International Journal of Science Engineering and Advance Technology

Location Privacy And Geo Based Applications

Yadlapalli S S Lakshmi Ramani [1], D.Ramesh [2]
[1]PG Scholar, Pydah College of Engineering, Kakinada, AP, India, E-mail: ramani95cse@gmail.com.
[2]Assistant Professor, Pydah College of Engineering, Kakinada, AP, India.

Abstract— Using Geo-social networking like Apple's i Groups and Hot Potato, many people communicate with their neighbouring locations through their associates and their suggestions. Without sufficient location protection, however, these systems can be easily misused, in this paper, we introduce, a technique that provides location secrecy without adding complexity into query results. Our idea here is to secure user-specific, coordinate conversion to all location data shared with the server. The associates of a user share this user's secret key so they can apply the same conversion. This allows all spatial queries to be evaluated correctly by the server, but our privacy mechanisms guarantee that servers are unable to see or infer the actual location data from the transformed data or from the data access.

Key words:Location privacy, security, location-based social applications, location transformation, efficiency.

I. Introduction

Geosocial networking application is using gps location services to provide a social interface to the physical world. Examples of popular social applications include freelancing networks are created with the specific purpose to allow users to find or post temporary employment opportunities. Users establish and operate a professional profile and are able to connect with past and possible employers, employees, colleagues, classmates and friends and social rendezvous [1], local friend recommendations for dining and shopping [2], [3], as well as collaborative network services and games [4], [5]. the explosive popularity of mobile social networks such as scvngr [6] and four square (3 million new users in 1 year) likely indicate that in the future, social recommendations will be our primary source of information about our surroundings. This functionality comes with significantly increased risks to privacy of the users. For current services with minimal privacy mechanisms, these data can be used to infer a user's detailed activities, or to track and predict the user's daily movements. In fact, there are numerous real-world examples where the unauthorized use of location information has been misused for economic gain [7], physical stalking [8], and to gather legal evidence [9]. Even more disturbing, it seems that less than a week after Facebook turned on their popular "Places" feature for tracking users' locations, such location data were already used by thieves to plan home invasion

Clearly, mobile social networks of tomorrow require stronger privacy properties than the open-to-all policies available today. Existing systems have mainly taken three approaches to improving user privacy in Geosocial systems: 1) Introducing uncertainty or error into location data relying on trusted servers or intermediaries to apply Anonymization to user identities and private data and 3) relying on heavy-weight cryptographic or private information retrieval (PIR) techniques. None of them, however, have proven successful on current application platforms. The challenge, then, is to design mechanisms that efficiently protect user privacy without sacrificing the accuracy of the system, To limit misuse, our goal is to limit accessibility of location information from global visibility to a user's social circle. We identify two main types of queries necessary to support the functionality of these Geosocial applications: point queries and nearest neighbour (kNN) queries. Point queries query for location data at aparticular point, whereas kNN queries query for k nearestdata around a given location coordinate.To address this challenge, in this paper, we proposelocation to index mapping), a novel approach to achievinguser privacy while maintaining full accuracy in locationbasedsocial applications (LBSAs from here onward).Our insight is that many services do not need to resolvedistance-based queries between arbitrary pairs of users, butonly between friends interested in each other's locationsand data. Thus, we can partition location data based onusers' social groups, and then perform transformations onthe location coordinates before storing them on untrustedservers. A user knows the transformation keys of all herfriends, allowing her to transform her query into the virtualcoordinate system that her friends use. Our coordinatetransformations preserve distance metrics, allowing anapplication server to perform both point and nearest neighbour queries correctly on transformed data. However,the transformation is secure, in that transformed valuescannot be easily associated with real-world locationswithout a secret, which is only available to the members ofthe social group. Finally, transformations are efficient, inthat they incur minimal overhead on the LBSAs. Thismakes the applications built on LocX lightweight andsuitable for running on today's mobile devices.The challenge, then, is to design mechanisms that efficientlyprotect user privacy without sacrificing the accuracy of thesystem, or making strong assumptions about the security ortrustworthiness of the application servers. More specifically,we target geo-social applications, and assume that servers (andany intermediaries) can be compromised and, therefore, areuntrusted. To limit misuse, our goal is to limit accessibility of circle. We identify two main types of queries necessary tosupport the functionality of these geo-social applications: pointqueries and nearest-neighbour (kNN) queries. Point queriesquery for location

