Location Privacy Policy Management System

Arej Muhammed[1], Dan Lin[1], and Anna Squicciarini[2]

[1] Department of Computer Science
Missouri University of Science & Technology
{aamcq9,lindan}@mst.edu
[2] Information Sciences & Technology
Pennsylvania State University
asquicciarini@ist.psu.edu

Abstract. As location-based services become more and more popular, concerns are growing about the misuse of location information by malicious parties. In order to preserve location privacy, many efforts have been devoted to preventing service providers from determining users' exact locations. Few works have sought to help users manage their privacy preferences; however management of privacy is an important issue in real applications. This work developed an easy-to-use location privacy management system including functions of policy composition, policy conflict detection and policy recommendation.

Keywords: Location privacy, policy management.

1 Introduction

With the advance of mobile devices and positioning systems, location-based services (LBSs) have become prevalent. While enjoying the convenience brought by LBSs, consumers have begun worrying about their location privacy due to the very nature of LBSs which typically require the disclosure of the users' locations. Undesired exposure of location information may render users an easy target of criminal behaviors. For example, kidnappers could take advantage of LBSs to acquire a target's daily travel route.

Many efforts [7,8,10] have been devoted to preventing service providers from knowing users' exact locations. However, few works [13–15] have sought to help users manage their privacy preferences, which is yet an important issue in real applications and at the core of the success of these applications. Several exploratory studies [3, 5] have shown that most users are concerned about their location privacy, but when they are actually facing the location-based services, they either give up their privacy concerns or totally abandon the services. The main reasons behind such behavior are summarized as follows: (i) lack of understanding about the privacy implications of their behavior; (ii) lack of a proper method for them to control privacy options; (iii) overhead introduced by privacy protections. For example, existing access control policies like XACML [16] aim to cover a wide range of needs of access control for various applications, which are too complicated to be manipulated by non-expert end-users and contain functions that may not be necessary in location-based services. The complexity of general access control policies is also the main cause of the management overhead that has been shown to hinder the adoption of location privacy protection mechanism by the end users.

T.W. Chim and T.H. Yuen (Eds.): ICICS 2012, LNCS 7618, pp. 427–434, 2012.

To cope with the above issues, in this work, we present an overview of an easy-to-use location privacy policy management system. We define a succinct yet expressive policy language tailored for location privacy protection. We propose algorithms for detecting policy redundancy, policy conflict and policy merging that ensure the consistency of the access right being granted as well as efficient policy evaluation. We develop a policy recommendation function that generates recommended policies based on users' basic requirements in order to reduce user's burden.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed policy management system. Finally, Section 4 concludes the paper.

2 Related Work

There have been extensive efforts on location anonymization in order to prevent service providers from knowing end-users' exact locations [1, 2, 6, 7, 9–11]. There are few works on location privacy policy management. Snekkeness [15] is one of the earliest researchers to identify the concepts for formulating personal privacy policies. Smailagic et al. [14] proposed a privacy model which specifies location privacy using set theory and rules. Myles et al. [12] developed a middleware service to allow location-based applications to use multiple location positioning systems. A recent related work is by Sadeh et al. [13] who developed an application, namely PeopleFinder, to enable cell phone and laptop users to selectively share their locations with others. Unlike existing works, our proposed system considers more policy management related tasks such as policy composition assistance, policy redundancy and conflict detection.

In addition, it is worth noting that location privacy policies are relevant but different from the concept of location-based access control (e.g., [4]) in the sense that location data plays different roles.

3 Location Privacy Policy Management System

The Location Privacy Policy Management System (LPPM) system is installed at user side, such as users' smart phones. We assume that users subscribe to location-based service providers who are allowed to know each user's location. We also assume that users may have created groups of contacts (e.g., family, friends) for their use of installed location-based services. The group information will be leveraged by the LPPM system to simplify the specification of the location privacy policy.

Figure 1 illustrates the framework of the LPPM system. When a user adds a new contact to his/her installed location-based service application, the LPPM system takes the profile of the new contact (such as his/her relationship with the user, hobbies, etc.) and invokes the recommendation module to generate a candidate privacy policy for the user's consideration. If the user is satisfied with the recommended policy, the policy will be inserted into the policy repository and may be merged with other policies for storage efficiency as well as evaluation efficiency. If the user modifies the recommended policy, the revised policy will be checked by the policy conflict detection module before inserting to the policy repository.

