Empowering Mobile Crowdsourcing Apps with User Privacy Control

Lakhdar Meftah, Romain Rouvoy, Isabelle Chrisment

Version of Record: https://www.sciencedirect.com/science/article/pii/S0743731520303403
Manuscript_494f6e77b9ca6f52e738e019036f3c11

Empowering Mobile Crowdsourcing Apps with User Privacy Control

Lakhdar Meftah[a], Romain Rouvoy[b] and Isabelle Chrisment[c]

[a]*Inria / Univ. Lille, France*
lakhdar.meftah@inria.fr
[b]*Univ. Lille / IUF / Inria, France*
romain.rouvoy@univ-lille.fr
[c]*LORIA-TELECOM Nancy / Univ. Lorraine / Inria, France*
isabelle.chrisment@loria.fr

ARTICLE INFO

Keywords:
Distributed Applications
Location Privacy
Mobile Crowdsourcing
Location Privacy Protection Mechanism

ABSTRACT

Mobile crowdsourcing is being increasingly used by industrial and research communities to build realistic datasets. By leveraging the capabilities of mobile devices, mobile crowdsourcing apps can be used to track participants' activity and to collect insightful reports from the environment (*e.g.*, air quality, network quality). However, most of existing crowdsourced datasets systematically tag data samples with metadata (*e.g.*, time and location stamps), which may inevitably lead to user privacy leaks by discarding sensitive information in the wild.

This article addresses this critical limitation of the state of the art by proposing a software library that empowers legacy mobile crowdsourcing apps to increase user privacy without compromising the overall quality of the crowdsourced datasets. We propose a decentralized approach, named FOUGERE, to convey data samples from user devices to third-party servers. By introducing an *a priori* data anonymization process, we show that FOUGERE defeats state-of-the-art location-based privacy attacks with little impact on the quality of crowdsourced datasets.

1. Introduction

Mobile crowdsourcing platforms and applications are being widely used to collect datasets in the field for both industrial and research purposes [4, 14, 61]. By relying on a crowd of user devices, mobile crowdsourcing delivers an engaging solution to collect insightful reports from the wild. However, the design of such platforms presents some critical challenges related to the management of users, also known as *workers*. In particular, the privacy of the workers is often underestimated by the crowdsourcing platforms and it often fails to be addressed effectively in practice [44]. Official approvals from *Institutional Review Boards* (IRB) and regulatory bodies addressing worker privacy do not only require consent from workers to collect data from their mobile devices (*e.g.*, app permissions, privacy policies, end-user license agreements), but suggest the use of data anonymization mechanisms to minimize the risk of privacy leaks [30].

While data anonymization is commonly achieved *a posteriori* on the server side [17, 26, 35, 40], this approach is subject to adversarial attacks, even when protocols for the communication and the data storage are claimed to be secured [20, 21]. Furthermore, the workers may be reluctant to share *Sensitive Personal Information* (SPI) with third parties (*e.g.*, students contributing to a crowdsourcing campaign initiated by a professor). Gaining the confidence of workers is extremely difficult and we argue in this chapter that the adoption of *a priori* data anonymization mechanisms contributes to delivering a trustable component to better mitigate privacy leaks in the data shared by workers.

For example, the worker's location is not only the most requested but also the most sensitive data collected by mobile crowdsourcing platforms [8]. Our scheme therefore explores the physical proximity of workers to agree on a dissemination strategy for reporting the crowdsourced data. By altering the link between workers and data *consumers* on the server, our approach intends to mix data contributed by several workers within a collaborative data flow that exhibit similar crowd-scale properties and without discarding any SPI. In particular, we propose a system-level service that acts as a proxy within the mobile device for sharing crowdsourced data and from which workers can control their privacy settings. FOUGERE library is our implementation of this anonymization scheme and is available as an open source library[1] that can be used by legacy mobile crowdsourcing apps. We illustrate the benefits of FOUGERE by integrating it within the state-of-the-art MOBIPERF mobile crowdsourcing app as well as the APISENSE mobile crowdsourcing platform. We evaluate the effectiveness and the impact of our anonymization scheme on these two mobile crowdsourcing systems by deploying and orchestrating a crowd of 15 emulated mobile devices. More precisely, we replay the SFCABS cab mobility traces [60] and we show that FOUGERE defeats state-of-the-art privacy attacks [28, 43, 47] with little impact on the quality of the resulting datasets.

The remainder of this article is organized as follows. Section 2 provides background knowledge on mobile crowdsourcing platforms and discusses the related work in the domain to privacy applied to the areas of mobile crowdsourcing and location-based services. Section 3 provides an overview of the privacy threats in crowdsourcing apps and platforms. Section 4 introduces our anonymization scheme and the integration of *Location Privacy Protection Mechanisms* (LPPMs) to increase the workers' privacy. Section 5 describes the im-

ORCID(s): 0000-0002-0292-3795 (L. Meftah); 0000-0003-1771-8791 (R. Rouvoy); 0000-0002-8474-0019 (I. Chrisment)

[1] https://github.com/m3ftah/fougere

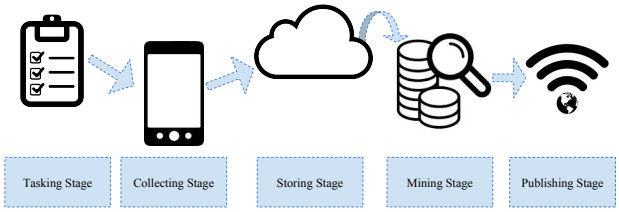

Figure 1: Anatomy of a mobile crowdsourcing campaign

plementation of the FOUGERE open source library on Android. Section 6 introduces our evaluation protocol of FOUGERE on the MOBIPERF mobile crowdsourcing app and discusses the results we obtained on an experimental setup involving 15 emulated workers. Section 7 discusses the threats to validity of our contribution. Finally, Section 8 concludes on this article.

2. Background on Mobile Crowdsourcing and User Privacy

With the ubiquity of the mobile devices, both scientific and industrial communities leverage the sensors of the mobile devices to collect insightful information about the user. As depicted in Figure 1, mobile crowdsourcing campaigns typically consist of several stages: *i)* the description of the data to be crowdsourced, *ii)* the deployment and the gathering of the dataset in the wild, *iii)* the aggregation and storage of datasets in the Cloud, *iv)* the processing and *v)* publication of the campaign results. However, along all these stages, *Sensitive Personal Information* (SPI) can be conveyed by the mobile app and potentially be subject to attacks from adversaries, therefore motivating the development of a better privacy support.

We identify 4 categories of SPI that can be collected by crowdsourcing campaigns and might be exploited by adversaries:

Identifiers group all persistent or transient identifiers that can take the form of a device ID (IMEI) or Google account ID, for example, to explicitly identify a worker from the perspective of a mobile crowdsourcing system. However, such identifiers may directly name the worker or be used to perform context linking attacks by combining several measurements;

Point of Interest (POI) groups all the forms of geolocated data that can deliver some spatial information on the location of a worker. This includes GPS locations, but also places' check-in, cell tower ID or location, which can be used to produce maps from crowdsourced measurements. However, these POIs may also reveal the home, office, shopping and/or leisure locations of workers that can uniquely identify them [27, 13, 48];

Routines concern any information that can be used to capture a recurrent activity of a worker. This category of SPI covers in particular any form of timestamp, no matter the format and the precision. While this precious information often appears as harmless, it may also be used by context linking attacks to group crowdsourced data and observe correlation along time (*e.g.*, nights, week-ends);

Markers finally focus on information whose entropy in terms of values can be exploited to detect outlier workers and thus be indirectly used as an identifier by an adversary. There can be a wide diversity of such markers depending on the purpose of the mobile crowdsourcing system. For example, in the case of MobiPerf (a mobile app for measuring network performances) [40], the properties of device manufacturer, model, OS version and network carrier can be considered as unique if a worker uses some original/old mobile device. Beyond individual values, the combination of markers can also lead to the disclosure of a unique fingerprint, which can be used by an adversary to extract insightful information from a group of workers [6].

We believe that this classification is sufficient to identify sensitive personal information and support the privacy of workers. While an ontology could be defined to further organize the diversity of SPI along these 4 categories, the definition of such an ontology remains out of the scope of this article.

2.1. Location Privacy

2.1.1. Location Privacy Threats

When studying the location privacy threats in crowdsourced datasets, we consider that the adversary can exploit two dimensions of knowledge [77]: *spatio-temporal information* and *context information*.

In the context of mobile crowdsourcing systems, *spatio-temporal information* refers to the capability of the adversary to access a history of crowdsourced data—*i.e.*, several measurements reported by a single worker. In the case of a compromised storage server (or connection to the storage server), such an assumption holds as the adversary can get access to sufficiently large volume of crowdsourced data to build some spatio-temporal knowledge.

Beyond spatio-temporal information, *context information* refers to any additional information that an adversary can exploit. This covers embedded knowledge that is included in the crowdsourced dataset (*e.g.*, markers) or side knowledge that an adversary can obtain from other information sources (*e.g.*, the number of involved workers).

Several location privacy attacks exploit crowdsourced data to reveal the identity of workers. For example, *identity matching* [5] can be used to attack several pseudonyms of a worker. The adversary can link several pseudonyms based on equal or correlating attributes to the same identity, this way, the provided privacy of the obfuscated pseudonyms is no longer useful.

A *location tracking attack* [34] makes use of several location updates known to the adversary. For example, this attack can be used against randomly changing pseudonyms without using mix zones. Here, the adversary can correlate succeeding pseudonyms by linking spatial and temporal in-

formation of succeeding location reports even if an obfuscation mechanism is used. For instance, the adversary can try to reconstruct the trajectory of a worker based on the provided locations of several pseudonyms.

In a *maximum movement boundary attack* [31], the adversary computes the maximum movement boundary area, where the worker could have moved between two succeeding position reports. For example, the position of the first measurement uploaded at time t_1 helps the adversary to increase the precision of the measurement uploaded at t_2. In this example, only a small part of the area reported at t_2 is reachable within the maximum movement boundary. Therefore, the remaining area of the position update can be excluded by the adversary.

Then, *location distribution attacks* [53] observe that workers are not often distributed homogeneously in space so they can use outlier locations in sparsely populated areas to link and extract the crowdsourced data belonging to the same worker. In particular, spatial-temporal clustering algorithms, like ST-DBSCAN [7], can be used in such a case to group the worker's traces together. In the case of dense areas, an adversary can use the speed of the workers to group the locations related to a given trajectory [16]. However, the performance issue of such techniques must be taken into consideration [75].

