International Journal of Security and Its Applications
Vol.9, No.9 (2015), pp.387-400
http://dx.doi.org/10.14257/ijsia.2015.9.9.33

Non-deterministic K-anonymity Algorithm Based Untrusted Third Party for Location Privacy Protection in LBS

Jinying Jia and Fengli Zhang

School of Computer Science & Engineering, University of Electronic Science and Technology of China, Chengdu, China
jiajinying@126.com

Abstract

When people use the LBS, they will leak their location information to an untrusted LBS provider. A new technology named location privacy protection has been well studied by scholars. But in their researches, they supposed that there was a trusted third party which could provide anonymous services for the query user. However, it is very difficult to find a trusted third party in practice. In this paper, we proposed a non-deterministic k-anonymity algorithm based untrusted third party for location privacy protection in LBS. It moved the process of the generating anonymous spatial region (ASR) from the third party to the users, thereby reduced the credibility of the third party from trusted to untrusted. And introduced an incremental query for KNN. The experiments demonstrate that our proposed algorithm has better performance than existing algorithms.

Keywords: *location based service; location privacy; spatial cloaking; k-anonymity*

1. Introduction

With the development of 3G net and 4G net, the smart phones have been becoming popular in the young people. At the same time, the location-based services have appeared and have become popular. While enjoying the convenience of location-based services, the people have to send their precise location information to the untrusted location service providers. The untrusted location service providers may betray the user's location information for economic benefits. In recent years, there have been disputes and lawsuits about the leakages of the user's location information, and even some films have appeared in the event which the man was tracked as his smart phone revealed his location information.

In order to let the people use location-based services without revealing their precise location information, a lot of scholars proposed many algorithms for location privacy protection. However, these algorithms did not consider the credibility of the centralized anonymizer. The attacker can obtain the users' location information by attacking and controlling the centralized anonymizer. Or the centralized anonymizer may betray the user's location information for economic benefits, too.

In order to solve the above problems, we have proposed twice anonymous algorithm. But that algorithm unified the centralized anonymizer's credibility for all the users. And it's processing efficiency was low for KNN query.

In this paper, we proposed a non-deterministic k-anonymity algorithm based untrusted third party for location privacy protection in LBS. It moved the process of the generating ASR from the third party to the users, thereby reduced the credibility of the third party from trusted to untrusted. And introduced an incremental query for KNN.

To summarize, our contributions in this paper are as follows:

- We moved the process of the generating ASR from the third party to the users. And used latitude and longitude grid instead of the grid-area in twice anonymous algorithm. The users did not have to report their single grid-area ID to the centralized

ISSN: 1738-9976 IJSIA

third party. Each user generated an ASR according to their own privacy requirements, and reported the ASR to the centralized third party. The ASR was composed of the grids of latitude and longitude. The users in the ASR were not less than k (k was user's privacy requirement) according to count using equivalent probability, and the ASR was not less than the minimum area of user's privacy requirement. Let the trusted centralized anonymizer become an untrusted anonymous assistant.

- Hybridized the twice anonymous algorithm with SpaceTwist algorithm, Solved the problem of inefficient of the twice anonymous algorithm for KNN query.

The rest of the paper proceeds as follows. Section 2 reviews existent work on location anonymity and privacy-aware LBS. Section 3 presents the system architecture. Section 4 presents the generated ASR algorithm and the proxy search algorithm at the centralized anonymous assistant end. The experimental results are shown in Section 5. The conclusion and future work is in Section 6.

2. Related Work

In mobile computing, especially for LBS, location anonymity has attracted intensive research as a solution to protect user privacy [1]. In order to allow a mobile user to request services without disclosing her position, location cloaking is proposed [2-4]. It sends to the server an ASR that contains the genuine user position and is large enough to satisfy some private metric [5-7]. The two most widely adopted metrics are k-anonymity — this region must contain at least k users so that the genuine requesting user is indistinguishable from at least k-1 other users who have the same cloaked region, and granularity — the area of this region must exceed a threshold.

Interval Cloak [8] designed by Gruteser and Grunwald is one of the first cloaking techniques. Their idea is to partition a region using a quad-tree. If the leaf node of the quad-tree where the inquirer is being in has not less than k users, then the leaf node region is returned. Otherwise, it will search the farther node of the leaf node for the next iteration. The Casper Cloak [9-10] algorithm proposed by Mokbel *et al.* is just the same with Interval Cloak, if the node has less than k users, it searches the brother nodes firstly, it still has less than k, it searches the further nodes. And it uses a B-hash to accelerate the search. Gedik and Liu proposed the algorithm Clique Cloak [11]. Users can choose different values for k to specify their personal privacy requirements in this algorithm. This is the major improvement. Kalnis P. et al. proposed the algorithm nnASR[12]. It finds the nearest k users from the inquirer firstly, after that it returns the area which can contain these users as the ASR.

However, these algorithms did not consider the credibility of the centralized anonymizer. The attacker can obtain the users' location information by attacking and controlling the centralized anonymizer. Or the centralized anonymizer may betray the user's location information for economic benefits, too. In order to solve the above problems, we have proposed twice anonymous algorithm [3-4]. It moved the judgment right of user's ownership of a small area from the centralized anonymizer to the users. And used the two-dimensional table instead of the quad-tree, let users report their grid area ID instead of their precise location, so it solved the problem of the user's precise location information may be leaked out from the centralized anonymizer. But it unified the centralized anonymizer's credibility for all the users, some users may feel that the grid area was too small for them. And it's processing efficiency was low for KNN query.

