{i,x}$←r_i*cos(i*θ)+ $\mathbf{pos}_x$
6: $pos_{i,y}$←r_i*sin(i*θ)+ $\mathbf{pos}_y$
7: append pos_i to **P**
8:end for
9: return **P**

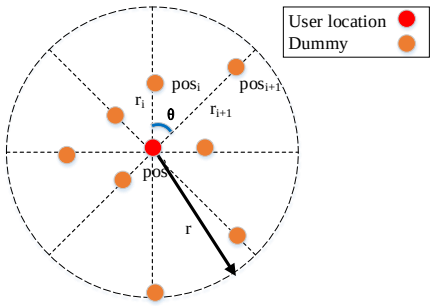


Figure 3. Distribution of Dummy Based on Circle

3.3. Anonymity Query

Under normal circumstances, by comparing the distance of the two points with predetermined threshold, we determine whether a spot is in close proximity to another point with a certain range. In the proposed method, however, in order to anonymize, without the user's true location, we can't directly calculate. More worse, due to the lack of true location, this method could not find any meet the requirements, so we have to consider other methods. In the paper, we suggest a method based on Bayesian statistics. Assuming that the two users' anonymization location distribution as shown in Figure.4, when the user A pos_i is used to inspect whether user B is within the region d, to find user B, we cannot arbitrarily determine whether the true location of B is in the range of A. And we need to further calculate the pos_{i+1}.By k^2 group of data comparison, we are still unable to accurately determine whether the true location of B is in the range of A. Fortunately, through the Bayesian estimation, we can use the transition probability to make the best decision, although there exists a certain error. Through establishing a Bayesian model and training it, we can hypothesize whether B is within A range. In the experiment part, we will in detail introduce the Bayesian model process and performance.

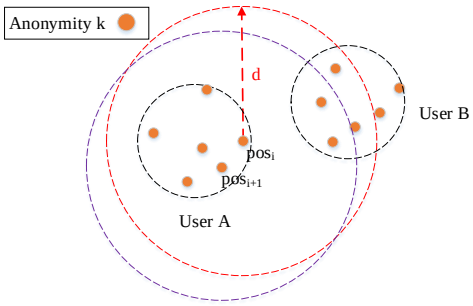


Figure 4. Anonymity Query

3. 4. Server-Side Processing

Having received a location privacy query $Q' = (pos_1, pos_2, \ldots, pos_k, r, d, K, Req)$ from a mobile client, the server processes this according to predicate Req, and returns K results.

Each user sends k anonymization locations to the server, if the server directly compares them the computational complexity will be $O(Nk^2)$ (N is the user scale), causing a great processing burden. Therefore, in order to speed up the server processing, the paper propose a telescopic search method. Telescopic search method consists of QuickSearch algorithm and AccurateCompare algorithm, and it works as follow. First, randomly select a position data from the target user and other users separately. Next, make a comparison in a larger range D. If distance is less than D, the next step will accurately compare. Otherwise, it is impossible to meet proximity, and to be discarded. Since most users do not meet proximity with the target user under the given conditions, the proposed method can cut down the computational complexity to tend to O(N), greatly reducing the calculation.

We have mentioned rough query radius D above, then how do we determine the D? To understand the idea how to determine the range of D, consider the example in Figure. 5. Red and brown solid is cloaking area, where hide the target user and other users separately, and red diagonal line is query range, and blue diagonal line is error range. As we know, we have:

$$D>d+r \quad (1)$$

If a certain distance is greater than D between the target user and other users, they will be impossible to meet proximity relationship. So this user won't be the user we need, and it does not make sense to make a further comparison. Obviously, we must meet the condition (1), when we consider the value D.

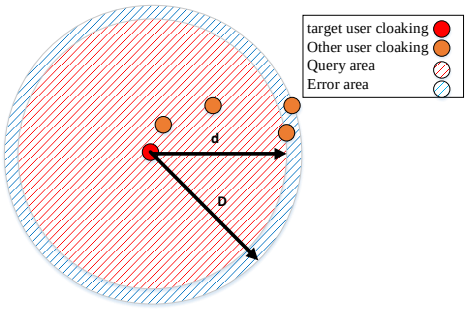


Figure 5. Quick Search

As described above, server-side processing is divided two parts. Next, we will discuss the implementation in detail.