Finally, given a dataset of workers' location traces with timestamps, *Mobility Markov Chain* (MMC) attack [27] reports on the workers favorite POIs with labels, such as home, work, or shopping. Then, by correlating these POIs with a map, it can reveal SPI about her work and her home, which combined with a public phone directory can deduce her full name and leave opportunity to query social networks, etc.

Most of these attacks are implemented within privacy evaluation tools [69, 49, 64, 11]. In particular, the *Location-Privacy and Mobility Meter* (LPM^2) tool [69] provides a reusable toolkit to evaluate location privacy.[2] LPM^2 uses known statistical methods (such as Bayesian inference, hidden Markov model, and Markov-Chain Monte-Carlo methods) to formalize and implement the location inference attacks that quantifies the location privacy of mobile users, given various *location-privacy preserving mechanisms*.

2.1.2. Location Privacy Protection Mechanisms (LPPMs)

LPPMs are an interesting mechanisms that can be used to limit user privacy leaks [12]. A large body of the related work has been devoted towards the latest stages of mobile crowdsourcing campaigns by improving the privacy properties of datasets once uploaded to remote servers [42, 58, 73, 17, 78, 63, 48]. These techniques contribute to preserving the privacy of workers while limiting the sanitization impact on the quality of the resulting dataset.

In particular, Gambs *et al.* [26] propose a toolkit called GEPETO to enhance the privacy of spatio-temporal datasets by applying a sanitization process. They provide a tool to sanitize spatio-temporal datasets and to measure the privacy and the utility trade-off of the resulting dataset.

Ma *et al.* [47] discuss some vulnerabilities found in mobility traces. In a formally detailed approach they explain how an adversary can identify workers using complementary metadata collected while observing workers.

Baik *et al.* [3] propose a new algorithm to achieve guaranteed anonymity in a spatio-temporal dataset; they focus on the home identification and user tracking privacy threats.

While all of these works focus on protecting the collected dataset, raw datasets stored on a remote server may be leaked through security breaches. Moreover, early stages of mobile crowdsourcing campaigns remain exposed to privacy attacks (*e.g.*, man-in-the-middle). This issue has been identified as an open challenge by Christin *et al.* [21]. Later, Christin *et al.* [20] have proposed a detailed approach that requires the involvement of participants in the anonymization process.

To ensure that workers will not upload any corrupted data, reputation systems [56, 54] have been proposed to build a trust between the collecting server and the workers, thus limiting the adversaries impact on the corrupted data that they want to upload to the server. In the crowdsourcing phase, Mousa *et al.* [55] introduce a new attack in crowdsourcing environments that are using reputation systems exploiting their attack, they can link all the data of one user without her identifier in the collected data. They propose a privacy-preserving reputation system, called PRIVASENSE, in which they integrate the reputation system while ensuring worker's anonymity against the reputation system itself. These works motivate the need to protect our users from the server (reputation system) as they show how they can link all the users' data. This highlights the need to protect the users from the collection server itself.

A lot of LPPMs have been proposed to anonymize the mobility traces. A major limitation of the existing LPPMs is that they are used on the server side and they require all the users' collected data as input, which forms a single point of failure from the privacy point of view, as our users do not have to trust the server. In our work, we use the most effective LPPMs and adapt them to work in mobile devices. We thus anonymize the data before they reach the server. In order to upload anonymized data to the server, we propose to send the data through other users (other than the data producer). Next, we will review some of the works that propose user collaboration to exchange data between users in a privacy friendly way.

2.1.3. Peer-to-Peer Collaborative Privacy-preserving

Some LPPMs provide collaborative privacy preserving methods which are particularly interesting in protecting the users' privacy in *Location-Based Services* (LBS) [62, 37, 46].

Some privacy protection mechanisms used in LBS can be adapted to mobile crowdsourcing platforms. As both domains share the same purpose from the location privacy view angle, which is hiding users' locations from the remote servers collecting users' locations. In LBS, users query the remote

[2] http://icapeople.epfl.ch/rshokri/lpm

Approach	Technique	LPPM	Communication
Show *et al.* [19] (2011)	Multi-Hop queries	Spatial cloaking	P2P protocols
Shokri *et al.* [67] (2011)	Shared buffer	MobiCrowd	WiFi *Access Point* connection
Shokri *et al.* [70] (2014)	Shared context	MobiCrowd	WiFi Direct
Peng *et al.* [59] (2017)	Fake queries	Spatial Cloaking	WiFi Direct

Table 1
Summary of related work using Peer-to-peer collaborative privacy-preserving solutions

server to get some context information about their locations. For example, the user must provide their location to a remote server to get the satellite image corresponding to their location. Most of the the proposed solutions would benefit from the nearby users to mix the user's location with theirs to confuse the adversary on the server side.

In Table 1 we consider the solutions where users collaborate to hide their locations data from the server [19, 69, 70, 59].

In particular, Show *et al.* [19] use spatial cloaking to hide users' locations from the LBS, the user's query passes through several hops before arriving to the server, providing some spatial cloaking on each hop, thus, they prevent the server from linking the query creator with the query itself, they use communication over *peer-to-peer* (P2P) protocols to disseminate the query over multiple hops to the server.

Shokri *et al.* [67] present a novel LPPM to hide the user location against LBS. They use a shared buffer to query the server only when the buffer lacks that information. A WiFi *Access Point* connection allows a collaboration between users. They need to change the LBS server architecture in order for their solution to work.

Shokri *et al.* [70] introduce a user-collaborative privacy-preserving approach for LBS. The users keep context information in a buffer and pass it to other peers. They only seek the LBS if the shared buffer does not contain the sought information. WiFi Direct communications are used to exchange the shared context between users. Their solution does not require changing the LBS server architecture nor involve third party servers.

Peng *et al.* [59] present a more complex scheme where the users obfuscate their actual trajectory by issuing fake queries to confuse the LBS. They propose to issue other users' queries along with the user real queries to confuse the LBS. WiFi Direct is used to exchange queries between users. They will need to issue a lot of fake queries to confuse the server, especially when they rely on just one location obfuscation LPPM.

Yet, such approaches are not widely adopted by LBS solutions as they fail to demonstrate their effectiveness and justify their cost in a realistic deployment. In particular, caching all the users' trajectories can take an important storage space in mobile devices, not to mention the delays added by such solutions. To these points, one can add the new threats evolving from including other adversaries in the scheme, like the *man-in-the-middle* (MITM), which has to be considered in each adversary model.

For all of the above reasons, we consider the following research work that proposes a cost effective and privacy preserving solution for peer-to-peer communication. In particular. Luxey *et al.* [45] propose a novel decentralized dissemination protocol that exploits opportunistic interactions between peer-to-peer users. Their solution called Sprinkler, a gossip-based protocol that enables an interaction between users while preserving their privacy. Bromberg *et al.* [9] present a peer-to-peer decentralized communication protocol called Cascade, to share the user's sessions between her devices in a seamlessly way while preserving her privacy. Aditya *et al.* [1] introduce EnCore, a peer-to-peer nearby communication for opportunistic encounters using Bluetooth communication to give the user the control over her privacy. Tsai *et al.* [74] propose enClosure, a peer-to-peer communication based on nearby encounters for mobile devices which ensures a privacy preserving communication between users. The above mentioned works provide a privacy preserving communications between users. The main intention of these work is either to reduce network costs or to explore the proximity of the users providing a secure and privacy preserving communication. In our work, we use the same opportunistic communication privacy preserving scheme, our intention is not to propose another communication scheme, but to extend existing communication schemes to disseminate crowdsourced data to help users hide from the collection server. Furthermore, in our work we anonymize the crowdsourced data on each encounter on their way to the server.

2.2. Mobile Crowdsourcing Platforms

Mobile crowdsourcing has gained a lot of attention for the last decade [38, 15], and this is due to the capabilities that offer these mobile phones. Scientists and decision makers can now run their experiments on the user mobile device to collect real life scenarios. The collected data can be of different types: geospatial data [38], behavioral data [41], environment monitoring [71, 50], Internet quality monitoring [33] and health data [65].

To help scientists, crowdsourcing platforms leverage all the technical kit to deploy their crowdsourcing collection campaign, to monitor the campaign and to store the collected data in the cloud servers.

In our work, we test our privacy preserving techniques on both APISENSE crowdsourcing platform and MobiPerf crowdsourcing mobile app:

2.2.1. APISENSE Platform

The crowdsourcing platform APISENSE [36] offers the scientists, with no technical background, the ability to ac-

Approach	Collaborative	Integration	Trust model
ANONYSENSE [22] (2008)	Yes	Library	Third-Party server
PEIR [57] (2009)	No	Platform	Server
PRISM [23] (2010)	No	Sandbox	Server
HP3 [39] (2010)	Yes	Platform	Third-Party server
SPEAR [32] (2014)	NO	Platform	Third-Party server

Table 2
Summary of related work in privacy of mobile crowdsourcing platforms

quire insightful data from the field. APISENSE manages the deployment of the dada acquisition campaigns as dedicated tasks described in JavaScript, which are remotely deployed to all the participants in real-time. APISENSE offers a generic mobile app that can be used by various scientists for different use cases. This mobile app is in charge of directly executing the deployed tasks on the participant's mobile device. The collected data comes from most of the sensors available on a mobile device, but the scientists can also ask the participant to perform a specific action whenever a given event happens, like answering a question whenever the participant comes back home.

The APISENSE platform provides a web interface to monitor and to control the crowdsourcing campaign. Through this web interface, the campaign manager (the scientist) can invite new users, update the data acquisition script and update it in real-time. The campaign manager can visualize the anonymized user data, as soon as they are uploaded to the server. Although, these received data do not contain any user identifiers, it needs to be further anonymized before starting to process the collected data.

2.2.2. MobiPerf App

MobiPerf is open source mobile app [40] for measuring network performances at regular intervals in the background. The MobiPerf mobile app use the open source library Mobilyzer [66] that facilitates the network measurement crowdsourcing campaigns. It contains most of the network measurement tools which can be used just by configuring a file that contains the measurement tasks (*e.g.*, *ping*, *traceroute*, *HTTP GET*) to be executed on the user mobile device. To use the MobiPerf mobile app, the scientists need to customize the app to meet their needs and to change the server address by their own server address.

2.2.3. Privacy-preserving Mobile Crowdsourcing Platforms

Mobile crowdsourcing platforms are used to collect data about the contributing users, these data include some *Sensitive Personal Information* (SPI), which introduce privacy issues. This is why researchers are actively working on privacy protection mechanisms for crowdsourcing platforms [8, 29].

Table 2 summarizes the work related to the privacy of mobile crowdsourcing platforms.

