Secure proximity detection

LI HONG PING
Hong Kong Baptist University
Kowloon Tong, Hong Kong

hpli@comp.hkbu.edu.hk

ABSTRACT

Nowadays, there is increasing number of mobile devices equipped with positioning capabilities (e.g. GPS), which ask location-dependent queries to Location Based Services (*LBS*). Normally, users expect LBS can help them find out the nearest point of interest and also share their location information through the social network web-site. (e.g. Twitter) Therefore, this raises a serious concern on location privacy of client. Previously, there are papers suggest number of methods (e.g. using dummies, cloaking, encryption) to preserve the location privacy of the client, while finishing the given mission by the user. In this paper, we will summarize the papers which are discussing this topic and propose a meaningful suggestion in the proximity detection area.

1. INTRODUCTION

More and more communication devices have implemented the user-positioning functions (e.g.GPS). Users can find out their location, by issuing location-dependent queries, for instance "find the nearest hospital" through the Location Based Services (LBS), which is provided by LBS provider (e.g. Google Maps)

Nevertheless, queries may disclose sensitive information about individuals, including user location, lifestyle and habits. Therefore people start to worry with the personal privacy, while they are using the Location Based Services (LBS).

Probably, there are three type of techniques can be used for protecting location privacy, including cloaking, dummy and encryption.

For the cloaking aspect, its main idea is to reduce the spatiotemporal resolution of user location. So the real user location is replaced by the cloaking region, in order to hide the user's location.

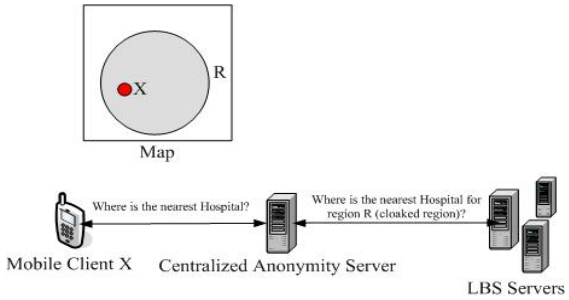

Fig. 1 Description of location cloaking

As Fig 1 show that, if mobile client X in location X, then he will sent out his location and convert into a cloaking region R(by himself or centralized anonymity server).Then send the query of region R to the Server. The concept of location k-anonymity was introduced in [9] where k is set to be uniform for all users. It focuses on the field of location privacy for mobile users through spatial and temporal cloaking of location and time information. We will give a further discussion in Section 2.

For the dummy aspect, the concept of this technique is user send her real location mixed with few dummy locations so that the attacker cannot tell which one is real.

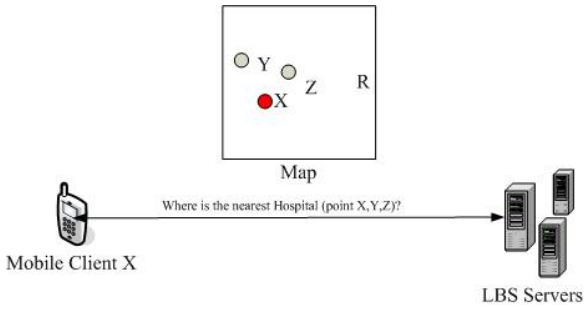

Fig. 2 Description of generating faked dummies

For example Fig. 2 show that when client X send his location service request to the LBS server, he will send out the faked dummies (Y,Z) simultaneously, in order to diversify the risk of the discovery of his actual location. Some kind of research [10, 11] focus on the user movement simulation, which create a more meaningful dummy, in order to increase the safety of protecting user location. Some of them [16,17] use *cryptographic* protocol-based approaches, which is based on a specific transformation fully known only to the clients, the server processes user queries without the ability to decipher exact user locations.

