Preventing Denial-of-request Inference Attacks in Location-sharing Services

Kazuhiro Minami[†1]

Location-sharing services, such as Google Latitute, have been popular recently. However, location information is sensitive and access to it must be controlled carefully. Although we can protect private location information with access-control policies, a denial of request itself implies a target user's visiting a private place. In this paper, we formally define this new inference problem and discuss possible solutions.

1. Introduction

Soon the vast majority of mobile devices will be equipped with some form of localization capability; already, most smart phones include a GPS receiver. This has led to the rise of location-based services on a number of mobile platforms, including Symbian, iPhone, and Android. Novel applications, such as Google Latitude[4], have opened up the possibili- ties of sharing location information with other users[3],[8],[10]. This theme has been picked up in social networks as well; e.g., Twitter recently announced support for embedding a location in each post[14].

Location sharing raises significant privacy concerns[1], since a location, such as a bar or a hospital, can be used to infer a users personal activities. Therefore, location- sharing services (LSSs) have introduced an access control mechanism that allows the user to specify what location data may be shared with whom. For example, Google Latitude allows a user to authorize others access to his or her location; it also allows a user to enter a decoy location manually. Glympse[3] specifies a time duration during which location information is shared. These interfaces provide coarse-grained controls. Researchers in pervasive computing have proposed more fine-grained access control schemes[6],[7],[9],[12],[13] that make use of context information such as loca- tion, time of day, and so on. These rules both better represent the users actual sharing desires and at least partially automate the decisions to provide seamless integration of location sharing into peoples daily lives.

One additional danger of sharing location information, however, is that it can lead to inference of previous or past locations. For example, a person traveling along a trajectory is likely to remain along that path. Things get significantly more complex as more background data is introduced. For example, walking and driving paths follow a predictable pattern, following streets and sidewalks; furthermore, each person exhibits more specific patterns in their activities. For example, Figure 3 shows two potential walking paths leading to a hospital and a library. Given background knowledge, it is possible to infer that a user traveling towards the intersection (black circles) is likely to visit one of these two places. A user who turns left at the intersection (white circles) may then be assumed to be going to the hospital. Therefore, if the user wishes to hide visits to the hospital, it is important to stop revealing his or her location earlier as well.

We previously propose to develop a new access-control scheme that prevents such inference attacks[11]. Our basic approach is to model an adversary as a location predictor that predicts future movements of a target user from his previous movements with certain probabilities. Intuitively, our access control scheme discloses a user's location information only if an unauthorized user cannot predict that the user moves to some private location with a sufficiently high probability. Our approach is the most conservative in the sense that we assume that an adversary knows all the previous movements of the target user.

However, our previous scheme does not consider indirect information disclosure through the *denial* of a service request. That is, if a request for a target user's location is denied, it is possible for the requester to infer that the target user is visiting the private location with a high probability. For example, suppose that Alice is leaving her office and is visiting a bookstore on the Main street next and that once Alice arrives at the bookstore, she will be very *likely* to next visit the hospital, which is her private location. Since disclosing the fact that Alice is at the bookstore allows Bob, the tracking user, to infer that Alice will visit the hospital

