SEARCH ME:PRIVACY-PRESERVING LOCATION QUERY SERVICE

Abhay Gaidhani[1], Rohan Sanap[2], Rajat Wadkar[3], Rushikesh Patil[4], Nayan Pawar[5]

[1, 2,3,4,5] *Computer Engineering, Sandip Institute of Engineering & Management, (India)*

ABSTRACT

Location-Based Service (LBS) becomes increasingly popular with the dramatic growth of smartphones and social network services (SNS), and its context-rich functionalities attract considerable users. Many LBS providers use users' location information to offer them convenience and useful functions. However, the LBS could greatly breach personal privacy because location itself contains much information. Hence, preserving location privacy while achieving utility from it is still a challenging question now. This paper tackles this non-trivial challenge by designing a suite of novel fine-grained Privacy-preserving Location Query Protocol (PLQP). Our protocol allows different levels of location query on encrypted location information for different users, and it is efficient enough to be applied in mobile platforms.

Keywords: Location-Based Service, LBS, Social network services, SNS, Privacy-preserving Location Query.

I. INTRODUCTION

Location Based Service (LBS) has become one of the most popular mobile applications due to the wide use of smartphones. The smartphones, equipped with GPS modules, have powerful computation ability to process holders' location information, and this brought the flood of LBS applications in the smartphone ecosystem. A good example is the smartphone camera: if one takes a photo with a smartphone camera, the location where the photo is taken is embedded in the picture automatically, which helps one's remembrance. Furthermore, the explosive growth of social network services (SNS) also assisted its growth by constructing connections between location information and social network. When a picture taken by a smartphone (location embedded) is uploaded to the Facebook album, the system automatically shows the location of the picture on the map, and this is shared with the owner's friends in the Facebook (unless the privacy setting specifies otherwise).

Many similar applications exploit both LBS and SNS. They offer several attractive functions, but location information contains much more information than barely the location itself, which could lead to unwanted information leakage. For example, when Alice and Bob both use check-in application in Facebook (which leaves a location record in one's webpage) in a nice restaurant, it is inferable that they are having a date and that they could be in a relationship. This inference might be an unintended information leakage from Alice's and Bob's perspective. Therefore, a privacy-preserving protocol is needed to prevent significant privacy breach resulted from the combination of LBS and SNS.

The simplest way, which most of applications adopted, is to exert group based access control on published locations: specify a group of user who can or cannot see them. Social photo sharing website Flickr only let users choose all users, neighbours, friends or family to allow the access to the locations, and SNS websites Facebook and Google+ additionally support custom groups to specify the accessible user groups. Mobile applications are much worse. Many mobile applications (e.g., Circle, Who's around and Foursquare) even do not offer group choices to the users, instead, they only ask users whether they want to disclose the location or not. Obviously, this is too simple to achieve what users need. First of all, from users' perspective, it is hard to explicitly determine a user group such that their locations are visible only to them. It is more natural to find a condition such that friends who satisfy it can or cannot see the location. Secondly, binary access control (can or cannot) is far beyond enough to properly configure the privacy setting. In the previous example of the two lovers Alice and Bob, Alice might want to share her date at the restaurant with her best friends and discloses the exact location to them. Besides, Alice might also want other friends to know that she is having a good time in downtown, but not detailed location. In this case, approximate settings between 'can' and 'cannot' are needed to fulfil her requirements.

Existing privacy control settings in LBS are 'coarse' in the sense that:

- Users can only explicitly specify a group of users who can or cannot access the location information.
- Access control policy supports binary choices only, which means users can only choose to enable or disable the information disclosure.
- The existing control strategies also suffer from privacy leakage in terms of the server storage. Even if one disables all of the location disclosure, his location is still open to the server, which in fact is users' top concern.
- Therefore, a fine-grained privacy control executable on encrypted location data is needed to further foster the LBS and its related business market.