For the encryption side, its concept is to transmit the data from one party to another party, while both parties do not know what is another party sending. Secure multiparty computation (SMC) is one of the major topics in this aspect. The researches have started, since the Yao's Millionaire Problem [18] is published. The Yao's Millionaire is able to compare two private numbers, while the participating entities (including client,server,non-colluding third party) unable to know what the exact information transmit by the other parties. Since then, many research efforts have been made to develop more efficient SMC protocols for specialized functions and tasks, including secure sum, scalar product, add vector and set operation.

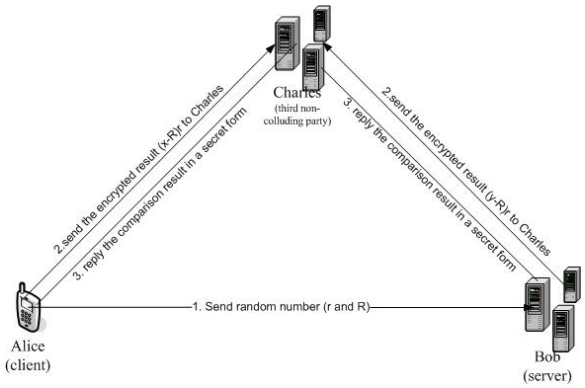

Fig. 3 Solution of Yao's comparison [3,4]

Fig. 3 is a graphical version of Yao's comparison solution. Alice (client) issues two random numbers to Bob (server), then Alice and Bob will both send their encrypted answer to Charles (third non-colluding party). Those paper [3,4] implement the solution of Yao's comparison, which can solve the problem in an efficient time and employ their own techniques to find the k-nearest neighbor(kNN) of the users.

Previously, there is most of the location privacy research focus on finding (kNN), the research in proximity detection start much later than previous one. The paper in this aspect [26, 27, 28, 29, 30] contribute different ideas in this area. For example [26] suggest VICINITYLOCATOR which do not require peer and peer communication with an adjustable region for proximity detection. However, that suggestion is still required to expose the approximate user location to the third party.

In this paper, we are going to suggest a solution, which can achieve the goal of proximity detection, while not expose the user's location in any manner.

2. RELATED WORK

Firstly, we review some previous work in the location privacy area and show the development idea of the solution of this paper.

Most existing solutions adopt one of the three technique (including cloaking, dummies and encryption) when handling the user location privacy problem.

The earliest proposal for location privacy protection is spatial cloaking [9]. Fig 4 shows us a brief picture of spatial cloaking by using a trusted anonymity server. Instead of sending a single user's exact location to the server, spatial cloaking techniques collect *k* user locations and send a corresponding (minimum) bounding region to the server as the query parameter.

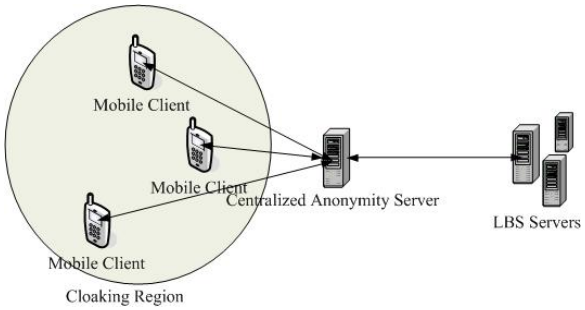

Fig. 4 Cloaking with trusted anonymity server

However, in the previous model, *k* is same for all users in the system. User cannot adjust the privacy level by themselves. It will seriously affect the quality of service when the user goes to an area which has low density of user. Because of the time consuming job on searching nearby users to form a cloaking region. After that, [5] extended this to a personalized k-anonymity model, that mean the user is enable to decide the balance between privacy and efficiency.

More recently, location cloaking algorithms advanced from cloaking of snapshot locations to continuous location updates [1, 6]. The cloaking of snapshot locations is not secure enough to prevent the leakage of location privacy, if an attacker (e.g., the service provider) can collect the user's historical cloaked regions as well as the user's mobility pattern (e.g., users' speed).