The algorithm, called QuickSearch, describes how to search, and it is shown in Algorithm 3.The algorithm first initializes an empty set S (line 1).Next, server randomly

selects one of positions in location privacy query Q′ from target user (lines 2-4).Similarly, server randomly selects one of positions from others (lines 6-8).Then, server calculates the distance between two selected positions, and compares whether their distance is less than D. If their distance is less than D, server will add user's ID and position data to S using hash table. Otherwise, server will discard it and calculate next one (lines 9-14). Finally, S is returned.

Algorithm3 QuickSearch

Input:
location privacy query **Q'**, anonymity **k**,quick query radius **D**,others' positions set **userDataSet**
Output:
quick search result set **S**
1:**S**←ϕ
2:i←random(0,**k**-1)
3: userQueryPos← get position from **Q′**
4:pos_i←userQueryPos[i]
5:foreach (userData in **userDataSet**) do
6: j←random(0,**k**-1)
7: userPos←get position from userData
8: pos_j←userPos[j]
9: if dist(pos_i,pos_j)<**D** then
10: userID←get user ID from userData
11: **S**[userID]←userPos
12: else
13: continue
14: end if
15: end foreach
16: return **S**

Obviously, the result S derives from a rough comparison. Users in S do not inevitable correspond to the expectations, so LBS server needs further comparison. The algorithm, called AccurateCompare, describes how to get result needed from S, and it is shown in Algorithm 4. In algorithm 4, we adopts the summation method to calculate distance. Alternatively, we can use the circle center method to calculate distance. In our experiments, we test both. Incidentally, we use Ritter's algorithm to calculate circle center.

Algorithm4 AccurateCompare

Input:
location privacy query **Q'**, anonymity **k**, query radius **d**, POIs number **K**, quick search result set **S**
Ouput:
K-nearest neighbor POIs set **R**
1: **R**←ϕ, T←ϕ
2: userQueryPos←get position from **Q'**
3: foreach (userID in **S**) do
4: count←0, distance←0
5: userPos←**S**[userID]
6: for i from 0 to k-1 do
7: pos_i←userQueryPos[i]
8: for j from 0 to k-1 do
9: pos_j←userPos[j]
10: if dist(pos_i,pos_j)<d

```
11:                     count←count+1
12:             else
13:                     continue
14:             end if
15:             distance←distance+ dist(pos_i,pos_j)
16:             end for
17:   end for
18:   if count>0 then
19:             t←(userID,count,distance)
20:             append t to T
21:   else
22:             continue
23:   end if
24:end foreach
25: T←sort T
26: R←T(1:K)
27: return R
```

The AccurateCompare algorithm first initializes empty set S and T (line 1). Next, server gets target user' position data from location privacy query Q′ (line 2). Then, for each user in S, server compares their distance with target user, records the repeat, and calculates the distance (lines 6-17). Then, if the repeat is more than 0, server adds a triple to T, otherwise continue (lines 18-23). Then, server sorts T by primary keyword repeat ascend and secondary keyword distance descend (line 25).Then, server packages the first K in T as a result set (line 26). Finally, R is returned.

3. 5. Communication Cost Analysis

3.5.1. Upstream Communication Cost: With a 2D location taking up 8 bytes, the size of a raw request containing a total of k locations is expressed as:

$$|Req_{raw}|= 8k \quad (2)$$

Restructuring a raw query request by putting all x coordinate values in front of y values, the cost can be reduced to roughly $16\sqrt{k}$.

3.5.2. Downstream Communication Cost: Let R_i be the result that corresponds to the i-th position in query Q′. In previous methods, the size in bytes of a raw result message without packing is:

$$res_{raw}=8\sum_{i=1}^{k} R_i \quad (3)$$

With the packing described in reference[3], the result message size shrinks to:

$$res_{pack}=(8+\lceil k/8\rceil)|\bigcup_{i=1}^{k} R_i| \quad (4)$$

However, in proposed method, the result message size is reduced to R, because it don't need to return independent data.