Yiu *et al.* proposed SpaceTwist algorithm[13], where the user repeatedly issues KNN queries from dummies, which they called anchors, until the KNN result for the genuine location was guaranteed. But this framework did not belong to the k-anonymity. It was not safe enough.

3. System Architecture

In this section, we give the definition for the gird of latitude and longitude, privacy threat model, and the system architecture.

Definition 1: Every point on the Earth can be represented by a latitude and longitude coordinates. For example, "(N 30°12.511′, E 103°22.379′)"is stand for a point in Chengdu, China. As the earth's polar radius is 6357km, one minute of latitude represents approximately 1.848km. Because the city of Chengdu is between N30°05 ′～N31°26′, and the equatorial radius of the earth is 6378km, one minute of longitude represents approximately 1.606km. We remove the fractional part of the minute for latitude and longitude coordinates, it can form a grid of latitude and longitude, it is abbreviated as GLL.

Every GLL is presented as the degree and minute of its latitude and longitude coordinates. For example, the GLL of the point "(N 30°12.511′, E 103°22.379′)"is "(N 30°12′, E 103°22′)".

Definition 2: Non-deterministic k-ASR is a rectangle which is made up of n adjoining GLL. And the users in it may be not less than k. It can be represented as $ASR_k=<(N_{ul},E_{ul}), (N_{dr},E_{dr})>$. $<(N_{ul},E_{ul}), (N_{dr},E_{dr})>$ is the GLL ID of the upper left and lower right corner of the rectangle.

Definition 3: The acreage of the Non-deterministic k-ASR can be calculated as below:

$$S(ASR_k) = (N_{ul}- N_{dr}+1)(E_{dr}- E_{ul})*S_{NE}$$

S_{NE} is stand for the acreage of a GLL.

Theorem 1: If all the ASR reported by the users were not less than 2 GLL, the number of users in the ASR is incalculable.

Proof: The ASR reported by one user may be completely or partly overlapping with the ASR reported by another user,and the user may be in any GLL of the ASR. So the centralized anonymous assistant cannot determine which GLL the user was in. The centralized anonymous assistant can only count the users of the GLL using equivalent probability. The user generates the ASR with this statistical result. The number of users in the ASR is just a statistical number using equivalent probability, not the real number of the users. So the number of users in the ASR is incalculable.

We can know that the number of users in the ASR is incalculable if all the ASR reported by the users were not less than 2 GLL from proof 1. Although the number of users in the ASR is incalculable, if the value of k is bigger, that means the real users in the ASR will be more.

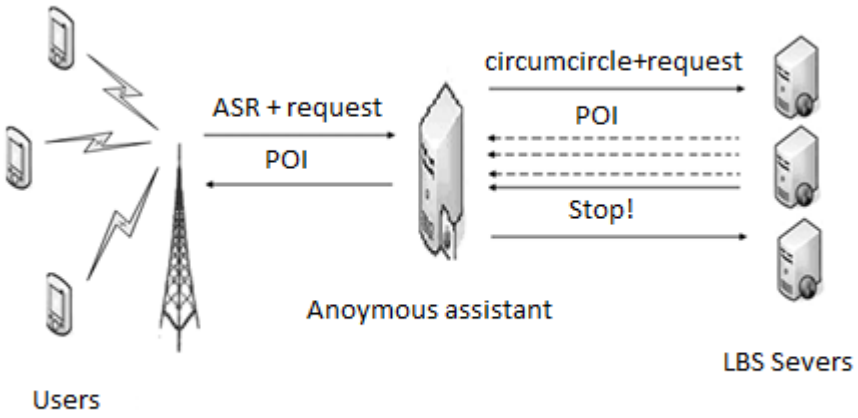

Figure 1. System Architecture

Figure 1 depicts the system architecture that consists of three entities, mobile users, anonymizer and LBS server. We will first discuss our privacy threat model and privacy settings in user privacy profiles, and then describe each entity in our system.

Privacy Threat Model: We assume that the centralized anonymous assistant and the LBS providers (LBS servers) are untrusted, both of them may betray user privacy information for economic benefits.

User Privacy Profiles: Each user specifies her privacy requirements in a privacy profile in a form of (k, A_{min}), where k indicates the required anonymity level, and A_{min} indicates the required minimum area of her cloaked areas. In other words, the user wants to find an ASR that includes at least k users and has an area of at least A_{min}. It is important to note that the query user can change her privacy profile at the begin time of any query to guarantee that her specified privacy settings achieve her desired privacy protection in different situations.

Mobile users: Each mobile user is equipped with a wireless network interface card for communicating with the anonymizer, e.g., GPRS, WCDMA, and CDMA2000. Each user is also equipped with a GPS or AGPS device, to determine her location that is represented as a coordinate (x, y). All the users in the system can calculate out the ID of the GLL which they are being in by themselves. They down load the user distributed information for each nearly GLL from the centralized anonymous assistant, and generate an ASR under their privacy requirements. If they move out the ASR they reported or the users in the ASR they reported are less than k, they will report their new ASR to the centralized anonymous assistant. When users want to query some things with k-anonymity, they send k, A_{min}, content of the query, K the number of target, and her ASR to the centralized anonymous assistant. When the results of the query come back from the centralized anonymous assistant, they filter the useless POI by their coordinates to get the last results.