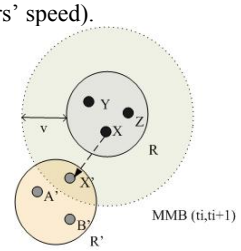

Fig. 5 Location Dependent Attacks

[21] shows that a circular cloaking region generally lead to a small result superset, so that less computational power and time is required for cloaking process. Fig 5. use circular cloaking region to show us the threat of snapshot locations cloaking. If we know that the maximum speed of client X, then we can calculate the maximum movement boundary (MMB) by creating a bigger circle which have the same center of the original cloaking region R. Then when client X emits his request for new cloaking region R'. We can find client X much more easily because client X must in the overlapping area of MMB and R'. Therefore we can see the importance of prevent the user location discovered by the continuous location tracking.

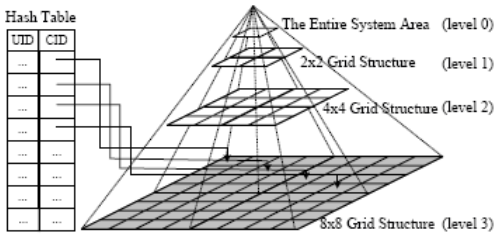

Fig. 6 Location Dependent Attacks

The Casper [13] consists of two main components, the anonymity server and the privacy-aware query processor. The anonymity server blurs a user's exact location information into a cloaking spatial region based on user specified privacy requirements. The privacy-aware query processor is embedded inside the AS to deal with the cloaking spatial areas rather than exact location information.

This framework uses a quad-tree data structure that maps the location information into grids with different levels and resolutions. The cloaking algorithm goes through the quad-tree in a bottom-up fashion to find a spatial region that meets the privacy requirements.

This approach also requires the anonymity server to dynamically keep track of the locations of mobile devices in a fine spatial resolution. This can easily lead the anonymity server become a performance bottleneck when there are large number of client updates and requests. Due to the limitations of the quad-tree structure, the calculated cloaking region is often larger than required, which may cause lower service quality.

While the above cloaking algorithms need a centralized trusted third party to perform location cloaking.
Chow *et a.* [20] proposed a peer-to-peer cloaking algorithm based on information exchanges among mobile clients. The paper point out two shortcomings of the centralized privacy-preserving framework.
When the entire location privacy queries processed by a single anonymity server, that server will become the system bottleneck. As cloaking function is quite time-consuming solution, therefore if there are many clients using the service, the response time for the server will be very slow. Also, if it is compromised by an attacker, or forced to cooperate with a government agency, all the users' movement records. All the protected information will be totally revealed.

Therefore some papers present some kinds of decentralized approach which can achieve anonymity without an anonymity server, so as to solve the high communication overload problem while using the cloaking method.
Peer-to-peer cloaking [20, 22] is one of the solution, which utilizes peer-to-peer communications Each moving object probes its neighborhood to look for other moving objects and anonymizes its own location using the information collected.

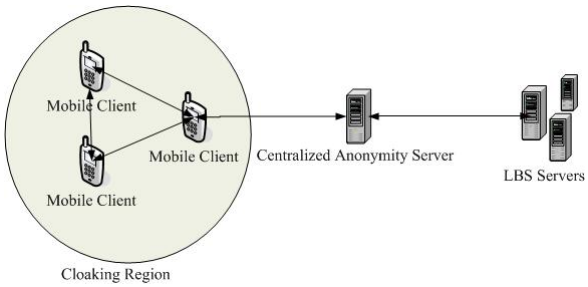

Fig. 7 Cloaking without trusted anonymity server

For instance the case of Fig 7 shows that a group of clients form a cloaking region by themselves. So that it can reduce the large amount of workload on the centralized server because the most time-consuming part (cloaking) has been done by the users.

However, the drawbacks of this approach are, for example it does not guarantee that the service requests can be fulfilled because there may not be enough peers nearby. As a result, the service availability is not consistent.