4. Performance Analysis

4. 1. Setting for Evaluation

In this section, we will present the performance of the proposed K-NN query. Table 1 shows the experimental environment.

Table 1. Experimental Environment

CPU	Intel Dual-Core E6700 3.2GHz
RAM Memory	2GB
Operation System	Window 7
Simulator	Python 2.7

During the experiments, we use the simulated data set. we compare our work with the exiting approach PAD[3](Since the method[5, 6] only consider the use of real situations, and not improves algorithm performance , and therefore we don't compare with them)in terms of response time and bandwidth size. The response time can be defined as how quickly the server returns the result set after receiving the query. The bandwidth can be defined as page size that encloses objects. The parameter settings are summarized in Table 2, with default values given in bold.

Table 2. Simulation Parameter Settings

Parameter	Setting
Spatial extents	10*10km^2
User cardinality N	10^4,10^5,2*10^5,**5*10^5**,10^6
Anonymity k	2,4,8,16,**32**,64
Cloaking radius r	50,**100**,200,500,1000
Query radius d	100,200, 500,800,**1000**
Nearest neighbor K	10,20,**30**,40,50

4. 2. Communication Cost

The proposed method just reduces the downstream communication cost, so only the downstream communication cost is analyzed here. Figure.6 shows that under different anonymity k, the server returns the message size in bytes against the PAD and proposed methods. The PAD not only returns the real requirement, also return extra k-1 useless results, thus resulting in a tremendous waste of bandwidth. However, the proposed method is able to avoid this problem very well, and only returns the results client needs, significantly reducing bandwidth utilization.

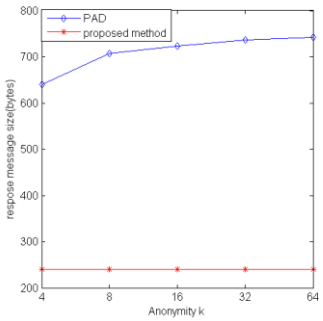
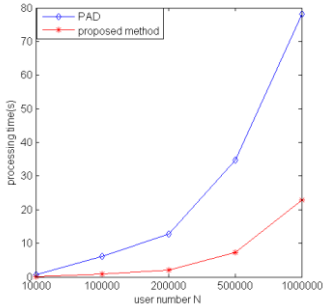


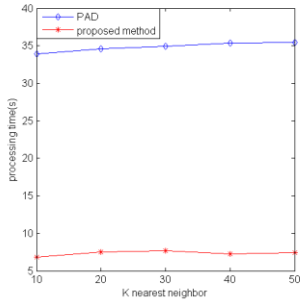
Figure 6. Downstream Communication Cost

4. 3. Server-Side Cost

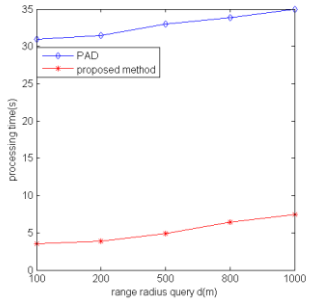
Figure .7 (a) (b) (c) (d) (e) report the processing performances of the PAD and the proposed method. Under different conditions, their computational efficiency shows a big difference. Using the proposed method can greatly optimize the processing time, and reduce the burden on the LBS server.



