Recommendations-based Location Privacy Control

Hongxia Jin, Gokay Saldamli, Richard Chow
Samsung Information Systems of America
San Jose, CA, USA
{hongxia.jin, gokay.s, richard.chow}@sisa.samsung.com

Bart P. Knijnenburg
Department of Computer Science
University of California, Irvine, USA
bart.k@uci.edu

***Abstract*—In this paper, we propose and investigate a user-centric device-cloud architecture for intelligently managing user data. The architecture allows users to keep their (private) data on their mobile devices and decide what to share with the service providers on the cloud, based on their individual privacy preferences, in order to get personalized services. Our architecture strives to help ease users' burden on managing privacy by giving automatic recommendations on how to configure their privacy profiles on devices. One focused contribution of this paper is that we instantiate this proposed general architecture to location-based service due to the privacy sensitivity of location data. We derive and validate our location-sharing recommendations using online user experiments. Our results show that the recommendations are accurate, and that they help users with the decisions involved in the privacy profile configuration process. Our results also demonstrate that the quality of personalized location-based services can be maintained even when the increased user privacy control leads to a situation where not all location data is shared with the service provider. These results lead the way to powerful location-based and other personalized services that improve user privacy.**

Keywords- mobile device-cloud environment, recommender systems, location-based systems, privacy, security

I. INTRODUCTION

With the explosive growth in popularity of the "always-on" devices, pervasive computing enters a new era. A huge amount of user data is continuously collected on their devices and may be sent to the service providers on the cloud. Location-Based Services (LBS) are a primary example of this. For example, a location-based social network allows users to "check in", telling their friends and the system where they are, through social media. Such a service helps users to stay socially connected, and enables the service providers to offer personalized services based on users' location. Despite these benefits, the adoption rate of LBS has been relatively low [8]. One explanation for this is the myriad of privacy concerns around location data and such concerns significantly limit users' amount of location-sharing [5].

Current industry-standard practices for location privacy are fairly primitive. Google Latitude, for example, only has a few basic privacy settings. In contrast, users of LBSs may encounter a wide variety of situations with different levels of sensitivity. Users may thus want to assess on a case-by-case basis to what extent they are willing to share their location with whom, and at what level of detail. An LBS can account for this variability by giving users *fine-grained options* to share their information. Moreover, recent research has suggested that fine-grained sharing options should increase users' level of disclosure[3].

However, location-sharing decisions are inherently constructive, contextual and personal. Increasing the number of sharing options may thus turn location-sharing into a rather complex decision that puts an unnecessary burden on the user. There is often a large discrepancy between users' expected sharing settings and their actual settings, as demonstrated for Facebook in [1].

In this paper, we reconcile these two opposing desires with a powerful, fine-grained privacy architecture that assists the user in making these complex privacy decisions. We explore some simple approaches that can be used in each component of the architecture, and conduct a series of experiments that demonstrate the viability and potential benefits of these approaches. Our work provides significant insights for building more practical and successful location-based services that at the same time improve user privacy. We argue that such systems will be superior in terms of user adoption.

II. RECOMMENDATION-BASED ARCHITECTURE

Throughout this paper we will use a running example of a mobile service that uses our proposed architecture, called "Hotspots". This service allows selective sharing of a user's location and also provides personalized location-based recommendations based on the user's previous "check-ins" (i.e., it combines features of Foursquare and Yelp).

Our proposed architecture is illustrated in Figure 1. The *Server-Side Privacy Recommendation Engine* (S-PRE) has information about the **anonymized** and **abstract** level sharing behavior of existing users in various sharing situations. The server may derive a default policy at an abstract level based on the collective knowledge from existing users. The default policy deployed on devices can ease the cold start problem for a new user. Periodically, S-PRE may update the default policy if the collective behavior changes/evolves.

The *Client-Side Privacy Recommendation Engine* (C-PRE) recommends a user to share (or hide) her location with other users and/or the HotSpots server itself. C-PRE can base this recommendation on the users' own sharing history (stored locally), or use the default policy in similar sharing situations from S-PRE. C-PRE may combine the considerations of default policy with the user's own sharing history.