Moreover, privacy leakage may happen because the requester tends to be in the center of the cloaking region.

Furthermore, as the user's location always changes when they are using the location-base system. Therefore we are facing the dilemma of changing the cloaking peers or enlarging the cloaking region. If we employ the former case, we need to update the cloaking set frequently. Because of the limited capabilities of current mobile, a peer-to-peer system may also pose too much computational and communicational overhead. If we employ the latter case, the cloaking region will become larger and larger. As a result, the quality of service will degrade rapidly.

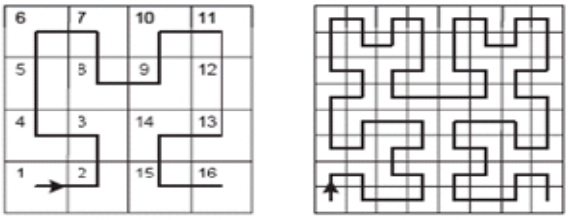

Fig. 8 Hilbert curve (Left (4x4) Right (8x8)

They also applied this algorithm to a distributed environment based on an annotated B+-tree index.
In [23], a decentralized approach based on Hilbert Curve is proposed to meet the reciprocity property requirements of the location cloaking algorithm. This approach guarantees the query anonymity even location information is disclosed to the adversary. However, each client needs to maintain relatively complex data structure and communication protocol as well as long range communication among peers. Therefore additional computation and communication cost may be posed to clients with limited capabilities.

Generating faked dummies is another kind of decentralized approach which does not involve any AS. For example, the paper [10, 11, 16, 17] lets the mobile clients generate false locations and send them along with the real locations to LBS. For every location update, a user would

send n different locations to the server. Only one of them is true. The rest are dummies. Thus, the server cannot know which location is the actual one. However, it is still possible to detect false dummies through data mining techniques if the algorithm used for dummy generation is not selected appropriately.

Another way to hide the client location is using cryptography solution. Yao's [18] present how to exchange secret by using some comparison method. After that paper [3] propose a kNN protocol for horizontal partitioned data and provide a privacy-preserving algorithms which can handle the large dimensionality and diversity of attributes common in vertically partitioned data.

[2] use Private Information Retrieval (PIR) implementation to build up a location privacy protection framework which does not require an anonymizer or collaborating trustworthy users. However, the limitation of this implementation is the cell contents have to match the query result that may cause a high storage overhead because the server is required to store different type of content. Also, it is not easy to find the optimal size of grid partition in order to minimize the computation and communication time.

Except the research of finding kNN, proximity detection is also a valuable topic for us to discuss. Sometimes we may want to find friend or place where is within in a certain distance. kNN may give us too much information when most of the point of interest (POI) are near us or give us too limit information when those POI are too concentrate together. As a result, we may not get what we want because of the uneven distribution of POI. Location cloaking may give us the solution, but we can until know the approximate location. In order to have a better protection for the users, we should create a better framework that none of us (including central server) know the other users' location except we are within a certain distance.

3. SECURE PROXIMITY DETECTION

Under an agreement, users in a social group may allow users in the same group to know where we are, when we are nearby. Nevertheless, tradition proximity detection is still requiring us to expose our approximate location, in order to maintain the operation. Our solution is able to solve such problem. The workflow of the system is shown in the Fig 9.

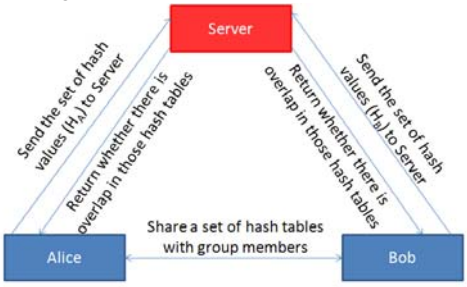

Fig. 9 System workflow

It is a solution which does not need to send the approximate location to any third party. Also, users in the same group can know you are nearby, if and only if you are within a certain distance. Therefore it can complete protect the user's location privacy, while providing the LBS.

