Privacy Protection for Wi-Fi Location Positioning Systems

Antoine Boutet, Mathieu Cunche

Privacy Protection for Wi-Fi Location Positioning Systems

Antoine Boutet
Univ Lyon, INSA Lyon, Inria, CITI, F-69621
VILLEURBANNE, France
antoine.boutet@insa-lyon.fr

Mathieu Cunche
Univ Lyon, INSA Lyon, Inria, CITI, F-69621
VILLEURBANNE, France
mathieu.cunche@insa-lyon.fr

ABSTRACT

With the democratization of mobile devices embedding different positioning capabilities, location information is used for a variety of applications. On mobile devices, the geolocation can be obtained via GPS or by leveraging surrounding network infrastructure such as Wi-Fi access points. Despite a lower accuracy, Wi-Fi based geolocation has several advantages over GPS such as reduced energy consumption and availability in indoor environments. To enable this network-based geolocation, mobile devices need to interact with a location positioning system that will resolve a list of visible Wi-Fi access points into a position. By doing so, mobile users are revealing their mobility to the location provider, potentially exposing sensitive information to an untrusted third-party.

In this paper, we propose a novel solution to preserve users' privacy when requesting users' location from Wi-Fi while supporting high utility. The key idea behind our online approach is to combine a *caching strategy* (for limiting the exposure of the user's position for already visited locations) and a *random sampling* (for controlling the precision of revealed information). We extensively evaluate our solution with a real dataset of mobility traces. We show that the proposed approach drastically reduces the exposure of the user's location to positioning systems (up to 95%). Indeed, by leveraging a caching strategy, requests are only sent when users visit new areas. Consequently, the capacity of positioning systems to infer points of interest of users from received requests is highly limited (a decrease of 50% on average). In addition, our privacy protection provides a trade-off between privacy (i.e., avoid revealing its true location) and utility (i.e., still benefiting from services such as places recommendation) fully controllable by the users.

KEYWORDS

Location privacy, Location data provider, Wi-Fi-based positioning

1 INTRODUCTION

With the democratization of positioning capabilities on mobile devices, location-aware computing is now exploited in most mobile applications. These applications are thus able to determine the location of users in real time and to provide them geolocated services, often called Location-Based Services (LBSs for short). These services provide contextual and personalised information depending on the current user's location. A multitude of LBSs have emerged these last years from venue finders, navigation, to social games (e.g., Pokemon GO [1]) or crowd-sensing applications [3].

While these LBSs require users to disclose their location to make the application working as expected, some mobile applications also collect the location of users through different sensors without their explicit consent [1, 4, 12]. This intrusive and abusing tracking for behavioral profiling purpose raises important privacy concerns from users.

The user's location can be retrieved by the mobile operating system from the GPS (fine-grained) or by requesting a location positioning system (coarse-grained) to convert surrounding Wi-Fi access points (APs for short), nearby cellular antennas, or an IP address into location. While GPS provides an accuracy of a few meters, it is not available in an indoor environment and may take some time to obtain geolocation information. An alternative to the GPS is to use the network infrastructure to pinpoint the location of a user. Indeed, Wi-Fi APs can be seen as landmarks that can be detected by the mobile system via native scanning process. Due to the high density of Wi-Fi APs in many areas, Wi-Fi based location constitutes a viable approach to build location systems. Consequently, many providers are available to serve in real time the location of users according to the surrounding Wi-Fi access points (e.g., WiGLE [2], Google [3], or Skyhook [4]). Location services offered by those providers are based on Wi-Fi access point BSSID (Basic Service Set Identifier), which is a globally unique identifier: a device obtains its location by transmitting the list of visible access points' BSSIDs. Kickstarted through wireless mapping (e.g. using the Google Car), the underlying databases are today continuously corrected and updated by the devices using the service in a crowdsourced fashion.

While relying on the GPS to retrieve the location is local and does not reveal the location of users to any third parties, requesting location positioning system with surrounding Wi-Fi APs obviously exposes its location to the provider of the positioning system. Nonetheless, how this provider actually exploits and shares the location of users (i.e., a sensitive personal information) remains not clear. The sensitivity of location data has been demonstrated by several works [8] and many research have focused on its protection [2, 9].

This privacy threat could easily be discarded by downloading the Wi-Fi location database on the mobile device. However, location providers have not considered to open this

[1] Pokemon GO: http://pokemongo.nianticlabs.com
[2] WiGLE: Wireless Network Mapping, http://wigle.net
[3] Google Maps Geolocation API: https://developers.google.com/maps/
[4] Skyhook: https://www.skyhookwireless.com

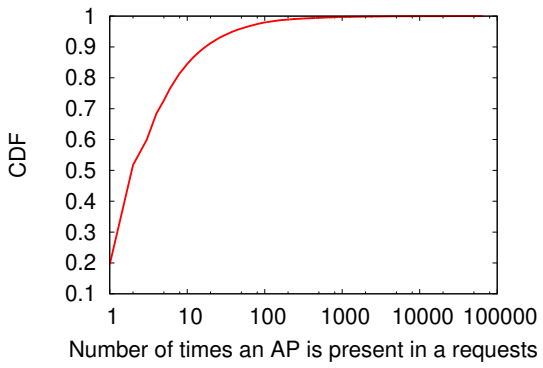


Figure 1: Caching strategy is motivated by the fact that many requests contain already visited access points (e.g., half of the access points have been already present in at least two requests).

valuable data: as this data is crowd-sourced, maintaining interactions with users is essential to keep the data up-to-date.

Protecting (or sanitizing) location information improves privacy but also have an inherent harmful impact on the utility of the protected information. For instance, introducing spatial noise obfuscates the real location of user (i.e., privacy gain), but reduces the accuracy of recommendations of places based on the protected data (i.e., utility loss). Privacy and utility metrics are very often dependent on the considered application.

In this paper, we propose a novel online solution to preserve users' privacy from location data providers when requesting the location of users from surrounding Wi-Fi access points, while supporting high utility. To achieve our goal, we combine a *caching strategy* for limiting the exposure of the user's position for already visited locations, and a *sampling strategy* for controlling the precision of revealed information in case of new locations.

