Securing Location of User in Geo Social Networking

Abinayaa.J[1], Abinaya.K[2],Abinaya.R[3], Abinaya.N[4]

[1234]Student, Department of Information Technology, Dr. Mahalingam College of Engineering and Technology, Tamil Nadu, India

Abstract: ***Using Geo-social networking like Apple's iGroups, FourSquare and Hot Potato., many people communicate with their neighboring locations through their neighbors and their suggestions. Without adequate location privacy protection, however, those systems can be easily misused. In this paper, we use a technique that provides improved location privacy without adding uncertainty into query results. Our main idea is to secure user-specific, coordinate transformations to all locations shared with the server. The users can share their secret key to apply for the same transformation. It allows all queries to be evaluated correctly by the user, this privacy mechanism guarantee that the servers are unable to see their own actual location data from the transformed data. We introduce LocX which provides privacy even against powerful opponent model and making it available for all mobile devices.***

Keywords: *Mobile Social Networks, Location privacy, Coordinate Transformation, Proxy.*

1. INTRODUCTION

Nowadays, Geosocial networking application is using GPS location services to provide a social interface to the physical world. Android are quickly becoming the dominant computing platform for today's user applications. Examples of popular social applications include social rendezvous [1], local friend recommendation for shopping and dining [2], [3], as well as collaborative network services and games [4], [5]. The major problem is to design mechanisms that efficiently protect user privacy without sacrificing the accuracy of the system, or making strong assumptions about the security or trust worthiness 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. Mobile social networks require stronger privacy properties than the open-to-all policies available today.

2. SCENARIOS AND REQUIREMENTS

2.1 GEOSOCIAL APPLICATION SCENARIOS

Here we describe several scenarios that 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.

Scenario 1. A and his friends are excited about exploring new activities in their city and leveraging the "friend referral" programs offered by many local businesses to obtain discounts. A is currently in town and is looking to try a new activity in his vicinity, but he also wants to try an activity that gives his 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 A is interested in finding out the businesses recommended by his friends and the discounts obtained through them, within his vicinity. In addition, he is also interested in checking if there are discounts available for his favorite restaurant at a given location.

Scenario 2. X and his 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 X to learn about the tasks available near his, he needs to query an application to find out all tasks from friends near his and the points associated with them. 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 those services provide any location privacy to users: all the locations visited by the users are known to these services and to its administrators.

2.2 NEW 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.

3. CURRENT SYSTEM

3.1 PRIVACY IN GENERAL LOCATION-BASED SERVICE(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, 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 is not transmitted for long periods at regular intervals. The key difference between these approaches and our work is that they rely on trusted intermediaries, or trusted servers, and reveal approximate realworld 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[7] with this approach is to accurately find all the real neighbors. Blind evaluation using Hilbert Curves [8], 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 [9]. 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's, is still much worse than all the other approaches, thus it is Unclear at present if this approach can be applied in real LBSs.

3.2 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 [10], using expensive cryptographic techniques such as secure two party computation. In contrast, LocX only uses inexpensive symmetric encryption and pseudorandom number generators.

The closest work to LocX is Longitude [11], which also transforms location 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 his location to only a subset of his friends. In contrast, LocX has a simpler threat model where all friends can access a user's information 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.

3.3 DESIGN OF LOCX

3.3.1 Basic Design

The server should support different types of 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. For example, when a user u wants to store a review r for a restaurant at (a, b), she would use her secrets to transform (a, b) to (a', b') and store encrypted review E(r) on the server. When a friend v wants to retrieve u's review for the restaurant at (a, b), she would again transform (a, b) using u's secret (previously shared with v), retrieve E(r), and then decrypt it using u's symmetric key to obtain r. Similarly, we would transform (a, b) according to each of his friends' secrets, obtain their reviews, and read them.

Limitation:

This basic design has one important limitation: the server can uniquely identify the client devices. Using this, the server can associate different transformed coordinates to the same user. Sufficient number of such associations can break the transformations. 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 [12]. 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.