First, we introduce the structure of our system. In order to simply the demonstration, we will assume in this system there are only 2 users, Alice and Bob. At the beginning Alice and Bob will communicate and create an agreement which contain an acceptable distance D. Then one of them will generate a set of hash tables which cell size is (D*D) and distribute those generated table to all other group members.

As 2 users are fallen in the same hashed cell, we can sure that they are within the distance D. However, there is still 75% of missing cases that we have not handled. Such as the example shown in the Fig. 10a. This problem can be solved by generating more same size hash tables that is randomly shifted to any direction Fig 10b.

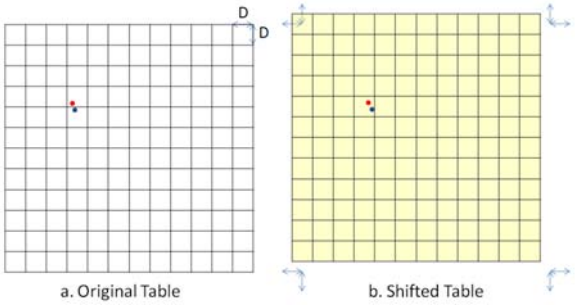

Fig. 10 Grid-based hash table

The original missing rate of a single hash table is 75%. However after 20 more randomly shifted table is added the missing rate will drop sharply to $(0.75)^{20} = 0.3\%$. After adding more tables, even there is only one overlap hash table we can still ensure that they are within a distance D.

3.1 Update handling

This solution is able to handle the continuous location update of the users. In the following picture Fig. 11, it shows the location of the user is at the center. Different color square represent different hash table. The user is required to update their location information to the server, whenever they enter or quit any grid cells in the hash tables.

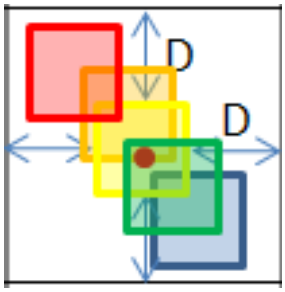

Fig. 11 Grid-based hash table

Based on the update rule mentioned previously, if the D is very small, the communication cost will become very high. Because of the frequent updates are required, when we always enter or quit the cell. Therefore we should have further improvement in the solution, so as to reduce the update cost.

3.2 Way to minimize the communication cost

We can adopt different sizes of hash table sets to help as to reduce the update cost. It is workable especially when 2 users are far away from each other, the distance between them longer, less update is required.

For instance we have already try 20 randomly shifted same size (size = D^2) hashed table, none of them are overlapped. Then we can conclude that they are very likely at least distance D apart from each other.

In the process of hash cell (size = D^2) randomly shift to different direction, it has also approximately cover the larger area (size = $(3D)^2$).

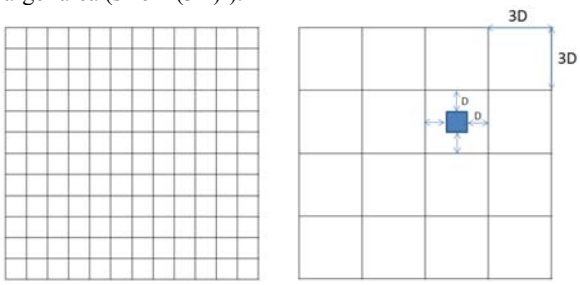

Fig. 12 Extension of from the base layer

In the example shown in Fig. 13, the distance between user A and B is 1.1 D, no matter how we shift the size D^2 hashed table. They will never fall in the same hashed cell. Therefore, we can at least ensure they are at least apart from each other more than distance D.