data *at* a particular point, whereas kNNqueries query for k nearest data *around* a given locationcoordinate (or up to a certain radius). Our goal is to supportboth query types in an efficient fashion, suitable for today'smobile devices.To address this challenge, in this paper, we propose *LocX*(short for location to index mapping), a novel approachto achieving user privacy while maintaining full accuracyin location-based social applications (LBSAs from here onwards).Our insight is that many services do not need toresolve distance-based queries between arbitrary pairs of users,but only between friends interested in each other's locationsand data. Thus, we can partition location data based onusers' social groups, and then perform *transformations* on thelocation coordinates before storing them on untrusted servers.A user knows the transformation keys of all her friends,allowing her to transform her query into the virtual coordinatesystem that her friends use. Our coordinate transformationspreserve distance metrics, allowing an application server toperform both point and nearest-neighbour queries correctly ontransformed data. However, the transformation is *secure*, inthat transformed values cannot be easily associated with realworld locations without a *secret*, which is only available tothe members of the social group. Finally, transformations areefficient, in that they incur minimal overhead on the LBSAs.This makes the applications built on LocX lightweight andsuitable for running on today's mobile devices.

II. Related Work

A. Prior Work on Privacy in General Location-Based Service

There are mainly three categories of proposals on providinglocation privacy in general LBSs that do not specificallytarget social applications. First is spatial and temporalcloaking, wherein approximatelocation and time is sent to the server instead of the exactvalues. The intuition here is that this prevents accurateidentification of the locations of the users, or hides the useramong k other users (called k-anonymity),and thus improves privacy. This approach, however, hurtsthe accuracy and timeliness of the responses from the server,and most importantly, there are several simple attacks onthese mechanisms that can still breakuser privacy. Pseudonyms and silent times areother mechanisms to achieve cloaking, where in deviceidentifiers are changed frequently, and data are nottransmitted for long periods at regular intervals. This,however, severely hurts functionality and disconnects users.The key difference between these approaches and our workis that they rely on trusted intermediaries, or trusted servers,and reveal approximate real-world location to the servers inplain text. In LocX, we do not trust any intermediaries orservers. On the positive side, these approaches are moregeneral and, hence, can apply to many location-basedservices, while LocX focuses mainly on the emerging Geosocial applications. The second category is locationtransformation, which uses transformed locationcoordinates to preserve user location privacy. One subtleissue in processing nearest neighbour queries with thisapproach is to accurately find all the real neighbours. Blindevaluation using Hilbert Curves [21], unfortunately, canonly find approximate neighbours. To find real neighbours,previous work either keeps the proximity of transformedlocations to actual locations and incrementally processesnearest-neighbour queries [28], or requires trusted thirdparties to perform location transformation between clientsand LBSA servers [29]. In contrast, LocX does not trust anythird party and the transformed locations are not related toactual locations. However, our system is still able todetermine the actual neighbours, and is resistant againstattacks based on monitoring continuous queries [30], [31].

The third category of work relies on PIR [16] to providestrong location privacy. Its performance, although improvedby using special hardwares [17], is still much worse than allthe other approaches, thus it is unclear at present if thisapproach can be applied in real LBSs.

B. Prior Work on Privacy in Geosocial Services

For certain types of Geosocial services, such as buddytracking services to test if a friend is nearby, some recentproposals achieve provable location privacy [18], [19] usingexpensive cryptographic techniques such as secure twoparty computation. In contrast, LocX only uses inexpensivesymmetric encryption and pseudorandom numbergenerators.The closest work to LocX is Longitude [32], [33], whichalso transforms locations coordinates to prevent disclosureto the servers. However, in longitude, the secrets fortransformation are maintained between every pair of friendsto allow users to selectively disclose locations to friends. Asin, longitude can let a user reveal her location to only asubset of her friends. In contrast, LocX has a simpler threatmodel where all friends can access a user's information andhence the number of secrets that users have to maintain isonly one per user. LocX can still achieve location and user unlinks ability. In addition, LocX can provide more versatile Geosocial services, such as location-based socialrecommendations, reminders, and others, than just buddytracking as in the above prior work.