[†1] National Institute of Informatics

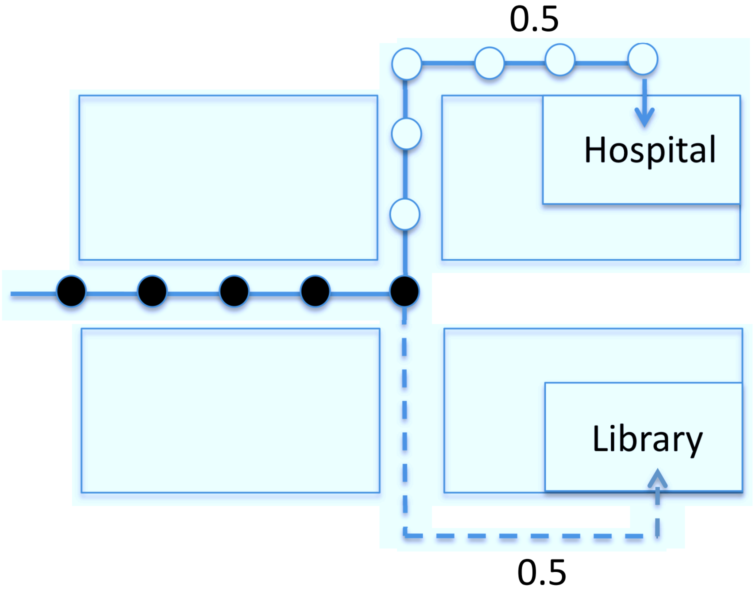

Fig. 1 Example safe disclosure of location information. The solid line represents an actual path of a user visiting a hospital. We assume that the hospital is a private place and the library is a public place. A safe LSS would disclose location points denoted by black nodes. We assume that the user has 50% chance of visiting of the library when he is at the intersection in the middle.

next, our access-control scheme does not disclose Alice's current location (i.e., the bookstore) to Bob. However, if Bob does not receive Alice's current location, he infers that Alice is getting close to her private location. If the bookstore is the only location near Alice's office, from where Alice visits her private location with a high probability, Bob can conclude that Alice is indeed visiting the hospital from the fact that he does not receive Alice's current location.

In this paper, we, therefore, consider such denial-of-request inference attacks on LSSs and propose a new access-control algorithm for preventing this new class of attacks. Our algorithm ensures that whenever a LSS does not disclose the location of a target user, there are sufficient uncertainty about the private location the user is visiting; that is, there are always multiple candidates of the private location the user will visit. Since the proposed algorithm possibly decides many of the location movements not to be disclosed and thus the utility of the system is significantly degraded, we discuss possible solutions that remedy such undesirable situations.

The rest of the paper is organized as follows. Section 2 introduces the system model and the location privacy metrics of LSSs in our previous research. Section 3 describes an inference on a denial of a request quantitatively and presents an algorithm for preventing such inferences. Section 4 possible solutions to maximize the amount of location information to be disclosed while preserving our location privacy metrics. We cover related work in Section 5 and finally states our concluding remarks and future plans in Section 6.

2. Background

In this section, we summarize our system model for LSSs and the metrics for location privacy in our previous research[11)].

2.1 System model

Figure 2 shows our system model for LSSs. We assume that user p_j is interested in receiving a target user p_i's location movements. User p_i carrying a GPS-enabled mobile device periodically sends LSS a series of location-timestamp pairs (loc_k, t_k) for $k \in \mathcal{N}$; LSS receives a set of all pairs

$$L = \{(loc_k, t_k) \mid k \in \mathcal{N}\}.$$

User p_i also defines its access-control policies in LSS so that LSS can protect p_i's location movements properly. We represent p_i's access-control policies with the function

$$acl : \mathcal{P} \times \mathcal{W} \rightarrow 2^{\mathcal{P}}$$

where $\mathcal{P}$ is a set of all users and $\mathcal{W}$ is a finite set of all locations. The function acl takes a user identity p_i and a location name l_k as inputs and outputs a set of users who are authorized to learn that "p_i is at location l_k." In other words, LSS releases p_i's location movement (l_k, t_k) to principal p_j only if p_j belongs to set $acl(p_i, l_k)$, and thus user p_j receives a subset of events $L' \subseteq L$

$$L' = \{(loc_k, t_k) \mid p_j \in acl(p_i, l_k)\}.$$

Notice that we only consider the case that p_i's access-control policies depend on p_i's location l_k to simplify our discussion in this paper, but we can easily support the general case where access-control policies also considers a timestamp t_k.

We next define which locations are *private* to user p_i formally.

Definition 1 (Private location.) We consider that a user p_i's location l is private with respect to another user p_j if:

$$l \in \{l' \mid p_j \notin acl(p_i, l')\}.$$

We consider that a LSS preserves a user p_i's privacy if p_j cannot infer that p_i was at some private location l from the information p_j receives from LSS. We formalize this concept below.