ANONYSENSE. Cornelius *et al.* [22] have proposed a mobile platform for opportunistic sensing called ANONYSENSE. Because the server hosting the collected dataset can be used to trace the worker's wireless access points, they use an anonymization network to hide the worker locations and they rely on a third-party server for routing the data. ANONYSENSE also supports reporting data with a statistical guarantee of k-anonymity. The workers' data are blurred and combined before being reported to the remote server.

PEIR. Mun *et al.* [57] introduce a platform for participatory sensing where they include an access control mechanism to let workers decide who can access to their data. In this approach, they control which sensors the mobile app has access to and who has access to the data on the server.

PRISM. Das *et al.* [23] present a *Platform for Remote Sensing using Smartphones* (PRISM). They use a sandbox environment to prevent mobile apps from accessing mobile sensors. Users participating in crowdsourcing campaigns will install the PRISM runtime and then the untrusted sensing app. This app can only fetch data after ensuring that the user cannot be identified using these data. PRISM acts thus as an intermediate to guarantee the user privacy preservation. As discussed in [23], both ANONYSENSE and PRISM suffer from similar privacy leaks as the mobile app collects data using local sensors made available by their mobile device, allowing data to be linked to the worker identifiers easily.

HP3. Hu *et al.* [39] present a collaborative privacy-preserving platform, which uses social networks to hide workers from the server. The user data will be uploaded through other devices (randomly chosen) before reaching the collection server, and the server thus cannot guess who is the data owner. In their approach, they rely on third-party servers (the social network) that can store all the exchanged locations along with workers identifiers.

SPEAR. Gisdakis *et al.* [32] introduce a privacy preserving architecture for crowdsourcing platforms. They address all the challenges of participatory sensing like security, privacy and worker incentives, as well as limiting the participation to legitimate workers and authenticate them using third-party servers.

These works use either a third-party server or propose a postoriori anonymization techniques. In our work, we have a different adversary model. As we do not trust the collection server or any third-party server, we apply a priori data anonymization scheme to hide workers from the crowdsourcing server.

2.3. Synthesis

Despite the prolific work that have been done to study and propose new data collection privacy preserving schemes, some of these approaches have become obsolete and require new privacy preserving schemes, especially with the new privacy attacks, the advance of the computing capabilities of the adversaries and the recent *(The EU General Data Protection Regulation)* legislation on handling user personal information. To the best of our knowledge, the state of the art fails to appropriately address the anonymization schemes along the earliest stages of a mobile crowdsourcing campaign in order to limit potential privacy threats, especially when the adversary model includes trusting a remote server. Therefore, in this article, we intend to address this limitation by proposing an a priori approach that uses privacy protection mechanisms on the mobile device level.

3. Privacy Threats in Mobile Crowdsourcing Systems

This section discusses the potential threats in mobile crowdsourcing systems along 2 axes: the *system model* and the *sensitive personal information*.

Mobile crowdsourcing system model. The architecture we consider is a mobile crowdsourcing campaign that involves three components, namely, *mobile devices*, *crowdsourcing apps*, and *storage servers*.

We consider that the mobile *crowdsourcing apps* can be trusted as we believe that the owner of the mobile crowdsourcing app or platform is interested in gathering insightful datasets with the consent of workers, especially if this mobile app is open sourced.

However, we consider that the *storage server* can be compromised and reveal some sensitive personal information on behalf of the owner and the workers. While crowdsourced datasets are expected to be anonymized prior to any online publication or reuse [12, 2], the potential threats we consider encompass a remote or physical access to the storage server to steal the raw crowdsourced dataset prior to the application of any anonymization technique. Another similar threat may involve a *man-in-the-middle* (MITM) attack to intercept the crowdsourced data while it is uploaded to the remote storage server. For example, no matter if they are deployed in the cloud or on-premise, the remote storage servers may suffer from security leaks that can be exploited by an adversary. Moreover, stakeholders who get the data before its anonymization can hold it without anonymizing it, knowing that anonymization will decrease data utility, or because they do not know exactly how to process the dataset. Thus, it may be stored on the server until they figure out how to anonymize it. Furthermore, storing the crowdsourced data on the server must comply with *The EU General Data Protection Regulation (GDPR)* and the *Privacy Act of 1974* of the USA. With crowdsourced data, it is difficult to comply with the regulations, for example: giving the users the right to delete their own data whenever they want. FOUGERE does resolve issues related to these regulations as it does not store personal identifiers on the server side and the users' data are anonymized before being uploaded to the server.

Sensitive personal information in mobile crowdsourcing. In Section **??** we considered 4 categories of Sensitive personal information (SPI): identifiers, Point of Interest (POI), routines and markers. These SPI can be collected by the mobile apps and can be subject to attacks from adversaries.

Existing mobile crowdsourcing systems may have collected some sensitive personal information. For example, the MOBIPERF mobile crowdsourcing app[3] publishes a privacy statement on the nature of data that is collected by the app:

> MOBIPERF regularly collects information about the network performance perceived from your device. This information includes the following:
> - Device properties:
> - Device manufacturer, model, OS, and OS version,
> - Google account ID, if the data is posted non-anonymously. You can choose to post anonymously, or not. Posting non-anonymously provides valuable data that can allow the MOBIPERF researchers to better understand networks and allows you to access your historical data.
> - Salted hash of device ID (*e.g.*, IMEI)
> - Coarse-grained Cell ID location information
> - Network properties:
> - Current network connection type (*e.g.*, HSPA or LTE)
> - Current carrier (*e.g.*, Verizon)
> - Current cell tower ID and signal strength
>
> We **DO NOT** collect any personally identifying information in addition to that listed above: no names, no private data from other apps, and so on.

While MOBIPERF claims that *data released to the public domain will not contain any identifying information*, some of these information chunks may contribute to indirectly identify the workers, like the cell tower ID or the carrier, or even some timestamp (not mentioned online but included in the source code[4]).

4. FOUGERE: Empowering Workers with LPPMs

To overcome the above privacy threats and strengthen the location privacy of workers, this chapter introduces FOUGERE, which allows the anonymization and dissemination of the workers' crowdsourced data across the network. This

[3] http://mobiperf.com

[4] https://github.com/Mobiperf/MobiPerf

section introduces the key design principles we adopted, a description of how crowdsourced data flows across multiple devices, as well as the core *Location Privacy Protection Mechanisms* (LPPMs) that are provided by FOUGERE.

Collaborating with apps & workers. In order to be trusted and gather a large crowd of workers, we assume that mobile crowdsourcing apps and platforms are doing their best to enforce privacy and security support. However, developers are not necessarily aware of privacy threats and implementing a comprehensive support for such a support might be time-consuming and error-prone. FOUGERE therefore offers mobile crowdsourcing apps the possibility to offload the management of the worker privacy settings and the data dissemination across the network, thus letting developers focus on the core business of the mobile app. By making FOUGERE available as a system service within the mobile device, a worker can configure her privacy settings for each installed mobile crowdsourcing app and decide upon the level of privacy she requires to be enforced for each of the apps integrating FOUGERE.

From the perspective of the mobile crowdsourcing app, FOUGERE mostly expects the developer to *i)* declare the SPI collected by the mobile app (*e.g.*, location, timestamp, identifier), *ii)* implement a data forwarding task in charge of connecting to the remote storage server and uploading the data, and *iii)* forward any crowdsourced data to the library, instead of uploading it directly to the server.

More specifically, FOUGERE offers the workers control over worker's privacy preferences, thus providing a preference panel to *i)* explore the list of mobile crowdsourcing apps and respective SPI, *ii)* monitor and control the volume of crowdsourced data reported by each app, and *iii)* configure the list of LPPMs to be enforced by a given mobile crowdsourcing app.

By following these principles, FOUGERE can collaborate with the mobile app and the worker to ensure the anonymization and the dissemination of crowdsourced data. Figure 2 overviews these principles and illustrates how a mobile app can disseminate crowdsourced data without and with FOUGERE. In particular, mobile crowdsourcing apps that do not fulfill the design principles—or do not integrate FOUGERE—will upload crowdsourced data directly to the remote server, thus exposing the workers to the privacy threats introduced in Section 3. By integrating FOUGERE, any mobile crowdsourcing app simply delegates the data dissemination to the library. FOUGERE enforces the worker's privacy settings and applies the appropriate LPPMs to the forwarded data. Such mechanisms include *privacy filters* (to discard the data), *privacy distortions* (to alter the data) and *privacy aggregation* (to group the data).

Enabling crowdsourced dissemination.

Algorithms 1, 2, 3 summarize our dissemination process, these three algorithms (SEND, ONDISCOVER, ONRECEIVE) are explained as follows:

1. **SEND** (Algorithm 1). If the crowdsourced data has not been discarded by one of the configured LPPMs, FOUGERE stores a message for dissemination that is composed of *i)* a *payload*, *ii)* a *configuration* of remote LPPMs, *iii)* a *bloom filter* of forwarder devices, and *iv)* a *time-to-live* (TTL) for the dissemination process. While the *payload* refers to the crowdsourced data, which has eventually been altered by the local LPPMs, the message also includes some *configuration* parameters for LPPMs that can be executed by remote instances of FOUGERE (*e.g.*, replacing the location of the source by the location of the forwarder). In order to avoid a given message to be forwarded by the same set of mobile devices, FOUGERE also includes a bloom filter that encodes the list of forwarder nodes, without discarding their identifiers. The *bloom filter* is a data structure that can tell us if the worker's *id* is present in the bloom filter or not, without discarding the ids of other workers. The *bloom filter* is configured with a false positive probability of 0.1 and a number of expected elements equals to the TTL. Finally, the message encloses a *TTL* to define the numbers of workers' hops requested by the worker to disseminate the message.

2. **ONDISCOVER** (Algorithm 2). This procedure is triggered by FOUGERE discovery service whenever other workers are detected in vicinity. FOUGERE filters out the known workers by querying the bloom filter, and randomly picks one candidate. FOUGERE updates the bloom filter with the worker's *id* so that worker will not receive the same data again. Once a data is forwarded, FOUGERE discards it from the forwarding queue.

3. **ONRECEIVE** (Algorithm 3). Upon receiving a forwarded data, a remote FOUGERE node eventually applies the LPPMs listed in the configuration of the data. If the TTL equals 0, then FOUGERE stores the payload in the uploading queue to be uploaded by the mobile crowdsourcing app to the remote storage server. Otherwise, FOUGERE decreases the TTL and stores the resulting data in the forwarding queue for further dissemination.

