Privacy-Aware Location Database Service for Granular Queries

Shinsaku Kiyomoto[1], Keith M. Martin[2], and Kazuhide Fukushima[1]

[1] KDDI R & D Laboratories Inc.
2-1-15 Ohara, Fujimino-shi, Saitama
356-8502, Japan
kiyomoto@kddilabs.jp
[2] Information Security Group,
Royal Holloway University of London
Egham, Surrey TW20 0EX, UK
Keith.Martin@rhul.ac.uk

Abstract. Future mobile markets are expected to increasingly embrace *location-based services*. This paper presents a new system architecture for location-based services, which consists of a location database and distributed location anonymizers. The service is privacy-aware in the sense that the location database always maintains a degree of anonymity. The location database service permits three different levels of query and can thus be used to implement a wide range of location-based services. Furthermore, the architecture is scalable and employs simple functions that are similar to those found in general database systems.

Keywords: *Location-based services, Privacy, k-anonymity*

1 Introduction

Future mobile markets are expected to increasingly embrace *location-based services*. These permit roaming users to locate local service providers (hotels, railway stations etc.), as well as enabling services that track a mobile user such as child safety applications. Although it creates many new service opportunities, the ability to locate and track mobile users also represents a threat to *location privacy* [3] if location information falls into the hands of unauthorized parties.

It is clear that location-based services cannot be offered without users providing some degree of location information, so the demand for services and the desire for location privacy have the potential to conflict. There are three common approaches to obfuscating location information to provide *privacy-aware* location-based services [13]:

1. Kido *et. al.* proposed a *false dummy method* [10], where a user sends n different locations to a location database server, with only one of them being correct (the rest are "dummies" that mask the true location). While effective for location privacy, this type of masking is obstructive for location-based service providers.
2. Hong and Landay introduce an architecture based on *landmark objects* [9], where users refer to the location of a significant object (landmark) in their vicinity, rather than sending an exact location. This scheme makes it difficult to control the granularity of location information and thus may not be suitable for some types of location-based services.
3. For many service providers it is sufficient to provide *approximate*, rather than *exact* location information. The idea of *location perturbation* is to blur the exact location information. Recent research [13] has focused on establishing location anonymity in a spatial domain. This approach uses a *location anonymizer*, which is a trusted server that anonymizes location information within a defined *anonymizing spatial region* (ASR). Location anonymity is provided to the extent that an attacker cannot determine precisely where a given user is in the ASR (although they do know that they are located in the ASR).

Location perturbation provides an attractive compromise approach, but most existing schemes have two potentially restrictive features:

1. They utilize a centralized location anonymizer which gathers location information for all users, thus creating a potential bottleneck.
2. They focus on *private queries*, which seek the approximate location of a specific user, but do not permit other types of location-based query such as personal tracking services (see Section 3.2).

In this paper, we propose a new system architecture for location-based services which avoids the bottleneck of a centralized anonymizer and can be used to support granular service queries. More particularly:

- Location information is registered in a common *location database*, but the task of anonymizing location information is distributed across a number of independent location anonymizers. This provides better scalability and reliability than the centralized approach.
- The location database accepts three types of queries, public, private, and personal, which can be used to implement a variety of different location-based services.
- Location anonymization uses structured location information and satisfies a notion of anonymity for public databases called k-*anonymity*

(see Section 3.3) even when there is an unbalanced distribution of users. Furthermore, users can control the granularity of location information provided to the other parties if desired.

2 Related Work

Various location perturbation techniques have been suggested for obfuscating location information. Gruteser and Grunwald [8] suggested "blurring" the user's location by subdividing space in such a way that each subdivision has at least $k-1$ other users. Gedik and Liu [6] adapted this to allow users to have personalized values of the masking parameter k. Mokbel *et. al.* presented a hierarchical partitioning method to improve the efficiency of location perturbation [14], however it was shown in [7] that this fails to provide location anonymity under non-uniform distribution of user locations. Selection of optimal subdivision spaces was investigated in [12, 1]. Finally, in [7] a decentralized approach without an anonymizer was considered in order to realize a good load balancing property, however communication between users is required to calculate anonymized location information.