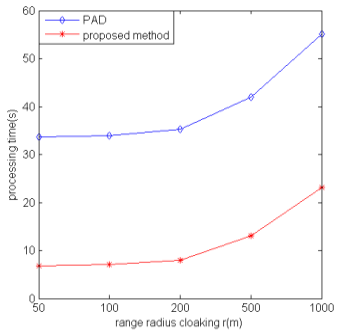
(a) K= 30, d=1000, r =100, k = 16



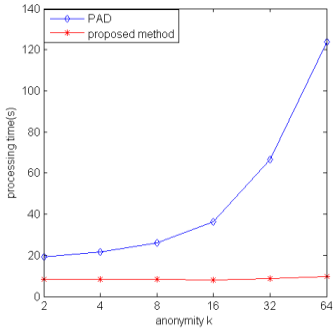
(b) N=5*105, d=1000, r=100,k= 16



(c) N= 5* 105,K =30, r =100, k= 16



(d) N = 5*105,K = 30, d = 1000, k = 16



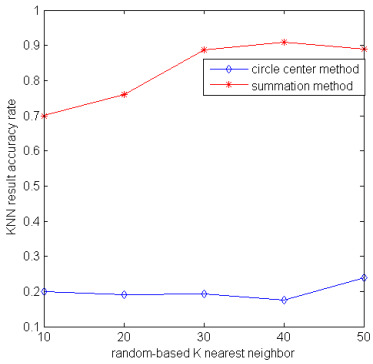
(e) N = 5 *105,K = 30, d = 1000, r =100

Figure 7. Algorithm Performance Comparison

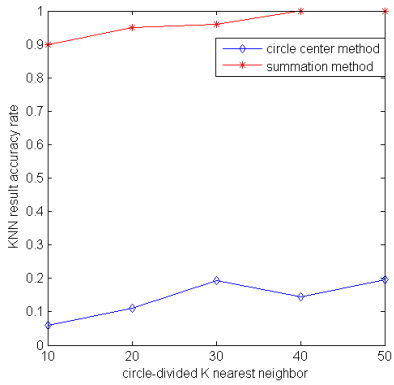
4. 4. Result Accuracy Rate

In the previous part, we describes that the results are calculated based on Bayesian statistics. Therefore, not all KNN results are just what client wants. In other words, it will perhaps bring some errors. Therefore, it is necessary to analyze the accuracy of the query results.

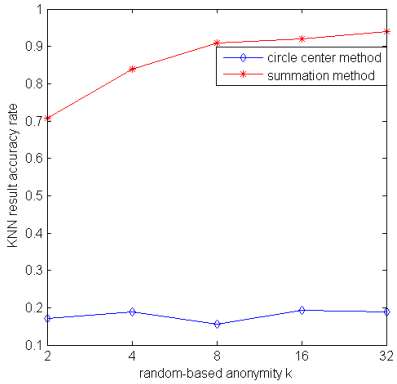
As is showed in Figure. 8, the dummy generation based on circle-divided (b) (d) (f) (h) (j) performs better than that based on random (a) (c) (e) (g) (i). Combined with summation method, the circle-divided dummy generation can get the higher accuracy. The reason is that the distribution from circle-divided dummy generation keep more stable and uniform than random-based, and corresponding distance is closer to the real one. Eventually, it has a better performance. However, the disadvantage is not hidden so well. In contrast, random-based dummy generation distributes more casual and hides better. Unfortunately, from Figure. 8(b), (d), (f), (h), (j), we can conclude that compared with summation method the circle center method performs very badly. To some extent, it fully proves that the proposed method have a good cloaking.



