International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

A Novel Approach of Preserving Location Privacy in Geo-Based Social Applications

Mutahar Sultana[1]& Sayeed Yasin[2]

[1] M.Tech (CSE), Nimra College of Engineering & Technology, A.P., India.
[2]Head of the Department, Dept. of Computer Science & Engineering, Nimra College of Engineering & Technology, A.P., India.

Abstract *—*
Now a days, applications, such as FourSquare, using geo-social technology, enabling millions of people to interact with their surroundings through their friends and their recommendations. Without adequate privacy protection, however, these systems can be easily misused, e.g., to track users or target them for home invasion. In this paper, we introduce LocX, a novel alternative that provides significantly-improved location privacy without adding uncertainty into query results or relying on strong assumptions about server security. Our key insight is to apply secure user-specific, distance- reserving coordinate transformations to all location data shared with the server. The friends of a user share this user's secrets so they can apply the same transformation. This allows all location 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. We show that LocX provides privacy even against a powerful adversary model, and we use prototype measurements to show that it provides privacy with very little performance overhead, making it suitable for today's mobile devices.

Keywords — Location privacy; security; location-based social applications; location transformation; efficiency

I. INTRODUCTION

With the rapidly increasing rate in downloads and annual revenue, smartphone applications offered by Apple iTunes and Android are quickly becoming the dominant computing platform for today's user applications. Within these markets, a new wave of geo-social applications is fully exploiting GPS location services to provide a "social" interface to the physical world. Examples of popular social applications include 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 FourSquare (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.

Unfortunately, this new functionality comes with significantly increased risks to personal privacy. Geo-social applications operate on fine-grain, time-stamped location information. For current services with minimal privacy mechanisms, this 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 was already used by thieves to plan home invasions [10]. Clearly, mobile social networks of tomorrow require stronger privacy properties than the opento- all policies available today.

Existing systems have mainly taken three approaches to improving user privacy in geo-social systems: (a) introducing uncertainty or error into location data [11] relying on trusted servers or intermediaries to apply anonymization to user identities and private

data [12], and (c) relying on heavy-weight cryptographic or private information retrieval (PIR) techniques. None of them, however, have proven successful on current application platforms. Techniques using the first approach fall short because they require both users and application providers to introduce uncertainty into their data, which degrades the quality of application results returned to the user. In this approach, there is a fundamental tradeoff between the amount of error introduced into the time or location domain, and the amount of privacy granted to the user. Users dislike the loss of accuracy in results, and application providers have a natural disincentive to hide user data from themselves, which reduces their ability to monetize the data. The second approach relies on the trusted proxies or servers in the system to protect user privacy. This is a risky assumption, since private data can be exposed by either software bugs or configuration errors at the trusted servers or by malicious administrators. Finally, relying on heavy-weight cryptographic mechanisms to obtain provable privacy guarantees are too expensive to deploy on mobile devices, and even on the servers in answering queries such as nearest-neighbor and range queries.

The challenge, then, is to design mechanisms that efficiently protect user privacy without sacrificing the accuracy of the system, or making strong assumptions about the security or trustworthiness of the application servers. More specifically, we target geo-social applications, and assume that servers (and any intermediaries) can be compromised and, therefore, are untrusted. 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 geo-social applications: point queries and nearest-neighbor (*kNN*) queries. Point queries query for location data at a particular point, whereas *kNN* queries query for k nearest data around a given location coordinate (or up to a certain radius). Our goal is to support both query types in an efficient fashion, suitable for today's mobile devices.

To address this challenge, in this paper, we propose LocX (short for location to index mapping), a novel approach to achieving user privacy while maintaining full accuracy in location-based social applications (LBSAs from here onwards). Our insight is that many services do not need to resolve distance-based queries between arbitrary pairs of users, but only between friends interested in each other's locations and data. Thus, we can partition location data based on users' social groups, and then perform transformations on the location 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 coordinate system that her friends use. Our coordinate transformations preserve distance metrics, allowing an application server to perform both point and nearest-neighbor queries correctly on transformed data. However, the transformation is secure, in that transformed values cannot be easily associated with real world locations without a secret, which is only available to the members of the social group. Finally, transformations are efficient, in that they incur minimal overhead on the LBSAs. This makes the applications built on LocX lightweight and suitable for running on today's mobile devices.

II. SCENARIOS AND REQUIREMENTS

Here we describe several scenarios we target in the context of emerging geo-social applications that involve heavy interaction of users with their friends. We use these scenarios to identify the key requirements of a geo-social location privacy preserving system.

Geo-social Application Scenarios

Scenario 1. Alice and her friends are excited about exploring new activities in their city and leveraging the "friend referral" programs offered by many local businesses to obtain discounts. Alice is currently in downtown and is looking to try a new activity in her vicinity. But she also wants to try an activity that gives her the most discount. The discounts are higher for a user that refers more friends or gets referred by a friend with high referral count. As a result Alice is interested in finding out the businesses recommended by her friends and the discounts obtained through them, within her vicinity. In addition, she is also interested in checking if there are discounts available for her favorite restaurant at a given location. global public key for this user. However, the CA is not involved in any attribute management and the creation of secret keys that are associated with

attributes. For example, the CA can be the Social Security Administration, an independent agency of the United States government. Each user will be issued a Social
Security Number (SSN) as its global identity.

