International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

A Novel Approach to Enhance the Privacy of LBS Users

[1]P.Ameena
Dr.K.V.Subba Reddy College of Engg for Women, Kurnool, A.P
[2]Md.Asim
Asst.Professor, Dr.K.V.Subba Reddy College of Engg for Women, Kurnool, A.P

Abstract:
We propose a novel framework for measuring and evaluating location privacy preserving mechanisms in mobile wireless networks. Within this framework, we first present a formal model of the system, which provides an efficient representation of the network users, the adversaries, the location privacy preserving mechanisms and the resulting location privacy of the users. This model is general enough to accurately express and analyze a variety of location privacy metrics that were proposed earlier. By using the proposed model, we provide formal representations of four metrics among the most relevant categories of location privacy metrics. We also present a detailed comparative analysis of these metrics based on a set of criteria for location privacy measurement. Finally, we propose a novel and effective metric for measuring location privacy, called the distortion-based metric, which satisfies these criteria for privacy measurement and is capable of capturing the mobile users' location privacy more precisely than the existing metrics. Our metric estimates location privacy as the expected distortion in the reconstructed users' trajectories by an adversary.

Keywords: Mobile Networks; Location-based Services; Location Privacy; Bayesian Inference Attacks; Epidemic Models

----------***----------

1. INTRODUCTION

An increasing number of communication devices (e.g., mobile phones, PDAs), feature positioning capabilities (e.g., GPS). Users can ask location-dependent queries, such as "find the nearest hospital", which are answered by Location Based Services (*LBS*) like Mapquest or Google Maps. However, queries may disclose sensitive information about individuals, including health condition, lifestyle habits, political and religious affiliations, or may result in unsolicited advertisement (i.e., spam). Privacy concerns are expected to rise as LBSs become more common. Observe that privacy is not protected by replacing the real user identity with a fake one (i.e., pseudonym), because, in order to process location-dependent queries, the LBS needs the exact location of the querying user. An attacker, which may be the LBS itself, can infer the identity of the query source by associating the location with a particular individual. This can be easily performed in practice, with the help of a public telephone directory, for instance, which contains subscribers' addresses. Most existing solutions adopt the *K-anonymity* [22] principle: a query is considered private, if the probability of identifying the querying user does not exceed $1/K$, where K is a user-specified anonymity requirement. To enforce this principle, a trusted third-party *anonymizer* is employed [11, 14, 17, 20] (see Figure 1). The anonymizer maintains the current locations of all subscribed users. Instead of sending the Nearest Neighbor (*NN*) query to the LBS, the user (Alice in our example) contacts the anonymizer, which generates a Cloaking Region (*CR*) enclosing Alice as well as $K - 1$ other users in her vicinity. In Figure 1, $K = 3$ and the CR contains $u1$ and $u2$ in addition to Alice. The CR is sent to the LBS, which cannot identify Alice with probability larger than $1/K$. The LBS computes a candidate set that includes all points of interest (*POI*) which may potentially be the