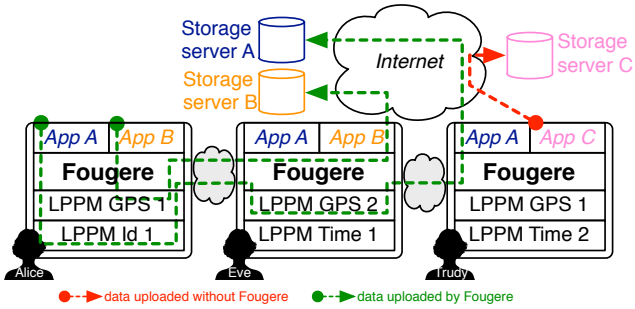

Figure 2: Overview of FOUGERE

The adoption of this dissemination scheme prevents server-side adversaries (including peers) to attack the forwarded

data in order to reveal some worker's SPI. In particular, by mixing the origins of crowdsourced data uploaded by a single device, FOUGERE confuses adversaries who would run attacks on a specific worker.

Algorithm 1 FOUGERE Send algorithm.

Ensure: $Q_{forward}$: Messages to be forwarded to devices.
  procedure SEND(app, $data$)
    for all $lppm_{app} \in$ LPPM(app) **do** ▷ applying local LPPM
      $data \leftarrow$ APPLY($lppm_{app}$, $data$)
      if $data = \emptyset$ **then**
        return ▷ $data$ discarded by LPPM
      end if
    end for
    $ttl \leftarrow$ TTL(app)
    if $ttl > 0$ **then** ▷ worker enabled the dissemination
      $cfg \leftarrow$ REMOTELPPM(app)
      $bloom \leftarrow$ BLOOMFILTER($this$)
      $Q_{forward} \leftarrow Q_{forward} \cup \langle data, cfg, bloom, ttl \rangle$
    else ▷ worker disabled the dissemination
      $Q_{upload} \leftarrow Q_{upload} \cup data$
    end if
  end procedure

Algorithm 2 FOUGERE OnDiscover algorithm.

Require: $Q_{forward}$: Messages to be forwarded to devices.
  procedure ONDISCOVER($peers$)
    for all $\langle data, cfg, bloom, ttl \rangle \in Q_{forward}$ **do**
      $candidates \leftarrow peers$
      repeat
        $candidate \leftarrow$ RANDOM($candidates$)
        if $candidate \notin bloom$ **then**
          $Q_{forward} \leftarrow Q_{forward} \setminus \langle data, cfg, bloom, ttl \rangle$
          $bloom \leftarrow bloom \cup candidate$
          FORWARD($candidate$, $\langle data, cfg, bloom, ttl \rangle$)
          break
        end if
        $candidates \leftarrow candidates \setminus candidate$
      until $candidates = \emptyset$
    end for
  end procedure

Mobile crowdsourcing apps share similarities with *Delay Tolerant Networks* (DTN) by considering that the crowdsourced data does not have to be immediately uploaded to the remote server and can tolerate delays ranging from minutes to hours. We exploit this property to adopt a multi-hop forwarding scheme in FOUGERE, which ensures that at least k neighboring devices with the same mobile app are also potentially collecting data in the same area, thus preventing the worker to be spotted as an outlier. This opportunistic dissemination schemes does not assume any specific network protocol and can be implemented atop of legacy discovery protocols, such as WiFi-Direct or Google Nearby connections.

Furthermore, FOUGERE complements existing privacy-preserving mechanisms, like TOR anonymity network [24] that prevents the server from tracing back the worker. Which

Algorithm 3 FOUGERE OnReceive algorithm.

Ensure: $Q_{forward}$: Messages to be forwarded to devices.
Ensure: Q_{upload}: Data to be uploaded by the app.
  procedure ONRECEIVE($data$, cfg, $bloom$, ttl)
    for all $lppm_{cfg} \in$ LPPM(cfg) **do** ▷ applying LPPM
      $data \leftarrow$ APPLY($lppm_{cfg}$, $data$)
      if $data = \emptyset$ **then**
        return ▷ $data$ discarded by LPPM
      end if
    end for
    $ttl \leftarrow$ TTL(m) $- 1$
    if $ttl > 0$ **then** ▷ $data$ to be sent to another device
      $Q_{forward} \leftarrow Q_{forward} \cup \langle data, cfg, bloom, ttl \rangle$
    else ▷ $data$ to be uploaded by the app
      $Q_{upload} \leftarrow Q_{upload} \cup data$
    end if
  end procedure

can also be used by FOUGERE to upload the crowdsourced data to the remote server. Using TOR, therefore, hides workers from the remote server, but it loses the physical proximity information that is useful for local LPPMs. For example, when an isolated worker is contributing from within the countryside, she can still report data using TOR but she will remain exposed to location privacy attacks. To further improve the privacy of the workers, TOR can be used after the data dissemination phase when the data is ready to be uploaded to the server, thus, the server will not be able to identify even if a worker is participating to the crowdsourcing campaign or not.

Controlling LPPMs from devices. In order to give the worker more control over her own data, FOUGERE includes several LPPMs that can be configured by the worker to decide upon the quality and the volume of crowdsourced data to be obfuscated. In particular, we consider 3 classes of LPPMs: *filters*, *distortions*, and *aggregations*, which can be implemented within a mobile device and used to obfuscate one of the SPI of the user.

Privacy Filters are a group of LPPMs that can decide autonomously if a crowdsourced data can be shared with the crowdsourcing platform or not. For example, a LocationFilter applies to *points of interests* and can be configured by the worker to define *white areas* or *black areas* that delimit zones where the mobile crowdsourcing app can or cannot collect data, respectively. Similarly, a TimeFilter rather applies on *routines* and is used with configured periods along which a mobile crowdsourcing app can or cannot collect data. Finally, a QuotaFilter is a more generic filter that can accept a worker-defined quota of crowdsourced data to be uploaded before discarding once this quota is reached.

Privacy Distortions are another class of LPPMs that can modify the value of an enclosed SPI in the crowdsourced data to be shared. For example, a IdentifierDistortion will change the value of an identifier at a given frequency (every request, hour, day), while a location distortion adds a controlled random noise to the worker's location (depending on radius r with a level of privacy that depends on r)

into the reported coordinates [25]. These distortions can be generalized to a wider set of SPI, such as *routines*, to reduce the accuracy of the data (*e.g.*, rounding the timestamp to the nearest hour) [19]. Based on this principle of noise injection, *samples* can also be generated by cloning a crowdsourced data and applying a distortion to one of the declared SPI. This results in the dissemination of several, almost similar, crowdsourced data samples that can be used by the worker to fuzz her location [76].

Privacy Aggregations reflect the last class of LPPMs that are supported by FOUGERE and propose to delay the dissemination of crowdsourced data by grouping them along a given criteria. For example, a TimeAggregation will group data per hour and apply an aggregation operator (like the average, the median, the min or the max) to the enclosed timestamp in order to report the same value for all the aggregated samples before reporting them. A MarkerAggregation is an example of remote LPPMs that will be encapsulated with the crowdsourced data and wait for a given marker (*e.g.*, the ISP name) to appear at least k times before being uploaded. This LPPM is an example of a distributed implementation of the k-anonymity algorithm [18, 72] that we can apply on a wide diversity of SPI, including GPS coordinates.

Summary. By combining an opportunistic dissemination scheme with worker-defined LPPMs, FOUGERE aims at leveraging the privacy properties of legacy mobile crowdsourcing apps and platforms. Before assessing the efficiency of FOUGERE, we now report on the implementation of these principles on the Android platform.

5. Implementation Details on Android

On Android, FOUGERE is packaged as an open source library that deploys system service within the mobile device of a worker. This system service currently builds on the Wi-Fi Direct network interface to exchange crowdsourced data between nearby devices of workers. It can be shared by multiple crowdsourcing apps of a given device to centralize the control of privacy settings, which are exposed to the worker as a dedicated preference panel. Thanks to its modular architecture, FOUGERE can be further extended with additional LPPMs, which are not covered by this work.

Application programming interface. Any mobile crowdsourcing app can integrate FOUGERE through a simple API that exposes the following operations:

- `hasFields(...)` is called by the mobile crowdsourcing app to declare any SPI as a `PrivacyField`, that refers the classes `IDENTIFIER`, `POI`, `ROUTINE`, and `MARKER`;
- `forward(...)` enlists a task in charge of uploading a crowdsourced data sample to the remote server when the TTL expires;
- `send(...)` delegates the dissemination of a crowdsourced data to FOUGERE.

All these operations are grouped within the interface `Fougere`, which is the facade used by a mobile app (cf. Listing 1). As introduced, in Section 3, FOUGERE provides a privacy support for 4 categories of SPI: `IDENTIFIER`, `POI`, `ROUTINE`, and `MARKER`.

Listing 1: FOUGERE API

```
public enum PrivacyFields {
  IDENTIFIER, POI, ROUTINE, MARKER;
}

public interface PrivacyField {
  PrivacyFields type();
}

public interface Fougere<D extends Serializable> {
  Fougere<D> hasField(PrivacyField f);
  void upload(Consumer<D> task);
  void send(D data);
}
```

Integrating FOUGERE within a mobile app requires to request 2 specific permissions: `fougere.permission.SHARE_DATA` and `fougere.permission.CONTROL_PRIVACY`, which are intended to inform the workers of the compatibility of the mobile app with FOUGERE and the possibility to adjust the privacy settings for this app.

Opportunistic dissemination. The current implementation of the FOUGERE dissemination module builds on the WiFi-Direct technology to discover nearby devices. When a mobile crowdsourcing app forwards a message, FOUGERE triggers the configured LPPMs and accumulates the data in the forwarding queue. For each data accumulated in the forwarding queue, FOUGERE picks a random peer that has never received this data and forwards it.

The second device receiving a connection request will receive the message `onReceive()` (cf. Algorithm 3). If the message reaches the configured number of device hops ($ttl = 0$), then the forwarded data is placed in an uploading queue, which will be emptied as soon as the remote mobile crowdsourcing app runs by invoking the upload handler registered by the app.

LPPM integration. FOUGERE combines the implementation of a decentralized dissemination scheme with the integration of LPPMs that can filter out data or alter its content depending on the worker's privacy settings. More generally, FOUGERE intends to leverage the integration of additional LPPMs to better control the data uploaded by any compatible crowdsourcing app. FOUGERE organizes these LPPMs along the 4 categories of SPI it supports. An LPPM complies to an interface `Lppm<T extends PrivacyField>` that declares the category `T` of SPI it considers and implements a method to apply a privacy mechanism on the uploaded data, which eventually returns the anonymized data to be further processed by FOUGERE.

To introspect and eventually modify the crowdsourced data sent by the mobile app, the Android implementation of FOUGERE uses the JXPATH library provided by Apache.[5]

[5] http://commons.apache.org/proper/commons-jxpath

This means that, using JXPATH, an LPPM can query the input data, changes its value and inject it back into the original crowdsourced data or discard it. Listing 2 provides simplified examples of location privacy filter (`WhiteAreaFilter`) and location privacy distortion (`LocationNoiseDistortion`) that are implemented in FOUGERE by using this LPPM API, according to the heuristics we introduced in Section 4.