Anonymous Assistant: The centralized anonymous assistant has a table or an array to record the numbers of users for each GLL. When users send their new ASR, it counts the users of the GLL in the ASR using equivalent probability. Figure 2 gives an example, user a reports an ASR which has 8 GLL, the number of users for each GLL is 0.125, user b reports an ASR which has 4 GLL, the number of users for each GLL is 0.25, the number of users in the overlapping GLL is 0.375. And the centralized anonymous assistant updates the total of users for related GLL. When users want to query, the centralized anonymous assistant generates the ASR for them, and then sends the query with the circumcircle of the ASR to the LBS servers. The LBS server sends back the results of the query to the anonymizer. The centralized anonymous assistant checks whether the results contain K targets, if the results contain K targets, it sends a message to the LBS server to stop search and sends the result to the query users, or it will wait for more results.

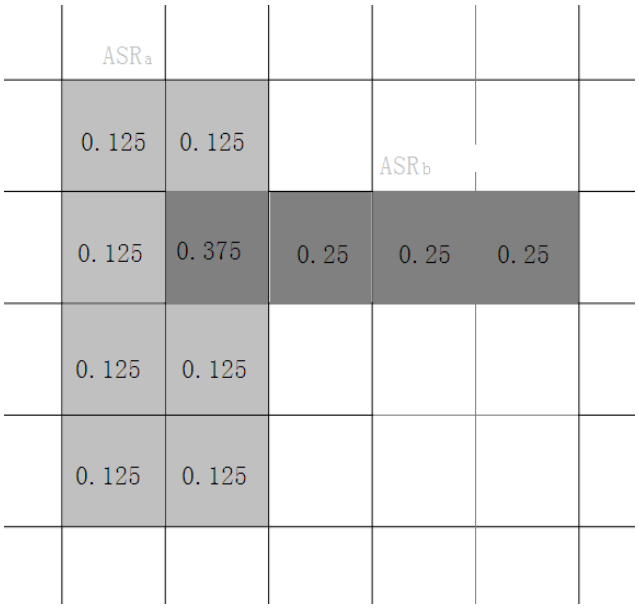

Figure 2. Counts Using Equivalent Probability

LBS servers A privacy-aware query processor embedded inside the LBS servers has the capability for incremental query processing. It can search the POI by the center and radius, and incrementally expands the radius to search the POI, sends back the searched POI to the centralized anonymous assistant batch to batch.

2. Algorithms

In this section, we give the generated ASR algorithm at the mobile end and the proxy query algorithm at the centralized anonymous assistant end.

Algorithm 1: Generated ASR at the mobile end

```
1:  function generateASR( ID of the user's GLL, k, A_min)
2:      ASR = { the user's GLL };
3:      Boolean updown = true;   //expends the ASR for up or down.
4:      if(Initialization==true){ // the system is in initialization.
5:          while( ASR < 2*A_min ){
6:              updown = ! updown;   //Anti-op，true becomes false, false becomes true.
7:              if(updown)
8:                  Extends the ASR one GLL for up or down randomly;
9:              else
10:                 Extends the ASR one GLL for left or right randomly;
11:             Adds the extended GLL into ASR;
12:         }// end while for line 5.
13:     }// end if for line 4.
14:     else {
15:         Down loads the two-dimensional sub-table from the anonymous assistant;
            // it contains the number of users in the GLL nearby the user.
16:         while( ASR < A_min || users in ASR < k ){
17:             updown = ! updown;
18:             if(updown)
19:                 Extends the ASR one GLL for up or down randomly;
20:             else
21:                 Extends the ASR one GLL for left or right randomly;
```

```
22:                Adds the extended GLL into ASR;
23:            } // end while for line 16.
24:        } // end else for line 14.
25:        return ASR;
26:    EndFunciton
```

Algorithm 1 depicts the pseudo code of our generating ASR algorithm at the mobile end. First of all, it add the user's GLL into the ASR, and sets the switch variable updown = true. After that, if the system is in initialization, the centralized anonymous assistant has not counted the number of users for each GLL. It generates the ASR just satisfying the user's twice the minimum area. Before extending the ASR, it changes the switch variable updown. If the updown is true, it will extend the ASR one GLL for up or down randomly, or it will extend the ASR one GLL for left or right randomly. Then it adds the GLL to the ASR. If the system is not in initialization, it down loads the two-dimensional sub-table from the anonymous assistant, the two-dimensional sub-table contains the number of users in the GLL nearby the user. Then extends the ASR using the same method to satisfy the user's the minimum area and the minimum number of users with the two-dimensional sub-table. At the end, it returns the ASR and the user's ASR.