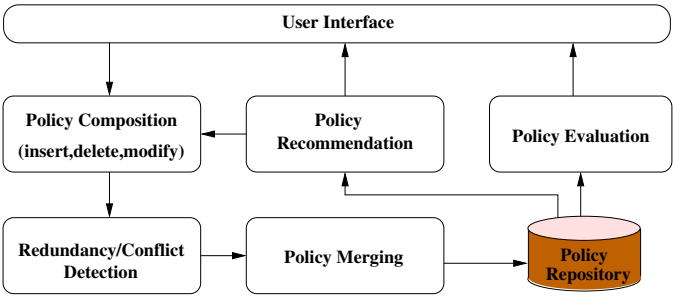

Fig. 1. Overview of the LPPM System

When someone (say Jack) sends a request to a location-based service provider to ask for his friend (Bob)'s location, the service provider will direct the request to Bob's mobile device. Bob's LPPM system checks the stored location privacy policies to see if Jack is allowed to view Bob's location. The decision is then forwarded back to the service provider. If Jack is granted the access right, the service provider will display Bob's current location on Jack's device. Otherwise, Jack will receive a message that his request is denied.

3.1 Location Privacy Policy

We define a policy language (as shown in Definition 1) that is able to specify the common components and requirements that are related to location privacy concerns.

Definition 1. *A location privacy policy P consists of the following components:*

- U, E *specify the policy target which are defined by a set of user IDs and roles.* U/E *excludes users in* E *from* U.
- T *is a Boolean expression on the time* t*, the day* d_1 *and the date* d_2*, which is the time when a location request is received.*
- L *specifies a set of policy owner's locations which are defined by either ranges of location coordinates, or semantic locations.*
- G *specifies the granularity of the location disclosure in a five scale system: exact location, district, city, state, country.*

$P\langle U/E, T, L, G\rangle$ *specifies that users in* U *but not* E *are allowed to view the policy owner's location at granularity* G *if the policy owner is within the region* L *during the time period defined by* T.

A user can define one location privacy policy for an individual user or a group of users. The access to the policy owner's location will be granted only when the policy is satisfied. Otherwise, the location request will be denied. The policy evaluation consists of the following four steps:

1. User u_1 wants to query user u_2's location and u_1 composes a location request in the form of $Q : \langle RID = u_1, QID = u_2\rangle$, where RID is the requester's ID and QID is the user being queried.

2. User u_1 sends the location request Q to the location-based service which has installed the location privacy policy management (LPPM) system.
3. The LPPM system searches u_2's policies that are applicable to u_1. Policies are considered applicable to u_1 if u_1 satisfies the policy target in that u_1 is listed in the policy target or u_1's relationship (role) to u_2 is specified in the policy. For example, the policy target $U = \{Alice, Bob\}$ and $u_1 = Alice$, or the policy target $U = \{Family\}$ and u_1 is one of u_2's family members.
4. For each applicable policy, the LPPM system first checks if the current time is within the time period specified by the time range T. If so, the LPPM system further checks if u_2's current location is within the location range L specified in the policy. To check the location, the LPPM system will convert the semantic locations (such as a name of a company) defined in the policy into location coordinates to be compared to u_2's current location coordinates. If the location check is also satisfied, the access to u_2's location will be granted to u_1, and the policy evaluation stops. Otherwise, the LPPM system continues to evaluate the remaining applicable policies. If none of the applicable policies are satisfied, u_1's request to view u_2's location will be denied, i.e., u_1 will not be able to know u_2's current location.

For example, suppose that Bob allows his colleagues to know his exact location only when he is in the company during work hours from 8am to 5pm on weekdays. To achieve this, Bob can use the following policy: $P_1\langle\{Colleague\}, (8am < t < 5pm)$ AND $(d_1 = \{Mon, .., Fri\})$, $companyLoc$, $exactLoc\rangle$. Given P_1, if one of Bob's colleagues, Jack, is looking for Bob for a meeting at 10am and Bob is in the company at that time, Jack will be able to view Bob's location according to the policy. If Jack wants to know where is Bob at 12pm while Bob is at lunch outside the company, Jack will not be able to see Bob's location in this case.

