Building Privacy-preserving Location-based Apps

Brian Sweatt‡, Sharon Paradesi†, Ilaria Liccardi†*, Lalana Kagal†, Alex (Sandy) Pentland‡
brian717@mit.edu, {paradesi, ilaria, lkagal}@csail.mit.edu pentland@media.mit.edu

‡Media Lab
Massachusetts Institute of Technology
Cambridge, MA

†Computer Science and
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA

*INRIA Saclay Île-de-France
Orsay, France

***Abstract*—Social apps usually require a lot of personal information in order to be tailored to the needs of individual users. However, the inherent social exchange of data exposes a user's personal data to other app users or publicly for anyone to see.**

In this paper, we present an app that enables users to determine the optimal location and time to meet without exposing their information to other users. We compare this app to other research-based and commercial social apps and show that ours is the only one where the risk of exposure is not present. In order to provide such improved privacy protections, we use openPDS, a decentralized and open-source framework. openPDS enables users to store their data on their own servers and participate in group computations without exposing their raw data.

Keywords—Privacy, Location Data, Social Apps.

I. Introduction

Todays' smartphones collect different types of data about people through a multitude of built-in sensors that could be used to infer a data owner's behavioral patterns. Location information is one type of sensor data available for collection on current smartphones, which research has shown can be used to discover different types of personal behaviors and details, such as activity patterns [11], profile behaviors [5], and likes and dislikes [16]. While inferring a users' behavior and details could be beneficial for creating targeted services tailored to the user' needs and personalities, it can also present harms and risks.

Users willing to share their location with the public or specific third parties might be risking robbery, identity theft, etc. Websites like "PleaseRobMe"[1] have shown how easy it is to extract users' home information from tweets. Hence robbers might target specific people by looking at recent tweets and learning exactly which place to rob and when it would be empty. Foursquare[2] is another application that lets users share their current locations with the public. While this might seem innocuous, it could - as in the "PleaseRobMe" example - present risks and harms. People are often unaware of the nature and extent to which their information is collected since the apps installed on smartphones can silently collect data even when the device is idle [17]. Further, in most commercial location-based services, users do not have the ability to control (modify, permanently delete, or limit the use of) their data. We demonstrate building privacy-preserving location-based apps on top of the openPDS platform [6]. Though tools for privacy-preserving distributed computing are not novel [2], apps created using our approach have the ability to preserve users' privacy by utilizing the *question and answer* and *group computation* techniques of openPDS. These techniques enable the apps to function without sharing users' raw or fine-grained personal information with other participants or the apps themselves.

II. Related Work

With the widespread use of mobile devices, highly accurate sensor data (such as location) are being collected and often shared without users' knowledge. Location information (such as checkins in Foursquare) has been shown to be useful in determining users' activity patterns [15]. In fact, only four spatio-temporal data points are sufficient to uniquely identify individuals in a set of de-identified data [4].

Due to the re-identifiable nature of location data, various approaches and metrics have been created to safeguard users' anonymity and prevent re-identification. Techniques developed to prevent leakage of location information involve obscuring the location data by: degrading its quality [7], injecting fake location points [18], adding uncertainty [8], providing frequently changing pseudonyms [1], sharing opaque identifiers using symmetric key encryption [3], and cloaking it thereby providing k-anonymity [9].

However pseudonymity and cloaking have been proven to be inadequate for protecting users' location data and preventing re-identification [14], [19]. Krumm [14] demonstrates that subjects' home locations could be identified within a 60-meter radius, when pseudonyms were used. By using commercially-available reverse geocoders, 13% of these subjects could be re-identified by name. Researchers [19] also suggest that cloaking does not necessarily depend on k for k-anonymity. If the k users were present in a small region, it would be easy to identify their exact location. Further, intersections among the cloaked regions of nearby users could be used to infer the locations of those users.

Decentralized approaches include having various parties piece encoded information from participants together to create triggers instead of relying on a centralized server [13] or having the participants communicate mutual distances to each other using homomorphic encryption [21].