1.1 Related Work

There are several works achieving privacy-preserving location query [1]–[4], which are based on k-anonymity model. The k-anonymity model [5] has been widely used to protect data privacy. The basic idea is to remove some features such that each item is not distinguishable among other k items. However, relevant techniques which achieve k-anonymity of data cannot be used in our case for the following four reasons:

1. Those techniques protect the privacy of the data stored in servers. In our PLQP, we do not store the data at all.
2. In LBS, location data is frequently updated, and this dynamic behaviour introduces huge overhead to keep the data k-anonymous.
3. As analysed in Zang *et al.* [6], achieving k-anonymity in location dataset significantly violate the utility of it even for small k, so it is not suitable for our location query protocol.
4. k is generally a system-wide parameter which determines the privacy level of all data in the system, but our goal is to leave the decision of privacy level to each user.

Kido *et al.* [7] proposed a scheme which appends multiple false locations to a true one. The LBS responds to all the reports, and the client only collects the response corresponding to the true location. They examined this dummy-based technique and predicted how to make plausible dummy locations and how to reduce the extra communication cost. However, their technique protects the users' location privacy against LBS provider. We are

also interested in a user's location privacy against other users. In the mix zone model proposed by Beresford *et al.* [8], users are assigned different pseudonyms every time he enters the mixzone, and users' paths are hidden by doing so. Several works [9]–[11] are based on this model, but they guarantee the privacy only when the user density is high and user behaviour pattern is unpredictable. Also, most of them require trusted servers.

There are also works related to CR (cloaking region) [12]–[15]. In these works, the LBS receives a cloaking region instead of actual users' locations. Gedit *et al.* proposed spatial cloaking and temporal cloaking in [12]. Each query specifies a temporal interval, and queries within the same interval, whose sources are in the vicinity of the first query's source, are merged to a single query. Otherwise, the query is rejected because it has no anonymity. Kalnis *et al.* [13] used the Hilbert space filling curve to map the two dimensional locations to one dimensional values, which are then indexed by a B+ tree. Then, they partition the one dimensional sorted list into groups of n users, which is the CR of their scheme. Since this Hilbert Cloaking is not based on geometric space, it guarantees privacy for any location distribution. However, a certain range, where the user is located, is disclosed in CR-based approaches, and this is out of users' control. It is more desirable to allow users themselves to configure it.

1.2 Contribution

This paper proposes a fine-grained Privacy-preserving Location Query Protocol (PLQP) which enables queries to get location information (e.g., Searching a friend's approximate location, Finding nearest friends) without violating users location privacy. This is not a trivial job since simple anonymization makes it impossible to utilize them for queries. Also, if one directly applies queries or functions on the raw location information, privacy leakage is inevitable. Main contributions of our work are three-fold.

- Fine-Grained Access Control: Our protocol allows users to specify a condition instead of a group and exert access control over the users who satisfy this condition. This is more scalable since users can simply add a new condition for new privacy setting instead of hand-picking hundreds of users to form a new group. Also, this is more user-friendly because users themselves do not clearly know which of their friends should or should not access the information most of time.
- Multi-levelled Access Control: The protocol also supports semi-functional encryption. That is, the protocol enables users to control to what extent (or level) others can learn his location. The lowest level corresponds to nothing, and the highest level corresponds to one's exact location. Levels between them correspond to indirect information about one's location.
- Privacy-Preserving Protocol: In our protocol, every location information is encrypted and queries are processed upon cipher texts. Therefore, a location publisher's friends learn nothing but the result of the location query, which is under the location publisher's control. In addition, since every location is encrypted, even the server who stores location information does not learn anything from the cipher text.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We denote every person engaged in the protocol as a user U_i (we do not differentiate smartphone users and PC users), the user who publishes his location as a publisher P_i and the user who queries the location information of other user as a querier Q_i. Note that a user can be a querier and a publisher at the same time. When he queries on others, he acts as a querier and when he is queried, he acts as a publisher. That is, $U_i = P_i = Q_i$ for the same i. Also, mobile applications or SNS applications which support LBS are denoted as service providers SP. Q and