With user explicit control on what data to share with the service providers, the personalized service is more privacy friendly. However, the quality of the service may be affected due to the potentially less data available to the service providers. In this paper we investigate the effect of restricted sharing on the quality of a personalized service (i.e. a location recommender), and demonstrate that it is still possible to achieve decent recommendation accuracy.

978-1-4673-5077-8/13/$31.00 ©2013 IEEE

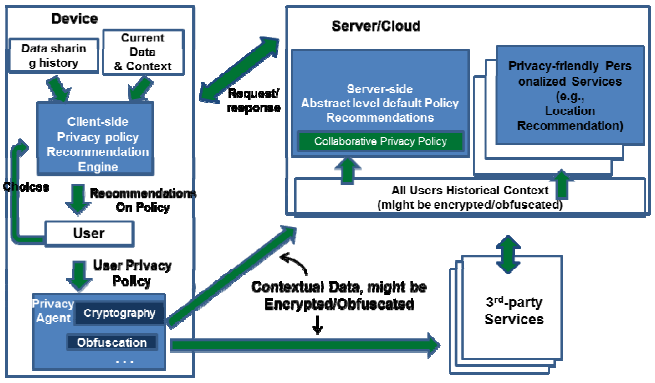

Figure 1. System Architecture

III. Recommendation Engine (Server & Client)

Our architecture assists users in making decisions on fine-grain sharing by recommending one or a subset of the sharing actions as being most appropriate in the current situation. Users are free to accept/reject any of these recommendations. Here are some example scenarios where such a recommendation may be beneficial:

a) The user stops on the way home at an adult bookstore. Because this is a sensitive location and the user feels uncomfortable to share this activity, the engine recommends disclosing only an approximate location, or keeping it private.

b) The user is at a restaurant with a group of school friends. The engine is aware of the user's social groups and recommends sharing the location only with the relevant social group/circle.

c) The user is alone in a bar. The engine suggests disclosing her location to her nearby friends only, so that they are invited to join without bothering those who are further away.

A. *Abstract level sharing policy recommendation*

When users decide whether or not to disclose their location, their reasoning typically considers not so much the location itself, but the activity that takes place there. For example, a user may share her location on a trip to Chicago because she finds it exciting, but not share a visit to a nude beach because she finds it embarrassing.

Based on this observation, we use "activity evaluation" to support an abstract level sharing policy in our architecture. For example, a user may set her policy as: "I felt excited about my current activity, and I chose to share this location with all my friends", or: "This activity makes me look interesting and I choose to share the location with a circle of my friends". Such an anonymized and abstract policy is safe to share with the server, because the server does not know the specific details of the activity, nor who the user's friends are. But such policy expressions, when shared with server, enable the server to derive a default policy at an abstract and canonical level (e.g. "If the user is excited about the activity, she wants to share her location with all her friends").

We also believe the question "What do you think about this activity?" is arguably easier to answer than the question "How do you want to share this location?" Moreover, if users' evaluation of the activity is indeed strongly related to their sharing behavior, the system can infer the correct sharing option from this evaluation, or at least reduce the number of plausible options. In other words: the system first converts the difficult sharing decision question into the activity evaluation question, which is much easier for the user to answer. Subsequently, based on the user evaluation of the activity, the system could recommend a (set of) sharing action(s).

Two requirements need to be met for this system to work. First of all, the server-side component (S-PRE) needs to analyze the evaluations of activities and sharing behavior from previous users to come up with recommended sharing actions. Secondly, the client-side component (C-PRE) needs to learn the user's evaluation of the current activity (through inference or by simply asking the user) and present the recommendation in a helpful yet unobtrusive way. We conducted two studies that show these requirements can be met.

IV. Experiments and Validations

To validate the above privacy recommenders for an LBS, several questions need to be answered:

a) Is the evaluation of the activity indeed a good predictor of the way/policy users want to share their location?

b) If so, would such a policy recommender indeed help users in making their sharing decisions?

c) And finally, would it still be possible to provide good personalized services to the user despite the potentially decreased amount of the input user data?