[1] http://pleaserobme.com/

[2] https://foursquare.com/

III. Preserving Users' private data

Social location-based apps enable users to interact with each other by sharing personal information. While these apps might provide an engaging and enjoyable experience, they have not been designed to preserve users' privacy. In our approach, we want to protect raw data and safeguard users' privacy while still providing most of the features that common popular apps provide. We use openPDS, an open-source platform that provides decentralized storage and computation space for users' raw personal sensor data. Under this system, users do not have to disclose their raw data to apps or third parties; each participant's openPDS instance accepts code to be run within the users' openPDS instance to compute answers to questions based on their raw data and can use these computed answers to contribute to group computations between other participants' openPDS instances. This privacy-preserving infrastructure allows apps using this approach to simulate social interaction while allowing users to store their own personal data.

A. Privacy-preserving Store: openPDS

Users' personal data are usually collected and stored disparately across a multitude of services. A user's social graph, calendar, and location history are typically stored by, and under the control of, the respective service used to generate the data. For this reason, any entity seeking a more holistic view of an individual's online life (as well as offline via sensor readings) must obtain authorization to access the raw data and perform analysis outside of the individual's control. This authorization to access the data in its raw form may be granted by the user (via consent mechanisms) or the service that stores the information. In typical scope-based authorization, individuals provide this authorization based on the source and type of data, with little to no insight or control over how that raw data is used or stored.

openPDS seeks to solve this problem by providing personalized services based on a complete view of the individual's data controlled and stored by each user (as opposed to the service-centric storage model that is ubiquitous today). In this manner, openPDS provides a unified location and interface for an individual to store and control their own personal data. Services (authorized third parties) which require users' personal data to function can access data based not only on the type and source of the raw data, but also the purpose for which the party intends to use the data. In order to use openPDS, a user must either pull down the open-source code on a server they control and register it with a service they hope to use, or seek a hosted openPDS solution operated on their behalf (as is the case with this paper).

B. Question and Answer Framework

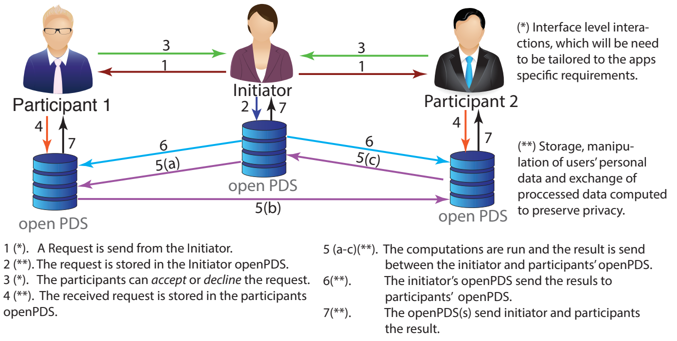
openPDS enforces purpose-based authorization by providing a question and answer framework on the data store and prohibiting outside access to and sharing of the underlying raw data. It provides an endpoint for third parties to submit questions (as Python code) that analyze the raw data within the individual's trusted PDS. Thus, computation on openPDS is restricted to code residing on the individual PDS instances and third-party endpoints or APIs cannot be used. Questions have a scope for each type of raw data they require access to, as well as a human readable purpose. Individuals must authorize access to each type of raw data a service requires, as well as the purpose it declares before the service can run questions within the individual's PDS. However the raw data is never directly accessed or seen by the service itself.

An internal API for reading and analyzing authorized raw data within an individual's PDS populates answers that can be consumed externally via a REST API. Each answer is stored in this external API with a corresponding key for retrieval at a later time. These keys correspond to additional authorization scopes, allowing individuals to revoke access to generated answers as well as the raw data used to compute them. As computed answers to questions are the only means of external data access to the PDS, users' raw data is never exposed.