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

NN for any point within the entire CR [15]. The candidate set (i.e., *{p*1, *p*2, *p*3, *p*4*}*) is sent back to the anonymizer, which filters the false hits and returns the actual NN (i.e., *p*3) to Alice. We discuss these methods in Section 2. Existing methods have several drawbacks: *(i)* The anonymizer is a single point of attack: if an attacker gains access to it, the privacy of all users is compromised. It is also a bottleneck, since it must process the frequent updates of user locations. *(ii)* A large number of users must subscribe to the service, otherwise CR cannot be constructed. It is assumed that all users are trustworthy. However, if some of them are malicious, they can easily collude to compromise the privacy of a targeted user. *(iii)* It is assumed that the attacker has no background information about the users, but in practice it is difficult to model the exact knowledge. Assume that Alice is searching for the nearest women's clinic, and the CR contains Alice and Bob. From the query content, the attacker can identify Alice as the query source. *(iv)* Privacy is guaranteed only within a static snapshot of user locations; users are not protected against correlation attacks. For example, if Alice asks the same query from different locations as she moves, she can be easily identified because she will be included in all CRs. We propose a framework for private location-dependent queries that solves these problems. Our framework is based on the theory of Private Information Retrieval (*PIR*) and does *not* need an anonymizer. Recent research on PIR [4, 19] resulted in protocols that allow a client to privately retrieve information from a database, without the database server learning what particular information the client has requested. Most techniques are expressed in a theoretical setting, where the database is an *n*-bit binary string *X* (see Figure 2). The client wants to find the value of the *ith* bit of *X* (i.e., *Xi*). To preserve privacy, the client sends an encrypted request *q*(*i*) to the server. The server responds with a value *r*(*X*, *q*(*i*)), which allows the client to compute *Xi*. We focus on *computational* PIR, which employs cryptographic techniques, and relies on the fact that it is computationally intractable for an attacker to find the value of *i*, given *q*(*i*). Furthermore, the client can easily determine the value of *Xi* based on the server's response *r*(*X*, *q*(*i*)). PIR theory is discussed in Section 3. In this paper we show that PIR can be used to compute privately the nearest neighbor of a user with acceptable cost, by retrieving a small fraction of the LBS' database. Consider the example of Figure 3.a, where *u* is the querying user and the LBS contains four points of interest *p*1, *p*2, *p*3, *p*4. In an off-line phase, the LBS generates a kd-tree index of the POIs and partitions the space into three regions *A,B,C*. To answer a query, the server first sends to *u* the regions *A,B,C*. The user finds the region (i.e., *A*) that contains him, and utilizes PIR to request all points within *A*; therefore, the server does not know which region was retrieved. The user receives the POIs in *A* in encrypted form and calculates *p*4 as his NN. The method can be used with a variety of indices. In Section 4 we present implementations based on the Hilbert curve and on an R-tree variant. Note that the result is approximate; in our example the true NN is *p*3. We show experimentally that the approximation error is low.

We also propose a method for finding the exact NN. In a pre-processing phase, the server computes the Voronoi diagram for the POIs (see Figure 3.b). Each POI *pi* is assigned to its Voronoi cell; by definition, *pi* is the NN of any point within that cell. The server superimposes a regular grid of arbitrary granularity on top of the Voronoi diagram. Each grid cell stores information about the Voronoi cells intersecting it. For example *D*1 stores *{p*4*}*, whereas *C*3 stores *{p*3, *p*4*}*. Upon asking a query, the client first retrieves the granularity of the grid, and calculates the grid cell that contains him (i.e., *C*2). Then, he employs PIR to request the contents of *C*2. He receives *{p*3, *p*4*}* (encrypted) and calculates *p*3 as his exact NN. The method is described in Section 5. Note that the cost is typically higher compared to approximate NN. PIR has been criticized of being too costly to be applied in practice. [24] showed that the computational time of PIR may be longer

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

than the time required for an oblivious transfer of the database. [24] assumes that the server agrees to surrender the entire database to the client. In practice, this is rarely the case, since the database is a valuable asset for the server, who charges the client, either directly or indirectly (e.g., advertisements), based on the actual usage. Moreover, most queries do not need to retrieve the entire database. Still, the CPU cost of PIR can be high, since it involves numerous multiplications of large numbers. Nevertheless, we show that much of the computation is redundant. In Section 6 we develop a query optimizer which employs data mining techniques to identify such redundancy. The resulting execution plan is up to 40% cheaper in terms of CPU cost. Moreover, we show that the required computations are easily parallelized. Summarizing, our contributions are:

1. We propose a novel framework for private location dependent queries, which uses PIR protocols and eliminates the need for any trusted third party. Our work is the first to provide provable privacy guarantees against correlation attacks.
2. We develop algorithms for approximate and exact private nearest neighbor search. We utilize data mining techniques to optimize query execution.
3. We show experimentally that the cost is reasonable; hence our methods are applicable in practice.