Pretrieves keys from SP, which are used for access control. For simplicity, we consider only one SP here. We assume an independent semi-honest model for users and service providers. That is, they all behaveindependently and will try to extract useful information from the ciphertexts, but they will follow the protocol in general and will not collude with each other. We further assume that every user communicate with each other via an anonymized network (e.g., Tor: https://www.torproject.org) or other anonymized protocol such that the privacy is not compromised by the underlying network protocol. We assume the origin of a packet is successfully hidden, which is out of this paper's scope (otherwise any attacker can achieve the location based on the origin of the packet).

2.1 Location Assumption

For simplicity, we assume the ground surface is a plane,and every user's location is mapped to a Euclidean spacewith integer coordinates (with meter as unit). That is, everyone'slocation can be expressed as a tuple of coordinatesrepresenting a point in a grid partition of the space. This doesnot affect the generality since there exists a bijection betweenspherical locations and Euclidean locations. By approximatingthe coordinates in the Euclidean space to the nearest grid point,we can show that it results in errors of the Euclidean distancebetween two locations at most $\sqrt{2}$meters when the space ispartitioned using grid of side-length 1 meter.The Euclidean distance between two users with locations$x_1 = (x_{11}, x_{12}, x_{13})$ and $x_2 = (x_{21}, x_{22}, x_{23})$ is,

$$dist\ (U_1, U_2) = |x_1 - x_2| = \sqrt{\sum_{i=1}^{3}(x_{1i} - x_{2i})^2} \quad ..(1)$$

Given a real location on the surface of the earth, we need to compute the surface distance, denoted as$SD(U_i, U_j)$, between these two points. By assuming that the earth is a sphere with radius R meters, it is easy to show that,

$$SD(U_i, U_j) = 2 \arcsin\left(\frac{dist(U_i U_j)}{2R}\right).R \quad ..(2)$$

Then the surface distance can be quickly computed from the Euclidean distance. To check if the surface distance satisfies certain conditions, we can convert it to check if the Euclidean distance satisfying corresponding conditions. For example, $dist(U_1, U_2) \geq D$ is equivalent as $SD(U_i, U_j) \leq 2 \arcsin\left(\frac{dist(U_i U_j)}{2R}\right)$. For simplicity and convenience of presentation, in this paper, we will focus on the Euclidean distance instead of the surface distance. Notice that although we consider only Euclidean space here, our protocol works for any system where distance is a polynomial of location points$\boldsymbol{x}$'s, where $\boldsymbol{x}$ is a vector.

2.2 System Implementation Models

2.2.1 Data-Flow Diagram

Through data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It is common practice to draw the context-level data flow diagram first, which shows the interaction between the system and external agents which act as data sources and data sinks. This helps to create an accurate drawing in the context diagram. The system's interactions with the outside world are modelled purely in terms of data flows across the system

boundary. The context diagram shows the entire system as a single process, and gives no clues as to its internal organization.