Listing 2: LPPM API

```
public interface Lppm<T extends PrivacyField> {
  <D extends Serializable>
  Optional<D> apply(D data, T field);
}

public class WhiteAreaFilter implements Lppm<PoiField> {
  @Override
  public <D extends Serializable> Optional<D> apply(D data,
       PoiField field) {
    int lat = field.getLatitude(data), lon =
         field.getLongitude(data);
    return checkLocation(lat,lon)?
         Optional.of(data):Optional.empty();
} }

public class LocationNoiseDistortion implements
     Lppm<PoiField> {
  @Override
  public <D extends Serializable> Optional<D> apply(D data,
       PoiField field) {
    field.setLatitude(data, randomNoise(radius,
         field.getLatitude(data)));
    field.setLongitude(data, randomNoise(radius,
         field.getLongitude(data)));
    return Optional.of(data);
} }
```

In order to effectively apply the worker's privacy settings, FOUGERE operates by first applying the privacy filters, before proceeding with privacy distortions and finally privacy aggregations. In addition to that, privacy distortions and aggregations can also be triggered remotely to implement decentralized algorithms that build on neighboring samples to increase the privacy of workers [2].

5.1. Privacy Settings

FOUGERE aims at informing and supporting workers in the control of their privacy settings. In particular, FOUGERE provides a preference panel to be used by the worker to change her privacy settings (cf. Figure 3). Through this preference panel, a worker can list all the mobile crowdsourcing apps installed on her mobile device that requested to use FOUGERE. For each of the listed apps, a worker can add and configure one or several LPPMs she intends to apply on the crowdsourced data, depending on her acquaintance to share with some specific research centers, public institutes, or individuals [10]. To ease the configuration process, instead of requesting raw LPPMs configuration parameters, FOUGERE adopts a slider to configure the privacy level to be considered for a given LPPM and maps to identify POI (cf. Figure 3). Finally, the effects of these LPPMs are then depicted to the worker as analytics on the volume of crowdsourced data that has been produced/filtered/forwarded/uploaded by the mobile crowdsourcing app.

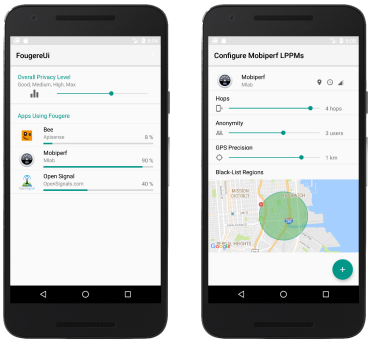

Figure 3: Preference Panel of FOUGERE

6. Evaluations of FOUGERE

In our evaluation, we want to answer the following research question:

RQ1: Can we increase the user's location privacy while maintaining a quality crowdsourced data?

RQ2: Can we put the user in center of her privacy control?

To effectively answer this research question, we propose the following evaluation protocol.

6.1. Evaluation Protocol

Beyond the challenges related to the integration in legacy mobile crowdsourcing systems, FOUGERE intends to deliver an efficient adoption of LPPMs in a decentralized context. The validation of such a capability requires consideration of a realistic deployment of mobile devices in order to assess the benefits of FOUGERE. Given that we are interested in providing a proof of feasibility for FOUGERE, we are not interested in simulating the behaviour of LPPMs, but rather in assessing the reference implementation of FOUGERE. However, testing mobile applications that make use of opportunistic communications is hard to achieve and reproduce with real mobile devices. We propose to deploy a cluster of emulated devices to reproduce the behavior of a crowd of workers who contribute to a mobile crowdsourcing campaign. We use mobility datasets that are publicly available to control the emulated devices and we collect their interactions to trace their actions *a posteriori*. The crowdsourced dataset collected on the remote server are evaluated by the LPM^2 toolkit [69] to evaluate the preservation of workers' privacy. By adopting such an empirical validation, we can evaluate real applications integrating FOUGERE and we can observe the impact of changing the parameters of FOUGERE (number of hops, LPPMs' specific parameters).

In the remainder of this section, we select the legacy MOBIPERF [40] mobile app as the mobile crowdsourcing app that we considered to assess FOUGERE.

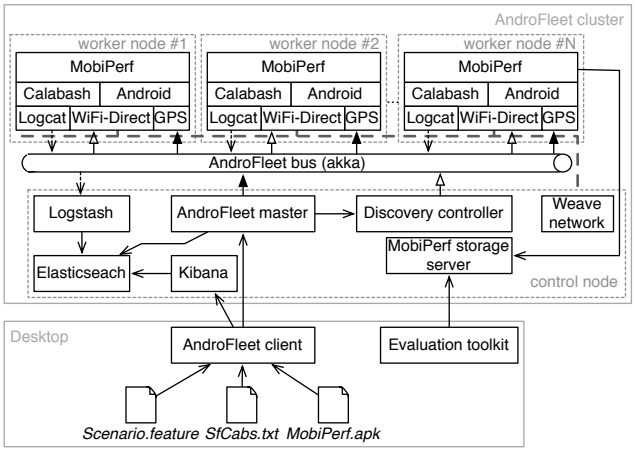

Figure 4: Overview of FOUGERE integration in the ANDROFLEET cluster

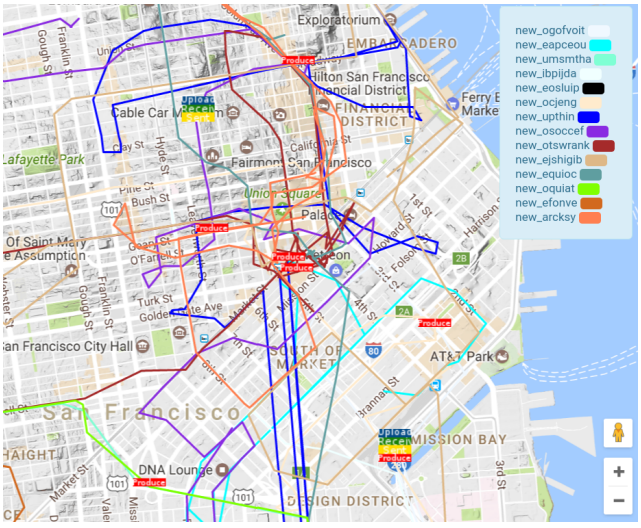

Figure 5: The ANDROFLEET log analytics interface

Emulating crowds of workers. The assessment of our opportunistic dissemination scheme and the associated LPPMs requires the consideration of a crowd of workers who installed a mobile crowdsourcing app that integrates FOUGERE. While running an emulator on a single machine is rather resource-consuming and cannot scale, we propose to consider the deployment of a cluster of servers to host multiple Android emulators. As Android emulators do not provide any support for ad hoc communications, such as WiFi-Direct, we use ANDROFLEET [51] to control the discovery of nearby peer-to-peer devices within a cluster of emulators.

In Figure 4, we show the integration of FOUGERE in the ANDROFLEET cluster. In particular, the AndroFleet master node takes as input an *app binary* (`MobiPerf.apk`), a *mobility dataset* (`SfCabs.txt`), and a set of *interaction scenarios* (`Scenario.feature`). It deploys the *app binary* and the set of *interaction scenarios* within each emulator and loads the *mobility dataset* into the Elasticsearch service. The ANDROFLEET master collaborates with the Discovery controller to control the virtual location of emulated devices. Each emulated device triggers a scenario autonomously and logs their actions via the Android logcat interface, which is automatically forwarded to Elasticseach using Logstash parser. Figure 5 reports on the map that can be obtained by querying the logs from Elasticseach using the Kibana UI service. In particular, this map depicts the path followed by crowdsourced messages from the time a crowdsourcing app delegates to FOUGERE to the time the data is effectively uploaded by the crowdsourcing app.

Controlling crowds of workers. To assess the efficiency of FOUGERE using ANDROFLEET, the emulated devices are required to be controlled in order to update their location and eventually internal state, to reproduce the mobility of a crowd of workers. While the choice of a mobility dataset can be challenging depending on the category of mobile crowdsourcing app, we used the widely used `epfl/mobility` dataset that is publicly available from CRAWDAD [60] to emulate 15 workers who are performing network measurements with the MOBIPERF mobile app. The crowdsourced dataset contains network measurements reported every 5 minutes by the workers moving in the San Francisco bay area.

Attacking crowdsourced datasets. To evaluate the impact of FOUGERE on the privacy of workers, we use the LPM^2 toolkit [69], which is a state-of-the-art tool for measuring location privacy. In particular, LPM^2 covers the evaluation of the LPPMs that are supported by FOUGERE, like the *privacy filters*, *privacy distortions* and *privacy aggregation*. To validate FOUGERE against privacy attacks, for each configuration, we run an experiment that follows these steps:

1. Run ANDROFLEET with MOBIPERF and FOUGERE (incl. privacy settings),
2. Assign tasks to workers during 3 days, and wait 4 more days for the data dissemination to complete,
3. Gather the logs of data exchanges between workers to evaluate the opportunistic dissemination scheme,
4. Retrieve all the raw crowdsourced data stored on the remote server,
5. Construct the adversary knowledge by tagging all the crowdsourced data of one worker (as required by LPM^2),
6. Evaluate the privacy support of FOUGERE with the LPM^2 toolkit,
7. Report on performance, utility, robustness and uncertainty, as proposed by Verykios *et al.* [75] to assess LPPMs.

6.2. Empirical Evaluation

In this section, we instantiate the above experimental protocol to assess FOUGERE as a practical support to improve the location privacy of workers.