Scenario 2. Alice and her friends are also interested in playing location-based games and having fun by exploring the city further. So they setup various tasks for friends to perform, such as running a few miles at the Gym, swimming certain laps, taking pictures at a place, or dining at a restaurant. They setup various points for each task, and give away prizes for the friends with most points. In order for Alice to learn about the tasks available near her, she needs to query an application to find out all tasks from friends near her and the points associated with them. The scenarios above, while fictitious, are not far from reality. Groupon and LivingSocial are some example companies that are leading the thriving business of local activities. SCVNGR [6] offers similar services as location-based games. But none of these services provide any location privacy to users: all the locations visited by the users are known to these services and to its administrators.

Our goal is to build a system that caters to these scenarios and enables users to query for friends' data based on locations, while preserving their location privacy. We want to support: a) point query to query for data associated with a particular location, b) circular range query to query for data associated with all locations in a certain range (around the user), and c) nearest-neighbor query to query for data associated with locations nearest to a given location. Finally, while it is also useful to query for data that belongs to non-friends in certain scenarios, we leave such extensions for future.

System Requirements

The target scenarios above bring out the following key requirements from an ideal location-privacy service.

- Strong location privacy: The servers processing the data (and the administrators of these servers) should not be able to learn the history of locations that a user has visited.
- Location and user unlinkability: The servers hosting the services should not be able to link if two records belong to the same user, or if a given record belongs to a given user, or if a given record corresponds to a certain real world location.
- Location data privacy: The servers should not be able to view the content of data stored at a location.
- Flexibility to support point, circular range, and nearest neighbor queries on location data.

The need for each of these requirements becomes clearer when we describe the related work and their limitations in more detail in the next section. In our proposed system, LocX, we aim to achieve all these requirements.

III. System Design

Terminology and Attacker Model

Terminology. Location coordinates refer to the longitude, latitude pairs associated with real-world locations. A pair of coordinates is returned from a GPS, and is used to associate data with a location. Location data or location information refers to such data associated with a location. For example, when reviews (and referral point details) are written for a given restaurant, the reviews are the location data associated with the restaurant's location coordinates.

System and Attacker Model. In this paper, we assume that the companies that provide LBSA services manage the servers. Users store their data on the servers to obtain the service. The companies are responsible for reliably storing this data, and providing access to all the data a user should have access to. The companies can get incentives via displaying ads, or charging users some usage fees. In our attacker model, we assume that the attacker has access to the LBSA servers. This attacker could be an employee of the company running the service or an outsider that compromises the servers. The attacker might even be an oppressive regime or a government that obtains data from the providers via subpoenas. As a result, in our model, the attacker can access all

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

the data stored on the servers, and can also monitor which user device is accessing which pieces of information on the servers. Our goal is to design a system that preserves the location privacy of users in this setting. We assume that the attacker does not perform any attacks on the consistency or integrity of data on the servers, but aims only to learn users' location information. Finally, like all prior social systems, we assume that the friends of a user are trusted and do not collude with the servers in breaking the user's privacy.

A Basic Design

To clarify the need for each component in LocX, we start the design description with a basic, simple design.

As listed in our requirements, the server should support different types of queries (point, circular range and nearest neighbor queries) on location data. For the server to be able to do this, we need to reveal the location coordinates in plain text. But doing so would allow the malicious server to break a user's location privacy

To resolve this problem, we propose the idea of coordinate transformation. Each user *u* in the system chooses a set of secrets that they reveal only to their friends. These secrets include a rotation angle θ_u, a shift b_u, and a symmetric key $symm_u$. The users exchange their secrets via interactions when friends meet in person, or via a separate trusted channel, such as email, phone etc. The secret angle and shift are used by the users to transform all the location coordinates they share with the servers. Similarly, the secret symmetric key is used to encrypt all the location data they store on the servers. These secrets are known only to the friends, and hence only the friends can retrieve and decrypt the data.

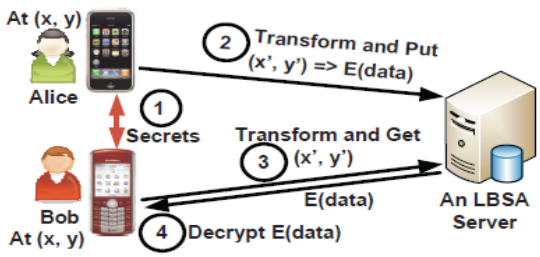

Figure 1 A Basic Design

For example, when a user u wants to store a review r for a restaurant at (x, y), she would use her secrets to transform (x, y) to (x1, y1) and store encrypted review E(r) on the server. When a friend v wants to retrieve u's review for 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 her friends' secrets, obtain their reviews, and read them. We only focus on point queries for now. Figure 1 depicts this basic design.