As another example, assume that Bob usually allows his family members to know his locations according to the following policy: $P_2\langle\{Family\}, Anytime, Anywhere, exactLoc\rangle$. One day, Bob needs to shop for a gift for one of his family members, say Alice. In order to surprise her, Bob may want to block Alice from knowing his locations by temporarily changing the policy P_2 to $P_2'\langle\{Family\}/\{\mathbf{Alice}\}, Anytime, Anywhere\rangle$. P_2' excludes Alice from the policy target and hence Alice request to viewing Bob's location will be denied.

3.2 Policy Maintenance

For a given new policy, it is important to check if the access right granted by the new policy has already been included in some existing policies. If so, it is unnecessary to insert the new -redundant- policy. For example, suppose that Alice is Bob's family member. A new policy says that Alice is allowed to view Bob's location anytime on Saturday: $P_4\langle\{Alice\}, d1 = \{Saturday\}$, Anywhere, exatLoc$\rangle$; while there is an existing policy which says that family members are allowed to view Bob's location anytime during weekend: $P_3\langle\{Family\}, d1 = \{Weekend\}$, Anywhere, exatLoc$\rangle$. It is obvious that P_4 is covered by the existing policy P_3 and does not need to be inserted to the system. Policy redundancy is formalized as follows.

Definition 2. *(Policy Redundancy) Let $P_i\langle U_i/E_i,\ T_i,\ L_i,\ G_i\rangle$ be a new policy composed by user u, and S_p be a set of existing policies belonging to the user u. P_i is redundant if there exists a policy $P_j\langle U_j/E_j,\ T_j,\ L_j\rangle \in S_p$, and $\{U_j/E_j\} \supseteq \{U_i/E_i\}$ and $T_j \supseteq T_i$, $L_j \supseteq L_i$ and $G_j = G_i$.*

Based on Definition 2, we can see that P_4's target, time constraint and location range are all subsets of that of the existing policy P_3, and they are specifying at the same location disclosure granularity, hence, P_4 is redundant.

If the above policy P_4 is slightly modified to P_4' which specifies a different location disclosure granularity: $P_4'\langle\{Alice\},\ d1\ =\ \{Saturday\}$, Anywhere, **city**$\rangle$, P_4' is not considered redundant but conflict with P_3. This is because P_4' does not allow Alice to see Bob's exact location but only the city of the location, while P_3 allows family members including Alice to see Bob's exact locations. In a nutshell, the conflict may occur when the new policy grants access to a user which is denied by an existing policy, or vice versa. Its formal definition is the following.

Definition 3. *(Policy Conflict) Let $P_i\langle U_i/E_i, T_i, L_i, G_i\rangle$ be a new policy composed by user u, and $P_j\langle U_j/E_j, T_j, L_j, G_j\rangle$ be an existing policy belonging to the user u. P_i conflicts with P_j if one of the following conditions is satisfied:*

- *$E_j \bigcap U_i \neq \emptyset$ and $T_j \bigcap T_i \neq \emptyset$ and $L_j \bigcap L_i \neq \emptyset$;*
- *$E_i \bigcap U_j \neq \emptyset$ and $T_j \bigcap T_i \neq \emptyset$ and $L_j \bigcap L_i \neq \emptyset$.*

After passing the policy redundancy and conflict check, a new policy will be considered whether it can be merged with existing policies. Merging related policies not only helps enhance the presentation of policies to users but also improves the efficiency of policy management and evaluation since fewer policies need to be checked given a location request.

Before the formal definition, let us first exam an example when two policies can be merged. Policy P_7 states that Jack is allowed to view Bob's location on Monday when Bob is at Chicago: $P_7\langle\{Jack\}$, d1=Monday, Chicago, exactLoc$\rangle$, and another policy specifies that Alice is allowed to view Bob's location when Bob is at Chicago: $P_8\langle\{Alice\}$, d1=Monday, Chicago, exactLoc$\rangle$. P_7 and P_8 has the same location, time constraints and location disclosure granularity, but only differ in the policy targets. P_7 and P_8 can then be merged into one policy $P_m\langle\{Jack, Alice\}$, d1=Monday, Chicago, exactLoc$\rangle$. In general, two policies can be merged if they are specified at the same location disclosure granularity and have only one different component. The following definition summarizes the scenarios when two policies can be merged.