A. *Evaluations of activity as a predictor for sharing actions*

We conducted an online user experiment with 100 participants recruited through Amazon Mechanical Turk. Participants were asked to imagine using HotSpots as a Facebook "check-in" and location recommendation service. Participants were presented with 10 location-sharing scenarios:

S1. Looking for a hairdresser near your home
S2. A vacation in Chicago
S3. Donating used clothes to a charity
S4. Inviting nearby friends to a new bar
S5. Find a legal counselor to help with neighbor issues
S6. Bar-hopping after work
S7. Lending your phone to your out-of-state aunt
S8. Finding a nude beach on Hawaii
S9. A trip to Las Vegas, finding a restaurant
S10. Buying a gift at a sex shop for a bachelor party

For each of these scenarios, participants chose one of the following 8 disclosure actions:

A1. fully use the system (recommendations+Facebook)
A2. restrict Facebook posts to friends that are nearby
A3. restrict Facebook posts to certain friends only
A4. restrict Facebook posts to only share the city
A5. restrict Facebook posts to only share the city block
A6. use recommendations, but don't post on Facebook
A7. "private mode" (temporarily anonymous)
A8. don't use the system in this situation (turn it off)

Participants subsequently chose one of the following 10 evaluations of the activity ("select the best one"):

E1. is exciting
E2. is interesting for others

E3. makes me proud
E4. makes me look interesting
E5. needed a good recommendation
E6. is private
E7. embarrasses me
E8. isn't useful for everyone
E9. doesn't really represent me
E10. may have unintended consequences when shared

We observe that despite the variety in chosen actions between scenarios and between users, there is a significant relation between the disclosure action and the evaluation of the activity, as shown in Table I. Given the evaluation, it is therefore possible to *recommend* a subset of all available actions to the user. For instance, if we recommend only one action for each evaluation, we are correct 43.2% of the time. If we recommend the top-3 actions per evaluation, we are correct 84.3% of the time. Increasing the number of recommended actions to just under 4 actions per evaluation (dark and light gray cells in Table I), we can get 95.1% recall.

TABLE I. ACTIVITY EVALUATIONS VESUS SHARING DECISIONS

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
A1	34	88	14	25	24	0	0	1	1	1
A2	6	25	1	6	6	3	0	32	0	4
A3	5	16	6	9	6	17	3	41	1	8
A4	1	8	1	11	6	4	2	10	0	2
A5	0	3	4	1	1	2	1	1	1	5
A6	2	5	0	1	23	112	17	58	16	36
A7	0	0	1	0	0	80	18	20	19	40
A8	0	0	0	0	0	34	14	27	4	26

B. *Testing the policy recommenders*

We also want to know: how many actions should it recommend? How should these recommendations be presented? How will this affect the recommender accuracy?

We recruited 368 U.S. participants using Amazon Mechanical Turk. Each participant was assigned to one of 6 conditions. In each condition (except for C1), the system first asks the user to evaluate the activity presented in the scenario using one of 6 options (E1-E10, with certain options combined, namely E2 and E4, E6 and E10, E7 and E9). Each recommender then tailors the display of the 8 sharing actions to the selected evaluation. Below are the 6 conditions:

C1. No recommendation
C2. Long list, rest hidden
C3. Short list, rest hidden
C4. One item, rest hidden
C5. Short list, highlighted
C6. One item, highlighted

Our results show that these recommendations are indeed accurate. In fact, as shown in Table II, the recommenders in this second study had a higher recall than would be expected from the first study: a clear indication that the recommendations were persuasive: users are disproportionally more likely to choose a recommended sharing action (C2-C6) over a sharing action that is not recommended (C1). In other words, the recommender persuaded them to choose one of the recommended sharing actions. The fact that this recommender is persuasive gives companies the opportunity to influence users' sharing actions through these recommendations.