Figure 3 depicts an example of our generating ASR algorithm at the mobile end. The open circle is stand for the user A, the ID of A's GLL is (N30°13′, E 103°24′), the A's privacy requirements is (k=40,A_{min}=4*s), the s is stand for the acreage of one GLL. It presumes the system is not in initialization. The ASR is initialized as ASR={(N30°13′, E 103°24′)}, the number of users and the acreage are both not satisfied the user's privacy requirements. So it extends the ASR one GLL for left or right randomly, if it extends to right, the ASR is {(N30°13′, E 103°24′),(N30°13′, E 103°25′)}. As the privacy requirements are both not satisfied after the first extending, it extends the ASR one GLL for up or down randomly. If it extends to down, the ASR is {(N30°13′, E 103°24′),(N30°13′, E103°25′),(N30°12′, E 103°24′),(N30°12′, E 103°25′)}. As the privacy requirements of the users is not satisfied after the second extending, it extends the ASR one GLL for left or right randomly. if it extends to right, the ASR is { (N30°13′, E 103°24′),(N30°13′, E103°25′),(N30°13′, E 103°26′),(N30°12′, E 103°24′),(N30°12′, E 103°25′),(N30°12′, E 103°26′) }. As the privacy requirements are both satisfied after the third extending, the last ASR is { (N30°13′, E 103°24′),(N30°13′, E103°25′),(N30°13′, E 103°26′),(N30°12′, E 103°24′),(N30°12′, E 103°25′),(N30°12′, E 103°26′) }, which is shown in light gray shaded area in figure 3.

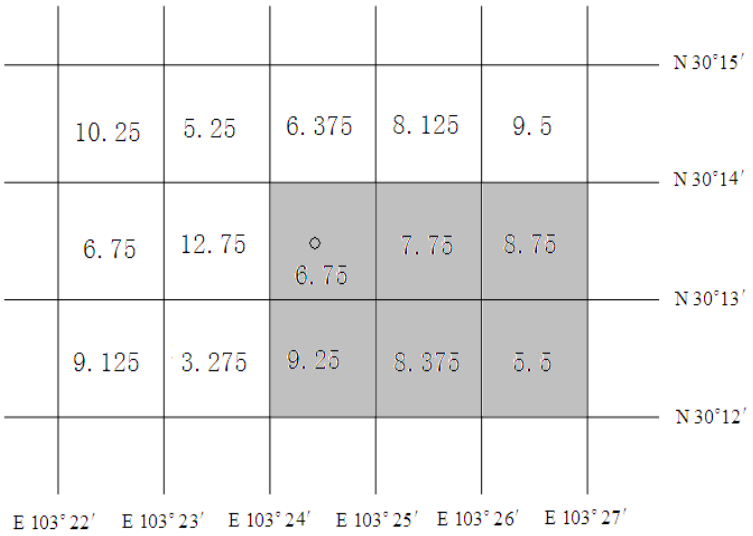

Figure 3. An Example for Generating ASR

When the algorithm 1 extends the ASR, it uses a switch variable updown. This tactics can make the ASR extended alternately for up/down and right/left, so the generated ASR will be a square or a rectangle with the smallest ratio of its length and its width. In the geometry, the smaller the ratio of the rectangle's length and the rectangle's width, the smaller the acreage ratio of the rectangle's circumcircle and the rectangle. So this tactics can decrease the acreage for searching the POI at the LBS server end.

Algorithm 2: Proxy query at the centralized anonymous assistant end.

```
1:  function proxyInquiry(ASR，targets K，content of query)
2:   circle(O, R)= the circumcircle of ASR；
3：   New set cs = {null};
4:   Sends circle(O, R) and content of query to LBS server, starts an incremental query;
5:   while(true){
6：    Receives rcs from the LBS server; // The rcs is made up POI for the query.
7：    cs = cs U rcs;
8:    if(cs contains K targets){
9:      Sends a message to the LBS server to stop search;
10:     Sends the cs to the query users;
11：    break;
12:   } //end if.
13:   else {
14:     if(LBS Server has searched all the area) {
15:       Sends a message to the LBS server to stop search;
16:       Sends the cs to the query users;
17：      break;
18:     } //end if for line 15.
19：    else
20:       Sends a message to the LBS server to extend the radius for new searching;
21：   } //end else for line 13.
22:  } //end while.
23：EndFunciton
```

Algorithm 2 depicts the pseudo code of our Proxy query at the centralized anonymous assistant end. At first, it generates the circumcircle of the ASR, and creates a candidate set cs whose initial value is null. After that, it sends the circumcircle and the content of the query to the LBS server to start an incremental query. It does not stop receiving the incremental candidate set rcs from the LBS server, and adding the rcs into cs. At the end, it checks whether there are K targets in the cs. If there are not less than K targets in cs, it stops the receiving, sends message to LBS server to stop the query, and sends the cs back to the query user. If there are not enough targets and the LBS server has searched all the space, it stops the receiving, sends message to LBS server to stop the query, and sends the cs back to the query user. Or it sends a message to the LBS server to extend the radius for more search.

5. Experiment

5.1 Experimental Configurations

We uses a computer with CPU 1.73GH, 2G RAM as the centralized anonymous assistant and the LBS server. Its operation system is windows XP sp3, the JVM is JDK 6.0, all the algorithms and programs are written with java.

We uses 60 × 60 GLL in our simulation. The length of Each GLL is about 1.85km. The width of each GLL is about 1.61km. We uses Java API to generate 3000 users and 7000 POI in 3600 GLL with uniform distribution and normal distribution.

The grid area in the twice anonymous algorithm is overlapping with the GGL, the random factor in the twice anonymous algorithm is 50. The LBS server in the twice anonymous algorithm extends 2km of the ASR to search POI, and the LBS server extends the radius 100m for one time to search.