Firstly, we leverage the caching of location retrieved from positioning systems. This cache is furthermore exploited before to request positioning systems for nearby places already visited. More precisely, when the location of the user is requested, we first evaluate the similarity between the list of currently surrounding Wi-Fi APs and the entries of a local cache containing the lists of Wi-Fi APs already converted to a location by a positioning system. If the similarity with one entry is higher than a threshold, the associated location present in the cache is used, otherwise a new request is sent to the positioning system. Specifically, a high similarity threshold will produce both an accurate utility and more requests sent to the positioning system. Inversely, a low similarity threshold will improve privacy by limiting the exposure of the user's location to the positioning system through requests, but the retrieved location can be less accurate. This caching strategy is motivated by the fact that a large number of requests to positioning systems concerns places where the user has been already close in the past. For instance, Figure 1 depicted the distribution of the number of times an AP has been present in the request of one user in our dataset (described Section 4.1). This plot shows that only 20% of the APs have been used in requests only once. This low percent is explained in part by the high regularity observed in the human mobility [21]. Consequently, a caching strategy can drastically reduce the number of requests sent to positioning systems and consequently significantly reduce the exposure of the user's location to an untrusted party.

Secondly, for controlling the precision of revealed information we sample the surrounding APs. More precisely, our protection mechanism picks at random a limited number of Wi-Fi APs in all surrounding APs to be part of the request. The number of samples included in the request impacts the precision of the location approximation obtained from positioning systems, the more samples, the more accurate.

These two mechanisms are complementary and have an impact of both privacy and utility. Moreover, the value of both system parameters (i.e., the similarity threshold controlling the data exposure of the caching mechanism and the sampling rate of surrounding APs) are user-driven and allow the end-user to control the expected privacy and utility trade-off.

We exhaustively evaluate our privacy-preserving scheme with a real dataset and show that our solution meets our expectations. Specifically, our solution drastically reduces the exposure of the user location by reducing the number of requests sent to the positioning system (from 40% to 95%). By leveraging caching, our privacy protection only sends requests when users visit new places or areas. However, as human mobility is highly repetitive, our solution limits the exposure of the user location on a regular basis. Consequently, the positioning system receives only a partial user location update that limits its capacity to extract points of interest from requests (by 50% on average). In addition, a high utility is preserved by only reducing the accuracy of the location by 25 meters on average. We show that the impact of this slightly decreased of accuracy on place recommendations remains limited and can be driven by the users. Finally, we also discuss the integration of our protection mechanism in mobile operating systems.

The remaining of this paper is organised as follows. We first present background information in Section 2 before to describe our protection mechanism and discuss its integration in the mobile operating system in Section 3. Finally, we introduce the experiment setup and the evaluation used to assess our protection mechanism in Section 4 and 5, respectively. Lastly, we conclude this paper in Section 6.

2 BACKGROUND

In this section, we first describe the problem statement associated to revealing Wi-Fi information (Section 2.1) before to present the considered adversary model (Section 2.2) and review related work (Section 2.3).

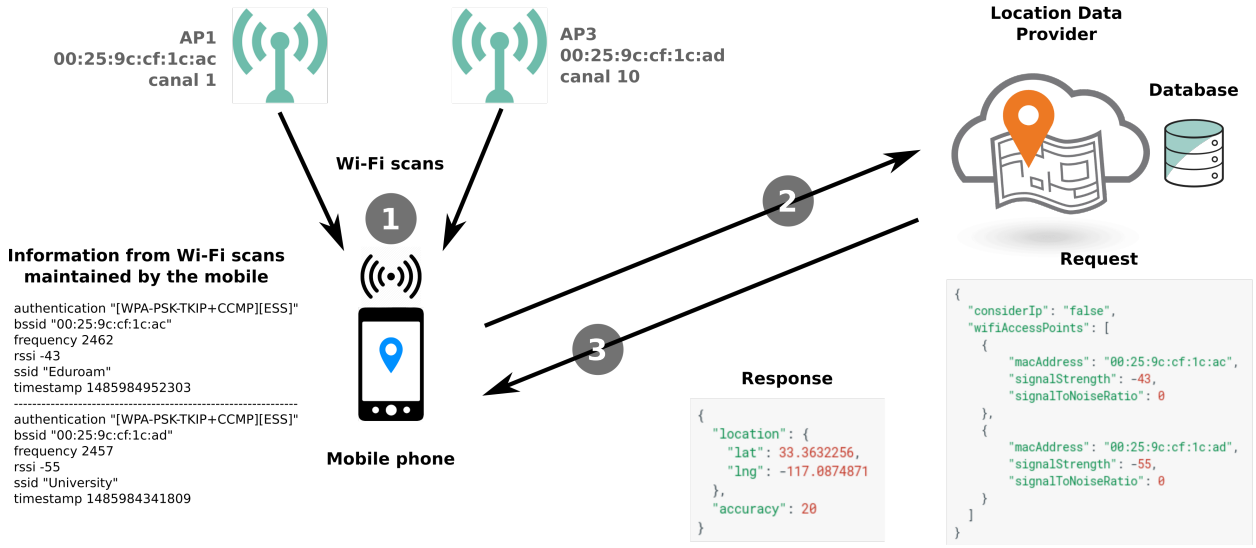


Figure 2: The user's location is exposed to the location data provider via the list of surrounding Wi-Fi APs.

2.1 Problem Statement

Most of the mobile phones nowadays embed a Wi-Fi interface. By regularly performing Wi-Fi scans, those Wi-Fi enabled devices maintain an up-to-date list of nearby APs. Consequently, through these network discovery operations, a mobile phone is always aware off the surrounding Wi-Fi APs.

The collected information about the surrounding Wi-Fi APs can be used to locate the user. Indeed, many location positioning systems offer online API to convert this information into location. Specifically, these positioning systems collect and maintain a database with the physical location of a large amount of Wi-Fi APs and use position estimator [13] to translate a list of Wi-Fi APs into a location. Once transmitted to a mobile system, this location is then spread to permitted mobile applications to provide a geolocated service.