C. Group Computation

Since each individual's personal data store only has access to the raw data for that individual, questions that aggregate data from multiple individuals must engage in a group computation with other openPDS instances. This implies that each PDS must either provide raw data access to a trusted centralized server for analysis, or contribute their respective portion of the computation, starting at an initiating data store, and proceeding in a ring to each participant. The Koi platform [10] uses the former technique by employing separate entities (matcher and combiner) to compute the location-based matches.

We have opted for the latter as it provides assurances that raw data access only occurs within the individual's PDS. Figure 1 shows a walkthrough example involving 3 participants' data. Each participants' PDS uses the data stored within it to update a running aggregate answer that is passed along between participating stores (Figure 1 [steps 5(a), 5(b) and 5(c)]). The time to complete a computation scales linearly in the number of contributing data stores; in general, each PDS incurs a 200ms overhead, on average, in addition to the computation time within the PDS. After the final participant has contributed to the answer, the result is broadcast to all participants' PDS (Figure 1 [step 6]).



(*) Interface level interactions, which will be need to be tailored to the apps specific requirements.

(**) Storage, manipulation of users' personal data and exchange of proccessed data computed to preserve privacy.

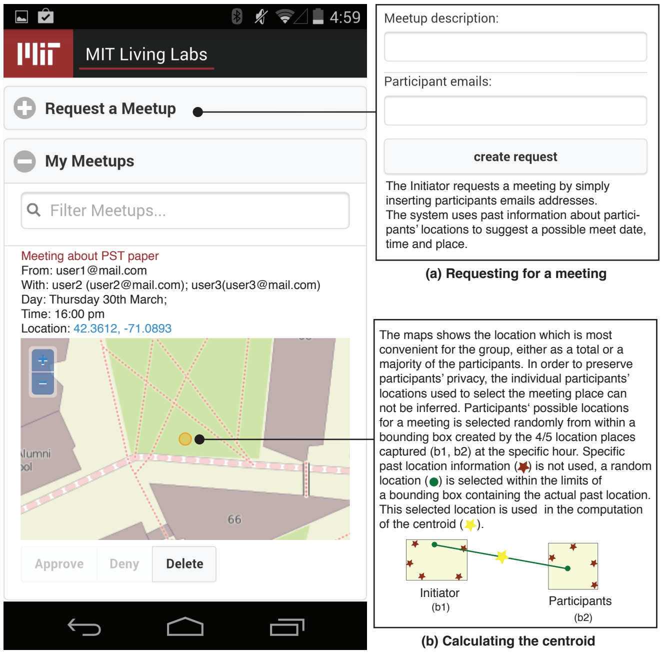
1 (*). A Request is send from the Initiator.
2 (**). The request is stored in the Initiator openPDS.
3 (*). The participants can *accept* or *decline* the request.
4 (**). The received request is stored in the participants openPDS.

5 (a-c)(**). The computations are run and the result is send between the initiator and participants' openPDS.
6(**). The initiator's openPDS send the resuls to participants' openPDS.
7(**). The openPDS(s) send initiator and participants the result.

Fig. 1. Example of the interactions, storage and manipulation of users' personal data and exchange of processed data between personal PDS instances underlining the privacy-by-design structure of the framework.

IV. ScheduleME: A Privacy-preserving App

We created ScheduleME, an app developed on openPDS, to demonstrate how our approach can help preserve users' privacy while still maintaining most of the functionality needed for their interactions (Figure 2). ScheduleME allows users to create and schedule meetings with one or more participants without directly disclosing any personal location information to them. Currently, when ScheduleME is installed by users, their respective PDS instances are configured and hosted for them.



Meeting about PST paper
From: user1@mail.com
With: user2 (user2@mail.com); user3(user3@mail.com)
Day: Thursday 30th March;
Time: 16:00 pm
Location: 42.3612, -71.0893

The Initiator requests a meeting by simply inserting participants emails addresses.
The system uses past information about participants' locations to suggest a possible meet date, time and place.