5.2 POI Experiment

The non-deterministic k-anonymity algorithm not only provides individual privacy needs to the third-party for the user, but also optimizes the processing efficiency for k-nearest neighbor anonymous query. As the interactive control information between the centralized anonymous assistant and the LBS server is very small, it can be neglected. The more POI to transport from the LBS server to the centralized anonymous assistant, the heavier load for the net and the LBS server. So we can compare the number of the POI which are transported in the two algorithm to compare and analyze the performance of the two algorithms.

Figure 4,Figure 5,Figure 6 and Figure 7 are the result of the two algorithms when the targets K of the K-nearest neighbor is smaller. Figure 8,Figure 9,Figure 10 and Figure 11 are the result of the two algorithms when the targets K of the K-nearest neighbor is bigger.

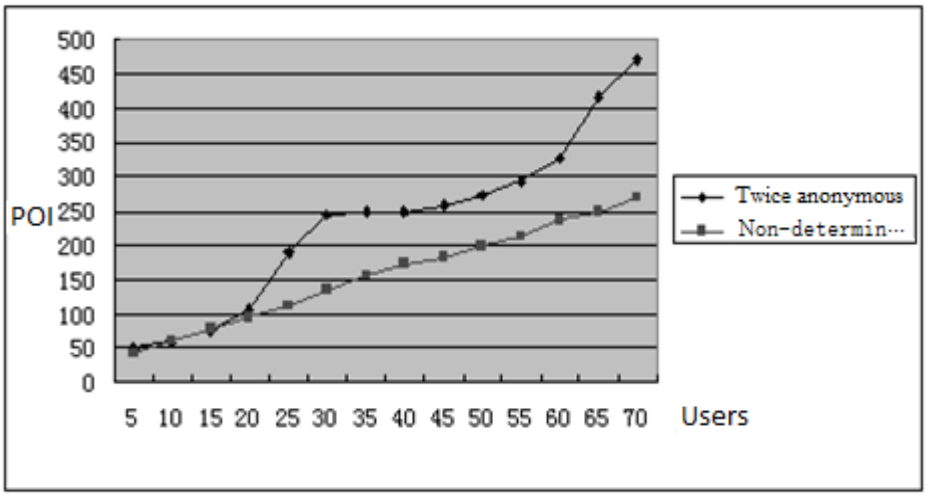

Figure 4. Experiment 1 for K=8

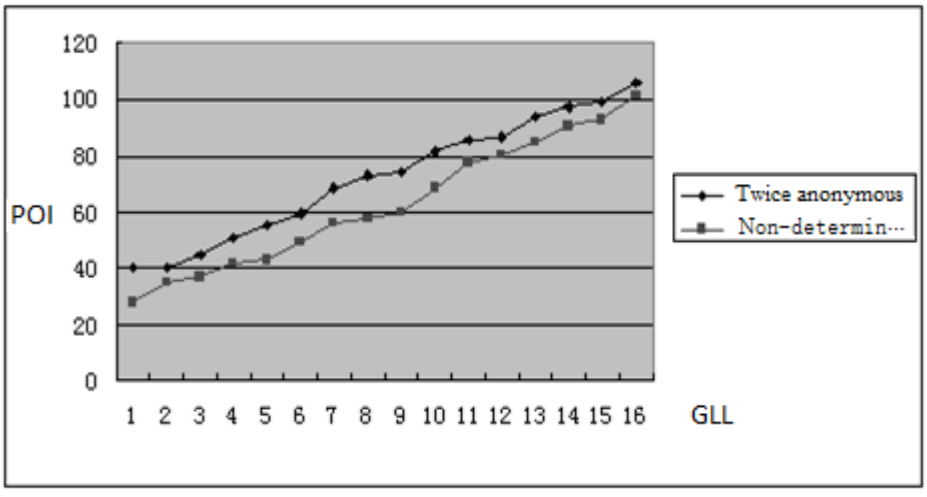

Figure 5. Experiment 2 for K=8

Figure 4 depicts the average value of 100 times experiments for the two algorithms when the users and the POI are both uniform distribution, the targets are 8, the A_{min} is one GLL, and the value of k is from 5 to 70. The number of the POI which are transported increases as the k increases for the both two algorithms, the number of the POI almost equal for two algorithms when k is less than 15, the number of the POI for the non-deterministic k-anonymity algorithm is less than the twice anonymous algorithm when k is more than 15.

Figure 5 depicts the average value of 100 times experiments for the two algorithms when the users and the POI are both uniform distribution, the targets are 8, the value of k is 4 users, GLL, and the A_{min} is from 1 GLL to 16 GLL. The number of the POI which are transported increases as the k increases for the both two algorithms, but the non-deterministic k-anonymity algorithm is less.