C. Anonymous Communication Systems

These systems, including Tor [34], provide anonymity tousers during network activity. One might ask, then, whyusing Tor to anonymously route data to LBSA servers is not sufficient. This approach seems to provide privacy as theserver only sees location data but not the identity of the userbehind that data. However, recent research has revealed thathiding the identity of the users alone is not sufficient toprotect location privacy. Even if Tor is used, it is possiblefor an attacker with access to the location data to violate ourprivacy and unlinkability requirements. For example, usinganonymized GPS traces collected by the servers, it has beenshown that users' home and office locations, and even useridentity can be derived [23], [24], [25], [26]. LocX defendsagainst such attacks and meets all our requirements.

D. Systems on Untrusted Servers

In the context of databases, recent systems proposedrunning database queries on encrypted data (stored onuntrusted servers), using heavy-weight homomorphic [35]or asymmetric encryption [36] schemes. These approachesare suitable for spatial data outsourcing or data miningscenarios where the data are static and are owned by limitednumber of users. But they are less suitable for LBSAs,where the data are dynamic and personal, and thus cannotbe encrypted under a single secret key. In the context oflocation and social applications, Persona [37] and Adeona

[38] Also relied on encrypting all data stored on untrustedservers to protect user privacy. Persona focused on privacyin online social networks, and Adeona focused on privacyin device tracking systems where there is no data sharingamong users. Applying Persona's mechanisms to LBSAsdirectly would encrypt all location coordinates, makingLBSAs unable to process nearest-neighbour queries. But iflocation is not encrypted, attacks using anonymized GPStraces, mentioned above, can succeed, and making Personainsufficient to protect location privacy. Similarly, Adeona isuseful for a user to retrieve her own data, but not the datafrom her friends. Our contributions complement thesesystems. Some techniques in these papers can help LocX aswell, for example, Persona's approach to partition datashared with friends into fine-grained groups, and Adeona'shardware-assisted approaches to speed up crypto processing.

Main Modules:

1. **Locx module**
2. **proxy server**
3. **index server**
4. **Data Server**

LOCX Module:

Loc X builds on top of the basic design, and introduces two new mechanisms to overcome its limitations. First, in Loc X, we split the mapping between the location and its data into two pairs: a mapping from the transformed *location to anencrypted index* (called **L2I**), and a mapping from the *index to the encrypted location data* (called **I2D**). This splitting helps in making our system efficient. Second, users store and retrieve the L2Is via *untrusted proxies*. This redirection of data via proxies, together with splitting, significantly improves privacy in LocX. For efficiency, I2Ds are not proxied, yet privacy is preserved (as explained later).

Proxying L2Is for location privacy:

Users store their L2Ison the index server via *untrusted proxies*. These proxies can be any of the following: Planet Lab nodes, corporate NAT sand email servers in a user's work places, a user's home and office desktops or laptops, or Tor [34] nodes. We only need a one-hop indirection between the user and the index server. These diverse types of proxies provide tremendous flexibility in proxying L2Is, thus a user can store her L2Is via different proxies without restricting herself to a single proxy. Furthermore, compromising these proxies by an attacker does not break users' location privacy, as (a) the proxies also only see transformed location coordinates and hence do not learn the users' real locations, and (b) due to the noise added toL2Is (described later). To simplify the description, for now, weassume that the proxies are non-malicious and do not collude with the index server. But we will later describe our solution in detail to even defend against colluding, malicious proxies. With this high-level overview, we now describe our solution to store and query data on the servers in detail. We also explain the challenges we faced, and the tradeoffs we made in makingour solution secure and efficient.

Storing L2I on the index server:

First consider storing L2I on the index server. This transformation preserves the distances between points1, so circular range and nearest neighbour queries for a friend's location data can be processed in the same way on transformed coordinates as on real-world coordinates. Then the user generates a random index (i) using her random number generator and encrypts it with her symmetric key to obtain at the transformed coordinate on the index server via a proxy. The L2I is small in size and is application independent, as it always contains the coordinates and an encrypted random index. Thus the over head due to proxying is very small.

Storing I2Ds on the data server:

The user can directly storeI2Ds (location data) on the data server. This is both secure and efficient.

1) This is secure because the data server only sees the index stored by the user and the corresponding encrypted blob of data. In the worst case, the data server can link all the different indices to the same user device, and then link these indices to the retrieving user's device. But this only reveals that one user is interested in another user's data, but not any information about the location of the users, or the content of the I2Ds, or the real-world sites to which the data in the encrypted blob corresponds to.