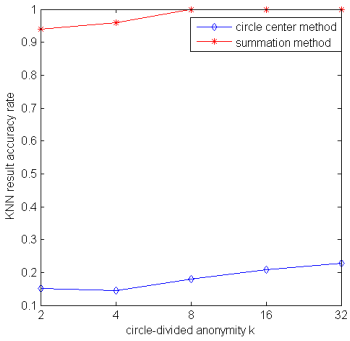
(a) N = 5*105, d = 1000,r = 100,k = 16



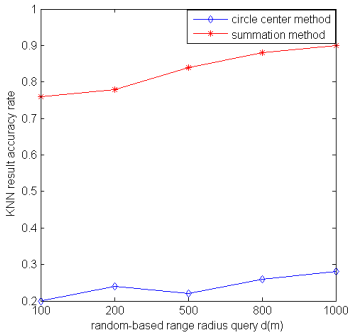
(b) N = 5*105,d = 1000,r = 100,k = 16



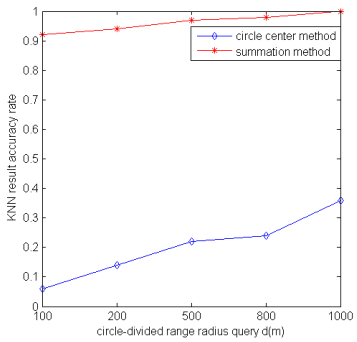
(c) N = 5*105,K = 30,d = 1000,r =100



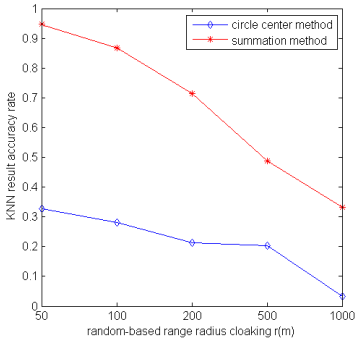
(d) N = 5*105,K = 30, d = 1000, r =100



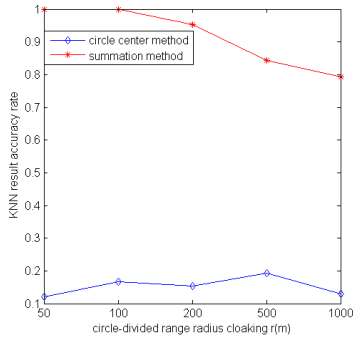
(e) N = 5 *105,K = 30, r = 100, k = 16



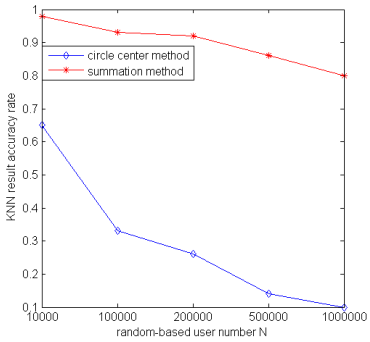
(f) N = 5 *105,K = 30, r = 100, k = 16



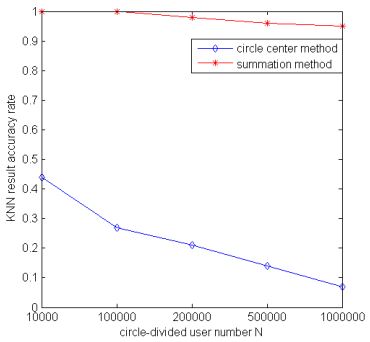
(g) N= 5*105,K = 30, d = 1000, k = 16



(h) N = 5* 105,K = 30, d = 1000, k = 16



(i) K = 30, d = 1000, r = 100, k = 16



(j) K = 30, d = 1000, r = 100, k = 16

Figure 8. Result Accuracy Rate

As previously mentioned, the accuracy of the query results is influenced by user density, anonymity k, cloaking radius r, query radius d and nearest neighbor number K. Next, we will analyze these factors how to work. Figure 8. (a) (b) shows that as K increases, the accuracy of results is also rising. The reason is that when K is large enough

the random impact becomes very weak. Next, Figure 8. (c) (d) report that as the anonymity requirement k grows, the result accuracy doesn't improves significantly. But when k is too small, the accuracy would produce a big fluctuations. Because too small k cannot search the whole proper points in the scope, resulting in an unstable results. Then, Figure 8. (e) (f) (g) (h) indicate that as the ratio of r/d is larger, the result accuracy becomes the higher. However, a bad situation is when the ratio r/d is more than 1, the accuracy will drop dramatically. The reason is that when r is very close to d, it will cause great probability that generating dummy locations deviate from user original location, and an extreme case is that all the dummies fall on the cloaking area border. While the ratio does not weaken the effect, results will cause errors dramatically. Figure 8 (i) and (j) describe that user density has an important influence on the query results for circle center algorithm, but not very clear for summation algorithm when combined with the circle-divided dummy generation.