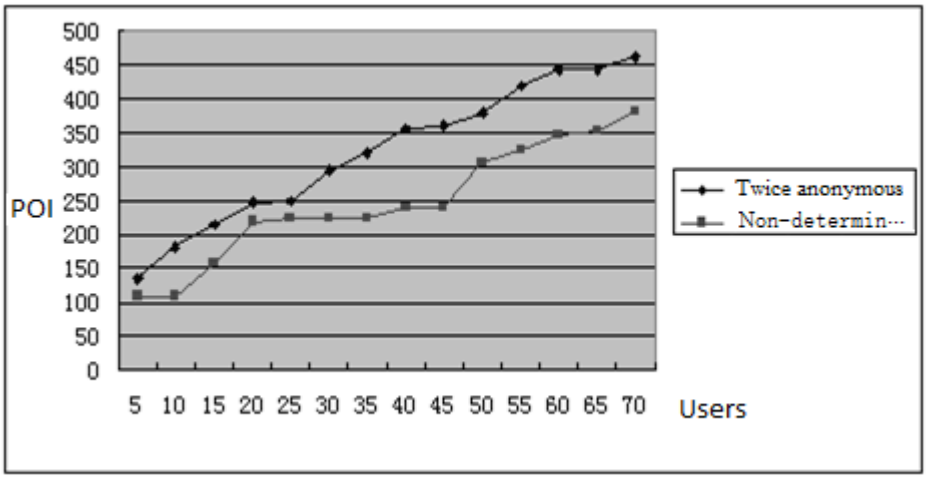

Figure 6. Experiment 3 for K=8

Figure 6 depicts the average value of 100 times experiments for the two algorithms when the users and the POI are both normal distribution, the targets are 8, the A_{min} is one GLL, and the value of k is from 5 to 70. The number of the POI which are transported increases as the k increases for the both two algorithms, but the non-deterministic k-anonymity algorithm is less.

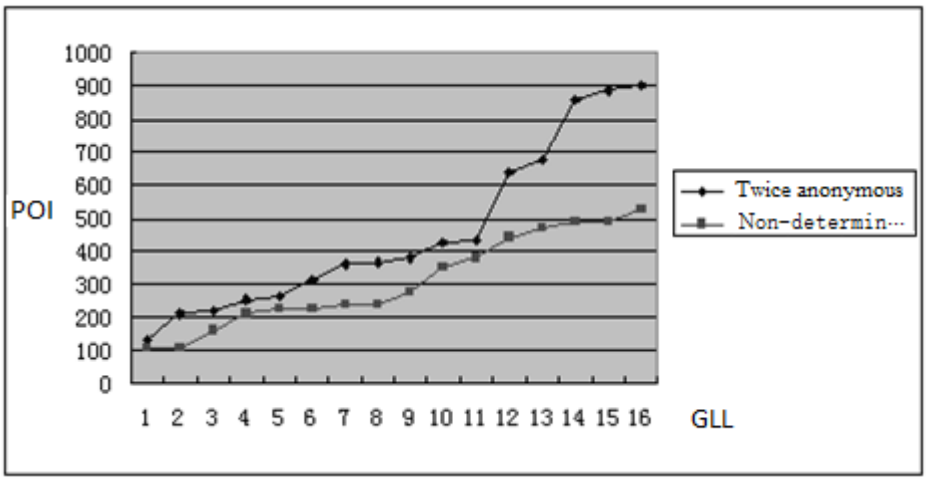

Figure 7. Experiment 4 for K=8

Figure 7 depicts the average value of 100 times experiments for the two algorithms when the users and the POI are both normal distribution, the targets are 8, the value of k is

4 users, GLL, and the A_{min} is from 1 GLL to 16 GLL. The number of the POI which are transported increases as the k increases for the both two algorithms, but the non-deterministic k-anonymity algorithm is less.

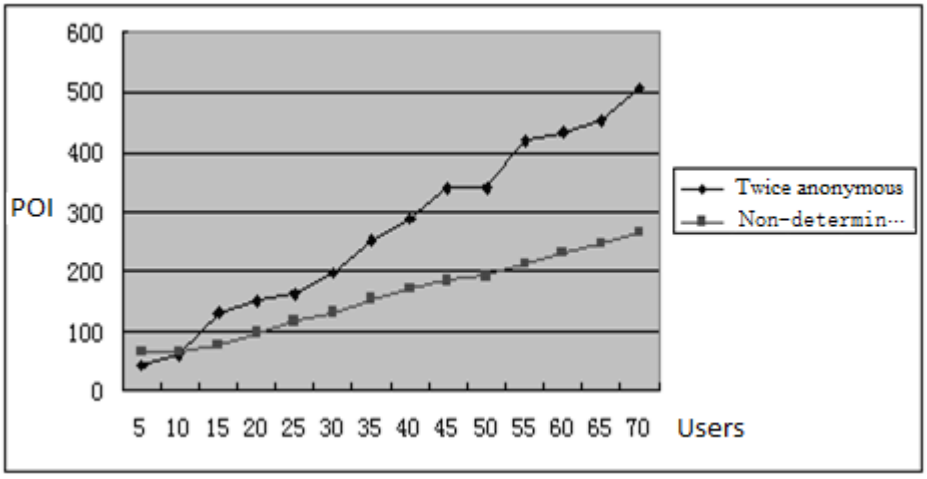

Figure 8. Experiment 1 for K=64

Figure 8 depicts the average value of 100 times experiments for the two algorithms when the users and the POI are both uniform distribution, the targets are 64, the A_{min} is one GLL, and the value of k is from 5 to 70. The number of the POI which are transported increases as the k increases for the both two algorithms, but the non-deterministic k-anonymity algorithm is less. The twice anonymous algorithm is failures for some experiments as the result does not include 64 targets when the k is less than 15.