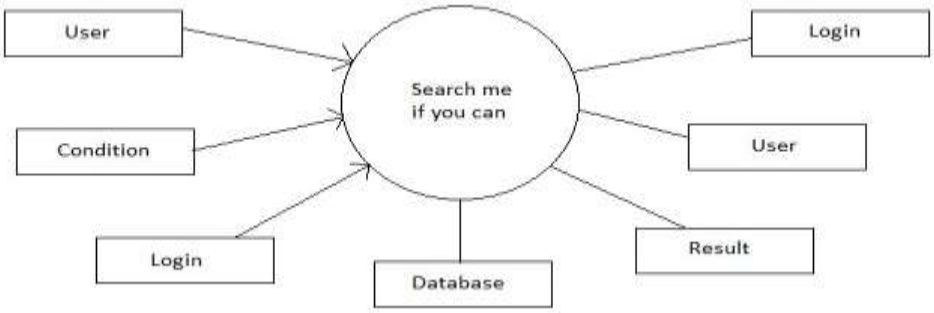

Fig. 1: Data-Flow Diagram

- User: one user sends a query and another user will accept that query and give the result as per his/her priority.
- Condition: For the authentication login provision is present so that querier will authenticate him/herself. If the authentication is appropriate then one can proceed forward.
- Login: login format should be standard so that one can authenticate it easily.
- Database: this section will store the information about persons for their location searching.
- Result: Once the authentication process is right then the querier will get appropriate result for next step.

2.2.2 Use-Case Diagram

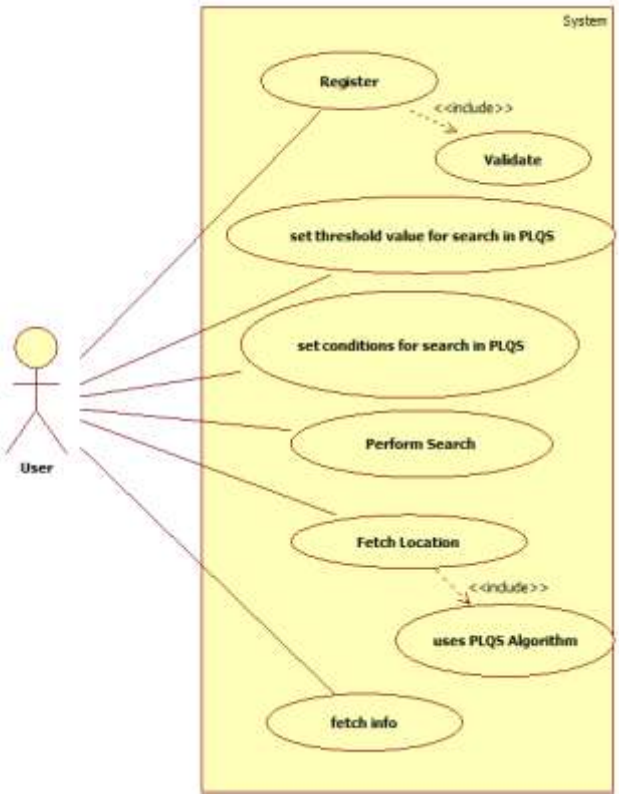

Fig. 2: Use-Case Diagram

2.2.3 Flow Chart

The flow chart explains the work flow of the system. Following is the flow chart of the proposed system that we have designed:

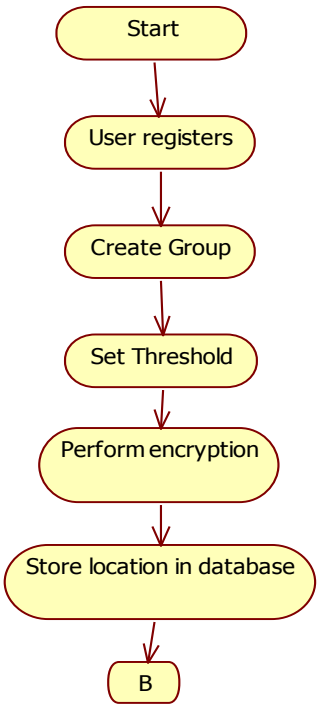

Fig. 3: Flow Chart

2.2.4 Activity Diagram

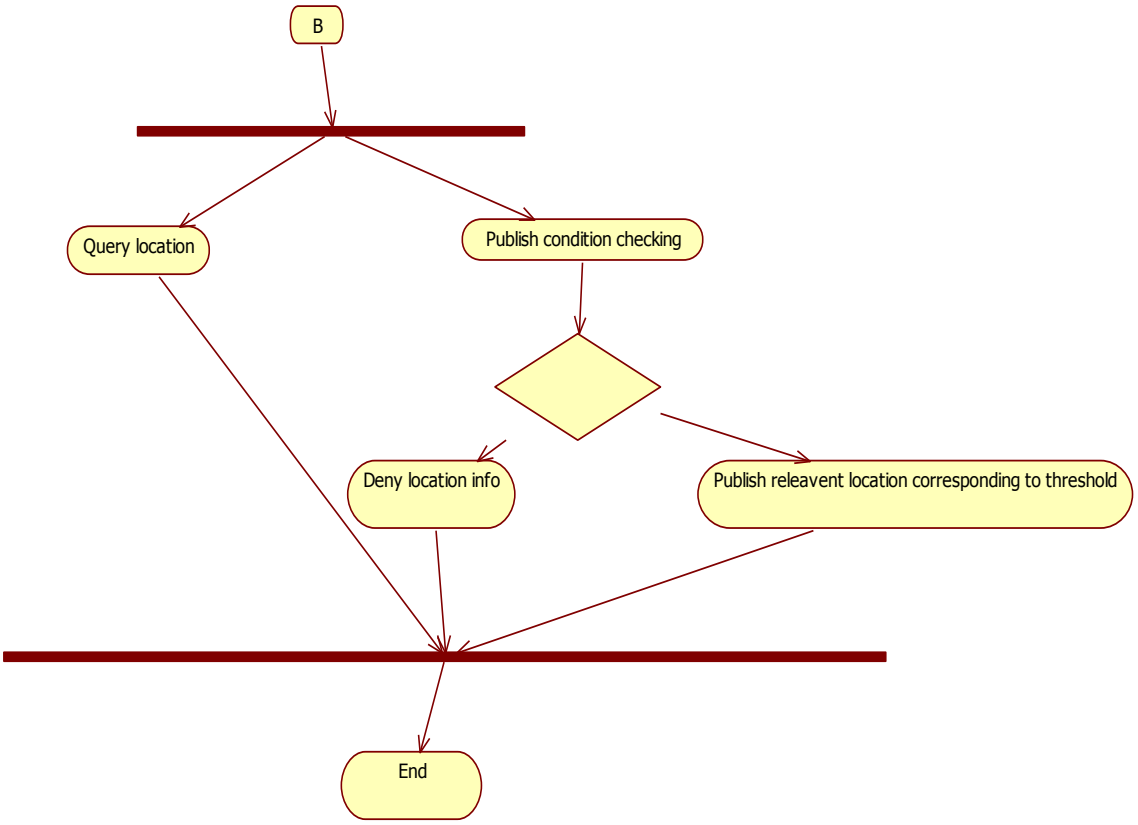

Fig. 4: Activity Diagram

III. SYSTEM IMPLICATION

3.1 Advantages

- Proposed protocol allows users to specify a condition instead of a group and exert access control over the users who satisfy this condition.
- The protocol also supports semi-functional encryption. That is, the protocol enables users to control to what extent (or level) others can learnhis location.
- In proposed protocol, location information is encrypted and queries areprocessed upon cipher texts.
- Disadvantages
- Proposed system is going to be developed for android users. So only android users can take advantages of proposed system.
- Proposed system is based on GPS and Location Based Services (LBS), so mobile tracker like application still able to access location information.

Well question may be raised over our disadvantages but the fact is that the mobile tracking applications are used by very high authorities to keep track on the people doing unlawful activities. Such authorities involve police department and military department. Our project will keep user's location safe from the spamers who supply malwares and unnecessary advertisements. As we are procceding on one thing at a time, in future, we will also get over this disadvantage and keep the user's location totally intact from outrer world and sharing based on user's choice only.

3.2 Applications

There are various applications of our project. The biggest example is that a user will be having good control over his location sharing as it is not available now. A normal user can use it to avoid his problems created by locations. His privacy will be not penetrated.

This application can also be used keep the user location safe of getting embedded in various files such as photos, videos, etc. This is done by the system without user's permission.