4. 5. Anonymous Area Achieving Variance

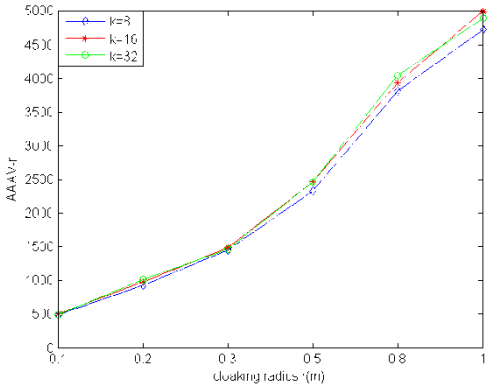
The paper adopts two evaluation criteria to measure the performance of summation method in terms of satisfaction with anonymity.

4.5.1. Anonymous Area Achieving Variance-r (AAAV-r): This metric was aimed at measuring the satisfaction rate for anonymous area requirements. We discuss the anonymity of service requests where the size of the anonymous area during the simulation time. The AAAV-r is defined as the variance of the location of the speculation (different r) and the true location. Thus, the greater the AAAV-r, the better the location anonymity.

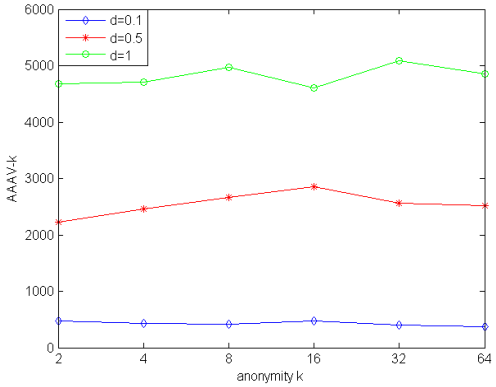
4.5.2. Anonymous Area Achieving Variance-k (AAAV-k)

This metric was aimed at measuring the satisfaction rate for anonymous area requirements. We discuss the anonymity of service requests where the number of the dummy location. The AAAV-k is defined as the variance of the location of the speculation (different k) and the true location. Thus, the greater the AAAV-k, the better the location anonymity.

Figure 9 (a) shows that when using Ritter's algorithm to estimate target's true location with dummies, the greater cloaking area, the greater AAAV-r. Obviously, the large cloaking radius provides more uncertain, and make a good hide. In reality, when cloaking radius up to 1km, the proposed method can protect user location privacy very well. However, according to Figure 9 (b), data shows that the greater k don't contribute to cloaking with effect, so it is no mean producing more dummies. But in order to keep the accuracy, enough dummies are necessary.



(a) Anonymous Area Achieving Variance-r (AAAV-r)



(b) Anonymous Area Achieving Variance-k (AAAV-k)

Figure 9. Anonymous Area Achieving Variance

5. Conclusion

In this paper we propose a novel dummy-based KNN query anonymization method in LBSs with Bayesian model, which can effectively reduce communication cost. In addition, we report two dummy generation algorithms that take into account privacy. At the same time, we propose a telescopic search method for the result query, which can greatly improve the processing efficiency of the server. At last, we show an empirical evaluation and validate the proposed method is effective in offering area-based location privacy. Of course, the proposed KNN query is based on transition probability and Bayesian statistics, so there exists a certain degree of error. The paper in the experiment section analyzes how the factors affect the error. As for how to choose the better parameters to further minimize the errors in specific cases, it will be the next direction we focus on.

Acknowledgements

This work was supported in part by the Scientific Research Plan of Hunan Provincial Science and Technology Department of China (2014FJ4161), supported by the Scientific Research Plan of Changsha Science and Technology Department (k1403027-11).

References

[1] Herrmann M, Troncoso C, Diaz C, Preneel B. Optimal sporadic location privacy preserving systems in presence of bandwidth constraints. Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society: ACM; 2013. p. 167-78.
[2] Kido H, Yanagisawa Y, Satoh T. An anonymous communication technique using dummies for location-based services. Pervasive Services, 2005 ICPS'05 Proceedings International Conference on: IEEE; 2005. p. 88-97.
[3] Lu H, Jensen CS, Yiu ML. Pad: privacy-area aware, dummy-based location privacy in mobile services. Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access: ACM; 2008. p. 16-23.
[4] Olumofin F, Tysowski PK, Goldberg I, Hengartner U. Achieving efficient query privacy for location based services. Privacy Enhancing Technologies: Springer; 2010. p. 93-110.
[5] Suzuki A, Iwata M, Arase Y, Hara T, Xie X, Nishio S. A user location anonymization method for location based services in a real environment. Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems: ACM; 2010. p. 398-401.
[6] Kato R, Iwata M, Hara T, Suzuki A, Xie X, Arase Y, et al. A dummy-based anonymization method based on user trajectory with pauses. Proceedings of the 20th International Conference on Advances in Geographic Information Systems: ACM; 2012. p. 249-58.
[7] Hu H, Xu J. 2PASS: Bandwidth-optimized location cloaking for anonymous location-based services. Parallel and Distributed Systems, IEEE Transactions on. 2010;21:1458-72.