Figure 2 gives an overview of the process. First, the mobile operating system performs regular Wi-Fi scans to discover and maintain an up-to-date list of nearby APs (❶). From these Wi-Fi scans the mobile system gathers several pieces of information such as the authentication mode, the MAC address of the AP (BSSID), the operating channel, the Service Set Identifier (SSID), a timestamp and a Received Signal Strength Indication (RSSI), the higher the stronger. When the operating system decides to update its location, it requests the API of a location positioning system (❷). This request contains the MAC address of a list of Wi-Fi APs and may include the associated RSSI. Obviously, requesting this service is a privacy threat as it reveals to the positioning system information related to the location of the users. Lastly, this positioning service responds to the mobile system by providing an estimation of the location from the request (❸) which stores and spreads this information to permitted applications. Note that in order to avoid malicious collection of the location of one specific AP, those positioning services require that at least two Wi-Fi APs are provided in the request. Location data providers usually can also provide an estimation of the location from the IP address and other wireless networks such as surrounding Cell Towers or Bluetooth networks. Although the storage capacity of smartphones increases, it is hardly possible to consider storing a database containing all the mappings of surrounding APs to associated locations.

2.2 Adversary Model

In this paper, we address the problem of privacy related to requesting location positioning systems only with information from Wi-Fi. In this context, the untrusted provider of the positioning system is considered as the adversary. It knows which request has been sent from which users and the location that has been returned. In other words, the adversary knows the location of the user every time it uses the Wi-Fi location provider to get its current location. Based on all these information, its goal is to conduct inference attacks to deduce information about the users and their locations such as their points of interest or demographics [2, 10, 23–25].

2.3 Related Work

Location privacy has been deeply surveyed and organised in [24]. Wireless location privacy has been early discussed by Schilit et al. [27]. Privacy issues associated with Wi-Fi location providers have been considered by Bellavista et al. [6]. They exploit a proxy-based architecture to dynamically adapt the granularity of the user's location exposed to LBSs. While attractive, using a proxy running near users to interact with LBSs on behalf of users has many limitations in terms of deployment.

Li et al. [17], in turn, introduced a scheme based on homomorphic encryption to protect both the client's location

privacy and the service provider's data privacy. However, the communication and computation costs linearly depend on both the number of access points and the number of location points, making this solution impractical for large scale applications.

The protection of Wi-Fi-based positioning information has been considered in the context of releasing a dataset for a challenge [14, 16]. In those works, the authors proposed a sanitization process to reduce the exposure of location data. However, this process includes many manual operations which are difficult to fully understand and reproduce. In addition, the evaluation of both the privacy and the utility is not reported. Furthermore, those approaches consider the offline anonymization of a dataset which does not correspond to our use case (i.e., online scenario).

Enhancing privacy through caching mechanisms has already been considered in [5]. Similarly, in the context of LBSs, CaDSA [22] also caches location information to reduce the number of requests sent to the service provider. However, this solution presents several substantial differences compared to ours. First, CaDSA requests the LBS with both its real location and dummy requests. Second, this caching mechanism is not implemented on the user device but instead on the network infrastructure such as Wi-Fi APs. Finally, CaDSA has been only evaluated through simulation and does not measure the utility loss.

Konstantinidis et al. proposed a privacy-preserving Wi-Fi localisation for indoor scenarios. This mechanism firstly relies on k-anonymity [30] to send multiple sets of APs in order to not allow the server to identify the real one among the $k-1$ fake ones. To do that, this mechanism uses bloom filters for 1-to-k matching set of APs. In addition, to generate continuous requests reflecting a natural mobility pattern, the candidates to be part of the fake set of APs are biased towards APs neighboring the previous location of the user. This work has multiple limitations. Several contributions have shown the limits of k-anonymity. However, this solution could leverage new models proposed to overcome these limits to guarantee privacy such as l-diversity [20] or t-closeness [18]. In addition, the proposed scheme requires multiple exchanges between the user's smartphone and the untrusted server of the positioning service consuming more energy than leveraging local cached information as in our solution. Lastly, even if local movement patterns seem natural, the persistent repetitiveness trait of individual's mobility [21] can be leveraged to identify the fake information in the requests.

3 PROTECTION MECHANISM

Our mechanism aims at protecting the requests sent to the positioning system to get the location of the users from the surrounding Wi-Fi access points. Specifically, to avoid revealing a fine-grained information about the location of users while maintaining a high utility, our protection mechanism combines two techniques: a *caching strategy* (Section 3.1) and a *random sampling* (Section 3.2). Lastly, we discuss implementation issues (Section 3.3).

3.1 Caching for Privacy

Our privacy-preserving mechanism aims at limiting the exposure of the user's location to the positioning system. To achieve that, it tries to reduce the number of requests sent to this positioning system required to update the position of the users from the surrounding Wi-Fi APs. Specifically, our mechanism caches the requests sent to the positioning system and the associated retrieved location, and furthermore leverages this cache to approximate the position of the user. More precisely, before sending a new request to the positioning system, we parse the cache to find an entry containing similar Wi-Fi APs. If the similarity between the current surrounding Wi-Fi APs and a list of APs stored in the cache is higher to a certain threshold, named p, we return the retrieved location for the associated cache entry (the first entry found is used). Otherwise, a new request to the positioning system is emitted. This query as well as the associated returned location are then added to the cache.

The cache, named C, is organized as a list of tuples where each tuple contains both the list of APs presents in the request ($APs =< AP_0, ..., AP_z >$) and the associated retrieved location ($l =< lat, lng >$) ; $C =< [APs_0, l_0], ..., [APs_n, l_n] >$. The similarity threshold p measures the overlap between the current surrounding Wi-Fi APs and the list of APs of a past request stored in the cache. Formally, this similarity threshold is defined as follow:

$$p = \frac{|APs_{current} \cap APsi|}{|APs_{current}|},$$

where APs_i represents the list of APs stored at the i^{th} element in the cache.

Our approach is user-driven, according to the expected privacy level, users define the similarity threshold p between 0.1 and 1. We do not consider a similarity threshold at 0 as the retrieved location will be not related to the current location of the user in this case. The closer to 0.1, the more likely to find a similar entry in the cache, and consequently less new requests will be emitted to the positioning system resulting in better privacy. Inversely, $p = 1$ means that all the surrounding APs must be present in at least one entry stored in the cache. In this case, new requests will be more likely to be sent as the probability to find an entry with the same APs in the cache is smaller.

Our privacy-preserving scheme leverages the high regularity of human mobility. Indeed, new requests will be sent only when the users visit a new place or do a new ride. Once cached, no new requests will be sent if the users follow roughly the same path where the precision of the new path compared to the previous one depends on the value of parameter p.