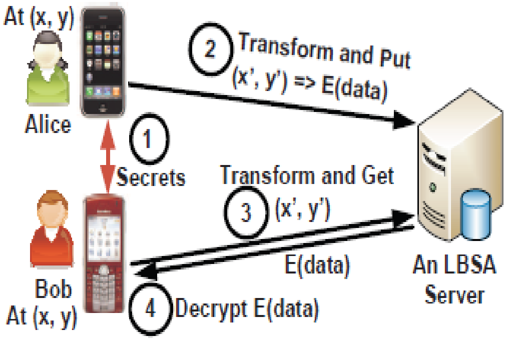

Fig. 1 Basic Design

In the basic design,

- ❖ Alice and Bob exchange their secrets
- ❖ Alice stores her review of the restaurant (at (a, b)) on the server under transformed coordinates
- ❖ Bob later visits the restaurant and queries for the reviews on transformed coordinates
- ❖ Decrypts the reviews obtained

3.3.2 Overview of LocX System

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*.

1. *Decoupling a location from its Data:* Location data (a, b) corresponding to the real world location (a, b) is stored under (a, b) on the server. But in LocX, the location (a, b) is first transformed to (a', b'), and the location data is encrypted into E (data (a, b)). Then the transformed location is decoupled from the encrypted data using a random index i via two servers as follows:*1)* an L2I = [(a1, b1), E(i)], which stores E(i) under the location coordinate (a', b'), and *2)* an I2D = [i, E(data(a, b))], which stores the encrypted location data E(data(a, b)) 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*.

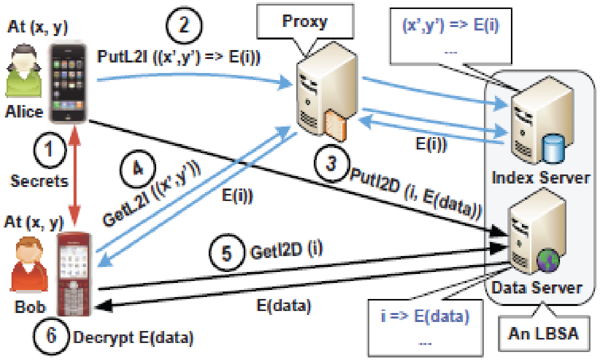

Fig. 2 Design of LocX

2. *Proxying L2Is for location privacy: Users* store their L2Is on the index server via *untrusted proxies.* These proxies can be any of the following: email servers in a user's work places, a user's home and office desktops or laptops, or Tor [13] nodes. Furthermore, compromising those proxies by an attacker doesn't break users location privacy, as (a) the proxies also only see transformed location coordinates and hence do not earn the users real locations, and (b) due to the noise added to L2Is.

4. NEW SYSTEM DESIGN

4.1 PRIVACY PRESERVING DATA STORAGE

When a user generates the location data corresponding to allocation (a,b) uses the secrets to decouple it into a L2I and an I2D. Now we describe how they are stored on the index and the data servers, respectively.

Storing L2I on the index server: First consider storing L2I on the index server. To perform this, the user transforms her real-world coordinate (a,b) to a virtual coordinate (a', b') using his secret rotation angle θ_u and secret shift bu. The user generates a random index (i) using her random number generator and encrypts it with her symmetric key to obtain (Esymm$_u$, (i)). The user then stores this L2I, [(a',b'), Esymm(i)], at the transformed coordinate on the index server via a proxy.

Storing I2Ds on the data server: The user can directly store I2Ds 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 data. 2) The content of I2D is application dependent.

4.2 PRIVACY PRESERVING DATA RETRIEVAL

Retrieving location data from the server in LocX is a more challenging problem. In particular, we need to 1) maintain location privacy, and 2) ensure that the retrieval is efficient. Consider the following simple design for data retrieval. A user takes the location coordinate he is interested in, transforms it according to all his friends' secrets, and sends a query to the server containing all the transformed locations via a proxy. The index server then fetches all the L2Is at the locations in the query and returns them. The user then decrypts all the returned L2Is, and queries the data server for the I2Ds he cares about. There might be collisions on the indices generated by different users. However, as the data in I2D are encrypted using shared symmetric keys, collisions do not lead to unauthorized data access.

4.3 PRIVACY ANALYSIS

We describe here about the intuition behind LocX's privacy and how it meets all the requirements.