Definition 4. *Two policies $P_i\langle U_i/E_i,\ T_i,\ L_i,\ G_i\rangle$ and $P_j\langle U_j/E_j,\ T_j,\ L_j,\ G_j\rangle$ can be merged if they satisfy one of the following conditions:*

- *Two policies have the same policy targets, time constraints, i.e., $U_i/E_i = U_i/E_i$, $T_i = T_j$, and $G_i = G_j$.*
- *Two policies have the same policy targets and location constraints, i.e., $U_i/E_i = U_i/E_i$, $L_i = L_j$ and $G_i = G_j$.*
- *Two policies have the same time constraints and location constraints, i.e., $T_i = T_j$, $L_i = L_j$, and $G_i = G_j$.*

The result of the policy merge will be: $P_m\langle(U_i \bigcup U_j)/(E_i \bigcup E_j),\ (T_i \bigcup T_j),\ (L_i \bigcup L_j),\ G_i\rangle$.

3.3 Policy Recommendation

The policy recommendation is based on the analysis of the privacy level of the existing policies. The privacy level dictates visibility of a user's location on a level hierarchy. The less the visibility of the user's location, the higher the privacy level is. In order to quantify the visibility level, we consider the following parameters:

- N_u: denotes the total number of contacts of the policy owner.
- N_p: denotes the number of contacts specified in the policy target.
- D_t, D_{d_1}, D_{d_2}: denote the range of the time constraint in the policy.
- $Space$: denotes the total area that covers the policy owner's recorded locations.
- G_d: maps the location disclosure granularity to numbers to quantify their visibility level: exactLoc, district, city, state, country are represented as number 1, 2, 3, 4, 5 respectively.

By comparing each policy component with its corresponding domain (i.e., all possible values that the policy component may have), we define the privacy level PL as follows:

Definition 5. *The privacy level (PL_p) of P is defined as the weighted sum of the ratio between each component value and its domain, where w_u, and w_t and w_l are the weights.*

$$PL = w_u \frac{N_u - N_p}{N_u} + w_t(1 - \frac{D_t}{24} \cdot \frac{D_{d_1}}{7} \frac{D_{d_2}}{12}) + w_l(1 - \frac{D_L}{Space} \frac{1}{G_d})$$

The privacy level PL consists of three parts. The first part is the total number of users in the policy compared to the total number of contacts of the policy owner. If the policy owner allows more users to view his/her locations, that means the privacy owner has lower level of privacy concerns, and hence the value of $\frac{N_u - N_p}{N_u}$ will be smaller. The second part of PL considers the time constraints in terms of hours, days and date. The longer the time that the policy owner's locations are disclosed, the lower the privacy level will be. The last part of the PL integrates the effect of both the range of the space and the disclosure granularity. The larger the range of the locations and the finer the granularity, the lower the privacy level will be. Finally, the weight values are used for the need to adjust the impact of each component if any prior knowledge is available. By default, the weight values are equal for all components.

We now proceed to introduce the process of policy recommendation which includes three phases: (1) a preparation phase, (2) a policy generation phase and (3) the finalization phase.

Phase 1: The preparation phase aims to build the knowledge base. The LPPM system needs to have a few policies input by the users to be used as the base of the recommendation. For the first few policies, the LPPM system groups them based on the relationship between the policy target and the policy owner. In other words, policies regarding the same role of users will be placed in the same group. For example, if Alice and Jack are Bob's family members, the policies regarding Alice, Jack, and family members, will be in the same group. The reason of such grouping is that individuals usually maintain

different rules for different types of contacts. For instance, the privacy policies for family members may usually allow the disclosure of the exact locations while the privacy policies for colleagues may just allow the disclosure of locations at city level. Next, in each group, the policies are further classified into three categories: low, medium, and high, according to their privacy levels. In particular, let $max(PL)$ denote the maximum PL of all existing policies of a user. If a policy's privacy level is lower than $\frac{max(PL)}{3}$, the policy is considered to have low privacy protection level. If a policy's privacy level is greater than $\frac{2 \cdot max(PL)}{3}$, the policy is assigned a high privacy protection level. The remaining policies are at the medium level.

Phase 2: With the aid of the knowledge base, the second step is to generate the recommendation policy based on the user input. When a user needs a policy for a certain scenario, the user just needs to input part of the information that he/she knows and desired privacy level. The LPPM system will fill in the remaining information. The LPPM system requires the users to specify at least two items when using the recommendation system: (1) the desired privacy protection level; (2) either the policy target or the locations to be protected.