3.2 Random Sampling for Utility

While the caching limits the data exposure, the sampling of surrounding Wi-Fi APs limits and controls the precision of the users' location revealed to the service provider. More precisely, we select a random sample of size s from the set of all surrounding Wi-Fi APs available, where each AP in the list has the same chance of being included in the sample.

Obviously, the larger s, the most accurate will be the estimation of the location provided by the location data provider. We empirically define three different values for the size s, 2 (the smallest accepted size by the service provider for the list of APs informed in the request), 5, and 10 for a high, medium, and low protection of the location, respectively. We do not consider larger values as the precision does not change significantly from s=10. Our approach is user-driven, according to the expected utility level, users control the size of the sample. However, for the sake of simplicity for end users, they do not define any value of s but choose among three levels of utility corresponding to the three predefined values of s. If the number of available Wi-Fi APs is smaller than the expected threshold, we use all available Wi-Fi APs.

3.3 Integration Issues

To be effective at protecting privacy, our proposal needs to be added to the mobile system. Adding this new feature may present several challenges regarding storage and the integration within the OS.

Most mobile devices are shipped with a version of the mobile operating system (OS) that cannot be easily modified. The elements in charge of performing the geolocation are integrated within the OS and their modification or replacement cannot be considered without involving a collaboration of the OS developers.

Having the collaboration of the OS developers would be the most straightforward and easy approach. Having full control on the elements of the OS in charge of the geolocation, it would be possible for the developers to modify those elements in order to include our mechanism. However, the company developing the OS may be reluctant to integrate our mechanism even if our mechanism provides the same level of accuracy as the current positioning system (as shown Section 5.2). Indeed, location providers' systems are constantly being updated and corrected thanks to the request of users, reducing the number of requests would hamper this mechanism.

If the collaboration of the OS developers is not possible, it could still be possible to integrate our proposal by relying on a local proxy that would intercept the requests made to the location provider. This kind of proxy could be implemented by a VPN proxy deployed by a mobile application [26]. Once intercepted, a request would be either answered locally or transparently forwarded to the location provider. Relying on this kind of local VPN proxy can face several issues. For instance, certificate pinning may prevent the proxy to set a Man-in-the-Middle. In addition, using a proxy can have a negative impact on the network performances [26] or can require rooted devices exposing users to limitations [29]. Nevertheless, the local proxy approach could be considered as an intermediary step before a full integration in the OS once the benefits of the approach have been demonstrated.

4 EXPERIMENTAL SETUP

In this section, we present the dataset (Section 4.1), the methodology and the evaluation metrics we used to conduct our experiments (Section 4.2).

4.1 Dataset

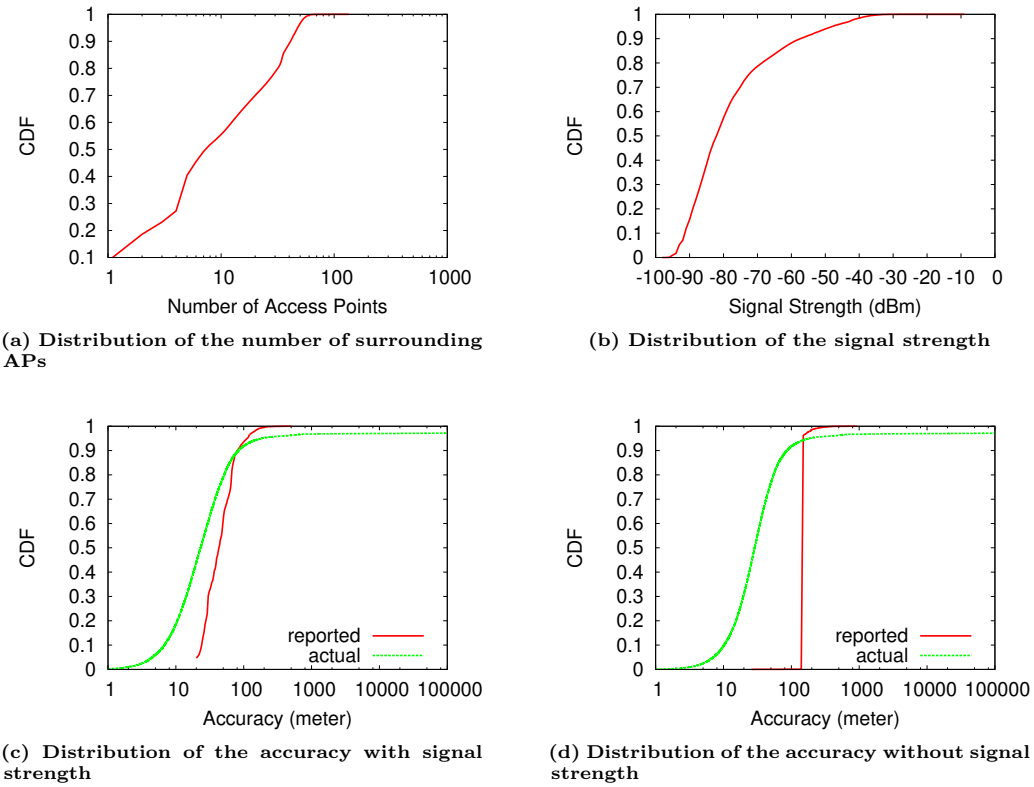
The PRIVAMOV dataset [7] involves 100 students and staff from various campuses in the city of Lyon (France) equipped with smartphones running a data collection software. The data collection took place from October 2014 to January 2016 and gathers information from many sensors such as GPS, Wi-Fi, GSM, accelerometer to name a few. In this paper, we use the records from the GPS periodically collected and the logs from the Wi-Fi scan as presented in Section 2.1. These two data collections gather 156M and 25M of records, respectively.

To compare both the location of the user inferred from the Wi-Fi and the location measured from the GPS, we first identify the information from the Wi-Fi scan that are combined to a GPS record with less than 1 second difference. We identify $849,776$ Wi-Fi scans (i.e., a list of surrounding Wi-Fi APs) associated to a GPS coordinates for $86,741$ different APs. Figure 3 presents the cumulative distribution (through CDF) of different properties of this dataset such as the number of APs in each Wi-Fi scan (Figure 3a), the signal strength of each Wi-Fi AP (Figure 3b), and the accuracy of location (reported by Google Maps as well as computed from the real location of the user collected through the GPS) when requests use signal strength (Figure 3c) and without signal strength (Figure 3d). These distributions show that on average a Wi-Fi scan contains 8 APs and has a signal strength smaller than -80dbm (the closer to 0, the stronger). Distribution Figure 3c also shows that on average the accuracy of the location reported by the Google Geolocation service is around 40 meters when we inform the signal strength in the request. Without this information (Figure 3d), the accuracy reported by Google is coarse-grained and is 150 meters for almost all requests. When the accuracy is computed from the collected gps-based location, the accuracy is on average around 25 meters when signal strength is used and around 30 meters without.