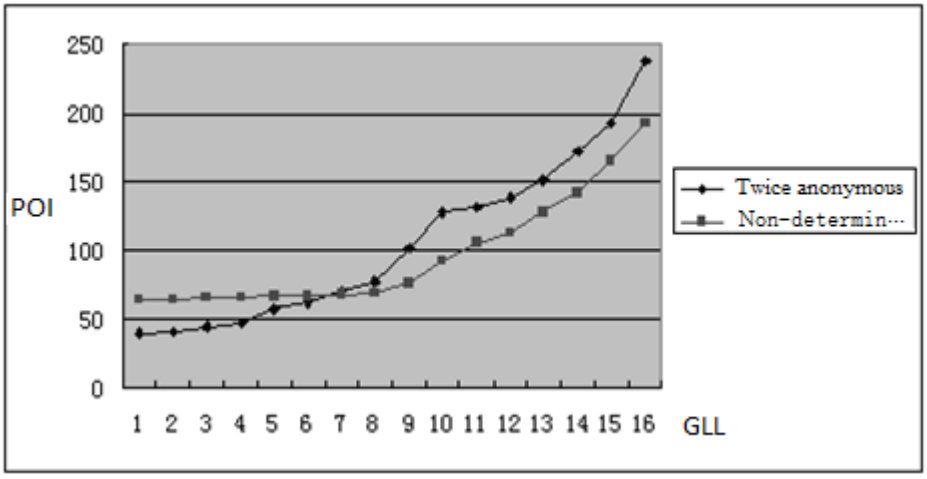

Figure 9. Experiment 2 for K=64

Figure 9 depicts the average value of 100 times experiments for the two algorithms when the users and the POI are both uniform distribution, the targets are 64, the value of k is 4 users, GLL, and the A_{min} is from 1 GLL to 16 GLL. The number of the POI which are transported increases as the k increases for the both two algorithms, but the non-deterministic k-anonymity algorithm is less. The twice anonymous algorithm is failures for some experiments as the result does not include 64 targets when the A_{min} is less than 8 GLL.

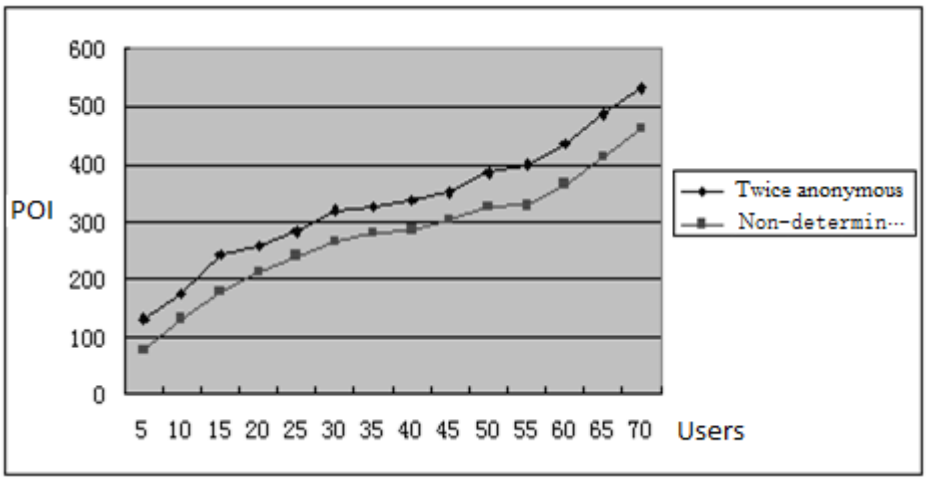

Figure 10. Experiment 3 for K=64

Figure 10 depicts the average value of 100 times experiments for the two algorithms when the users and the POI are both normal distribution, the targets are 64, the A_{min} is one GLL, and the value of k is from 5 to 70. The number of the POI which are transported increases as the k increases for the both two algorithms, but the non-deterministic k-anonymity algorithm is less.

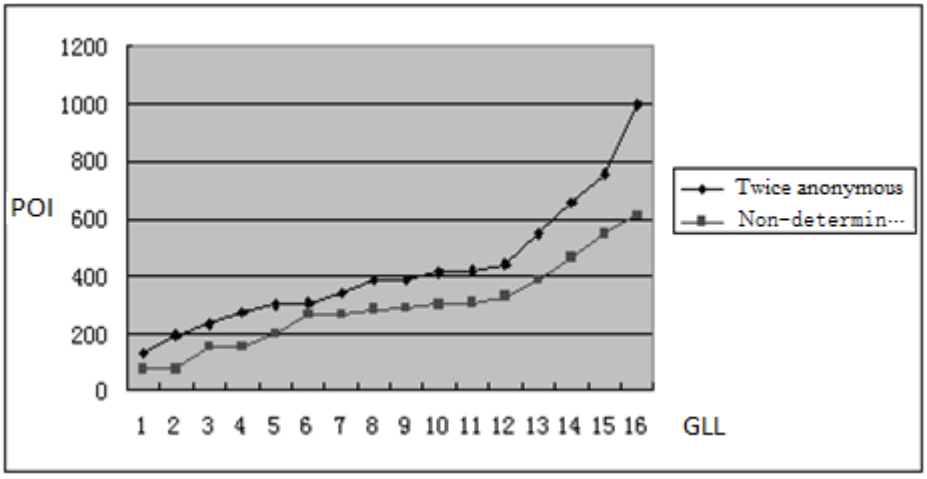

Figure 11. Experiment 4 for K=64

Figure 11 depicts the average value of 100 times experiments for the two algorithms when the users and the POI are both normal distribution, the targets are 64, the value of k is 4 users, GLL, and the A_{min} is from 1 GLL to 16 GLL. The number of the POI which are transported increases as the k increases for the both two algorithms, but the non-deterministic k-anonymity algorithm is less.