2) The content of I2Dis application dependent. For example, a location-based video or photo sharing service might share multiple MBs of data at each location. Since this data is not proxied, LocX still maintains the efficiency of today's systems.

Mechanisms:

In this we use Locx Mechanisms is used in this project.

1) Alice and Bob exchange their secrets,

2) Alice generates and L2I and I2D from her review of the restaurant (at (x, y)), and stores the L2I on the index server via a proxy.

3) She then stores the I2D on the data server directly.

4) Bob later visits the restaurant and fetches for L2Is from his friends by sending the transformed coordinates via a proxy.

5) He decrypts the L2I obtained and then queries for the corresponding I2D, 6) finally Bob decrypts Alice's review.

III.System Design

A. Basic Design

The server should support different types of queries (point,circular range and nearest neighbour queries) on location data.For the server to be able to do this, we need to reveal thelocation coordinates in plain text. But doing so would allowthe malicious server to break a user's location privacy.To resolve this problem, we propose the idea of *coordinatetransformation*. Each user u in the system chooses a set ofsecrets that they reveal only to their friends. These secretsinclude a rotation angle θu, a shift bu, and a symmetric keysymmu. The users exchange their secrets via interactionswhen friends meet in person, or via a separate trustedchannel, such as email, phone etc. The secret angle and shiftare used by the users to transform all the locationcoordinates they share with the servers. Similarly, the secretsymmetric key is used to encrypt all the location data theystore on the servers. These secrets are known only to thefriends, and hence only the friends can retrieve and decryptthe data. For example, when a user u wants to store areview r for a restaurant at (x, y), she would use her secretsto transform (x, y) to (x1, y1) and store encrypted

review E(r)on the server. When a friend v wants to retrieve u's reviewfor the restaurant at (x, y), she would again transform (x, y)using u's secret (previously shared with v), retrieve E(r),and then decrypt it using u's symmetric key to obtain r.

Similarly, v would transform (x, y) according to each of herfriends' secrets, obtain their reviews, and read them. Weonly focus on point queries for now. Figure 1 depicts thisbasic design.

A limitation.

This basic design has one important limitation: the servercan uniquely identify the client devices (for *e.g.*, using theIP address). Using this, the server can associate differenttransformed coordinates to the same user (using the IP).

Sufficient number of such associations can break thetransformations (as we show in Section 5). So maintaining

Unlinkability between different queries is critical. Oneapproach to resolve this limitation is to route all queriesthrough an anonymous routing system like Tor [34]. Butsimply routing the data through Tor all the time will be in efficient. Especially in the context of recent LBSAs, thatadds larger multimedia files (pictures and videos) at eachlocation. So we need to improve this basic design to be bothsecure and efficient.

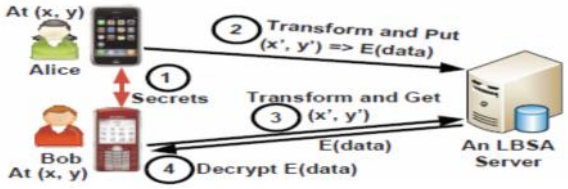

Fig. 1 A basic design

In the basic design

1) Alice and Bob exchange their secrets

2) Alice stores her review of the restaurant (at (x, y)) on theserver under transformed coordinates,

3)Bob later visits the restaurant and queries for the reviewson transformed coordinates,

4) Decrypts the reviews obtained.

B. Overview of LocX

LocX builds on top of the basic design, and introduces twonew mechanisms to overcome its limitations. First, inLocX, we split the mapping between the location and itsdata into two pairs: a mapping from the transformed*location to an encrypted index* (called **L2I**), and a mappingfrom the *index tothe encrypted location data* (called **I2D**). This splittinghelps in making our system efficient. Second, users storeand retrieve the L2Is via *untrusted proxies*. This redirectionof data via proxies, together with splitting, significantlyimproves privacy in LocX. For efficiency, I2Ds are notproxied, yet privacy is preserved

1. Decoupling a location from its data:

Location data data(x, y) corresponding to the realworldlocation (x, y) is stored under (x, y) on theserver. But in LocX, the location (x, y) is firsttransformed to (x1, y1), and the location data isencrypted into E(data(x,y)). Then the transformedlocation is decoupled from the encrypted data using arandom index i via two servers as follows:

1) AN L2I = [(x1, y1),E(i)],which stores E(i) under thelocation coordinate (x1, y1), and

2) AN I2D = [i,E(data(x,y))], which stores the encryptedlocation data E(data(x,y)) under the random index i.The index is generated using the user's secret randomnumber generator. We refer to the server storing L2Isas the *index server* and the server storing I2D as the*data server*. We describe these two as separate serversfor simplicity, but in reality they can be on the sameserver, and our privacy properties still hold. Thisseparation of location information into two components(L2I and I2D) helps us continue to efficiently rundifferent types of location queries on L2Is and retrieveonly relevant I2Ds.Figure 2 depicts the design of LocX.