TABLE II. ACTUAL, EXPECTED, AND COMPARED RECALL FOR EACH OF THE 5 RECOMMENDER CONDITIONS.

	C2	C3	C4	C5	C6
Recall	97.9%	91.4%	75.5%	84.9%	62.8%
Expected (study 1)	95.1%	81.5%	43.2%	81.5%	43.2%
Compared (C1)	86.6%	66.3%	36.0%	66.3%	36.0%

C. *Privacy-Preserving Personalized Services*

We want to evaluate how the quality of personalized services would be affected with less user data. We took location recommendation as an example service and modeled a location recommender using actual location check-ins on Gowalla from June to October 2010. Previously, this dataset was studied in [9] to evaluate the feasibility for location recommendation using collaborative filtering.

Our goal was to evaluate how the predictive quality of the dataset decreased as data were reduced. The dataset consisted of 104,875 users and 7,718,911 check-ins. Using Austin, Texas as an example, it has 4,871 users and 380,685 check-ins. The number of locations in Austin was 9,577.

To evaluate a location recommendation system, we generated a binary user rating for each location. A location was marked with "1" for a user only if the user checked into that location at least once. Otherwise, mark with "0". For our set of 4,871 users and 9,577 locations, we obtained 245,153 ratings of "1".

We divided our dataset into a training set and test set by randomly choosing 20% of each user's ratings as a held-out test set, giving 49008 test ratings in all. We trained a Top-N recommender based on the remaining 196145 ratings. The recommender's task is to predict the likelihood of visiting the remaining 46453422 (=4871*9577 - 196145) spots, of which 49008 are test spots which have been visited.

In our experiments, we used standard user similarity collaborative Filtering (CF) [2], a popular class of recommender algorithms. The algorithm computed the user's similarity with all other users (using cosine similarity), and the probability of visiting a spot was the fraction of users who have visited the spot, weighting the users by similarity.

Our first finding is that the location data from 2am-4am decreases the recommendation accuracy. In other words, while 2am-4am location data might be privacy sensitive, it is however not useful for recommender accuracy. We then define and measure a "user hardship" for each data point and separate the data into 10 pieces or "deciles". A user's location traces are clustered according to surface-of-the-earth distance, and points are ranked according to their distance to the set of cluster centroids. Intuitively those centroids correspond to "home" and "work" locations which are arguably privacy sensitive. We experiment the recommender accuracy with some deciles of the data removed successively.

The result for city of Austin is shown in Figure 2. Surprisingly the accuracy after deleting 60% of the all data is still comparable with accuracy having all the data. A big portion

of the privacy sensitive data is removed within the 60% data removal. The results for other cities are very similar.

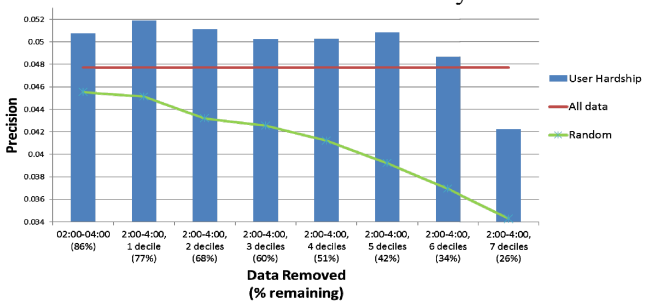

Figure 2. Accuracy as data is successively removed.

V. Discussion on Privacy Agent

In our architecture, as shown in Figure 1, depending on the user's sharing decision, the location data will be processed before uploading it to the server. For instance, part of the data may be filtered out and decided to stay private on users' device and not be shared with the service providers. Some of the data may be only shared with a specific group people (a circle of friends, or friends who are nearby/at the same location). In our architecture, the main goal of the *Privacy Agent* is to enable these types of limited scope sharing and allow the user to selectively and securely share her location with other users and/or service providers.

Depending on the exact application that our architecture is applied to and the exact sharing options offered for the application, cryptographic protocols may need to be implemented to enforce those fine-grain limited scope sharing options. To make the cryptographic protocols lightweight, one prefers as few encryptions as possible.