Experimental Setup. Thanks to the ANDROFLEET [51, 52] emulation platform, we can reproduce the execution of a deployment of 15 mobile instances emulating a one-week

crowdsourcing campaign, thus proposing a realistic input dataset to evaluate FOUGERE. Then, we compare the behaviors of 6 configurations of the MOBIPERF app:

1- VANILLA refers to the reference implementation of the MOBIPERF Android app, as it can be downloaded from *http://www.mobiperf.com*. This configuration is used to demonstrate the vulnerability of legacy mobile crowdsourcing apps with regards to potential privacy threats. It is also used as a witness to evaluate the benefits of the other configurations including FOUGERE;

2- FOUGERE *with no LPPM* refers to the extension of MOBIPERF with the FOUGERE library. This configuration is used to isolate the properties of our opportunistic dissemination schemes independently of the impact of LPPMs. In particular, we consider the following worker configurations for the number of required hops to disseminate the crowdsourced data and the WiFi-Direct discovery scans: (a) $\langle 1\,hop, 5\,min\rangle$, (b) $\langle 4\,hops, 5\,min\rangle$ (default configuration), and (c) $\langle 4\,hops, 10\,min\rangle$;

3- FOUGERE *with LPPMs* refers to the FOUGERE library with the default configuration 2-b : 2 privacy distortions (*location noise* and *time noise*) and 1 privacy aggregation (*k-anonymity*), which are representative LPPMs used by the state-of-the-art. To configure these LPPMs, we consider 2 worker profiles, which are mapped to the following values:

   (a) *weak privacy profile* where location noise is set to $\langle 1, 0.1, 0.05\rangle$, thus reducing the location precision by 1 digit with a probability of 0.1 and possibly removing the location with a probability of 0.05. Time noise is set to $\langle 30, 0.1, 0.05\rangle$, thus reducing the time precision to half an hour with a probability of 0.1 and possibly removing the timestamp with a probability of 0.5, and finally k-anonymity is set to $\langle 2\rangle$, meaning that at least 2 samples should be produced in the same area to be forwarded;

   (b) *strong privacy profile* which is configured with location noise = $\langle 2, 0.2, 0.1\rangle$, time noise = $\langle 60, 0.2, 0.1\rangle$ and k-anonymity = $\langle 4\rangle$ as privacy settings.

None of these configurations includes a privacy filter, as these LPPMs are expected to be used to hide the living and working places of workers and the input dataset does not include this information. Furthermore, this article does not aim at evaluating the efficiency of individual LPPMs, but rather demonstrating the benefit of combining them in an open framework like FOUGERE.

Performance analysis. FOUGERE implements an opportunistic dissemination scheme to improve the privacy of workers. By doing so, FOUGERE exploits the physical proximity of workers to exchange crowdsourced data and to guarantee that the uploaded data has been forwarded along a number of hops requested by the worker. Figure 6 depicts the *time to converge* as a metrics to evaluate *i)* the impact of integrating FOUGERE on a legacy mobile crowdsourcing app like MOBIPERF, and *ii)* the effect of the number of hops and the WiFi-Direct discovery duration parameters. One can observe that, by using FOUGERE, not all the crowdsourced data is reported back to the remote storage server. This can be explained by the fact that some workers are contributing in sparsely populated areas, which prevents FOUGERE from disseminating the collected measurements. This result is actually a strength of FOUGERE as it automatically protects the isolated workers from adversaries who would apply some location distribution attacks to identify them.

Regarding the parameters of FOUGERE, one can note that the delay to upload data and the volume of reported data is more affected by the discovery duration than the number of hops required to upload the crowdsourced data. By increasing the delay of peer discovery, mobile devices miss some other workers in their vicinity in order to improve the time to converge. Therefore, we privilege the configuration 2-b (4 hops and 5 minutes) as the default configuration for FOUGERE. However, the worker remains free to adjust each of these parameters.

The *traveling distance* is another interesting metrics to evaluate the efficiency of the dissemination process and the relevance of peer-to-peer communications. Increasing this data traveling distance with FOUGERE contributes to better shuffle crowdsourced data produced by a crowd of workers. Figure 7 reports on this distance traveled by the crowdsourced data before being uploaded back to the remote storage server. In particular, the default configuration of FOUGERE maximizes the traveled distance with 20 % of data that traveled at least 10 km (6.2 $miles$), thus ensuring that the data was conveyed by FOUGERE as far as possible from the location where it has been produced.

Utility analysis. While FOUGERE aims at improving the location privacy of workers, the utility of the resulting dataset should not be neglected. The data utility is calculated using the similarity provided by the LPM^2 toolkit. Then, the data utility corresponds to the number of similar locations between the locations of data received on the server and the locations of the original raw data before anonymization.

Figure 8 reports on the tradeoff between utility and the anonymity of the configurations we considered. While the vanilla configuration (1) offers the highest utility with no anonymity, one can observe that the integration of FOUGERE seriously improves the anonymity of workers without seriously impacting the utility of the resulting dataset. As mentioned in Figure 6, the loss of 20 % utility is mainly due to crowdsourced data in sparsely populated areas that were retained by FOUGERE. Furthermore, adding some LPPMs (configurations 3-a and 3-b) strongly increase the anonymity of workers.

Interestingly, one can observe that the *weak privacy profile* offers a good privacy/anonymity tradeoff compared to the *strong privacy profile*, which reduces the dataset utility without bringing any further improvement over anonymity.

Robustness analysis. Regarding the effective privacy support offered by FOUGERE, we used the LPM^2 toolkit to evaluate the robustness of crowdsourced datasets that are uploaded through FOUGERE, this metric correspond to the LPM^2

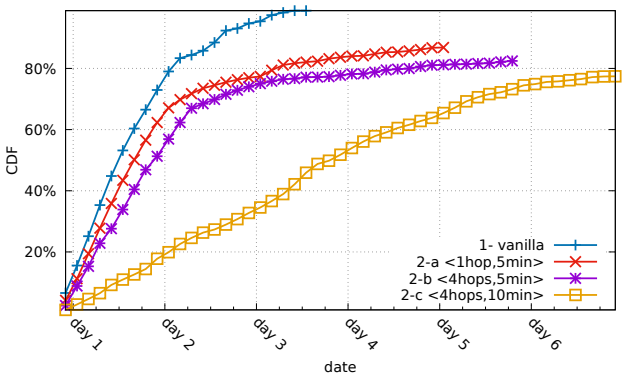

Figure 6: Measurements' time to converge

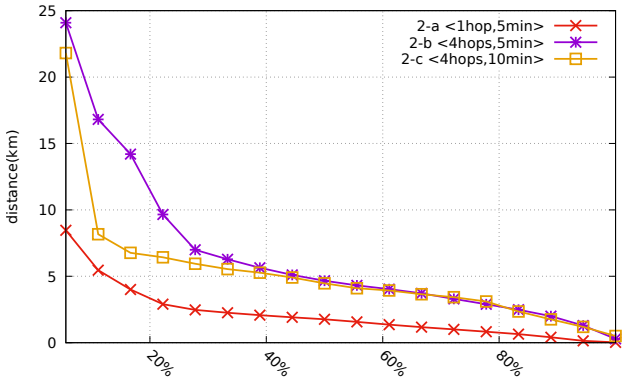

Figure 7: Distance traveled by measurements

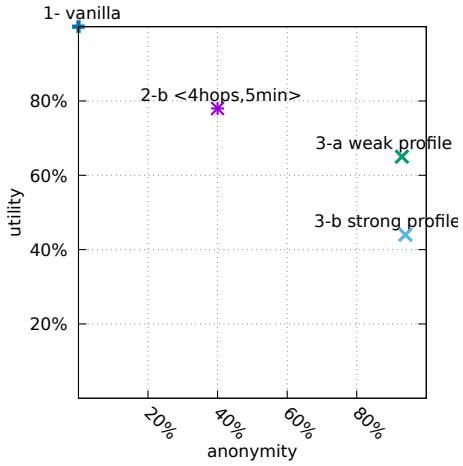

Figure 8: Dataset utility

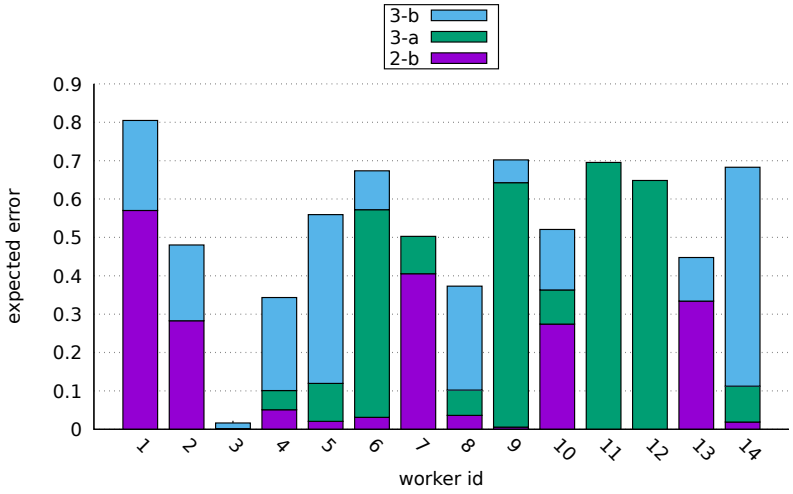

Figure 9: Robustness against location privacy attacks

toolkit's distortion metric[6] [68], which measures the location-privacy of the worker at each time instance. We randomly select the crowdsourced data reported by one of the workers as the adversary knowledge required by LPM^2 to apply location privacy attacks and we depict in Figure 9 the reported robustness for 14 workers. While LPM^2 successfully defeats worker 3 (used as the adversary knowledge), the other 14 workers clearly benefit from the integration of FOUGERE. In particular, we can observe that the integration of LPPMs complements efficiently our opportunistic dissemination scheme by supporting workers who are not located in a dense area and by offering similar privacy guarantees. Successfully location privacy attacks requires to combine different strategies to cope with the profile of workers.

While FOUGERE offers the worker the possibility to manually adjust her privacy settings, one of the perspectives of this work consists in leveraging this configuration process by delivering privacy risk feedback that would guide her settings accordingly. By recommending the privacy settings of FOUGERE, we aim at maximizing the individual privacy of workers, while preserving the overall utility of the crowdsourced dataset (cf. Figure 8).

Uncertainty analysis. Finally, the uncertainty metric evaluates the uncertainty that our LPPMs introduce to the adversary when she tries to reconstruct the hidden information from the crowdsourced dataset on the server. Figure 10 reports on the uncertainty metrics computed by LPM^2 [69]. One can observe that FOUGERE succeeds to increase the uncertainty of adversaries when it combines the opportunistic dissemination scheme with LPPMs, which confirms our previous observation. Furthermore, it also assesses that adopting a *weak privacy profile* already brings a reasonable level of privacy that puts adversaries in difficulties.

Overhead analysis. To analyze the overhead induced by our data dissemination process, we report in Table 3 the statistics related to an experiment involving 500 emulated workers for 24 hours. Along the experiment, the workers adopt the default configuration of FOUGERE $\langle 4\,hops, 5\,min\rangle$ (2-b). The overhead per user and at the scale of the crowd does not ex-

[6] https://github.com/rzshokri/quantifying_location_privacy

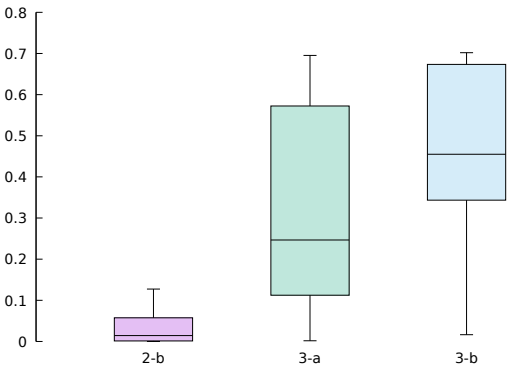

Figure 10: Adversary uncertainty

Table 3
Overhead analysis for 500 workers

Indicator	Value
Crowdsourced data size	29,712
Exchanged messages	113,785
Contributions per user	59
Messages forwarded per user	227
Detected neighbors	1,730,827
Established connections	127,545
Isolated users	8

ceed 4 times the initial volume of contributions. FOUGERE also discards 8 users considered as isolated and thus identifiable by tools like LPM[2].