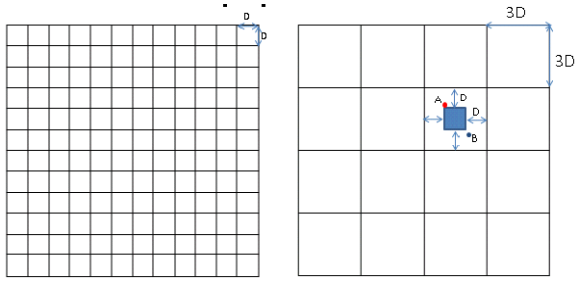

Fig. 13 Example in D^2 layer

Another example is the extended vision of previous example is shown in Fig. 14, the distance between user A and B is 1.1 D, no matter how we shift the size D^2 hashed table. They will never fall in the same hashed cell. Therefore, we can at least ensure they are at least apart from each other more than distance D.

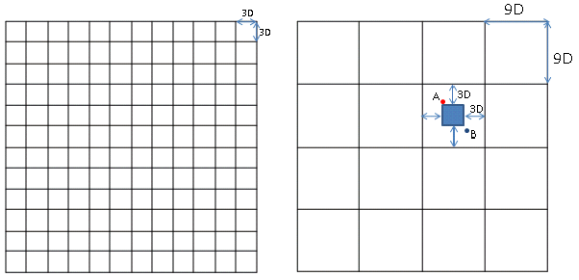

Fig. 14 Example in $(3D)^2$ layer

Finally, this concept can be further extend to $D^2 \rightarrow (3D)^2 \rightarrow (9D)^2 \rightarrow (27D)^2$ …etc. layers. Therefore less update is required if we know the larger distance we are apart from each other.

3.3 Structure of hash tables layers

For a certain group, they will exchange n sets of hashed tables (layer 1 – n) between each other.

If n = 8, then group member will need to share 160 hashed table when each has 20 randomly shifted table.

Dist.	D^2	$(3D)^2$	$(9D)^2$	$(27D)^2$	$(81D)^2$	$(243D)^2$	$(729D)^2$	$(2187D)^2$
LAYER	1	2	3	4	5	6	7	8

We start the checking in a bottom-up manner and stop until the first overlap layer is found. Then the system will work as follow:

Let x is the current layer

1. no overlap in layer x ∩ no overlap in layer x+1
   increase the layer level (bigger cell)
2. no overlap in layer x ∩ no overlap in layer x+1
   remain in the same layer
3. overlap in layer x
   decrease the layer level (smaller cell)

4. CONCLUSION

This solution provides a safe and flexible ways in proximity detection. Also, the efficient layer structure reduce the communication cost to a reasonable level, which can be used in many practical situation.

REFERENCE

[1] T. Xu and Y. Cai. Location Anonymity in Continuous Location-based Services. In ACM GIS'07, pages 300--307, November 2007.

[2] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: Anonymizers are not necessary," in Proc. ACM SIGMOD Int. Conf. Manage. Data, Vancouver, Canada, Jun. 2008, pp. 121–132.

[3] Artak Amirbekyan and Vladimir Estivill-Castro. Privacy-preserving k-nn for small and large data sets. In Proceedings of the ICDM Workshops, 2007.

[4] Processing Private Queries over Private and Indexed Data

[5] Buğra Gedik , Ling Liu, Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms, IEEE Transactions on Mobile Computing, v.7 n.1, p.1-18, January 2008

[6] Xian Pan, Jianliang Xu, Xiaofeng Meng**,** Protecting location privacy against location-dependent attack in mobile services, Proceeding of the 17th ACM conference on Information and knowledge management, 2007