4.2 Utility and Privacy Metrics

Using a location-privacy preserving mechanism (LPPM for short) improves the user privacy but inherently impacts the utility of the resulting protected data [11]. Many utility and privacy metrics have been proposed [19, 24, 28, 31].

In this paper, to reflect the level of exposure of the user's location as a privacy assessment, we consider the number of requests sent to the positioning system. More precisely, we measure the ratio of requests actually sent to the positioning system to the total number of accesses to location (i.e., through requests and caching). This privacy measurement is then normalized between [0:1], the closest to 0, the better. This metric named exposure is defined as follow:



(a) Distribution of the number of surrounding APs

(b) Distribution of the signal strength

(c) Distribution of the accuracy with signal strength

(d) Distribution of the accuracy without signal strength

Figure 3: We use a real dataset (Privamov [7]) to evaluate our privacy-preserving scheme. For instance, the cumulative distribution functions show that the accuracy of the location provided by the Google Geolocation service is around 20 meters on average.

$$Exposure = \frac{\#Requests}{(\#Requests + \#CacheAccesses)}$$

This privacy metric is however largely dependent on the user's mobility. Indeed, a user who remains in the same area could save a large number of queries compared to a user who always moves to new places. To take the user mobility into account we also analyse the area covered by users. To do that, we split the map by cells of 100 square meter and we computed the number of different cells visited by each user. The area coverage is reported as square kilometer.

To evaluate privacy, we also consider the extraction of points of interest (POIs for short). POIs are spatially delimited places where users spend some time. POIs can be home or workplace, but also a school, a library or a museum for instance. POIs can be extracted from mobility traces by using clustering spatio-temporal algorithms parametrized with a maximum POI diameter Δ_d and a minimum stay time Δ_t (we implemented a POI detection scheme similar to [32] and we consider $\Delta_d = 200$ meters and $\Delta_t = 30$ minutes in our experiments). Specifically, to measure the privacy leakage, we extract POIs from the requests sent to the positioning system with and without using our privacy protection and evaluate the number of detected POIs, their duration and the number of associated records.

The utility, in turn, is evaluated through two metrics. The first one quantifies the precision (i.e., the spatial distortion) between the real location of the user (i.e., the location collected by the GPS) and the approximation of the location retrieved with our solution (i.e., either from a new request to a positioning system or from the local cache). The second one reflects the quality of the recommendations.

Specifically, we measure the completeness (i.e., the recall) of the recommendations provided by Google Places API associated to the real location of the user compared to the recommendations associated to the approximation of the user location (i.e., from the protected request). These recommendations correspond to the nearest places (e.g., restaurants, shops) to the user's location inside a certain radius. We consider a radius of 50 meters when we request the recommendations from Google Places API. These utility metrics named accuracy and recommendation-quality respectively are defined as follow:

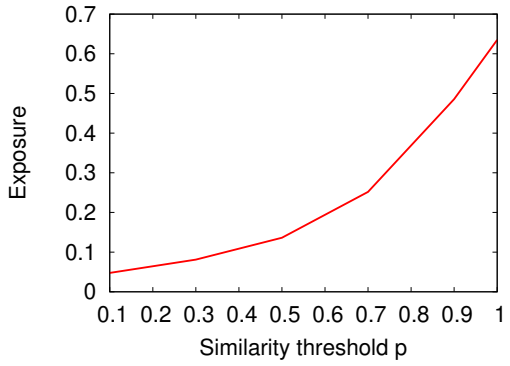


Figure 4: By leveraging our caching strategy, the level of exposure (i.e., the number of requests sent to the position system) is drastically reduced (e.g., only 5% of the requests are actually sent with $p = 0.1$).

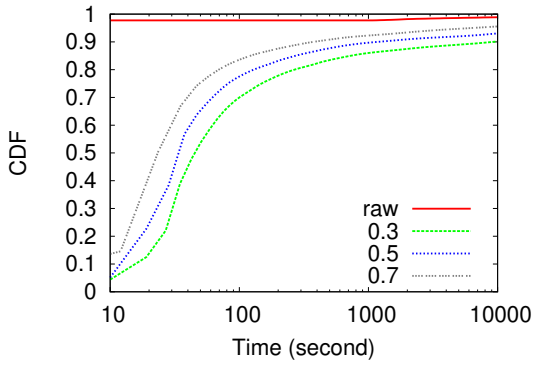


Figure 5: By sending request only when new places are visited, our caching scheme drastically increases the time between two consecutive requests.

$$Accuracy = \Delta(coord_{gps}, coord_{Wi-Fi})$$

$$Recommendation - Quality = \frac{|Reco_{gps} \cap Reco_{Wi-Fi}|}{|Reco_{gps}|}$$

where, $\Delta(a, b)$ provides the distance in meter between the coordinate a and b, and $coord_{wifi}$ the coordinate retrieved from our solution. In turn, $Reco_{gps}$ and $Reco_{Wi-Fi}$ are respectively the list of recommendations associated to the real location of the user (i.e., the GPS coordinates) and the location approximation retrieved from the protected requested sent to Google Places API. Although these proposed utility metrics are affected by the accuracy of the Google Map Geolocation service, this analysis is comparative and well reflects the fact that utility assessment is often application dependent.

5 EVALUATION

In this section we evaluate the capacity of our protection mechanism to preserve the privacy of users when requesting a location positioning system (Section 5.1) while limiting the associated utility loss (Section 5.2).