We can see that the number of the POI which are transported increases as the k increases for the both two algorithms, but the non-deterministic k-anonymity algorithm is less, the twice anonymous algorithm may be failures when the user's privacy requirement is lower and the target is more from the 8 figures.

The reason for the above situation is that the non-deterministic k-anonymity algorithm adopts the incremental search tactics. In order to get the K targets, it extends the radius for the searching more POI until the requirement is satisfied. While the twice anonymous algorithm extends the ASR 2km just for one time to search the POI. The incremental search tactics let the number of POI be less and the requirement can be satisfied all the time. While the tactics of the twice anonymous algorithm make the more useless POI

when the target is lower. And the ASR is small when the user's privacy requirement is low, if the target is large, the number of the searched POI may be less than the target K.

This section provides the number of POI experiment. We can see that the number of POI for the non-deterministic k-anonymity algorithm is less than the twice anonymous algorithm from the results of the experiments. So the loads of the net and the LBS server for the non-deterministic k-anonymity algorithm are both lighter than the twice anonymous algorithm. The non-deterministic k-anonymity algorithm has better performance than the twice anonymous algorithm.

6. Conclusions

This paper proposed a non-deterministic k-anonymity algorithm for location privacy protection in LBS. It moved the generation process of the ASR from the third party to the users, let the trusted anonymizer become untrusted anonymous assistant. It leaded into a method to count the users for each GLL using equal probability, so the number of users in the generated ASR was non-deterministic. It leaded into the incremental search tactics for KNN anonymous query, so its performance for KNN anonymous query is higher.

It was a pity that our non-deterministic k-anonymity algorithm could not resist the attack of continuous queries[14], could not satisfy the road network environment, and did not solve the single point of failure for the anonymous assistant. How to improve our algorithm to meet these needs is our work in the future.

Acknowledgements

This work is supported in part by Important National Science and Technology Special Project of China (No. 20112X03002-002-03). This work is also supported partly by the National Nature Science Foundation of China under Grant (No. 60903157 and No. 61133016), and the National High Technology Joint Research Program of China (863 Program, Grant No. 2011AA010706).

References

[1] J. Y. Jia and F. L. Zhang, "Overview of location privacy protection technology", Application Research of Computers, vol. 30, no. 3, **(2013).**
[2] J. Y. Jia, F. L. Zhang and R. C. Wu, "An Encryption-based K-anonymity Approach for Location Privacy Protection in LBS", 2013 International conference on mechatronic sciences, electric engineering and computer, **(2013)**,Shenyang.
[3] J.Y. Jia and F. L. Zhang, "Twice Anonymity Algorithm for LBS in Mobile P2P Environment", Journal of Computational Information Systems, vol. 9, no. 9, **(2013).**
[4] J. Y. Jia and F. L. Zhang, "Nonexposure Accurate Location K-Anonymity Algorithm in LBS", Scientific World Journal, doi: 10.1155/2014/619357, **(2014).**
[5] X. Y. Duan, "Research on intrusion event sequence correlation method for privacy protection", International Journal of Security and its Applications, vol. 8, no. 6, **(2014).**
[6] H. Kim, "P_PAKA: Privacy preserving authenticated key agreement protocol in smart grid", International Journal of Security and its Applications, vol. 8, no. 6, **(2014).**
[7] X. M. Sun, L. Zhou, Z. J. Fu and J. Wang, "Privacy-preserving multi-keyword ranked search over encrypted cloud data supporting dynamic update", International Journal of Security and its Applications, vol. 8, no. 6, **(2014).**
[8] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking", In: Proc. 1st international conference on Mobile Systems, applications, and services (MobiSys), USENIX Association, **(2003).**
[9] CY. Chow, MF. Mokbel and WG. Aref, "Casper: query processing for location services without compromising privacy", ACM Trans Database Syst., vol. 34, no. 4, **(2009).**
[10] MF. Mokbel, CY. Chow and WG. Aref, "The new casper: a privacy-aware location-based database server", In: Proc. 23rd International Conference on Data Engineering (ICDE). IEEE Computer Society, **(2007).**
[11] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: architecture and algorithms", IEEE Trans Mob Comput., vol. 7, no. 1, **(2008).**

[12] P. Kalnis, G. Ghinita, K. Mouratidis and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries", IEEE Trans Knowl Data Eng., vol. 19, no. 12, **(2007).**
[13] 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: Proc. of ICDE,**(2008).**
[14] T. Zhou, "Examining continuous usage of location-based services from the perspective of perceived justice", Information Systems Frontiers, vol. 15, no. 1, **(2013).**

Authors

Jinying Jia, The School of Computer Science & Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China (e-mail: jiajinying@ 126.com). His interests include: LBS, GIS and Information Security.

Fengli Zhang, The School of Computer Science & Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China. Her interests include: Computer Applications and Information Security.