ISSN:2348-2079

Volume-5 Issue-2

International Journal of Intellectual Advancements and Research in Engineering Computations

Preserving location privacy in geo social applications

[1]Mr.S.Sambasivam M.C.A., M.Phil., Associate Professor,
[2]Mr.G.Vinothkumar Final MCA,
Department of MCA, Nandha Engineering College (Autonomous) Erode-52
Email ID: Sammy2173@gmail.com, vinothkumarplus@gmail.com

Abstract - Using geosocial applications, such as FourSquare, millions of people interact with their surroundings through their friends and their recommendations. Without adequate privacy protection, however, these systems can be easily misused, for example, 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-preserving 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

1. INTRODUCTION

With billions in downloads and annual revenue, smartphone applications offered by Apple iTunes and Android are quickly becoming the dominant comput-in platform for today's user applications. Within these markets, a new wave of geosocial 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 recom-mendations will be our primary source of information about our surroundings.

Unfortunately, this new functionality comes with sig-nificantly increased risks to personal privacy. Geosocial applications operate on fine-grain, time-stamped location information. 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 exam-ples 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 invasions [10].

Clearly, mobile social networks of tomorrow introducing uncertainty or error into location data relying on trusted servers or intermediaries to apply anonymization to user identities and private data and 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 funda-mental 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 global visibility to a user's social circle. Wedentify two main types of queries necessary to support the functionality of these geosocial 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 on-ward). 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 transforma-tions 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, al-lowing 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.

2. SCENARIOS AND REQUIREMENTS

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

2.1.Geosocial 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.

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. 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: 1) point query to query for data associated with a particular location, 2) circular range query to query for data associated with all locations in a certain range (around the user), and 3) 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 nonfriends in certain scenarios, we leave such extensions for future.

2.2.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.

Efficiency in terms of computation, bandwidth, and latency, to operate on mobile devices.

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

3. RELATED WORK

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, 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), 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 are other mechanisms to achieve cloaking, where in device identifiers are changed frequently, and data are 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 geosocial 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. 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 transfor-mation 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 PIR to provide strong location privacy. Its performance, although im-proved by using special hardwares , is still much worse than all the other approaches, thus it is unclear at present if this approach can be applied in real LBSs.

3.1.Prior Work on Privacy in Geosocial Services

For certain types of geosocial services, such as buddy tracking services to test if a friend is nearby, some recent proposals achieve provable location privacy using expensive cryptographic techniques such as secure two party computation. In contrast, LocX only uses inexpensive symmetric encryption and pseudorandom number genera-tors. The closest work to LocX is Longitude, which also transforms locations coordinates to prevent disclosure to the servers. However, in longitude, the secrets for transformation are maintained between every pair of friends to allow users to selectively disclose locations to friends. As in, longitude can let a user reveal her location to only a subset of her friends. In contrast, LocX has a simpler threat model where all friends can access a user's informa-tion and hence the number of secrets that users have to maintain is only one per user. LocX can still achieve location and user unlinkability. In addition, LocX can provide more versatile geosocial services, such as location-based social recommendations, reminders, and others, than just buddy tracking as in the above prior work.

3.2Anonymous Communication Systems

These systems, including Tor , provide anonymity to users during network activity. One might ask, then, why using Tor to anonymously route data to LBSA servers is not sufficient? This approach seems to provide privacy as the server only sees location data but not the identity of the user behind that data. However, recent research has revealed that hiding the identity of the users alone is not sufficient to protect location privacy. Even if Tor is used, it is possible for an attacker with access to the location data to violate our privacy and unlinkability requirements. For example, using anonymized GPS traces collected by the servers, it has been shown that users' home and office locations, and even user identity can be derived. LocX defends against such attacks and meets all our requirements.

3.3 Systems on Untrusted Servers

In the context of databases, recent systems proposed running database queries on encrypted data (stored on untrusted servers), using heavy-weight homomorphic or asymmetric encryption schemes. These approaches are suitable for spatial data outsourcing or data mining scenarios where the data are static and are owned by limited number of users. But they are less suitable for LBSAs, where the data are dynamic and personal, and thus cannot be encrypted under a single secret key.

In the context of location and social applications, Persona and Adeona also relied on encrypting all data stored on untrusted servers to protect user privacy. Persona focused on privacy in online social networks, and Adeona focused on privacy in device tracking systems where there is

no data sharing among users. Applying Persona's mechanisms to LBSAs directly would encrypt all location coordinates, making LBSAs unable to process nearest-neighbor queries. But if location is not encrypted, attacks using anonymized GPS traces, mentioned above, can succeed, making Persona insufficient to protect location privacy. Similarly, Adeona is useful for a user to retrieve her own data, but not the data from her friends. Our contributions complement these systems. Some techniques in these papers can help LocX as well, for example, Persona's approach to partition data shared with friends into fine-grained groups, and Adeona's hardware-assisted approaches to speed up crypto processing.

4. SYSTEM DESIGN

In this section, we describe the design of LocX in detail.

4.1.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 incentivesdisplaying 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 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.

4.2A 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 $_u$, 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 e-mail, phone, and so on. 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.

One approach to resolve this limitation is to route all queries through an anonymous routing system like Tor. But simply routing the data through Tor all the time will be Fig. 1. A basic design. In this design, (1) Alice and Bob exchange their secrets, (2) Alice stores her review of the restaurant (at ðx; yÞ) on the server under transformed coordinates, (3) Bob later visits the restaurant and queries for the reviews on transformed coordinates, and (4) decrypts the reviews obtained.