Definition 2 (Preservation of location privacy.) We say that a LSS preserves a user p_i's location privacy against another user p_j if p_j cannot infer p_i's movement (l, t) where l is p_i's private location from a set of location-timestamp pairs L'.

In next section, we describe how an unauthorized user p_j performs inference with a location predictor based on the Markov model.

2.2 Metrics of location privacy

We consider a Markov chain with a sequence of random variables

$$X_1, X_2, X_3, \ldots$$

where each X_i has a value drawn from the finite set of locations $\mathcal{W}$. We here assume that location l_k is published periodically, and we thus omit timestamp t_k in tuple (l_k, t_k). We also assume that the Markov chain is time-homogeneous. So, if we consider a Markov chain of order 1,

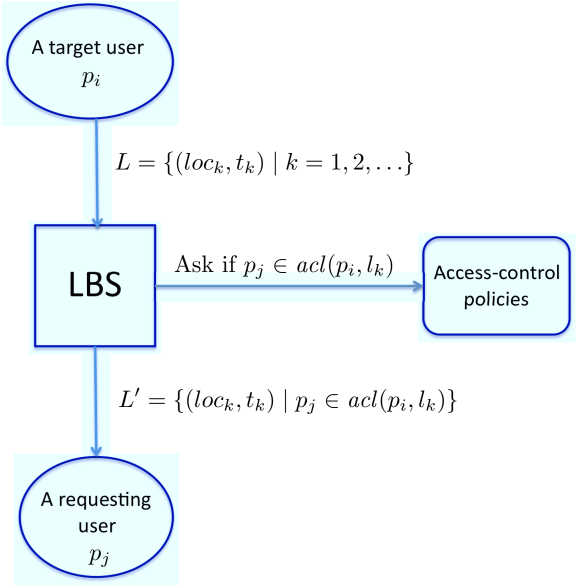
A target user p_i
$L = \{(loc_k, t_k) \mid k = 1, 2, \ldots\}$
LBS
Ask if $p_j \in acl(p_i, l_k)$
Access-control policies
$L' = \{(loc_k, t_k) \mid p_j \in acl(p_i, l_k)\}$
A requesting user p_j

Fig. 2 System model.

$$Pr(X_{n+1} = l_i | X_n = l_j) = Pr(X_n = l_i | X_{n-1} = l_j).$$

We maintain the probability of moving from location l_i to l_j in (i, j)th element of a state transition matrix $M_{i,j}$ as follows:

$$Pr(X_{n+1} = l_i | X_n = l_j) = M_{i,j}.$$

for every pair of l_i and l_j in set $\mathcal{W}$. The probability of moving from location l_i to l_j in n time steps can be computable by multiplying the transition matrix M n times as follows:

$$Pr(X_n = l_i | X_0 = l_j) = M^{(n)}_{i,j}.$$

Since it is likely that we can improve the accuracy of location predictions by considering multiple previous movements, we also consider a location predictor based on a Markov model of a higher order. If we use a Markov model of 2 order, a state transition matrix M must maintain the probability $Pr(X_{n+1} = l_i | X_n = l_j, X_{n-1} = l_k)$ in $((j,k),i)$th element of M; that is,

$$Pr(X_{n+1} = l_i | X_n = l_j, X_{n-1} = l_k) = M_{(j,k),i}.$$

We make the most conservative assumption that an adversary can observe all the previous movements of a target user and compute a state transition matrix M of an arbitrary order n before predicting the target user's next movement. We now define the preservation of location privacy against an adversary with a state transition matrix M of the 1-order Markov model as follows:

Definition 3 (Preservation of (M,t)-location privacy.) Suppose that a user p_i' current location is l_i and that t is a probability threshold where $0 \leq t \leq 1$. We say that a LSS preserves a user p_i's (M,p)-location privacy against another user p_j if, for every private location $l_k \in \mathcal{W}$ with respect to p_j, the following condition holds

$$M_{i,k}^{(n)} \leq t \text{ for } n = 1, 2, \ldots.$$

Intuitively speaking, the above definition requires that an unauthorized user p_j cannot predict that the target user p_i is at some private location l_k in some future time with probability p, which is greater than the threshold value t. Although the above definition only covers the case with the 1-order Markov model, we can easily generalize the definition to consider a Markov model of order n.