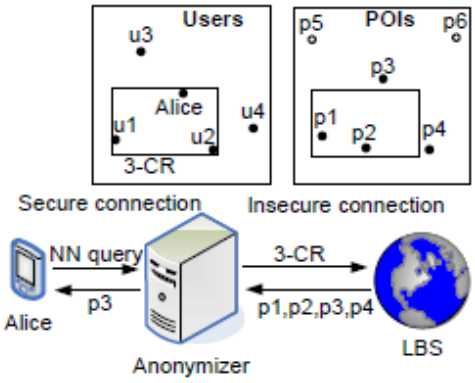

Figure 1: Existing three-tier Architecture

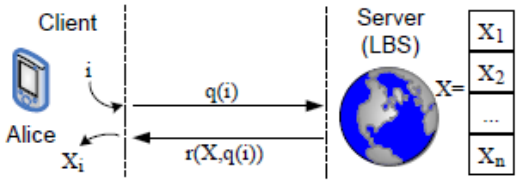

Figure 2: PIR framework

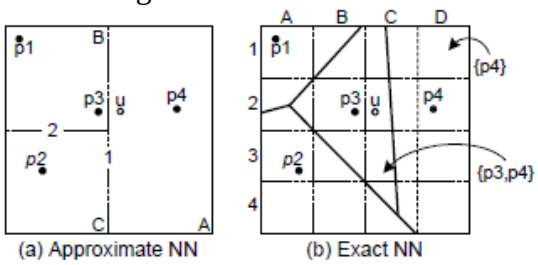

Figure 3: Finding the nearest neighbor of u

2. RELATED WORK

We provide a brief overview of cloaking- and PIR-based approaches for location privacy. A survey and classification of methods for location privacy in LBS can be found in [33]. Similarly, in a position paper in 2008 [11], Ghinita introduced a taxonomy for LBS privacy techniques.

2.1 Location cloaking techniques

Location cloaking in general seeks to prevent an attacker from being able to match queries to particular users and to thus compromise their privacy. The attacker may be in a position to observe traffic flowing through the network or even be situated at the LBS provider endpoint. One popular cloaking technique is based on the principle of k-anonymity, where a user is hidden among k-1 other users. Queries from multiple users are typically aggregated at an anonymity server which forms an intermediary between the user and the LBS provider. This central anonymity server can provide spatial and temporal cloaking functions, so that an attacker will encounter

difficulty matching multiple queries that are observed with users at particular locations and at particular points in time. Many cloaking solutions for location privacy suggest either a central anonymity server as described [18, 34], or other

means such as decentralized trusted peers [9] or distributed k-anonymity [35]. The chief problem is that the anonymity server must normally be part of the trusted computing environment and represents a single point of vulnerability. If it is successfully attacked, or collusion with the LBS server occurs, then the locations of all users may be divulged. It is also observed that although a cloaking technique by itself is advantageous in that it does not result in increased computational cost on the server, it can carry with it a high communication cost from the LBS provider to the client. This can mean a large and unacceptable penalty for mobile phone users. Finally, if a reduced sample population results from the number of active users in a particular geographic area, it may not suffice to satisfy the desired degree of anonymity. If the anonymity server delays execution of a request until the k-anonymity condition is satisfied, then this delay may prove to be unacceptable to the user from a feature interaction point of view.

2.2 PIR-based techniques

A PIR technique can be used to ensure that queries and their results are kept private. Specifically, PIR provides a user with a way to retrieve an item from a database, without the database (or the database administrator) learning any information about which particular item was retrieved. PIR satisfies our requirements for privacy and low communication cost. However, existing PIR techniques have drawbacks of high computational cost for applications that require low latency. The PIR database is typically organized as an n-bit string, broken up into r blocks, each n/r bits long. The user's private input or query is typically an index i 2 {1, ..., r} representing the ith block of bits. A trivial solution for PIR is for the database to send all r blocks to the user and have the user select the desired block at index i, but this carries a maximum cost of communication and is unsuitable in a resource-constrained environment such as a wireless network. When the PIR problem was first introduced in 1995 [7], it was proven that a single-database solution with information theoretic privacy and a sub-linear communication complexity (between the user and the database) is impossible to achieve. Information theoretic privacy assures user privacy even for an adversary with unlimited computational capability. Using at least two replicated databases, and some form of restrictions on how the databases can communicate, PIR schemes with information theoretic privacy are possible, and sometimes hold attractive properties like byzantine robustness [3, 15]. The first single-database PIR proposal was in 1997 [5]; its PIR scheme only assures privacy against an adversary with limited computational capability (i.e., polynomially bounded attackers). The type of privacy protection known as computational privacy, where computational capability is expected to be limited, is a weaker notion of privacy compared to information theoretic privacy. Nonetheless, computational PIR (CPIR) [5, 22] offers the benefit of fielding a single database. Basic PIR schemes place no restriction on information leaked about other items in the database that are not of interest to the user; however, an extension of PIR, known as Symmetric PIR (SPIR) [24], adds that restriction. The restriction is important in situations where the database privacy is equally of concern. The only work in an LBS context that attempts to address both user and database privacy is [12]. Although, not strictly an SPIR scheme, it adopts a cryptographic technique to determine if a location is enclosed inside a rectangular cloaking region. The goal of the paper was to reduce the amount of POIs returned to the user by a query. Unlike ours, the approach fails to guarantee a constant query result size which defeats correlation attacks, and it requires dynamic partitioning of the search space which may be computationally intensive. It also requires two queries to be executed, whereas a single query-response pair is sufficient in ours. PIR has been applied to solving the problem of keeping a user's location private when retrieving location-based content from a PIR database. This content typically consists of points of interest