The increased battery usage regarding the use of the WiFi-Direct following several discovery parameters has been reported in ANDROFLEET [51, 52], in FOUGERE we have used 5 *min* and 10 *min* discovery periods which are respectively responsible for $\simeq$ 15% and $\simeq$ 11% decrease in battery life for a *Galaxy Tab A 2016* mobile device running on *Android 5.1.1* OS. These battery usage values can be acceptable given the provided anonymity for the users.

Synthesis. To answer **RQ1**, given the preserved privacy for our workers and the obtained data utility we had for our crowdsourced data using FOUGERE. We can say that using FOUGERE, we can increase the user's privacy while maintaining a quality crowdsourced data. While we have introduced an overhead for the data dissemination, we believe that this overhead is acceptable tradeoff to preserve the users' privacy while maintaining the data utility.

To answer **RQ2**, in our experiment we have used different privacy settings, given that these privacy settings are set from the user's mobile device (Figure 3), we can say that the user is the center of control of her privacy.

7. Threats to Validity

This section analyzes the factors that may threaten the validity of our results.

Internal validity concerns the relation between theory and observations. In this work, they could be due to measurement errors reported during the experimentation. That is the reason why we did several experiments and we tried to reduce as much as possible external factors as explained in our experimental protocol in Section 6.2. We also performed our experiments on a crowd of emulated devices equipped with real mobile apps, instead of a simulation, to reduce the threats that could be due to an integration of the proposed approach in a real mobile crowdsourcing app or platform.

External validity relates to the possibility to generalize our findings. We believe that further validations should be done on different mobile crowdsourcing apps and with different configurations to broaden our understanding of the impact of LPPMs on the privacy of workers. Thus, we are not assuming that our results can be used to generalize the impact of a specific LPPM on privacy. However, we believe that this work contributes to prove that there is a clear positive impact for the privacy threats we considered.

Reliability validity focuses on the possibility of replicating our experiments and results. We attempt to provide all the necessary details to replicate our study and our analysis. Furthermore, the reference implementation of FOUGERE, the input datasets, case studies and testing environment are made available online to leverage its reproduction by the research community.

Construct validity has been covered by considering the convergent validity of privacy and utility properties. We observed that these two properties are related in practice, as the application of LPPMs tends to decrease the utility of the crowdsourced dataset. This observation calls for the identification of a privacy and utility trade-off in the context of mobile crowdsourcing systems, as acknowledged by [12].

Conclusion validity refers to the correctness of the conclusions reached in this work. The empirical evaluation we reported confirms our initial assumption that *a priori* anonymization techniques can be used to leverage the privacy of workers. We were also careful with our conclusion with regards to the impact on the utility of crowdsourced dataset.

8. Conclusion

Mobile crowdsourcing apps and platforms are more and more challenged to protect their workers' privacy. To address this challenge, we have introduced FOUGERE to increase worker's privacy in mobile crowdsourcing systems. FOUGERE operates a system-level service that collaborates with a mobile crowdsourcing app to declare SPI and delegate the dissemination of crowdsourced data by leveraging the physical proximity of workers. This opportunistic dissemination scheme is complemented by the integration of LPPMs that can be configured by the workers, independently of the installed mobile crowdsourcing apps.

Finally, we consider the deployment of FOUGERE in a realistic Android environment by emulating a crowd of 15

mobile devices hosting different versions of MOBIPERF and FOUGERE to assess our contribution. We show that FOUGERE succeeds in improving the workers' privacy by defeating location privacy attacks implemented by the LPM[2] toolkit.

References

[1] Aditya, P., Erdélyi, V., Lentz, M., Shi, E., Bhattacharjee, B., Druschel, P., 2014. Encore: Private, context-based communication for mobile social apps, in: Proceedings of the 12th annual international conference on Mobile systems, applications, and services, ACM. pp. 135–148.

[2] Andrés, M.E., Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C., 2013. Geo-indistinguishability: differential privacy for location-based systems, in: Proc. of CCS'13, pp. 901–914.

[3] Baik Hoh, Gruteser, M., Hui Xiong, Alrabady, A., 2010. Achieving Guaranteed Anonymity in GPS Traces via Uncertainty-Aware Path Cloaking. IEEE Transactions on Mobile Computing 9, 1089–1107. doi:10.1109/TMC.2010.62.

[4] Balan, R.K., Misra, A., Lee, Y., 2014. LiveLabs: Building An In-Situ Real-Time Mobile Experimentation Testbed, in: ACM HotMobile, ACM Press, New York, New York, USA. pp. 0–5. doi:10.1145/2565585.2565597.

[5] Beresford, A.R., Stajano, F., 2004. Mix zones: User privacy in location-aware services, in: Proc. of Percom'04.

[6] Bettini, C., Wang, X.S., Jajodia, S., 2005. Protecting Privacy Against Location-based Personal Identification, in: Proc. of VLDB'05. doi:10.1007/11552338_13.

[7] Birant, D., Kut, A., 2007. ST-DBSCAN: An algorithm for clustering spatial-temporal data. Data & Knowledge Engineering 60.

[8] Boutsis, I., Kalogeraki, V., 2016. Location privacy for crowdsourcing applications, in: Proc of. UbiComp'16.

[9] Bromberg, Y.D., Luxey, A., Taïani, F., 2018. Cascade: Reliable distributed session handoff for continuous interaction across devices, in: 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), IEEE. pp. 244–254.

[10] Brush, A., Krumm, J., 2010. Exploring end user preferences for location obfuscation, location-based services, and the value of location, in: Proc. of Ubicomp'10.

[11] Cerf, S., Mokhtar, S.B., Bouchenak, S., Marchand, N., Robu, B., 2018a. Dynamic modeling of location privacy protection mechanisms, in: IFIP International Conference on Distributed Applications and Interoperable Systems, Springer. pp. 26–39.

[12] Cerf, S., Primault, V., Boutet, A., Mokhtar, S.B., Birke, R., Bouchenak, S., Chen, L.Y., Marchand, N., Robu, B., 2017. PULP: Achieving Privacy and Utility Trade-off in User Mobility Data, in: Proc. of SRDS'17.

[13] Cerf, S., Robu, B., Marchand, N., Mokhtar, S.B., Bouchenak, S., 2018b. A control-theoretic approach for location privacy in mobile applications, in: 2018 IEEE Conference on Control Technology and Applications (CCTA), IEEE. pp. 1488–1493.

[14] Chatzimilioudis, G., Konstantinidis, A., Laoudias, C., Zeinalipour-Yazti, D., 2012a. Crowdsourcing with smartphones. IEEE Internet Computing 16.

[15] Chatzimilioudis, G., Konstantinidis, A., Laoudias, C., Zeinalipour-Yazti, D., 2012b. Crowdsourcing with smartphones. IEEE Internet Computing 16, 36–44.

[16] Chau, M., Cheng, R., Kao, B., Ng, J., 2006. Uncertain data mining: An example in clustering location data, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer.

[17] Chen, R., Fung, B.C.M., Mohammed, N., Desai, B.C., Wang, K., 2013. Privacy-preserving trajectory data publishing by local suppression. Information Sciences 231.

[18] Chow, C.Y., Mokbel, M.F., Liu, X., 2006. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. Proc. of ACM SIGSPATIAL .

[19] Chow, C.Y., Mokbel, M.F., Liu, X., 2011. Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. GeoInformatica 15.

[20] Christin, D., Bub, D.M., Moerov, A., Kasem-Madani, S., 2015. A distributed privacy-preserving mechanism for mobile urban sensing applications, in: Proc of. ISSNIP'15.

[21] Christin, D., Reinhardt, A., Kanhere, S.S., Hollick, M., 2011. A survey on privacy in mobile participatory sensing applications. Journal of Systems and Software 84, 1928–1946. doi:10.1016/j.jss.2011.06.073.

[22] Cornelius, C., Kapadia, A., Kotz, D., Peebles, D., Shin, M., Triandopoulos, N., 2008. Anonysense: privacy-aware people-centric sensing. Mobisys08 Proceedings of the Sixth International Conference on Mobile Systems Applications and Services , 211–224doi:10.1145/1378600.1378624.

[23] Das, T., Mohan, P., Padmanabhan, V.N., Ramjee, R., Sharma, A., 2010. PRISM: Platform for Remote Sensing using Smartphones. Proc. of MobiSys'10 .

[24] Dingledine, R., Mathewson, N., Syverson, P., 2004. Tor: The second-generation onion router. SSYM'04 Proceedings of the 13th conference on USENIX Security Symposium 13, 21. doi:10.1.1.4.6896.

[25] Fawaz, K., Shin, K.G., 2014. Location privacy protection for smartphone users, in: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ACM. pp. 239–250.

[26] Gambs, S., Killijian, M.O., Cortez, M.N.d.P., 2010. GEPETO: A GEoPrivacy-Enhancing TOolkit, in: Proc. of AINA Workshops'10.

[27] Gambs, S., Killijian, M.O., Del Prado Cortez, M.N., 2012. Next place prediction using mobility Markov chains, in: Proceedings of the First Workshop on Measurement Privacy and Mobility MPM 2012, ACM Press, New York, New York, USA. pp. 1–6. doi:10.1145/2181196.2181199.

[28] Gambs, S., Killijian, M.O., Núñez del Prado Cortez, M., 2014. De-anonymization attack on geolocated data. Journal of Computer and System Sciences 80, 1597–1614. doi:10.1016/j.jcss.2014.04.024.

[29] Gao, S., Ma, J., Shi, W., Zhan, G., Sun, C., 2013. TrPF: A trajectory privacy-preserving framework for participatory sensing. IEEE Transactions on Information Forensics and Security 8.

[30] Garfinkel, S.L., 2008. IRBs and security research: myths, facts and mission creep. Proceedings of the 1st Conference on Usability, Psychology, and Security , 1–5.

[31] Ghinita, G., Damiani, M.L., Silvestri, C., Bertino, E., 2009. Preventing velocity-based linkage attacks in location-aware applications, in: Proc. of ACM SIGSPATIAL'09. doi:10.1145/1653771.1653807.