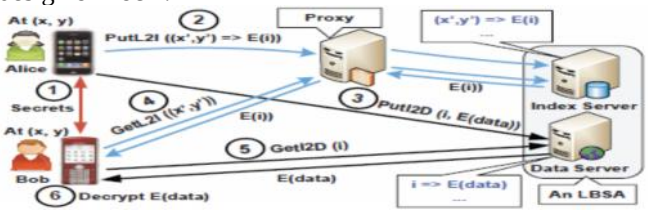

Fig. 2 design of LocX

1) Alice and Bob exchange their secrets, 2) Alice generatesand L2I and I2D from her review of the restaurant (at (x,y)), and stores the L2I on the index server via a proxy. 3)She then stores the I2D on the data server directly, 4) Boblater visits the restaurant and fetches for L2Is from hisfriends by sending the transformed coordinates via a proxy,

5) He decrypts the L2I obtained and then queries for thecorresponding I2D, 6) finally Bob decrypts Alice's review.

2. Proxying L2Is for location privacy:

Users store their L2Is on the index server via *untrustedproxies*. These proxies can be any of the following:PlanetLab nodes, corporate NATs and email serversin a user's work places, a user's home and officedesktops or laptops, or Tor [34] nodes. We only need aone-hop indirection between the user and the indexserver. These diverse types of proxies providetremendous flexibility in proxying L2Is, thus a user canstore her L2Is via different proxies without restrictingherself to a single proxy. Furthermore, compromisingthese proxies by an attacker does not break users'location privacy, as (a) the proxies also only seetransformed location coordinates and hence do notlearn the users' real locations, and (b) due to the noiseadded toL2Is. To simplify the description, for now, weassume that the proxies are non-malicious and do notcollude with the index server. But we will laterdescribe our solution in detail to even defend againstcolluding, malicious proxies. With this high-leveloverview, we now describe our solution to store andquery data on the servers in detail. We also explain thechallenges we faced, and the tradeoffs we made inmaking our solution secure and efficient.

Building Applications Using Locx

Here we sketch how to build LBSAs using LocX. We demonstratethe usage of our APIs by building three applications.In today's systems that provide these services, the data isentrusted to the server in plain-text, which

performs thecomputations in the application logic. Trust the server in LocX, the application logic that computeson the plain-text location data is moved to the client.

Location-based reminders.This application users place remindersfor friends at specific locations (for *e.g.* reminder tobuy milk near a grocery store), and when the friends are atthat location, an alert is generated on their device. To build thisapplication in our model, a user bundles all the details aboutthe reminder, such as the reminder text and time, encrypts thewhole bundle and generates a corresponding I2D. Then theuser transforms the reminder location based on the friend'ssecret and generates a corresponding L2I. These pieces arestored on the servers with a *putL2I* and a *putI2D* calls. Eachuser periodically runs a neighbourhood query for data from herfriends. First the user takes her current location, transforms itaccording to her secret, runs a neighbourhood query, and fetchesthe L2Is and I2Ds, if any, using the *getL2I* and *getI2D* calls.Then the device decrypts and reminds the user as appropriate.***Location-based recommendations.***This application aims torecommend nearby sites (restaurants, shopping malls, etc.)to users based on the reviews given to these sites by theirfriends. In our model, this application is built as follows.A user stores her reviews by generating a bundle containingall the information related to the review, such as the reviewtext, rating, etc., encrypts the bundle using her symmetric key,and generates a L2I and I2D using the data. The locationsof the sites are transformed, of course, while generating theL2Is. This information is then stored on the servers using the*putL2I* and *putI2D* calls. The application on each user's mobiledownloads the data from her friends at the user's currentlocation by running a neighbourhood query. Then it decrypts thereturned data, and plots the recommended sites on a map in thedevice. Thus, the application operates without even revealingusers' location to the servers.***Friend locator.***This application alerts a user whenever a friendis in the vicinity. When this application is built on LocX, userscheck-in at their current location periodically; then users checkfor friends in the vicinity by running a neighbourhood queryaround their current location and decrypting check-ins fromfriends in recent times (*e.g.* last ten minutes). Despite using neighbour query, this approach to building friend locator is stillefficient. Even a hotspot (*e.g.* a concert) in the real coordinatespace is usually *not a hotspot* in the transformed coordinatespace due to user-specific location transformations, and thuslimits the amount of (irrelevant) data received and processedby a user.