(a) Requesting for a meeting

The maps shows the location which is most convenient for the group, either as a total or a majority of the participants. In order to preserve participants' privacy, the individual participants' locations used to select the meeting place can not be inferred. Participants' possible locations for a meeting is selected randomly from within a bounding box created by the 4/5 location places captured (b1, b2) at the specific hour. Specific past location information (★) is not used, a random location (●) is selected within the limits of a bounding box containing the actual past location. This selected location is used in the computation of the centroid (★).

(b) Calculating the centroid

Fig. 2. ScheduleME app created using openPDS: showing (a) the interface to request a meeting; (b) the possible results of a group computation, explaining how the actual personal location information is preserved and the computed answer is shared among users' PDS to maintain participants' privacy.

An initiator sends a request to participants by entering their email addresses (Figure 2(a)). Each participant can either approve, deny, or delete the request. PDS instances use hierarchical clustering on each participant's location history to compute bounding boxes that represent where the user has spent most of their time for each hour of the day. Once each participant approves the request, the initiator's PDS randomly selects a location for each hour from the bounding box computed for that hour. This suggested location is then passed to the neighboring PDS in the ring. Each participants' PDS updates the incoming suggested locations by computing the centroid between the received location and a random location from its own bounding box for that hour of the day (Figure 2(b)). Additionally, each participant's PDS computes the distance from that centroid to their randomly selected location and adds this distance to a running score for each hour (Figure 1). Upon receiving the suggested locations and scores for all hours of the day from the last participant, the initiator's PDS selects and broadcasts the meeting place and time having the lowest score. All participants must then agree on the set location and time; if a consensus can not be reached, the meeting is not scheduled. In the future, ScheduleME could take into account users' appointments or iterate until a consensus is reached.

Since a centroid is passed around and not the actual location coordinates, the participants do not have to suggest possible places or disclose their past or current locations to other participants. Thus, privacy is preserved while a reasonably accurate estimation of the participants' locations is used for the computation. Further, since the participants' location data is stored within their own PDS instances, there is no risk of exposure or enabling inferences from raw data.

V. Discussion

There are a number of apps – both commercial and research-based – that use users' location to provide various types of services. Blendr[3] and Skout[4] allow discovery of people while Glympse[5], Owntracks[6] and Miataru - be found[7] share location information between users. Other apps such as Waze[8] and GPS Plus[9] provide crowd-sourced and personal traffic navigation respectively. Tag - You're It[10] and PrivateMeetUp [12] use location information to help coordinate meetups. These apps use centralized or decentralized structures to store users' data (Table I). Only one app, PrivateMeetUp [12], was developed to explicitly preserve users' location data privacy.

We have analyzed these apps in order to understand their privacy mechanisms with respect to disclosure of user's location data (Table I). In particular, we focus on (i) **Visibility:** who can view a user's location and who can users share their location with; (ii) **Real-time tracking:** whether the app allows users' location to be tracked in real-time; (iii) **Privacy techniques:** privacy-preserving mechanisms provided by the app; (iv) **Log-in access:** whether an app requires users to create a new account or allows them to use their existing social network profiles; (v) **Framework structure:** whether the app's framework is centralized or decentralized (Table I).

Location information has been shown to help users convey truthful signals and build trust when forging relationships [20]. Thus, it is not surprising that all referenced apps enable users to share their location information at some level (Table I). Blendr and Skout recommend people based on a user's proximity to them, allowing strangers to view their *inexact* locations. Glympse, Waze, and GPS Plus enable sharing location information for a user-specified amount of time. Tag allows users to privately broadcast their location and view a suggested route to meet with "tagged" friends. In PrivateMeetUp, users do not share actual location information, but send approximate distances to points of interest to their peers.

Visibility can be inferred from the type of social network allowed. Glympse, Waze, Owntracks, and Miataru allow real-time tracking of users' location. Glympse restricts the tracking privileges to a maximum of 4 hours. Users can be identified in different ways on these apps. Miataru, Owntracks, and GPS Plus identify users using their device IDs. Waze allows temporary accounts without requiring users to log in, though anonymous users need to log in to share their location with contacts. The remaining apps require users to log in by either creating new accounts or by using their existing social network profiles (Table I).