[32] Gisdakis, S., Giannetsos, T., Papadimitratos, P., 2014. SPPEAR: Security & Privacy- Preserving Architecture for Participatory-Sensing Applications, in: Proc of. WiSec'14.

[33] Gregori, E., Lenzini, L., Luconi, V., Vecchio, A., 2013. Sensing the internet through crowdsourcing, in: 2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), IEEE. pp. 248–254.

[34] Gruteser, M., Grunwald, D., 2003. Anonymous usage of location-based services through spatial and temporal cloaking, in: Proc. of MobiSys'03. doi:10.1145/1066116.1189037.

[35] Haderer, N., Rouvoy, R., Seinturier, L., 2013a. A preliminary investigation of user incentives to leverage crowdsensing activities. 2013 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2013 , 199–204doi:10.1109/PerComW.2013.6529481.

[36] Haderer, N., Rouvoy, R., Seinturier, L., 2013b. Dynamic deployment of sensing experiments in the wild using smartphones, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer. pp. 43–56. doi:10.1007/978-3-642-38541-4_4.

[37] Haitao, Z., Lei, Z., Weimiao, F., Chunguang, M., 2016. A users collaborative scheme for location and query privacy, in: 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS), IEEE. pp. 383–390.

[38] Heipke, C., 2010. Crowdsourcing geospatial data. ISPRS Journal of Photogrammetry and Remote Sensing 65, 550–557.

[39] Hu, L., Shahabi, C., 2010. Privacy assurance in mobile sensing networks: Go beyond trusted servers, in: 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM Workshops 2010, IEEE. pp. 613–619. doi:10.1109/PERCOMW.2010.5470509.

[40] Huang, J., Chen, C., Pei, Y., Wang, Z., Qian, Z., Qian, F., Tiwana, B., Xu, Q., Mao, Z., Zhang, M., Others, 2011. Mobiperf: Mobile network measurement system. Technical Report. University of Michigan and Microsoft Research .

[41] Kazai, G., Kamps, J., Milic-Frayling, N., 2011. Worker types and personality traits in crowdsourcing relevance labels, in: Proceedings of the 20th ACM international conference on Information and knowledge management, ACM. pp. 1941–1944.

[42] Kifer, D., 2006. l-Diversity : Privacy Beyond k -Anonymity. Proceedings of the 22nd International Conference on Data Engineering 1, 1–36. doi:10.1145/1217299.1217302.

[43] Krumm, J., 2007. Inference Attacks on Location Tracks. Pervasive Computing 10, 127–143. doi:10.1007/978-3-540-72037-9_8.

[44] Lin, J., Sadeh, N., Amini, S., Lindqvist, J., Hong, J.I., Zhang, J., 2012. Expectation and purpose: understanding users' mental models of mobile app privacy through crowdsourcing, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12, p. 501. doi:10.1145/2370216.2370290.

[45] Luxey, A., Bromberg, Y.D., Costa, F.M., Lima, V., da Rocha, R.C., Taïani, F., 2018. Sprinkler: A probabilistic dissemination protocol to provide fluid user interaction in multi-device ecosystems, in: 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE. pp. 1–10.

[46] Ma, C., Zhang, L., Yang, S., Zheng, X., 2017. Hiding yourself behind collaborative users when using continuous location-based services. Journal of Circuits, Systems and Computers 26, 1750119.

[47] Ma, C.Y., Yau, D.K., Yip, N.K., Rao, N.S., 2010. Privacy vulnerability of published anonymous mobility traces, in: Proc. of MobiCom'10.

[48] Maouche, M., Ben Mokhtar, S., Bouchenak, S., 2018. Hmc: Robust privacy protection of mobility data against multiple re-identification attacks. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2, 124.

[49] Maouche, M., Mokhtar, S.B., Bouchenak, S., 2017. Ap-attack: a novel user re-identification attack on mobility datasets, in: Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, ACM. pp. 48–57.

[50] Martí, I.G., Rodríguez, L.E., Benedito, M., Trilles, S., Beltrán, A., Díaz, L., Huerta, J., 2012. Mobile application for noise pollution monitoring through gamification techniques, in: International Conference on Entertainment Computing, Springer. pp. 562–571.

[51] Meftah, L., Gomez, M., Rouvoy, R., Chrisment, I., 2017. ANDROFLEET: Testing WiFi Peer-to-Peer Mobile Apps in the Large. Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering , 961–966doi:10.1109/ASE.2017.8115712.

[52] Meftah, L., Rouvoy, R., Chrisment, I., 2019. Testing Nearby Peer-to-Peer Mobile Apps at Large, in: Poshyvanyk, D., Malavolta, I. (Eds.), MOBILESoft 2019 - 6th IEEE/ACM International Conference on Mobile Software Engineering and Systems, Montréal, Canada. URL: https://hal.inria.fr/hal-02059088.

[53] Mokbel, M.F., 2007. Privacy in location-based services: State-of-the-art and research directions, in: Proc. of MDM'07. doi:10.1109/MDM.2007.45.

[54] Mousa, H., Benmokhtar, S., Hasan, O., Brunie, L., Younes, O., Hadhoud, M., 2017a. A reputation system resilient against colluding and malicious adversaries in mobile participatory sensing applications, in: 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE. pp. 829–834.

[55] Mousa, H., Mokhtar, S.B., Hasan, O., Brunie, L., Younes, O., Hadhoud, M., 2017b. Privasense: Privacy-preserving and reputation-aware mobile participatory sensing, in: Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, ACM. pp. 38–47.

[56] Mousa, H., Mokhtar, S.B., Hasan, O., Younes, O., Hadhoud, M., Brunie, L., 2015. Trust management and reputation systems in mobile participatory sensing applications: A survey. Computer Networks 90, 49–73.

[57] Mun, M.U., Reddy, S.U., Shilton, K., Yau, N., 2009. PEIR, the personal environmental impact report, as a platform for participatory sensing systems research. MobiSys .

[58] Ninghui, L., Tiancheng, L., Venkatasubramanian, S., 2007. t-Closeness: Privacy beyond k-anonymity and l-diversity, in: Proceedings - International Conference on Data Engineering, IEEE. pp. 106–115. doi:10.1109/ICDE.2007.367856.

[59] Peng, T., Liu, Q., Meng, D., Wang, G., 2017. Collaborative trajectory privacy preserving scheme in location-based services. Information Sciences 387.

[60] Piorkowski, M., Sarafijanovic-Djukic, N., Grossglauser, M., 2009. {CRAWDAD} dataset epfl/mobility (v. 2009-02-24). Downloaded from \url{anonymous}. doi:10.15783/C7J010.

[61] Prandi, C., Salomoni, P., Mirri, S., 2014. mPASS: Integrating People Sensing and Crowdsourcing to Map Urban Accessibility. Proc. of CCNC'14 .

[62] Primault, V., Boutet, A., Mokhtar, S.B., Brunie, L., 2016. Adaptive location privacy with alp, in: 2016 IEEE 35th Symposium on Reliable Distributed Systems (SRDS), IEEE. pp. 269–278.

[63] Primault, V., Boutet, A., Mokhtar, S.B., Brunie, L., 2018a. The long road to computational location privacy: A survey. IEEE Communications Surveys & Tutorials .

[64] Primault, V., Maouche, M., Boutet, A., Mokhtar, S.B., Bouchenak, S., Brunie, L., 2018b. Accio: How to make location privacy experimentation open and easy, in: 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), IEEE. pp. 896–906.

[65] Ranard, B.L., Ha, Y.P., Meisel, Z.F., Asch, D.A., Hill, S.S., Becker, L.B., Seymour, A.K., Merchant, R.M., 2014. Crowdsourcing—harnessing the masses to advance health and medicine, a systematic review. Journal of general internal medicine 29, 187–203.

[66] Rosen, S., Yao, H., Nikravesh, A., Jia, Y., Choffnes, D., Mao, Z.M., 2014. Mapping global mobile performance trends with mobilyzer and mobiperf, in: Proceedings of the 12th annual international conference on Mobile systems, applications, and services, ACM. pp. 353–353.

[67] Shokri, R., Papadimitratos, P., Theodorakopoulos, G., Hubaux, J.P., 2011a. Collaborative location privacy, in: Proceedings - 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS 2011, IEEE. pp. 500–509. doi:10.1109/MASS.2011.55.

[68] Shokri, R., Theodorakopoulos, G., Danezis, G., Hubaux, J.P., Le Boudec, J.Y., 2011b. Quantifying location privacy: the case of sporadic location exposure, in: International Symposium on Privacy Enhancing Technologies Symposium, Springer. pp. 57–76.

[69] Shokri, R., Theodorakopoulos, G., Le Boudec, J.Y., Hubaux, J.P., 2011c. Quantifying location privacy, in: Proc. of S&P'11.

[70] Shokri, R., Theodorakopoulos, G., Papadimitratos, P., Kazemi, E., Hubaux, J.P., 2014. Hiding in the mobile crowd: Location privacy through collaboration. IEEE Transactions on Dependable and Secure Computing 11, 266–279. doi:10.1109/TDSC.2013.57.

[71] Stevens, M., D'Hondt, E., 2010. Crowdsourcing of pollution data using smartphones, in: Workshop on Ubiquitous Crowdsourcing, pp. 1–4.

[72] Sweeney, L., 2002. k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY. International Journal on Uncertainty 10, 557–570. doi:10.1142/S0218488502001648, arXiv:10(5),2002;555-570.

[73] Terrovitis, M., Mamoulis, N., 2008. Privacy preservation in the publication of trajectories. Proceedings - IEEE International Conference on Mobile Data Management , 65–72doi:10.1109/MDM.2008.29.

[74] Tsai, L., De Viti, R., Lentz, M., Saroiu, S., Bhattacharjee, B., Druschel, P., 2019. enclosure: Group communication via encounter closures, in: Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, ACM. pp. 353–365.

[75] Verykios, V.S., Bertino, E., Fovino, I.N., Provenza, L.P., Saygin, Y.,

Theodoridis, Y., 2004. State-of-the-art in privacy preserving data mining. ACM SIGMOD Record 33.

[76] Wang, H., Su, H., Zheng, K., Sadiq, S., Zhou, X., 2013. An Effectiveness Study on Trajectory Similarity Measures. Proceedings of the Twenty-Fourth Australasian Database Conference-Volume 137 137, 13–22.

[77] Wernke, M., Skvortsov, P., Dürr, F., Rothermel, K., 2014. A classification of location privacy attacks and approaches. Personal Ubiquitous Comput. 18. doi:10.1007/s00779-012-0633-z.

[78] Zan, B., Sun, Z., Gruteser, M., Ban, X., 2013. Linking anonymous location traces through driving characteristics , 293–300.