Similarly there can be many such possibilities of this project if implemented on deep system level functionalities and if adapted by big system developers as the basic and fundamental protocol.

IV. CONCLUSION

In this paper, we proposed a fine-grained Privacy-preserving Location Query Protocol (PLQP), which successfully solves the privacy issues in existing LBS applications and provides various location based queries. The PLQP uses our novel distance computation and comparison protocol to implement semi-functional encryption, which supports multi-levelled access control, and used CP-ABE as subsidiary encryption scheme to make access control be more fine-grained. Also, during the whole protocol, unless intended by the location publisher, the location information is kept secret to anyone else. We also conducted experiment evaluation to show that the performance of our protocol is applicable in a real mobile network.

REFERENCES

Journal Papers

[1] T. Hashem and L. Kulik,, Safeguarding location privacy in wireless ad-hoc networks, *Ubicomp 2007: Ubiquitous Computing, pp. 372–390, 2007.*

[2] C. Bettini, X. Wang, and S. Jajodia, Protecting privacy against location based personal identification, *Secure Data Management, pp. 185–199,2005.*

[3] M. Mokbel, C. Chow, and W. Aref, The new casper: query processing for location services without compromising privacy, *in Proceedings of the 32nd international conference on Very large data bases, VLDB Endowment, 2006, pp. 763–774.*

[4] K. Vu, R. Zheng, and J. Gao, Efficient algorithms for k-anonymous location privacy in participatory sensing, *IEEE INFOCOM, 2012.*

[5] L. Sweeney *et al.*, k-anonymity: A model for protecting privacy, *International Journal of Uncertainty Fuzziness and Knowledge Based Systems, vol. 10, no. 5, pp. 557–570, 2002.*

[6] H. Zang and J. Bolot, Anonymization of location data does not work: A large-scale measurement study, *Proceedings of the 17th annual international conference on Mobile computing and networking, 2011, pp. 145–156.*

[7] H. Kido, Y. Yanagisawa, and T. Satoh, Protection of location privacy using dummies for location-based services, *21st International Conference on Data Engineering Workshops, 2005, pp. 1248–1248.*

[8] A. Beresford and F. Stajano, Mix zones: User privacy in location-aware services, *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004, pp. 127–131.*

[9] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J. Herrera, A. Bayen, M. Annavaram, and Q. Jacobson, Virtual trip lines for distributed privacy-preserving traffic monitoring, *in Proceeding of the 6th international conference on Mobile systems, applications, and services, ACM, 2008, pp. 15–28.*

[10] M. Li, K. Sampigethaya, L. Huang, and R. Poovendran, Swing & swap: user-centric approaches towards maximizing location privacy, *Proceedings of the 5th ACM workshop on Privacy in electronic society, 2006, pp. 19–28.*

[11] X. Liu, H. Zhao, M. Pan, H. Yue, X. Li, and Y. Fang, Traffic-aware multiple mix zone placement for protecting location privacy, *IEEE INFOCOM 2012.*

[12] B. Gedik and L. Liu, Location privacy in mobile systems: A personalized anonymization model, *Proceedings of 25th IEEE International Conference on Distributed Computing Systems, 2005, pp. 620–629.*

[13] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, Preventing location-based identity inference in anonymous spatial queries, *IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 12, pp. 1719–1733, 2007.*

[14] Y. Wang, D. Xu, X. He, C. Zhang, F. Li, and B. Xu, L2p2: Location aware location privacy protection for location-based services. *IEEE INFOCOM 2012*.

[15] C. Chow, M. Mokbel, and X. Liu, A peer-to-peer spatial cloaking algorithm for anonymous location-based service, *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems, 2006, pp. 171–178.*

[16] Y. Liu, J. Han, and J. Wang, Rumor riding: anonymizing unstructured peer-to-peer systems, *IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 3, pp. 464–475, 2011.*