A key differentiator for ScheduleME is that it does not allow contacts to either view or track raw locations of users; it differs from other similar-purpose apps (Tag and PrivateMeetUp) by providing users with a potential meeting place

[3] http://blendr.com/
[4] http://www.skout.com/
[5] https://www.glympse.com/
[6] http://owntracks.org/
[7] http://miataru.com/
[8] https://www.waze.com/
[9] https://itunes.apple.com/us/app/gps-plus-location-commute/id528698859
[10] https://itunes.apple.com/us/app/tag-youre-it/id829344553

TABLE I. COMMERCIAL & RESEARCH LOCATION-BASED APPS, SHOWING THE VISIBILITY WITHIN EACH APPS, WHETHER REAL-TIME LOCATION TRACKING IS POSSIBLE, THE PRIVACY TECHNIQUES USED TO SAFEGUARD USERS' LOCATION INFORMATION, TYPE OF LOG IN ACCESS (F. FACEBOOK, G. GOOGLE+, T. TWITTER, FS. FOURSQUARE, O. OTHERS), AND THE UNDERLINING STRUCTURE OF DATA STORAGE / COMMUNICATION.

		VISIBILITY			LOG IN ACCESS		FRAMEWORK
	APP NAME	(SHARING)	TRACKING	PRIVACY TECHNIQUES	NEW	S.N.	STRUCTURE
COMMERCIAL	**Blendr**	Public	No	Reveals inexact location	Yes	F.	Centralized
	Glympse	F. T.	Yes	Time-based (can expire early)	Yes	F., T.	Centralized
	GPS Plus	Private	No	Share past history (up to 24 hours old)	Device	N/A	Centralized
	Miataru	Private	Yes	Can use own server to store data	Device	N/A	Decentralized
	Owntracks	Private	Yes	Can use own server to store data	Device	N/A	Decentralized
	Skout	Public	No	Separate communities for adults and teens	Yes	F., G.	Centralized
	Tag	Private	No	Send location and privately view route to meetup	Yes	F.	Centralized
	Waze	F., T., FS.	Yes	Send ETA via email/SMS	Optional	F.	Centralized
RESEARCH	**Private-MeetUp**	F.	No	Only disclose approximate distance to a particular location	N/A	F.	Decentralized
	ScheduleME	None	No	individual private store, question and answer framework, group computation	Yes (email)	N/A	Decentralized

without disclosing their actual locations. It also does not allow peer-to-peer communication of raw or approximate location data. By comparing bounding boxes and alerting users when others have similar patterns, openPDS-based apps could provide unexpected discovery of nearby people (like Skout and Blendr). openPDS-based apps could simulate navigation-based apps (Waze and GPS Plus) using collected activity, bluetooth and other sensor data. Rather than track users' location like Glympse, Miataru and Owntracks, openPDS-based apps could broadcast plausible locations by randomly selecting the most-frequented location in recent history. Thus, we see that apps developed on openPDS can provide similar experiences to other apps while keeping user data private.

VI. CONCLUSION

In this paper, we demonstrate a privacy-preserving way to build location-based apps using openPDS framework. We outlined and compared our demo app (ScheduleME) and possible other apps built using this framework with popular location-based apps, their functionalities and their different privacy mechanisms. We have shown that privacy-preserving location-based apps built with openPDS can provide most of the functionalities of popular apps while still keeping users' personal location data private.

VII. ACKNOWLEDGEMENTS

This material is based upon work supported by the MIT Big Data Initiative at CSAIL. Ilaria Liccardi was supported by the European Commission Marie Curie International Outgoing Fellowship grant 2011-301567 *Social Privacy*.

REFERENCES

[1] Beresford, A. R., and Stajano, F. Location privacy in pervasive computing. *Pervasive Computing, IEEE 2*, 1 (2003), 46–55.