Location privacy during server access: Even the attacker with access to monitor both servers cannot link accesses to the index and the data server because the indices stored on the index server are encrypted, but the indices are not encrypted on the data server. Only the users know how to decrypt the encrypted indices. Without the decryption keys, the attacker cannot link these records to figure out even the transformed location of the users accessing the servers.

Location data unlinkability: The I2Ds are encrypted, and the users access them only via indices. Hence, users cannot be linked to any locations. The indices stored or accessed by a user are random numbers. The data server can link together the indices accessed by the same user, but this doesn't help the servers link the user to any locations. Finally, the users store and retrieve L2Is on the index server via proxies, so servers cannot link different transformed locations to the same user. Together, these provide location unlinkability.

4.4 OTHER ATTACKS

Fingerprinting using cookies in incoming connections. Assume that the proxies or the clients scrub with the outgoing connections, using tools such as Privoxy [14], to remove all user-identification information from the connection. This assumption is common to all anonymity-preserving systems, including Tor [15]. Thus, such attacks do not work on LocX.

Localization-based attacks. As the users want to connect to the data server directly, it can attempt to learn users' location using their IPs. So, to prevent these attacks, accessing the server via proxies helps, but it reduces the efficiency of the system. Recently proposed [16] mechanisms can also help in reducing the localization accuracy of the server and even defeating these attacks.

5. EVALUATION

We programmed LocX in Java. We used Blow fish for encryption and decryption. Blowfish has a 64bit block size and a variable key length from 32 bits up to 448 bits. Blowfish is a symmetric block cipher algorithm encrypts block data of 64bits at a time and it will follow the feistel network. To evaluate the overhead that our approach is adding to today's LBSAs with no privacy, we compared LocX with random tags, referred to as LocX, with an implementation of a today's service that has social network on the server and directly maps a location to its data, referred to as L2D. In L2D, data is in plain-text, thus no encryption or decryption is needed.

6. CONCLUSION

This paper describes about the securing of user's location from other users. LocX provides location privacy for users without uncertainty or errors into the system, and does not depend on any trusted servers. In LocX, users can transform all their locations shared with the server and encrypt all location data stored on the server using inexpensive symmetric keys. Only neighbors with the right keys can query and decrypt a user's data. Here we included several mechanisms to achieve both privacy and efficiency in this process. LocX also used for maintaining overall system efficiency, by leveraging the social data-sharing property of the target applications.

We find that Locx adds smaller computational and communication over the existing system by using evaluation based on both synthetic and real-world LBSA traces. In resource constrained mobile phones, LocX prototype runs more efficiently than the unconstrained resource mobile phones. Finally we conclude 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. ACM of MobiCom, 2005.*

[2] *M. Hendrickson, "The State of Location-Based Social Networking,"2008.*

[3] *P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: richmonitoring of road and traffic conditions using mobile smartphones,"inProc. 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/, 2013.*

[6] *A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy," in Proc.of SSTD, 2007.*

[7] *J Maruthi Nagendra Prasad et al, "Securing User Location in Geo Social Networking Using Coordinate Conversions", IJCSIT, Vol. 5 , 2014.*

[8] *M. L. Yiu, C. S. Jensen, X. Huang, and H. Lu, "Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services," in Proc. of ICDE, 2008.*

[9] *D. Lin, E. Bertino, R. Cheng, and S. Prabhakar, "Position transformation: a location privacy protection method for moving objects," in Proc. Of Security and Privacy in GIS and LBS, 2008.*

[10] *A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, and D.Boneh, "Location privacy via private proximity testing," in Proc. OfNDSS, 2011.*

[11] *S. Mascetti, C. Bettini, and D. Freni, "Longitude: Centralized privacy preserving computation of users' proximity," in Proc. OfSDM, 2009.*

[12] *R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second generation onion router," in USENIX Security Symposium, 2004.*

[13] *H. Hu, J. Xu, C. Ren, and B. Choi, "Processing Private Queries over Untrusted Data Cloud through Privacy Homomorphism," Proc. IEEE 27th Int'l Conf. Data Eng. (ICDE), 2011.*

[14] *"Privoxyweb proxy," http://www.privoxy.org/.*

[15] *R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," Proc. 13th Conf. USENIX Security Symp., 2004.*

[16] *P. Gill et al., "Dude where's that IP? Circumventing Measurement based IP Geolocation," in USENIX Security Symposium, 2010.*