In the first case when the user input the policy target and privacy level, the LPPM system will conduct the following steps. First, the LPPM system locates the group of policies which contain the same role of the input policy target. For example, Bob indicates that he would like to assign a medium level privacy policy to his new friend Tim. This input contains information about the privacy level, the role of the policy target (i.e., "friend"). The LPPM system will search the policy repository to find the group of policies for "friends". Within the retrieved policies, the LPPM system further looks for policies of user requested privacy level, e.g., medium level. Among the qualifying policies, the one with highest PL value will be selected. Finally, the LPPM system customizes the policy target to include the information from user input, e.g., the friend's name (i.e., Tim).

In the second case when the user input the locations to be protected and the desired privacy level, the LPPM system will search all the existing policies to find the ones at the required privacy level. Then, the LPPM system replaces the locations in the identified policies to the user input. For example, Bob wants to set up location privacy policies with high level protection when he is traveling at Chicago. The LPPM system finds that there are three policies at high level which are specified for family, friends and colleagues respectively. These three policies will be customized by modifying the locations to "Chicago" and present to Bob for review.

Phase 3: Finally, after the user decides the policies to be added to the system, the LPPM system will compute the privacy level of the newly inserted policies and store them for the future use. Note that it is possible that no matching policy is found by the recommendation function. In that case, the user needs to compose the policy by himself.

4 Conclusion

We developed a location privacy policy management system. The system supports an easy-to-understand yet expressive policy language. The system also automatically

detects policy conflict whenever there is a policy update. Moreover, the system generates recommended policies based on existing privacy policies so that users do not need to compose entire policy for every new friend. In the future, we plan to implement a prototype in smart phones to further verify the practical value of the proposed system.

References

1. Bamba, B., Liu, L., Pesti, P., Wang, T.: Supporting anonymous location queries in mobile environments with privacygrid. In: Proceeding of the 17th International Conference on World Wide Web, pp. 237–246 (2008)
2. Chow, C.Y., Mokbel, M.F., Liu, X.: A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In: Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems, pp. 171–178 (2006)
3. Cvrcek, D., Kumpost, M., Matyas, V., Danezis, G.: A study on the value of location privacy. In: Proc. of the ACM Workshop on Privacy in Electronic Society, pp. 109–118 (2006)
4. Damiani, M.L., Bertino, E., Silvestri, C.: Spatial domains for the administration of location-based access control policies. Journal of Network and Systems Management 16(3), 277–302 (2008)
5. Danezis, G., Lewis, S., Anderson, R.: How much is location privacy worth. In: Fourth Workshop on the Economics of Information Security (2005)
6. Gedik, B., Liu, L.: Protecting location privacy with personalized k-anonymity: Architecture and algorithms. IEEE Transactions on Mobile Computing 7(1), 1–18 (2008)
7. Ghinita, G., Kalnis, P., Skiadopoulos, S.: Prive: anonymous location-based queries in distributed mobile systems. In: Proceedings of the 16th International Conference on World Wide Web, pp. 371–380 (2007)
8. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, pp. 31–42 (2003)
9. Lin, D., Bertino, E., Cheng, R., Prabhakar, S.: Location privacy in moving-object environments. Transactions on Data Privacy 2(1), 21–46 (2009)
10. Mokbel, M.F.: Towards privacy-aware location-based database servers. In: 22nd International Conference on Data Engineering Workshops. Proceedings, p. 93 (2006)
11. Mokbel, M.F., Chow, C.Y., Aref, W.G.: The new casper: query processing for location services without compromising privacy. In: Proceedings of the 32nd International Conference on Very Large Data Bases, pp. 763–774 (2006)
12. Myles, G., Friday, A., Davies, N.: Preserving privacy in environments with location-based applications. IEEE Pervasive Computing 2(1), 56–64 (2003)
13. Sadeh, N., Hong, J., Cranor, L., Fette, I., Kelley, P., Prabaker, M., Rao, J.: Understanding and capturing people's privacy policies in a mobile social networking application. Personal and Ubiquitous Computing 13(6), 401–412 (2009)
14. Smailagic, A., Kogan, D.: Location sensing and privacy in a context-aware computing environment. IEEE Wireless Communications 9(5), 10–17 (2002)
15. Snekkenes, E.: Concepts for personal location privacy policies. In: Proceedings of the 3rd ACM Conference on Electronic Commerce, pp. 48–57 (2001)
16. OASIS Standard. Extensible access control markup language (XACML). version 2.0 (2005)