More generally, *private information retrieval* models privacy leakage from queries to a database. Several schemes have been proposed to hide query information from the database that receives the queries [5]. Thus these schemes focus on privacy of query senders, which are the service providers in our scenario. In contrast, we wish to protect privacy of users who register location data with a location database.

Another approach to protecting data privacy in a database is to use *searchable encryption schemes* [16, 4]. These provide secure search functions for encrypted data. However, these schemes require a cryptographic computation for each record in the database, making them relatively inefficient. Bellare *et. al.* [2] proposed an efficient searchable encryption scheme that retrieves data in time logarithmic to the size of the database, however it has a relatively high false-positive response.

Thus it would seem that the problem of providing privacy-aware location-based services in an efficient and scalable manner remains open.

3 Preliminaries

3.1 System Architecture

The system architecture is shown in Figure 1. The system consists of the following four entities:

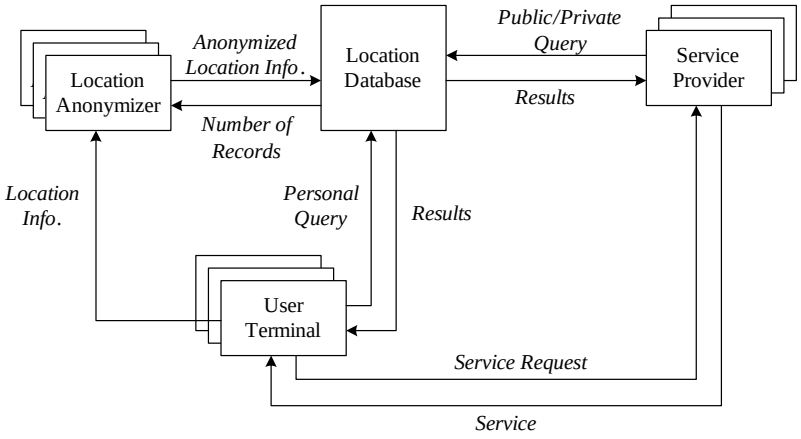

Fig. 1. System Architecture

- Location Database. This centralized database stores location information of users. The location database is assumed to be honest and allows public access to registered location information.
- User Terminals. We assume that users carry mobile devices with embedded positioning capabilities which have access to a wireless network. As in previous research, it is assumed that the network address of a user terminal is a virtual address that changes when the terminal moves and which cannot be used to identify its real location.
- Location Anonymizers. These trusted servers receive location information from user terminals and register/update anonymized location information to the location database. They are responsible for checking whether it is "safe" (from a privacy perspective) to register location information. We assume that secure and authenticated communication channels exist between location anonymizers and the other entities.
- Service Providers. Service providers provide a variety of location-based services to user terminals using location information from the location database.

3.2 Service Model

A general service model is now described consisting of three types of location database query. *Public* and *private* queries were previously defined in [13], however we extend the service model to include *personal* queries:

- *Public Queries.* These are general queries regarding user location information. An example is a query about the approximate number of users in a certain area. Public queries are the least threatening to location privacy since they often only require approximate answers and do not involve specific user identities.
- *Private Queries.* These relate to *approximate* location information of a specific user. Location privacy is clearly threatened by this type of query, hence the importance of applying suitable location perturbation techniques.
- *Personal Queries.* These relate to *exact* location information of a user. Personal queries should only be satisfied if the inquirer is authorized to extract such information from the location database. Personal queries can be used to enable user tracking.

3.3 Location Privacy Requirement

The main location privacy requirement of our system is that service providers should only know the granularity of location information of a user that they are entitled to. As a measure of anonymity of location information we adopt the notion of *k-anonymity* of a public database [15], which is satisfied if each database record is indistinguishable from at least $k-1$ other records with respect to specific identifying attributes. In our context, if the location database provides k-anonymity then any (private) query which requires location obfuscation should result in the attributes defined in the query (normally defining an ASR) matching at least k potential locations stored in the location database. This results in the inquirer having a predefined degree of uncertainty about the exact location of the target user.