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.

4.3 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

The key interfaces used by the applications to store and retrieve data on the LocX servers are listed in Table 1. Fig. 2 depicts the 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 e-mail 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 efficiency, I2Ds are not proxied, yet privacy is preserved (as explained later).

Decoupling a location from its data. In today's systems, location data $data_{ðx;yÞ}$ corresponding to the real-world location ðx; yÞ are stored under ðx; yÞ on the server. But in LocX, the location ðx; yÞ is first transformed to $ðx^0; y^0Þ$, and the location data are encrypted into $Eðdata_{ðx;yÞ}Þ$. Then, the transformed location is decoupled from the encrypted data using a random index i via two servers as follows: 1) an L2I ¼ ½ðx^0; y^{0}Þ; EðiÞ&, which stores EðiÞ under the location 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 coordinate ðx^0; y^{0}Þ, and 2) an I2D ¼ ½i; $Eðdata_{ðx;yÞ}Þ$&, which stores the encrypted location data $Eðdata_{ðx;yÞ}Þ$ under the random index i. The index is generated using the user's secret random number generator. We refer to the server storing L2Is as the index server and the server storing I2D as the data server. We describe these two as separate servers for simplicity, but in reality they can be on the same server, and our privacy properties still hold. This separation of location information into two components (L2I and I2D) helps us continue to efficiently run different types of location queries on L2Is and retrieve only relevant I2Ds the data are entrusted to the server in plain text, which perform the computations in the application logic. But since we do not trust the server in LocX, the application logic that computes on the plain-text location data is moved to the client.

Location-based reminders. This application users place reminders for friends at specific locations (e.g., reminder to buy milk near a grocery store), and when the friends are at that location, an alert is generated on their device. To build this application in our model, a user bundles all the details about the reminder, such as the reminder text and time, encrypts the whole bundle and generates a corre-sponding I2D. Then, the user transforms the reminder location based on the friend's secret and generates a corresponding L2I. These pieces are stored on the servers with a putL2I and a putI2D calls. Each user periodically runs a neighborhood query for data from her friends. First the user takes her current location, transforms it according to her secret, runs a neighborhood query, and fetches the 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 to recommend nearby sites (restaurants, shopping malls, etc.) to users based on the reviews given to these sites by their friends. In our model, this application is built as follows. A user stores her reviews by generating a bundle containing all the information related to the review, such as the review text, rating, and so on encrypts the bundle using her symmetric key, and generates a L2I and I2D using the data. The locations of the sites are transformed, of course, while generating the L2Is. This information is then stored on the servers using the putL2I and putI2D calls. The application on each user's mobile downloads the data from her friends at the user's current

location by running a neighborhood query. Then, it decrypts the returned data, and plots the recommended sites on a map in the device. Thus, the application operates without even revealing users' location to the servers.

Friend locator. This application alerts a user whenever a friend is in the vicinity. When this application is built on LocX, users check-in at their current location periodically; then users check for friends in the vicinity by running a neighborhood query around their current location and decrypting check-ins from friends in recent times (e.g., last 10 minutes). Despite using neighbor query, this approach to building friend locator is still efficient. Even a hotspot (e.g., a concert) in the real coordinate space is usually not a hotspot in the transformed coordinate space due to user-specific location transformations, and thus limits the amount of (irrelevant) data received and processed by a user.

5. CONCLUSIONS

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.

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 con-strained mobile phones. Overall, we believe that LocX takes a big step toward making location privacy practical for a large class of emerging geosocial applications.

REFERENCES:

[1] B. Schilit, J. Hong, and M. Gruteser, "Wireless Location Privacy Protection," Computer, vol. 36, no. 12, pp. 135-137, Dec. 2003.

[2] F. Grace, "Stalker Victims Should Check for GPS," http:// www.cbsnews. com, Feb. 2003.

[3] A. Gendar and A. Lisberg, "How Cell Phone Helped Cops Nail Key Murder Suspect. Secret 'Pings' that Gave Bouncer Away,"

[4] "Police: Thieves Robbed Homes Based on Facebook, Social Media Sites," WMUR News, http://www.wmur.com/r/24943582/ detail.html, Sept. 2010.

[5] M. Gruteser and D. Grunwald, "Anonymous Usage of Location-Based Services through Spatial and Temporal Cloaking," Proc. First Int'l Conf. Mobile Systems, Applications Services, 2003.

[6] M.F. Mokbel, C.-Y. Chow, and W.G. Aref, "The New Casper: A Privacy-Aware Location-Based Database Server," Proc. IEEE 23rd Int'l Conf. Data Eng., 2007.

[7] B. Gedik and L. Liu, "Location Privacy in Mobile Systems: A Personalized Anonymization Model," Proc. IEEE 25th Int'l Conf. Distributed Computing Systems, 2005.

[8]T. Jiang, H.J. Wang, and Y.-C. Hu, "Preserving Location Privacy in Wireless Lans," Proc. Fifth Int'l Conf. Mobile Systems, Applications Services, 2007.

[9]P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing Location-Based Identity Inference in Anonymous Spatial Queries," IEEE Trans. Knowledge Data Eng., vol. 19, no. 12, pp. 1719-1733, Dec. 2007.

[10] Necessary," Proc. ACM SIGMOD Int'l Conf. Management Data,