(POI's), with each entry consisting of a description of a place of interest as well as its geographical location. The only work cited for PIR in the survey from [33] which does not utilize a third party is [13]. Our approach differs from the PIR approach in [13] in three important ways. First, the approach is specifically based on the 1997 computational PIR scheme by Kushilevitz et al. [22]. It would require considerable re-invention before it could be used with recent and more efficient PIR schemes. For instance, it re-organizes a POI database into a square matrix M despite the reduced communications costs attainable from using a rectangular matrix. On the other hand, our approach is flexible and supports any block-based PIR schemes. Secondly, the costs of computation and communication with the approach are O(n) and O(pn), respectively, where n is the number of items, or POIs, in the database. The user has no flexibility for dealing with this linear computational cost for large n and it reveals too many POIs to the user; it is too costly for low-bandwidth devices. Our hybrid technique departs from this one-size-fits-all approach and enables users to negotiate their desired level of privacy and efficiency with LBS providers. Thirdly, the scope of the approach did not consider a privacy-preserving partitioning approach for the data set. It considers partitioning with kd-tree and R-tree in the general sense, without specific privacy considerations (see Section 4.2 in [13]). On the other hand, we will show how to use a different method of partitioning of POI data that permits cloaking, and offers privacy protection when used in conjunction with PIR. Most of the PIR-based approaches for location privacy rely on hardware-based techniques, which typically utilize a secure coprocessor (SC) at the LBS server host [1, 19]. This hardware creates a computing space that is protected from the LBS, to realize query privacy. A major drawback of SC-based PIR is that it requires the acquisition of specialized tamperproof hardware and it usually requires periodic reshuffling of the POIs in the database, which is a computationally expensive operation [1, 20].

2.3 Hybrid techniques

Hybrid techniques [11] permit privacy-efficiency tradeoff decisions to be made by combining the benefits of cloaking- and PIR-based techniques. Chor et al. [8] conjectured a tradeoff between privacy and computational overhead as a means of reducing the high computational overhead for some application areas of PIR. Our work concretizes and validates their conjecture in the context of LBS, and also realizes the future work left open in [11], which is to further reduce the performance overhead of PIR techniques. The authors' own optimization of PIR in [13] (paper previously mentioned above) reuses partial computation results (i.e., multiplications of large numbers) and parallelizes the computations. This optimization reduces CPU cost by 40%, but the overall query response time is still impractical [23, 29]. Ghinita [11] suggests improving the performance of PIR-based techniques for LBS privacy through a hybrid method that includes a PIR phase on a restricted subset of the data space. Our work answers the open question of how to reduce the processing cost of PIR, without requiring the LBS to have multiple CPUs to take advantage of parallelization. Parallel processors are not typically found on smart phones, either.