[7] Haibo Hu, Jianliang Xu, Non-Exposure Location Anonymity, Proceedings of the 2009 IEEE International Conference on Data Engineering
[8] Chengyang Zhang, Yan Huang, Cloaking locations for anonymous location based services: a hybrid approach, Volume 13, Issue 2 (June 2009) table of contents, Pages: 159 - 182
[9] M.Gruteser and D. Grunwald, "Anonymous usage of location-based service through spatial and temporal cloaking," Proc. Of the International Conference on Mobile Systems, Applications, and Services (MobiSys' 03), pp163-168, Scan Francisco, USA, 2003
[10] Hidetoshi Kido, Yutaka Yanagisawa, Tetsuji Satoh, An Anonymous Communication Technique using Dummies for Location-based Services, Pervasive Services, 2005. ICPS '05. Proceedings. International Conference
[11] Hidetoshi Kido, Yutaka Yanagisawa, Tetsuji Satoh, Protection of Location Privacy using Dummies Location-based Services, Data Engineering Workshops, 2005. 21st International Conference
[12] O. Goldreich. The Foundations of Cryptography, volume 2, chapter General Cryptographic Protocols. Cambridge University Press, 2004.
[13] M. F. Mokbel, C. Y. Chow, and W. G. Aref. The New Casper: Query Processing for Location Services without Compromising Privacy. In *Proc. of VLDB*, 2006.
[14] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing Location-Based Identity Inference in Anonymous Spatial Queries. *IEEE TKDE*, 19(12):1719–1733, 2007.
[15] Hua Lu, Christian S. Jensen, Man Lung Yiu, PAD: Privacy-Area Aware, Dummy-Based Location Privacy in Mobile Services, 2008.
[16] P. Indyk and D. Woodruff. Polylogarithmic Private Approximations and Efficient Matching. In *Proc. TCC*, 2006.
[17] A. Khoshgozaran and C. Shahabi. Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy. In *Proc. SSTD*, 2007.
[18] A.C. Yao. Protocols for secure computations. In Proceedings of the 23nd Annual IEEE Symposium on Foundations of Computer Science, 1982.
[19] A.C. Yao How to generate and exchange secrets. In Proceedings 27th IEEE Symposium on Foundations of Computer Science.
[20] C.-Y. Chow, M. F. Mokbel, and X. Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based services. *ACM GIS*, Arlington, VA, 2006.
[21] J. Xu, X. Tang, H. Hu, and J. Du. "Privacy-Conscious Location-Based Queries in Mobile Environments." *IEEE Transactions on Parallel and Distributed Systems (TPDS)*,
[22] H. Hu and J. Xu. "Non-Exposure Location Anonymity." Proc. IEEE 25th International Conference on Data Engineering *(ICDE '09)*, Shanghai, China, March 2009
[23] Ghinita, G., ,Kalnis, P., Skiadopoulos, S.:Mobihide: A mobile peer-to-peer system for anonymous location-based queries. In: SSTD '07: 10th International Symposium on Advances in Spatial and Temporal Databases, Boston, MA, USA, Springer (2007) 221–238
[24] Gedik, B., Liu, L.: Location privacy in mobile systems: A personalized anonymization model. In: ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems, Washington, DC, USA, IEEE Computer Society (2005) 620–629
[25] Xiaokui Xiao, Yufei Tao, Dynamic Anonymization: Accurate Statistical Analysis with Privacy Preservation. Proceedings of the 2008 ACM SIGMOD international conference on Management of data
[26] Laurynas Siksnys, Jeppe Thomsen, Simonas Saltenis, Man Lung Yiu, Private and Flexible Proximity Detection In Mobile Social Networks, 11th International Conference on Mobile Data Management (MDM 2010)
[27] K. Liu, C. Giannella, and H. Kargupta, "An Attacker's View of Distance Preserving Maps for Privacy Preserving Data Mining," in PKDD, 2006, pp. 297–308.
[28] S. Mascetti, C. Bettini, and D. Freni, "Longitude: Centralized privacy-preserving computation of users' proximity." in Secure Data Management, 2009, pp. 142–157
[29] S. Mascetti, C. Bettini, D. Freni, X. S. Wang, and S. Jajodia, "Privacy-aware proximity based services," in MDM, 2009, pp. 31–40.
[30] P. Ruppel, G. Treu, A. Küpper, and C. Linnhoff-Popien, "Anonymous User Tracking for Location-Based Community Services," in LoCA, 2006, pp. 116–133.