3.4 Representation of Location Information

We represent location information using an intuitive hierarchical structure, which could be instantiated by geopolitical boundaries (country, state, city, street, post code) or topographical coordinates (such as structured granularity of GPS data). Location information of user i is represented as an l-tuple of attributes $(A_i^1, A_i^2, \ldots, A_i^l)$, where A_i^j is more fine-grained than A_i^{j-1} in the hierarchy of location information. Users can thus control disclosure of their location information by concealing attributes beyond a specific layer of granularity. For example, if user i wishes to only reveal three layers of location information then they specifiy their location as $(A_i^1, A_i^2, A_i^3, -, \ldots, -)$, where $-$ is an empty field.

3.5 Anonymized Identifiers for Location Database

An important feature of our system is that each location database entry is indexed in the location database using an *anonymized identifier*, which is randomly generated. This anonymized identifier distinguishes the location database entry from all others, but the relationship between this anonymized identifier and the user should only be known by the user themselves and any parties (service providers) authorized by them. We will use the notation $I_{i_t}^j$ to denote an anonymized identifier associated with a location database entry for user i at time t that reveals j layers of location information. Hence the location database entry itself will be denoted by $(I_{i_t}^j, A_{i_t}^1, A_{i_t}^2, \ldots, A_{i_t}^j, -, \ldots, -)$.

4 Operation of the Location Database Service

We now explain how our privacy-aware location database service operates.

4.1 Functions of Location Database

The location database has five functions:

- **Register.** On inputting a record $(I_{i_t}^j, A_{i_t}^1, \ldots, A_{i_t}^j, -, \ldots, -)$, the register function adds this record to the location database.
- **Delete.** On inputting anonymized identifier $I_{i_t}^j$, the delete function removes the record in the location database that is indexed by $I_{i_t}^j$.
- **Count.** On inputting a query $(A_{i_t}^1, \ldots, A_{i_t}^j, -, \ldots, -)$, the count function outputs the number of records in the location database that have the same attribute values.
- **Search.** On inputting anonymized identifier $I_{i_t}^j$, the search function outputs the record in the location database that is indexed by $I_{i_t}^j$.
- **Refer.** On inputting a range (group) of attributes, the refer function outputs all records in the location database that have matching attributes.

4.2 Registration of Location Information

We now explain the critical process of registering location information on the location database. A user first contacts their chosen location anonymizer and submits location information. The location anonymizer then generates anonymized identifiers and prepares potential location database entries. However, these potential entries will only be registered

by the location anonymizer after they have checked that registering this location information does not contravene the required level of k-anonymity. If the location information does not conform to k-anonymity then the location anonymizer modifies the entry before registering it.

Prior to submitting location information to a location anonymizer, the user provides the location anonymizer with an initial anonymized identifier and secret keys that will be needed for the database entry randomization function (see Section 4.4). The process of registering location information is as follows:

1. User i at time t sends location information $(A^1_{i_t}, A^2_{i_t}, \ldots, A^l_{i_t})$ to the location anonymizer.
2. The location anonymizer generates l anonymized identifiers $I^j_{i_t}$ (for $j = 1, \ldots, l$), as described in Section 4.3.
3. The location anonymizer generates l potential database entries of the form $(I^j_{i_t}, A^1_{i_t}, A^2_{i_t}, \ldots, A^j_{i_t}, -, \ldots, -)$ (for $j = 1, \ldots, l$) from the location information.
4. For $j = 1, \ldots, l$:
   (a) The location anonymizer submits a count query for the potential database entry indicated by $I^j_{i_t}$ to the location database.
   (b) The location database returns the number of matching entries $z(I^j_{i_t})$ to the location anonymizer.
   (c) If $z(I^j_{i_t}) \geq k$ then the location anonymizer registers the tuple indexed by $I^j_{i_t}$ with the location database.
   (d) If $z(I^j_{i_t}) < k$ then for $\lambda = j, \ldots, l$ the location anonymizer replaces the potential database entry indexed by $I^\lambda_{i_t}$ with:

$$(I^\lambda_{i_t}, A^1_{i_t}, A^2_{i_t}, \ldots, A^{j-1}_{i_t}, f(k^j_{i_t}, A^j_{i_t}), \ldots, f(k^\lambda_{i_t}, A^\lambda_{i_t}), -, \ldots, -),$$

   where the generation of $k^j_{i_t}$ is explained in Section 4.4. The location anonymizer then registers the replacement database entries indexed by $I^j_{i_t}, \ldots, I^\lambda_{i_t}$ with the location database and sets $j = l+1$ (to halt the subprotocol).