3. PROPOSED SYSTEM

We have developed a hybrid solution that consists of PIR to achieve query privacy in the context of a location-based service, and a cloaking technique to reduce the computational cost of PIR to a feasible level. Our technique essentially describes how the user creates a cloaking region around his or her true location, and performs a PIR query on the contents of the cloaking region only. The benefits are numerous: the user's location is kept hidden from the server to an acceptable degree regardless of the number of other users in the area; there is no intermediary server that is responsible for cloaking and that would need to be trusted; and the computational cost of the cryptographic algorithms employed is still practical. We ensure that the user downloads only the POIs that are of interest to the

smartphone, keeping wireless traffic to a minimum to reduce costs and conserve the battery. We describe our solution in this section. The approach that we propose entails two phases. First, there is a pre- processing phase in which the system is set up for use. The pre-processing operation must be carried out whenever significant changes are made to the POI database on the server. In practice, it can occur every few months during a period of low usage on the server such as nighttime maintenance activities. Second, there is an execution phase, in which the LBS server responds to queries for POIs from users. At a high level, the pre-processing phase consists of the following steps:

1. A geographic region is projected onto a two-dimensional plane.
2. A suitable grid is formed on the plane.
3. A collection of POIs is saved in a database such that each row corresponds to one POI.
4. Each cell of the grid is mapped to a portion of the database, i.e., a particular set of database rows (each containing a POI).
5. The grid structure is transmitted and saved on the client device in a local mapping database so that it can be referenced in a subsequent query.

The execution phase, in which a query is made for a set of nearby POIs, consists of the following steps:

1. The user determines the area of interest, either based on the current physical position as determined through GPS, or some other arbitrary area that the user may be traveling to in the future.
2. The user chooses a desirable level of privacy.
3. The client creates a cloaking region corresponding to this level of privacy, which will enclose the area of interest.
4. The client sends the cloaking region to the server. Also, the client identifies which portion of the cloaking region contains the area of interest, in a way that is hidden from the server.
5. The server receives the request, and finds the database portion corresponding to the cloaking region. A block of rows is retrieved from this portion based on the user's specified location of interest. The POIs present in these rows are transmitted back to the client.
6. The client decodes the result, and automatically finds the nearest neighbour POI, or presents the full list of POIs returned to the user to choose amongst.

4. CONCLUSION

In this paper, we employed the Private Information Retrieval theory to guarantee privacy in location-dependent queries. To the best of our knowledge, this is the first work to provide a practical PIR implementation with optimizations that achieve reasonable communication and CPU cost. Compared to previous work, our architecture is simpler, more secure (i.e., does not require an anonymizer or collaborating trustworthy users), and is the first one to protect against correlation attacks. Currently, we are working on sophisticated heuristics to generate better optimized execution plans, in order to reduce further the CPU cost. In the future, we plan to investigate the extension of our framework to different types of queries, such as spatial joins.

5. REFERENCES

[1] G. Aggarwal, N. Mishra, and B. Pinkas. Secure Computation of the k th-Ranked Element. In *Proc. of Int Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, pages 40–55, 2004.

[2] R. Agrawal, T. Imielinski, and A. N. Swami. Mining Association Rules between Sets of Items in Large Databases. In *Proc. of ACM SIGMOD*, pages 207–216, 1993.

[3] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Preserving user location privacy in mobile data management infrastructures. In *Int. Workshop on Privacy Enhancing Technologies*, pages 393–412, 2006.

[4] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *IEEE*

Symposium on Foundations of Computer Science, pages 41–50, 1995.

[5] C.-Y. Chow and M. F. Mokbel. Enabling Private Continuous Queries for Revealed User Locations. In *Proc. of SSTD*, pages 258–275, 2007.

[6] C.-Y. Chow, M. F. Mokbel, and X. Liu. A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-based Services. In *ACM International Symposium on Advances in Geographic Information Systems*, 2006.

[7] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2nd edition, 2000.

[8] R. Fagin. Combining Fuzzy Information from Multiple Systems. In *Proc. of ACM PODS*, pages 216–226, 1996.

[9] J. Feigenbaum, Y. Ishai, T. Malkin, K. Nissim, M. Strauss, and R. N. Wright. Secure Multiparty Computation of Approximations. In *Int. Colloquium on Automata, Languages and Programming (ICALP)*, 2001.