5.1 Privacy Evaluation

We first evaluate the gain of privacy provided by our caching mechanism. As described in Section 3, our protection scheme leverages caching to avoid exposing the location of users to untrusted positioning systems. More precisely, if the surrounding Wi-Fi APs are similar to the ones included in a request found in the cache, the previous location retrieved from the positioning system is used instead of sending a request. Figure 4 plots the exposure of the user's location (i.e., the ratio of requests sent to the positioning system with and without the adoption of our solution) depending on the similarity threshold p (i.e., parameter which defines if an entry in the cache can be used or not). First, results show that privacy increases exponentially according to the similarity threshold. Indeed, as expected a smaller p increases the likelihood to find an entry in the cache close enough to the current surrounding APs and consequently generates less requests. Second, we show that even with a similarity threshold at 1 (i.e., the list of the current Wi-Fi APs is exactly contained in the list of APs of at least one entry in the cache), almost 40% of the requests exposing the location of user are avoided. Finally, only 5% of the requests are actually sent if we consider a lower similarity threshold (i.e., $p = 0.1$).

We then analyse the time spent between two requests sent to positioning systems. Figure 5 depicts the distribution of the time spent between two consecutive sent requests for various values of parameter p as well as without privacy protection (named raw on the figure). Results show that compared to without any protection where requests are mostly sent every second, our caching scheme drastically reduces the frequency of the data exposure. On average, 35 seconds are spent between two consecutive requests with $p = 0.5$, and this time drops to 20 seconds with $p = 0.7$. This behaviour is consistent with the previous result (Figure 4), parameter p controls the sending of requests, a p closer to 1 generates more requests than a p closer to 0.

This high reduction of data exposure provided by our caching scheme is a direct result of the important regularity in human mobility. Indeed, as repetitiveness is a persistent trait in an individual's mobility [21], our mechanism only sends requests to positioning systems when users visit new places or are located in an area with surrounding APs which are different enough to what they have been already exposed in the past. To comfort this assumption, Figure 6 reports the average number of requests sent to positioning systems per user according to the area covered by its mobility. Results show that the number of sent requests is largely correlated to the area covered by the user, the more explored places, the more sent requests.

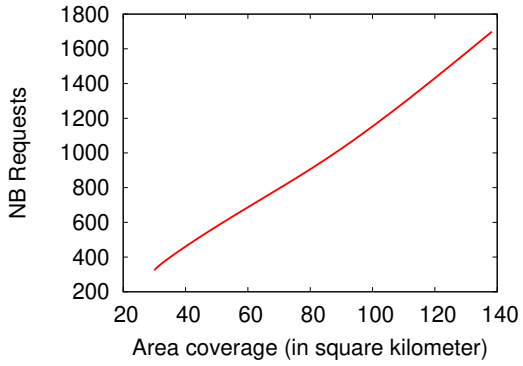
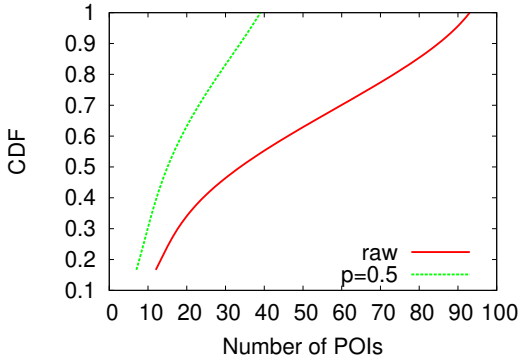


Figure 6: The number of sent requests is correlated to the mobility of users, the more explored places, the more sent requests.

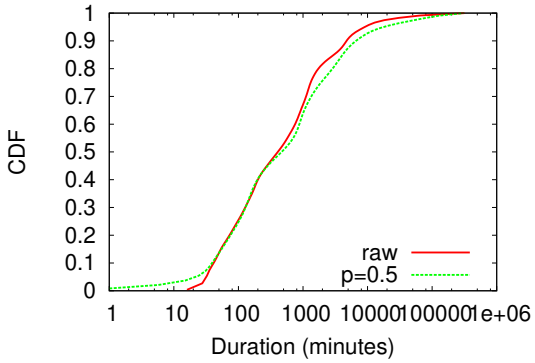
Lastly, we analyse how positioning systems can infer POIs from the received requests. By caching information, our privacy protection also reduces the capacity of the targeted positioning system to extract POI from the received data. Figure 7 compares different data distributions related to POI extraction without privacy protection (named raw in the figure) and with caching using a similarity threshold at 0.5. These distributions include the number of POIs (Figure 7a), their duration (Figure 7b) and the number of associated requests (Figure 7c). Results (Figure 7a) show that the positioning system is only able to extract roughly half of the users' POIs (13 on average against 32 without privacy protection). However, results (Figure 7b) show that our privacy protection does not impact the capacity of the positioning system to infer the duration that users stay in the same POIs. Finally, results (Figure 7c) also show that even inside a POI (i.e., a limited place where the user spent time) many requests have not been sent to the positioning system as they have been already present in the cache.

5.2 Utility Evaluation

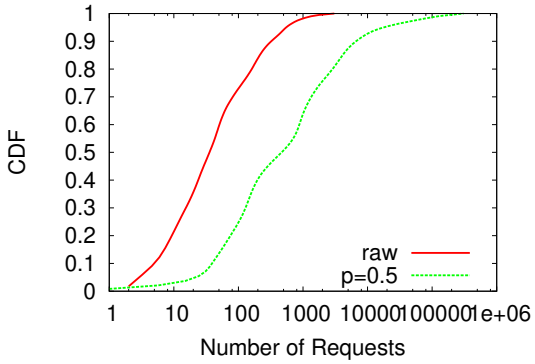
We start by evaluating the utility loss in terms of accuracy introduced by the caching scheme. Figure 8 depicts the distribution of the spatial distortion between the actual user's location and the value returned by our solution for different values of similarity threshold p. This figure also reports the accuracy provided by the Google Maps Geolocation API (named request only in the figure), means the difference between the actual location of the user and the location provided by the Google service (i.e., without our privacy-preserving mechanism). The average accuracy provided by Google is around 30 meters. For comparison purposes, the accuracy of the location retrieved from the GPS is around 3 meters, 10 times for precise. With the adoption of our caching scheme, results show that regardless of the value of the similarity threshold, the average spatial distortion is between 20 and 40 meters. This accuracy is similar to the actual accuracy provided by the positioning system of Google service.



(a) Distribution of the number of POIs



(b) Distribution of the duration of POIs



(c) Distribution of the number of records per POI

Figure 7: Our caching scheme drastically reduces the capacity of positioning systems to extract points of interest from the received requests.