[8] Gruteser M, Grunwald D. Anonymous usage of location-based services through spatial and temporal cloaking. Proceedings of the 1st international conference on Mobile systems, applications and services: ACM; 2003. p. 31-42.
[9] Mokbel MF, Chow C-Y, Aref WG. The new Casper: query processing for location services without compromising privacy. Proceedings of the 32nd international conference on Very large data bases: VLDB Endowment; 2006. p. 763-74.
[10] Chow C-Y, Mokbel MF. Enabling private continuous queries for revealed user locations. Advances in Spatial and Temporal Databases: Springer; 2007. p. 258-75.
[11] Talukder N, Ahamed SI. Preventing multi-query attack in location-based services. Proceedings of the third ACM conference on Wireless network security: ACM; 2010. p. 25-36.
[12] Yi X, Paulet R, Bertino E, Varadharajan V. Practical k nearest neighbor queries with location privacy. Data Engineering (ICDE), 2014 IEEE 30th International Conference on: IEEE; 2014. p. 640-51.
[13] Chow C-Y, Mokbel MF, Liu X. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems: ACM; 2006. p. 171-8.
[14] Duckham M, Kulik L. A formal model of obfuscation and negotiation for location privacy. Pervasive computing: Springer; 2005. p. 152-70.
[15] Šikšnys L, Thomsen JR, Šaltenis S, Yiu ML, Andersen O. A location privacy aware friend locator. Advances in Spatial and Temporal Databases: Springer; 2009. p. 405-10.
[16] Indyk P, Woodruff D. Polylogarithmic private approximations and efficient matching. Theory of Cryptography: Springer; 2006. p. 245-64.
[17] Khoshgozaran A, Shahabi C. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. Advances in Spatial and Temporal Databases: Springer; 2007. p. 239-57.
[18] Shokri R, Theodorakopoulos G, Troncoso C, Hubaux J-P, Le Boudec J-Y. Protecting location privacy: optimal strategy against localization attacks. Proceedings of the 2012 ACM conference on Computer and communications security: ACM; 2012. p. 617-27.

Authors

Huan Zhao. Was born in 1967, she is a professor at the School of Information Science and Engineering, Hunan University. She obtained her B.Sc. degree and M.S. degree in Computer Application Technology at Hunan University in 1989 and 2004, respectively, and completed her Ph.D. in Computer Science and Technology at the same school in 2010. Her main research interests include speech information processing, embedded system design and embedded speech recognition. She served as visiting scholar at the University of California-San Diego (UCSD), USA during the period of March 2008 to September 2008. The visiting scholarship was appointed and sponsored by the China Scholarship Council (CSC). Prof. Zhao is a Senior Member of China Computer Federation, Governing of Hunan Computer Society, China and China Education Ministry Steering Committee. **Email:** hzhao@hnu.edu.cn

Jiaolong Wan. Received his B.Sc. degree in Communication Engineering at the School of Information Science and Engineering, Hunan University, P. R. China in 2013. Currently, he is a M.S. candidate of Hunan University, P. R. China. His main research interests include location privacy and big data analytics. Email: hnu_long@hnu.edu.cn

Zuo Chen. Received his PhD degree in Hunan University in 2008. He is an assistant professor at College of Computer Science and Electronic Engineering, Hunan University. He is currently a postdoctoral of IIE CAS. His research interests include wireless sensor network, data mining and so on. Email: chenzuo@iie.ac.cn