In the above process, as soon as one prospective database entry fails the test for k-anonymity, both this entry and all entries containing more fine-grained location information (which will also by default fail the k-anonymity test) are randomized and registered with the location database.

Note that one problem with this process is that if the location database fails to satisfy k-anonymity with respect to certain location attributes

then it will never satisfy k-anonymity with respect to these since such entries are never registered with the location database. One solution to this is that location anonymizers cache failed prospective database entries along with their *deficiency* (how far short of k-anonymity the database was with respect to them). Once the location anonymizer receives enough new prospective database entries to account for this deficiency, they recheck the location database and, if appropriate, block registers the cached entries (and deletes the corresponding randomized entries). Such a technique could also be used when initially populating the location database.

4.3 Generation of Anonymized Identifiers

In our system the location anonymizers generate anonymized identifers, rather than have them sent to them by users. This is simply for reasons of efficiency, as it saves bandwidth. It is thus important for the generation of anonymized identifiers to be synchronized between users and location anonymizers. In addition, since user devices may have restricted power, an efficient process for generating anonymized identifiers is desirable. We thus use a *one-way hash function chain* [11] to generate anonymized identifiers, as follows:

$$I_{i_0}^j \leftarrow H(I_{i_0}^0 | r_i^j), \quad r_i^j \in_R \{0,1\}^n, \quad I_{i_t}^j \leftarrow H(I_{i_{t-1}}^j | r_i^j),$$

where n is a security parameter, $|$ is concatenation of data, $I_{i_0}^0$ is an initial identifier of user i, r_i^j is an n-bit random number for the j-th tuple of attributes, and H is a one-way hash function.

4.4 Randomization Function

We use a *randomization function* to transform an attribute of location information into a random value. The randomization function should be a trap door one-way function because authorized service providers or users should be able to obtain the original values from outputs of the randomization function. We assume that a symmetric key encryption scheme $f(k, A)$ is used for the randomization function, where the data A is encrypted using secret key k. A user prepares at most l keys for encrypting each tuple of location information. The j-th tuple is encrypted by a secret key $k_{i_t}^j$. Thus, only the parties that have received $k_{i_t}^j$ from the user can obtain the original values. The secret keys for encrypting tuples are also updated as $k_{i_t}^j \leftarrow H(k_{i_{t-1}}^j | r_i^j)$, where $k_{i_0}^j$ is the original secret key.

4.5 Queries

The procedures for each type of query, which it is assumed are conducted over secure channels, are described as follows:
Public Query. A service provider sends a public query using the refer function to the location database, and the location database forwards the results to the service provider.
Private Query. The principle behind private queries is that when a service provider requires j-level location information $(A_{i_t}^1, \ldots, A_{i_t}^j, -, \ldots, -)$ for user i in order to provide their service, the user should only inform the service provider of the anonymized identifier $I_{i_t}^j$. A private query proceeds as follows:

1. The user requests that a service provider provides a location-based service by providing the service provider with the appropriate anonymized identifier.
2. The service provider sends a request to the location database using the search function.
3. The location database forwards the record of the identifier. (In the event that the search fails, the service provider requests that the user updates their location information.)
4. The service provider checks whether the record includes the location information required for the service. If some of the attributes in the record are encrypted (randomized) and these attributes are needed for the service, the service provider requests that the user sends the corresponding secret keys. If the user accepts the request, the user sends the secret keys to the service provider, who then decrypts the attributes.
5. The service provider provides the location-based service to the user.

Personal Query. A personal query is essentially an l-level location information request. It can thus be conducted in exactly the same way as a private query, except that any user (service provider) conducting a personal query will need to be supplied by the target user with $I_{i_t}^l$ and, if necessary, the information required to decrypt any randomized location database entry fields.

5 Scheme Properties

In this section, we briefly summarize the main properties of the proposed scheme.