3. Prevention of denial-of-request inference attacks

In this section, we introduce denial-of-request inference attacks and present a new access-control scheme for preventing this class of attacks.

3.1 Denial-of-request inference attacks

We first informally describe a denial-of-request attack. Suppose that a target user who is currently located at l_i moves to location l_j next, the probability of moving from location l_j to l_k, $M_{j,k} = 1.0$, and that l_k is a private location of the target user. Then, our access-control scheme that preserves (M,t)-privacy where a threshold probability $t < 1.0$ in Section 3 does not disclose the user's location at l_j. However, it is possible to infer that the probability of the user's visiting location l_k is equal to the probability of moving from l_i to l_j, $M_{i,j}$, from the fact that the location after l_i is not disclosed.

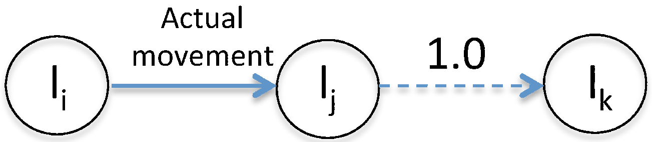

Fig. 3 Example of a denial of request. When a user moves from location l_i to l_j, it is certain that the user is moving to location l_k next. Therefore, the system does not desclose the fact that the user is at location l_j.

We next quantify this information leakage through a denial of request. We first introduce a few notations below.

Definition 4 ($M^*_{i,j}$) We denote by M^* the maximum among the probability of moving location l_i to l_j at steps $k = 1, \ldots, \infty$; that is,

$$M^*_{i,j} = \max_{k=1}^{\infty}(M^{(k)}_{i,j}).$$

Definition 5 (Function *neighbor*) Given a state transition matrix M and a location l_i, we denote by $neighbor(l_i)$ a set of locations the user possibly moves at a single step; that is,

$$neighbor(l_i) = \{l_j \mid M_{i,j} > 0\}.$$

We define the boolean function *releasable* that determines whether a system preserving (M,t)-location privacy grants a request for a given location l_j.

Definition 6 (Function *releasable*) The function *releasable* that takes a location l_j as an input returns *true* if the following statement holds.

$$\forall l_k \in (L \setminus L') : M^*_{j,k} < t.$$

If a system preserving (M,t)-location privacy releases the information that a target user is location l_i and then does not release her next movement, the information inferred from that denial of request is formulated as follows:

$$\exists l_j \in neighbor(l_i),\ \exists l_k : \neg releasable(l_k).$$

That is, an adversary can learn the above information through a denial-of-request inference.

3.2 Access-control condition for denial-of-request inferences

The conditional probability of moving to a private location l_k given a denial of request for location l_j after receiving the fact that the target user at location l_i is calculated as follows:

$$Pr[l_k \mid RELEASE(l_i), DENY] = (M_{i,j} \times M^*_{j,k}) / \sum_{\neg releasable(l_m)} M_{i,m}.$$

We denote by $RELEASE(l_i)$ and $DENY$ events of the system's releasing location l_i and denying the request respectively. Therefore, a system preserving (M,t)-location privacy must consider the following additional condition to prevent denial-of-request inferences if the system release the current location l_i of the target user.

$$\forall l_k \in L \setminus L' : Pr[l_k \mid RELEASE(l_i), DENY] < t.$$

4. Discussion

The access-control condition in Section 3.2 implies that if a system denies a request for a user's current location, there must exist multiple candidate locations that are not releasable in order to have sufficient uncertainty about the user's visiting a private location. Therefore, if a target user only defines a small number of private locations, there is a danger that the system ends up hiding most of the user's movements. To avoid such significant loss on the utility of the service, it is necessary to add some *artificial* private locations in some way. We plan to investigate such methods without distracting the utility of the service significantly.