We now evaluate the impact of random sampling. Figure 9 plots the distribution of the distance between the real location of users and the approximation inferred from Wi-Fi provided by Google Maps Geolocation API for different sizes of sample (parameter s where $s = \infty$ represents no sampling). Results show that a sample with 10 APs slightly increases

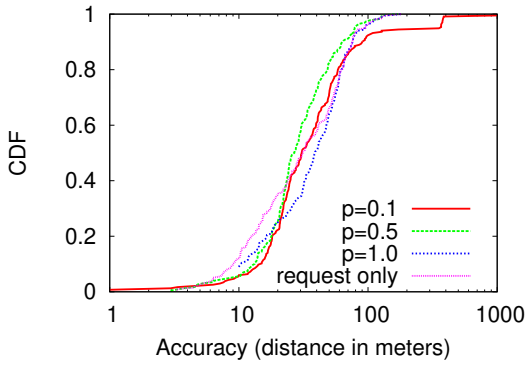


Figure 8: The spatial distortion provided by our caching scheme is low and similar to the actual accuracy provided by the positioning system of Google.

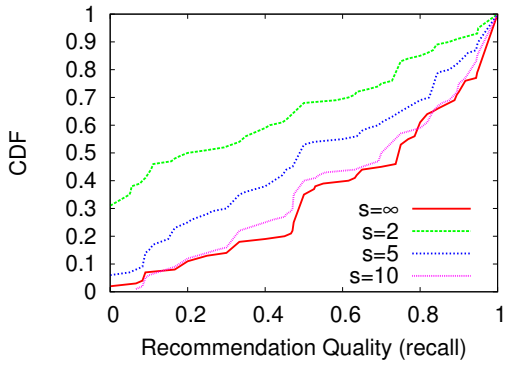


Figure 10: According to the sampling rate of our privacy-preserving mechanism, the quality of the places recommendation of Google varies.

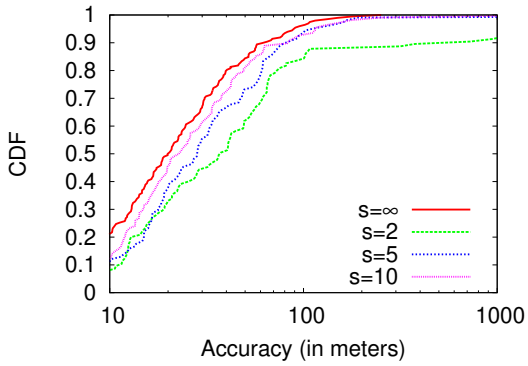


Figure 9: Our random sampling scheme controls the quality of the request and consequently the accuracy (i.e., the spatial distortion) of the location returned by the positioning system.

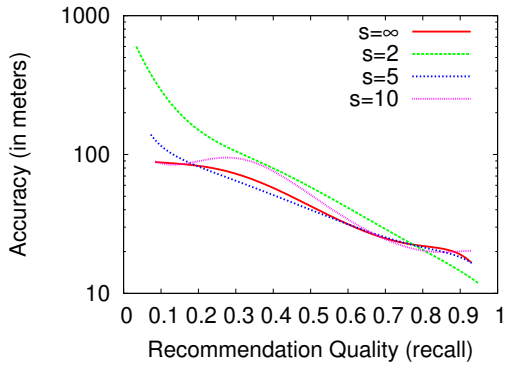


Figure 11: An exposition of more accurate information about the user's location provides better recommendations, a well-known conflicting privacy and utility trade-off.

the distortion of the location approximation returned by the positioning system (i.e., 25 meters on average against 20 meters if all APs are used in the requests). Reducing the size of this sample reduces the accuracy of the location returned by the positioning system (i.e., an accuracy of 40 and 50 meters for a size of 5 and 2, respectively). Interesting enough, this low spatial distortion is compliant with many applications.

In addition, we evaluate the impact of the random sampling on the recommendation quality. Figure 10 depicts the distribution of the recall (i.e., the ratio of the places recommendation received with the real user location over the recommendations received from the approximation received from the positioning system) for varying size of sample (parameter s), namely 2, 5, 10, and without sampling ($s = \infty$). Similarly, to the accuracy, retrieving the location of users from the Wi-Fi even without sampling inherently reduces the recommendation quality compared to a location retrieved from the GPS. For instance, 80% of the answers have more than 0.5 of recall and this number jumps to 80 of recall for 50% of the answers. In addition, results show that reducing the size of the sample has an important impact on the recommendation quality, the smaller size, the smaller recall. For instance, 50% of the answers have more than 0.7 of recall for a size of sample of 10 while this value drops to 0.35 for a size of 2.

Finally, we analyse the trade-off between accuracy and recommendation quality. Figure 11 presents this trade-off for a sample size of 2, 5, and 10, as well as without sampling ($s = \infty$). Obviously, results show that when an important recommendation quality is reached, the accuracy of the approximation of the user's location is fine-grained (accuracy of 10 meters for a recall of 0.95). Inversely, when the accuracy is coarse-grained, the recommendation quality is low (e.g., an accuracy around 75 meters gives a recall of 0.3). These curves illustrate the well-known conflicting trade-off between utility (i.e., the recommendation quality) and privacy (i.e., the precision of the revealed location).

6 CONCLUSIONS

In this paper, we proposed a practical mechanism to preserve the privacy of users by avoiding the disclosure of location information to positioning systems. More precisely, our privacy protection combines a caching scheme and a random sampling strategy. The caching scheme exploits already retrieved information from positioning systems to approximate the current user's position in order to reduce the exposure of its location (i.e., the number of requests sent to the positioning system). The caching mechanism is thus effective only in familiar locations, as requests are only generated for new visited location. Due to the high regularity of human mobility, in most cases, the location will be obtained from the cache, thus protecting the user. The random sampling strategy, in turn, controls the precision of the information revealed to the positioning system, and consequently the accuracy of the returned location approximation. We demonstrate the capacity of our solution with the use of a real dataset and through real exchanges with a positioning system. We show that our protection mechanisms can drastically improve privacy (i.e., reducing the number of requests emitted to the positioning system) while maintaining a high utility (i.e., a small spatial distortion).

REFERENCES

[1] Jagdish Prasad Achara, Franck Baudot, Claude Castelluccia, Geoffrey Delcroix, and Vincent Roca. 2013. Mobilitics: Analyzing Privacy Leaks in Smartphones. *ERCIM News* 2013, 93 (2013).