[2] Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X., and Zhu, M. Y. Tools for privacy preserving distributed data mining. *ACM SIGKDD Explorations Newsletter 4*, 2 (2002), 28–34.

[3] Cox, L. P., Dalton, A., and Marupadi, V. Smokescreen: flexible privacy controls for presence-sharing. In *Proc. of ACM MobiSys* (2007), 233–245.

[4] de Montjoye, Y.-A., Hidalgo, C. A., Verleysen, M., and Blondel, V. D. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports 3* (2013).

[5] de Montjoye, Y.-A., Quoidbach, J., Robic, F., and Pentland, A. S. Predicting personality using novel mobile phone-based metrics. In *Social Computing, Behavioral-Cultural Modeling and Prediction*. Springer, 2013, 48–55.

[6] de Montjoye, Y.-A., Wang, S. S., Pentland, A., Anh, D. T. T., and Datta, A. On the trusted use of large-scale personal data. *IEEE Data Eng. Bull. 35*, 4 (2012), 5–8.

[7] Duckham, M., and Kulik, L. A formal model of obfuscation and negotiation for location privacy. In *Pervasive computing*. Springer, 2005, 152–170.

[8] Gambs, S., Killijian, M.-O., and del Prado Cortez, M. N. Show me how you move and i will tell you who you are. In *Proc. of ACM SIGSPATIAL Workshop on Security and Privacy in GIS and LBS*, ACM (2010), 34–41.

[9] Gruteser, M., and Grunwald, D. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. of ACM MobiSys* (2003), 31–42.

[10] Guha, S., Jain, M., and Padmanabhan, V. N. Koi: A location-privacy platform for smartphone apps. In *Proc. of NSDI* (2012), 1–14.

[11] Hasan, S., Zhan, X., and Ukkusuri, S. V. Understanding urban human activity and mobility patterns using large-scale location-based data from online social media. In *Proc. of ACM SIGKDD Workshop on Urban Computing* (2013), 6.

[12] Hashem, T., Ali, M. E., Kulik, L., Tanin, E., and Quattrone, A. Protecting privacy for group nearest neighbor queries with crowdsourced data and computing. In *Proc. of ACM Ubicomp* (2013), 559–562.

[13] Jaiswal, S., and Nandi, A. Trust no one: a decentralized matching service for privacy in location based services. In *Proc. of ACM SIGCOMM Workshop on Networking, systems, and applications on mobile handhelds* (2010), 51–56.

[14] Krumm, J. Inference attacks on location tracks. In *Pervasive Computing*. Springer, 2007, 127–143.

[15] Noulas, A., Scellato, S., Mascolo, C., and Pontil, M. An empirical study of geographic user activity patterns in foursquare. *ICWSM 11* (2011), 70–573.

[16] Savage, N. S., Baranski, M., Chavez, N. E., and Höllerer, T. Im feeling loco: A location based context aware recommendation system. In *Advances in Location-Based Services*. Springer, 2012, 37–54.

[17] Shih, F., and Boortz, J. Understanding peoples preferences for disclosing contextual information to smartphone apps. In *Human Aspects of Information Security, Privacy, and Trust*. Springer, 2013, 186–196.

[18] Shokri, R., Theodorakopoulos, G., Danezis, G., Hubaux, J.-P., and Le Boudec, J.-Y. Quantifying location privacy: the case of sporadic location exposure. In *Privacy Enhancing Technologies*, Springer (2011), 57–76.

[19] Shokri, R., Troncoso, C., Diaz, C., Freudiger, J., and Hubaux, J.-P. Unraveling an old cloak: k-anonymity for location privacy. In *Proc. of ACM Workshop on Privacy in the electronic society* (2010), 115–118.

[20] Toch, E., and Levi, I. Locality and privacy in people-nearby applications. In *Proc. of ACM Ubicomp* (2013), 539–548.

[21] Zhong, G., Goldberg, I., and Hengartner, U. Louis, lester and pierre: Three protocols for location privacy. In *Privacy Enhancing Technologies*, Springer (2007), 62–76.