[10] D. E. Flath. *Introduction to Number Theory*. John Wiley &
Sons, 1988. [11] B. Gedik and L. Liu. Location Privacy in Mobile Systems: A Personalized Anonymization Model. In *Proc. of ICDCS*, pages 620–629, 2005.

[12] G. Ghinita, P. Kalnis, and S. Skiadopoulos. PRIVE: Anonymous Location-based Queries in Distributed Mobile Systems. In *Proc. of Int. Conference on World Wide Web (WWW)*, pages 371–380, 2007.

[13] O. Goldreich. *The Foundations of Cryptography*, volume 2.
Cambridge University Press, 2004.

[14]. GPSmagazine. Garmin nuvi 780 GPS Review. http://gpsmagazine.com.

[14]. GPSreview.net. POI– Points of Interest. http://www.gpsreview.net/pois/.

[15]. M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services, pages 31– 42, New York, NY, USA, 2003.

[16]. U. Hengartner. Hiding location information from location-based services. In Mobile Data Management, 2007 International Conference on, pages 268–272, May 2007.

[17]. A. Iliev and S. W. Smith. Protecting Client Privacy with Trusted Computing at the Server. IEEE Security and Privacy, 3(2):20–28, 2005.

[18]. M. Kennedy and S. Kopp. Understanding Map Projections. ESRI (Environmental Systems Research Institute) press, 2000.

[19]. E. Kushilevitz and R. Ostrovsky. Replication is not needed: single database, computationally-private information retrieval. In FOCS '97: Proceedings of the 38th Annual Symposium on Foundations of Computer Science, page 364, Wash- ington, DC, USA, 1997.

[20]. D. Lin, E. Bertino, R. Cheng, and S. Prabhakar. Position transformation: a location privacy protection method for moving objects. In SPRINGL '08: Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS, pages 62–71, New York, NY, USA, 2008.

[21]. S. K. Mishra and P. Sarkar. Symmetrically private information retrieval. In IN- DOCRYPT '00: Proceedings of the First International Conference on Progress in Cryptology, pages 225–236, London, UK, 2000.

[22]. M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new Casper: query processing for location services without compromising privacy. In VLDB '06: Proceedings of the 32nd international conference on Very large data bases, pages 763–774, 2006.

[23]. M. Naor and B. Pinkas. Oblivious transfer and polynomial evaluation. In STOC '99: Proceedings of the thirty-first annual ACM symposium on Theory of

computing, pages 245–254, New York, NY, USA, 1999.

[24]. F. Olumofin, P. K. Tysowski, I. Goldberg, and U. Hengartner. Achieving Efficient Query Privacy for Location Based Services. Technical report, CACR 2009-22, University of Waterloo, 2009.

[25]. A. Pingley, W. Yu, N. Zhang, X. Fu, and W. Zhao. CAP: A Context-Aware Pri- vacy Protection System For Location-Based Services. In 29th IEEE International Conference on Distributed Computing Systems, Jun 2009.

[26]. D. Riboni, L. Pareschi, and C. Bettini. Privacy in georeferenced context-aware services: A survey. In Bettini et al. [4].

[27]. F. Saint-Jean. Java implementation of a single-database computationally symmetric private information retrieval (CSPIR) protocol. Technical Report YALEU/DCS/TR-1333A, Yale University, New Haven, CT, USA, 2005.

[28]. R. Sion and B. Carbunar. On the computational practicality of private information retrieval. In Proceedings of the Network and Distributed Systems Security Symposium, 2007.

[29]. J. P. Snyder. Flattening the Earth, two thousand years of map projections. Uni- versity of Chicago Press, 1993.

[30]. A. Solanas, J. Domingo-Ferrer, and A. Mart´ınez-Ballest´e. Location privacy in location-based services: Beyond TTP-based schemes. In Bettini et al. [4].

[31]. T. Xu and Y. Cai. Location anonymity in continuous location-based services. In Proceedings of the 15th Annual ACM international Symposium on Advances in Geographic information Systems, pages 1–8, New York, NY, USA, 2007.

[32]. G. Zhong and U. Hengartner. A distributed k-anonymity protocol for location privacy. In Proceedings of Seventh IEEE International Conference on Pervasive Computing and Communication (PerCom 2009). Galveston, TX, pages 253–262, 2009.