In the example of a "private sharing protocol" between A-Alice and B-Bob without S-Server learning any data under sharing, one could perform one encryption for each private sharing between two parties. However, here we show a more lightweight design of the private sharing protocol. Let location information (i.e w_a, w_b) be represented by 32-bit numbers and encryption function E has 128-bit block size.

- Keys k_a and k_b are shared by A & S and B & S respectively; k_{ab} is the key shared by users A & B.
- *ctr* is a counter value set to zero in the very first handshake of users and incremented whenever needed.
- k_0, k_1, k_2 and k_3 are 32 bit parts of the encrypted *ctr* value.

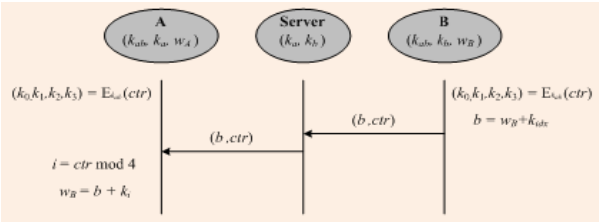

Figure 3. Private Sharing Protocol.

As shown in Figure 3, both parties only need to perform a single encryption after every four sharing instances. This type of efficient design is highly desirable in our architecture.

VI. Conclusion

In this paper we proposed a new recommendation-based device-cloud architecture that will automatically recommend user's desirable location sharing decision for each sharing case. This architecture allows users to maintain fine-grained control of their data, sharing with a service provider **only** data they are comfortable sharing. We have conducted two user studies to validate the policy recommender. The first study shows that the server-side component can indeed successfully come up with abstract-level policy recommendations: the evaluation of the activity is strongly related to the sharing behavior. Our second study evaluates the accuracy of our recommendation mechanism. The results of this study demonstrate that users indeed follow these recommendations, and perceive them as helpful. An interesting future work for us is to explore how the service providers can influence users' sharing actions through these recommendations.

We also experimented with an actual location dataset, and gave evidence that standard collaborative filtering algorithms perform well even with substantially less data. These results points out other potential interesting future research directions. For example, it may be possible to design a flexible strategy that protects different data to different degrees depending on its privacy sensitiveness and its importance to the recommender accuracy. We are also interested in exploring ways to encourage user sharing data that is important to the recommendation services.

References

[1] Liu, Y., Gummadi, K., Krishnamurthy, B., and Mislove, A. Analyzing Facebook Privacy Settings: User Expectations vs. Reality. IMC, (2011), 61–70.

[2] http://en.wikipedia.com/wiki/Collaborative_filtering

[3] Benisch, M., Kelley, P.G., Sadeh, N., and Cranor, L.F. Capturing location-privacy preferences: quantifying accuracy and user-burden tradeoffs. Personal Ubiquitous Comput. 15, 7 (2011), 679–694.

[4] Knijnenburg, B., Willemsen, M.C., Gantner,Z., Soncu,H., & Newell, C. Explaining the user experience of recommender systems. User Modeling and User-Adapted Interaction 22, 4-5 (2012), 441–504.

[5] Strater, K. and Lipford, H.R. Strategies and struggles with privacy in an online social networking community. BCS HCI, (2008), 111–119.

[6] Xu, H., Luo, X. (Robert), Carroll, J.M., and Rosson, M.B. The personalization privacy paradox: An exploratory study of decision making process for location-aware marketing. Decision Support Systems 51, (2011), 42–52.

[7] Xu, H., Teo, H.-H., Tan, B., and Agarwal, R. The Role of Push-Pull Technology in Privacy Calculus: The Case of Location-Based Services. J. of Management Information Syst. 26, 3 (2009), 135–174.

[8] Zickuhr, K. and Smith, A. 4% of online Americans use location-based services. Pew Research Center, 2010.

[9] Berjani, B. and Strufe, T. A Recommendation System for Spots in location-based Online Social Networks. *SNS*, (2011).