IV.Conclusions

This paper describes the design, prototype implementation,and evaluation of LocX, a system for building locationbasedsocial applications (LBSAs) while preserving userlocation privacy. LocX provides location privacy for userswithout injecting uncertainty or errors into the system, anddoes not rely on any trusted servers or components. LocXtakes a novel approach to provide location privacy whilemaintaining overall system efficiency, by leveraging thesocial data-sharing property of the target applications. In

LocX, users efficiently *transform* all their locations sharedwith the server and encrypt all location data stored on theserver using inexpensive symmetric keys. Only friends withthe right keys can query and decrypt a user's data. Weintroduce several mechanisms to achieve both privacy andefficiency in this process, and analyze their privacyproperties. Using evaluation based on both synthetic andreal-world LBSA traces, we find that LocX adds littlecomputational and communication overhead to existingsystems. Our LocX prototype runs efficiently even onresource constrained mobilephones. Overall, we believe that LocX takes a big steptowards making location privacy practical for a large classof emerging geo-social applications.

References

[1] M. Motani, V. Srinivasan, and P. S. Nuggehalli, "Peolenet:engineering a wireless virtual social network," in *Proc. of MobiCom*,2005.

[2] . Hendrickson, "The state of location-based social networking,"2008.

[3] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nercell: richmonitoring of road and traffic conditions using mobile smartphones,"in *Proc. of SenSys*, 2008.

[4] G. Ananthanarayanan, V. N. Padmanabhan, L. Ravindranath, and CA. Thekkath, "Combine: leveraging the ower of wireless peersthrough collaborative downloading," in *Proc. of MobiSys*, 2007.

[5] M. Siegler, "Foodspotting is a location-based game that willmakeyour mouth water," http://techcrunch.com/2010/03/04/foodspotting/.

[6] http://www.scvngr.com.

[7] B. Schilit, J. Hong, and M. Gruteser, "Wireless location privacyprotection,"*Computer*, vol. 36, no. 12, pp. 135–137, 2003.

[8] F. Grace, "Stalker Victims Should Check For GPS," Feb.2003,www.cbsnews.com.

[9] DailyNews, "How cell phone helped cops nail key murder suspectsecret'pings' that gave bouncer away," Mar. 2006.

[10] "Police: Thieves robbed homes based on facebook, social media
sites," WMUR News, September 2010,
http://www.wmur.com/r/24943582/detail.html.

[11] M. Gruteser and D. Grunwald, "Anonymous usage of location-basedservices through spatial and temporal cloaking," in *Proc. of Mobisys*,2003.

[12] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: Aprivacyaware location-based database server," in *ICDE*, 2007.

[13] B. Gedik and L. Liu, "Location privacy in mobile systems: Apersonalized anonymization model," in *Proc. of ICDCS*, 2005.

[14] T. Jiang, H. J. Wang, and Y.-C. Hu, "Preserving location privacy inwireless lans," in *Proc. of MobiSys*, 2007.

[15] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventinglocation-based identity inference in anonymous spatial queries,"*TKDE*, 2007.

[16] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan,"Private queries in location based services:

anonymizers are notnecessary," in *SIGMOD Conference*, 2008.
[17] S. Papadopoulos, S. Bakiras, and D. Papadias, "Nearest neighbour search with strong location privacy," *PVLDB*, 2010.
[18] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, and D.Boneh, "Location privacy via private proximity testing," in *Proc. OfNDSS*, 2011.

[19] G. Zhong, I. Goldberg, and U. Hengartner, "Louis, lester andpierre:Three protocols for location privacy," in *Proc. of PET*, 2007.

[20] N. Daswani and D. Boneh, "Experimenting with electroniccommerceon the palmpilot," in *Financial Cryptography*.Springer,1999.