- **Distributed Workload.** Most of the effort required to operate the scheme is conducted by a distributed network of location anonymizers, who take on the main effort required to prepare location database entries and conduct k-anonymity checks of the location database. By distributing this effort the scheme is more scalable than centralized approaches.
- **Anonymized Location Database Entries.** The location database satisfies k-anonymity of database entries at all times. If a new location database entry does not satisfy this property then it is randomized before being registered.
- **Anonymized Location Database Indexing.** Entries in the location database use anonymized identifiers, which means that it is impossible for an attacker to determine whether two database entries relate to the same user. On the other hand, these identifiers are efficiently computed by repeated applications of a hash function for both users and location anonymizers, who can also synchronize their computation.
- **User Control of Location Information.** Users have full control over the granularity of location information released to service providers. They execute that control by providing service providers with the appropriate anonymized identifier that references the relevant entry in the location database.
- **Support for Granular Queries.** The scheme supports a variety of types of location database query. By deploying trapdoor one-way functions to mask location database entries that do not satisfy k-anonymity, any service providers who require database entries to be unmasked can do so by being sent the trapdoor information directly by the user.

6 Conclusion

We have proposed a privacy-aware location database service that supports different types of location-based service. The architecture is scalable and employs simple functions that are similar to those found in general database systems.

Although the architecture distributes the main computational burden amongst location anonymizers, the scheme still relies on a centralized location database. One possible extension of this architecture would be to consider whether the location database could be structured to permit a degree of parallelization of the location database operations.

Another issue that could be investigated in more detail is the maintenance of k-anonymity of the location database over time. While location database entries are normally short-lived, it may still be desirable to update randomized database entries in the event that k-anonymity becomes satisfiable or unsatisfiable, having originally not been. The advantage of upgrading to full entries is that it saves the computational costs associated with unmasking a randomized database entry in the event that the location described in the entry is required by a service provider.

References

1. B. Bamba, L. Liu, P. Pesti, and T. Wang. Supporting anonymous location queries in mobile environments with privacygrid. In *Proc. of 17th International World Wide Web Conference (WWW 2008)*, pages 237–246, 2008.
2. M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In *Proc. of CRYPTO 2007, LNCS*, volume 4622, pages 535–552, 2007.
3. A. R. Beresford and F. Stajano. Location privacy in pervasive computing. In *IEEE Pervasive Computing*, volume 2 (1), pages 46–55, 2003.
4. D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *Proc. of TCC 2007, LNCS*, volume 4392, pages 535–554, 2007.
5. William Gasarch. A survey on private information retrieval. *Bulletin of the EATCS*, 82:72–107, 2004.
6. M. Gedik and L. Liu. A customizable k-anonymity model for protecting location privacy. In *Proc. of the 25th International Conference on Distributed Computing Systems (ICDCS 2005)*, pages 620–629, 2005.
7. G. Ghinita, P. Kalnis, and S. Skiadopoulos. PRIVÉ: Anonymous location-based queries in distributed mobile systems. In *Proc. of 16th International World Wide Web Conference (WWW 2007)*, pages 371–380, 2007.
8. M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. of the 1st International Conference on Mobile Systems, Applications, and Services (MobiSys 2003)*, pages 163–168, 2003.
9. J. I. Hong and J. A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *Proc. of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys 2004)*, pages 177–189, 2004.
10. H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. In *Proc. of IEEE International Conference on Pervasive Services 2005 (ICPS 2005)*, pages 88–97, 2005.
11. L. Lamport. Password authentication with insecure communication. In *Cummunications of the ACM 24.11*, pages 770–772, 1981.
12. S. Mascetti and C. Bettini. A comparison of spatial generalization algorithms for lbs privacy preservation. In *Proc. of the 1st International Workshop on Privacy-Aware Location-Based Mobile Services (PALMS 2007)*, pages 258–262, 2007.
13. M. F. Mokbel. Towards privacy-aware location-based database servers. In *Proc. of the 22nd Internationl Conference on Sata Engineering Workshops (ICDEW 2006)*, pages 93–102, 2006.

14. M. F. Mokbel, C. Y. Chow, and W. G. Aref. The new casper: Query processing for location services without compromising privacy. In *Proc. of the 32nd International Conference on Very Large Data Bases (VLDB 2006)*, pages 763–774, 2006.
15. P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information. In *Proc. of the 17th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (PODS'98)*, page 188, 1998.
16. Dawn Xiaodong Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *Proc. of IEEE Symposium on Security and Privacy 2000*, pages 44–55, 2000.