A limitation. This basic design has one important limitation: the server can uniquely identify the client devices (for e.g., using the IP address). Using this, the server can associate different transformed coordinates to the same user (using the IP). Sufficient number of such associations can break the transformations (as we show in Section 5). So maintaining unlinkability between different queries is critical. One approach to resolve this limitation is to route all queries through an anonymous routing system like Tor [34]. But simply routing the data through Tor all the time will be inefficient. Especially in the context of recent LBSAs, that adds larger multimedia files (pictures and videos) at each location. So we need to improve this basic design to be both secure and efficient.

Overview of LocX

LocX builds on top of the basic design, and introduces two new mechanisms to overcome its limitations. First, in LocX, we split the mapping between the location and its data into two pairs: a mapping from the transformed location to an encrypted 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).

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

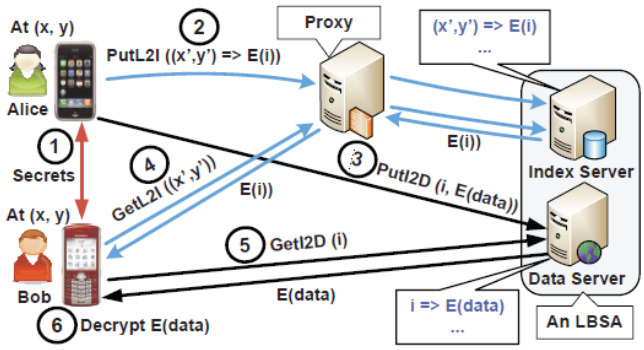

Figure 2 Design of LocX

Proxying L2Is for location privacy. Users store their L2Is on the index server via untrusted proxies. These proxies can be any of the following: PlanetLab nodes, corporate NATs and email servers in a user's work places, a user's home and office desktops or laptops, or Tor 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 to L2Is (described later). To simplify the description, for now, we assume 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 making our solution secure and efficient.

IV. Related Work

Prior work on privacy in general location-based services (LBS). There are mainly three categories of proposals on providing location privacy in general LBSs that do not specifically target social applications. First is spatial and temporal cloaking [11], [12], [13], [15], wherein approximate location and time is sent to the server instead of the exact values. The intuition here is that this prevents accurate identification of the locations of the users, or hides the user among k other users (called k-anonymity [12], [13]), and thus improves privacy. This approach, however, hurts the accuracy and timeliness of the responses from the server, and most importantly, there are several simple attacks on these mechanisms that can still break user privacy. Pseudonyms and silent times [14] are other mechanisms to achieve cloaking, where in device identifiers are changed frequently, and data is not transmitted for long periods at regular intervals. This, however, severely hurts functionality and disconnects users. The key difference between these approaches and our work is that they rely on trusted intermediaries, or trusted servers, and reveal approximate real world location to the servers in plain-text. In LocX, we do not trust any intermediaries or servers. On the positive side, these approaches are more general and, hence, can apply to many location-based services, while LocX focuses mainly on the emerging geo-social applications.

The second category is location transformation, which uses transformed location coordinates to preserve user location privacy. One subtle issue in processing nearest-neighbor queries with this approach is to accurately find all the real neighbors. Blind evaluation using Hilbert Curves, unfortunately, can only find approximate neighbors. In order to find real neighbors, previous work either keeps the proximity of transformed locations to actual locations and incrementally processes nearest-neighbor queries, or requires trusted third parties to perform location transformation between clients and LBSA servers. In contrast, LocX does not trust any third party and the transformed locations are not related to actual locations. However, our system is still able to determine the actual neighbors, and is resistant against attacks based on monitoring continuous queries.

The third category of work relies on Private Information Retrieval (PIR) to provide strong location privacy. Its performance, although improved by using special hardware, is still much worse than all the other approaches, thus it is unclear at present if this approach can be applied in real LBSs.

V. CONCLUSION

In this paper describes the design, prototype implementation, and evaluation of LocX, a system for building location-based social applications (LBSAs) while preserving user location privacy. LocX provides location privacy for users without injecting uncertainty or errors into the system, and does not rely on any trusted servers or components.

LocX takes a novel approach to provide location privacy while maintaining overall system efficiency, by leveraging the social data-sharing property of the target applications. In LocX, users efficiently transform all their locations shared with the server and encrypt all location data stored on the server using inexpensive symmetric keys. Only friends with the right keys can query and decrypt a user's data. We introduce several mechanisms to achieve both privacy and efficiency in this process, and analyze their privacy properties.

Using evaluation based on both synthetic and real-world LBSA traces, we find that LocX adds little computational and communication overhead to existing systems. Our LocX prototype runs efficiently even on resource constrained mobile phones. Overall, we believe that LocX takes a big step towards making location privacy practical for a large class of emerging geo-social applications.

REFERENCES

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

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

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

[4] G. Ananthanarayanan, V. N. Padmanabhan, L. Ravindranath, and C. A. Thekkath, "Combine: leveraging the power of wireless peers through collaborative downloading," in Proc. of MobiSys, 2007.

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

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

[7] B. Schilit, J. Hong, and M. Gruteser, "Wireless location privacy protection," 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 suspect secret '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-based services through spatial and temporal cloaking," in Proc. of Mobisys, 2003.

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

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

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

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

Student

Guide