5. Related work

Several researchers[6),7),9),12),13)] propose rule-based access-control schemes for protecting user location in pervasive environments. Hengartner[6)] supports access-control policies considering the granularity of location information and time intervals. Myles[12)] provides a XML-based authorization language for defining privacy policies that protect users location information. Users must trust a set of validators that collect context information and make authorization decisions. Those schemes allows a user to define fine-grained access-control policies. Apu[9)] provides users with an intuitive way of defining access control policies, which represent physical boundaries surrounding the users. However, no previous scheme considers the issue of inference based on the mobility patterns of users.

Location privacy has been studied heavily in the context of location data anonymization[2),5)]. The focus of research in this sequence is to ensure that no anonymized data is associated with an individual. For example, Gruteser[5)] proposes a scheme that changes the granularity of location information to ensure that each location contains at least k users (i.e., k-anonymity). However, the problem addressed in this paper is different since we consider inference on location data associated with a known individual.

6. Summary

In this paper, we address a new inference problem concerning a denial of service request in location-sharing services (LSSs). We precisely quantify information leakage through a denial of request and establish an access-control condition to prevent such inference attacks. We plan to conduct experiments involving actual mobile users and evaluate a trade-off between the utility and security of the system and to develop mechanisms for balancing the trade-off adequately.

Acknowledgments

This work was supported by a grant from the Promotion program for Reducing global Environmental loaD through ICT innovation (PREDICT) of the Ministry of Internal Affairs and Communications and by Grants-in-Aid for Scientific Research C, 11013869, of Japan Society for the Promotion of Science.

References

1) Anthony, D., Henderson, T. and Kotz, D.: Privacy in Location-Aware Computing Environments, *IEEE Pervasive Computing*, Vol.6, No.4, pp.64–72 (2007).

2) Beresford, A.R. and Stajano, F.: Location Privacy in Pervasive Computing, Vol.2, No.1, pp.46–55 (2003).

3) : Glympse, http://www.glympse.com.

4) : Google latitude, http://www.google.com/latitude.

5) Gruteser, M. and Grunwald, D.: Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking, *Proceedings of Mobisys 2003: The First International Conference on Mobile Systems, Applications, and Services*, San Francisco, CA, USENIX Associations (2003).

6) Hengartner, U. and Steenkiste, P.: Access control to people location information, *ACM Transactions on Information and System Security (TISSEC)*, Vol.8, No.4, pp.424–456 (2005).

7) Hong, J.I. and Landay, J.A.: An architecture for privacy-sensitive ubiquitous computing, *Proceedings of the 2nd international conference on Mobile systems, applications, and services (MobiSys)*, New York, NY, USA, ACM, pp.177–189 (2004).

8) : InstaMapper, http://www.instamapper.com.

9) Kapadia, A., Henderson, T., Fielding, J.J. and Kotz, D.: Virtual Walls: Protecting Digital Privacy in Pervasive Environments, *Proceedings of the Fifth International Conference on Pervasive Computing (Pervasive)*, LNCS, Vol.4480, Springer-Verlag, pp.162–179 (2007).

10) : Loopt, http://www.loopt.com.

11) Minami, K. and Borisov, N.: Protecting location privacy against inference attacks, *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, New York, NY, USA, ACM, pp.711–713 (2010).

12) Myles, G., Friday, A. and Davies, N.: Preserving Privacy in Environments with Location-Based Applications, *IEEE Pervasive Computing*, Vol.2, No.1, pp.56–64 (2003).

13) Sacramento, V., Endler, M. and de Souza, C.: A privacy service for location-based collaboration among mobile users, *Journal of the Brazilian Computer Society*, Vol.14, No.4, pp.41–57 (2008).

14) : How to Tweet With Your Location, http://twitter.zendesk.com/entries/122236.