[2] Berker Ağır, Kévin Huguenin, Urs Hengartner, and Jean-Pierre Hubaux. 2016. On the Privacy Implications of Location Semantics. In *PETS*, Vol. 2016. 165–183.

[3] Nadav Aharony, Wei Pan, Cory Ip, Inas Khayal, and Alex Pentland. 2011. Social fMRI: Investigating and Shaping Social Mechanisms in the Real World. *Pervasive Mobile Computing* 7, 6 (2011), 643–659.

[4] Hazim Almuhimedi, Florian Schaub, Norman Sadeh, Idris Adjerid, Alessandro Acquisti, Joshua Gluck, Lorrie Faith Cranor, and Yuvraj Agarwal. 2015. Your Location Has Been Shared 5,398 Times!: A Field Study on Mobile App Privacy Nudging. In *CHI*. 787–796.

[5] Shahriyar Amini, Janne Lindqvist, Jason Hong, Jialiu Lin, Eran Toch, and Norman Sadeh. 2011. Caché: Caching Location-enhanced Content to Improve User Privacy. In *MobiSys*. 197–210.

[6] P. Bellavista, A. Corradi, and C. Giannelli. 2005. Efficiently managing location information with privacy requirements in Wi-Fi networks: a middleware approach. In *ISWCS*. 91–95.

[7] Sonia Ben Mokhtar, Antoine Boutet, Louafi Bouzouina, Patrick Bonnel, Olivier Brette, Lionel Brunie, Mathieu Cunche, Stephane D 'alu, Vincent Primault, Patrice Raveneau, Herve Rivano, and Razvan Stanica. 2017. PRIVA'MOV: Analysing Human Mobility Through Multi-Sensor Datasets. In *NetMob*.

[8] V. Bindschaedler and R. Shokri. 2016. Synthesizing Plausible Privacy-Preserving Location Traces. In *S&P*. 546–563.

[9] Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2014. Optimal Geo-Indistinguishable Mechanisms for Location Privacy. In *CCS*. 251–262.

[10] Antoine Boutet and Sébastien Gambs. 2019. Inspect What Your Location History Reveals About You: Raising User Awareness on Privacy Threats Associated with Disclosing His Location Data. In *CIKM*. 28612864.

[11] Sophie Cerf, Vincent Primault, Antoine Boutet, Sonia Ben Mokhtar, Robert Birke, Sara Bouchenak, Lydia Y Chen, Nicolas Marchand, and Bogdan Robu. 2017. PULP: Achieving Privacy and Utility Trade-off in User Mobility Data. In *SRDS*. 164–173.

[12] Mojtaba Eskandari, Maqsood Ahmad, Anderson Santana de Oliveira, and Bruno Crispo. 2017. Analyzing Remote Server Locations for. Personal Data Transfers in Mobile Apps. In *PETS*. 118–131.

[13] D. Kelly, R. Behan, R. Villing, and S. McLoone. 2009. Computationally tractable location estimation on WiFi enabled mobile phones. In *ISSC*. 1–6.

[14] N. Kiukkonen, Blom J., O. Dousse, Daniel Gatica-Perez, and Laurila J. 2010. Towards rich mobile phone datasets: Lausanne data collection campaign. In *ICPS*.

[15] A. Konstantinidis, G. Chatzimilioudis, D. Zeinalipour-Yazti, P. Mpeis, N. Pelekis, and Y. Theodoridis. 2016. Privacy-preserving indoor localization on smartphones. In *ICDE*. 1470–1471.

[16] Juha K. Laurila, Daniel Gatica-Perez, Imad Aad, Jan Blom, Olivier Bornet, Trinh Minh Tri Do, Olivier Dousse, Julien Eberle, and Markus Miettinen. 2013. From Big Smartphone Data to Worldwide Research: The Mobile Data Challenge. *Pervasive Mob. Comput.* 9, 6 (2013), 752–771.

[17] Hong Li, Limin Sun, Haojin Zhu, Xiang Lu, and Xiuzhen Cheng. 2014. Achieving privacy preservation in WiFi fingerprint-based localization. In *INFOCOM*. 2337–2345.

[18] Ninghui Li and Tiancheng Li. 2007. t-Closeness: Privacy Beyond k-Anonymity and -Diversity. In *ICDE*. 106–115.

[19] Bo Liu, Wanlei Zhou, Tianqing Zhu, Longxiang Gao, and Yong Xiang. 2018. Location privacy and its applications: A systematic study. *IEEE access* 6 (2018), 17606–17624.

[20] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. 2006. L-diversity: privacy beyond k-anonymity. In *ICDE*. 24–24.

[21] Eduardo Mucceli, Aline Carneiro Viana, Carlos Sarraute, Jorge Brea, and José Ignacio Alvarez-Hamelin. 2016. On the Regularity of Human Mobility. *Pervasive and Mobile Computing* (2016).

[22] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li. 2015. Enhancing privacy through caching in location-based services. In *INFOCOM*. 1017–1025.

[23] A. Olteanu, K. Huguenin, R. Shokri, M. Humbert, and J. Hubaux. 2017. Quantifying Interdependent Privacy Risks with Location Data. *IEEE Transactions on Mobile Computing* 16, 3 (2017), 829–842.

[24] Vincent Primault, Antoine Boutet, Sonia Ben Mokhtar, and Lionel Brunie. 2018. The Long Road to Computational Location Privacy: A Survey. *Communications Surveys and Tutorials, IEEE Communications Society* (2018), 1.

[25] Ajaysinh Rathod and Vivaksha Jariwala. 2019. Investigation of Privacy Issues in Location-Based Services. In *Recent Findings in Intelligent Computing Techniques*. 55–65.

[26] Abbas Razaghpanah, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Christian Kreibich, Phillipa Gill, Mark Allman, and Vern Paxson. 2015. Haystack: In situ mobile traffic analysis in user space. *ArXiv e-prints* (2015).

[27] B. Schilit, J. Hong, and M. Gruteser. 2003. Wireless location privacy protection. *Computer* 36, 12 (2003), 135–137.

[28] Reza Shokri. 2012. *Quantifying and protecting location privacy*. Technical Report. EPFL.

[29] San-Tsai Sun, Andrea Cuadros, and Konstantin Beznosov. 2015. Android rooting: Methods, detection, and evasion. In *CCS Workshop*. 3–14.

[30] Latanya Sweeney. 2002. K-anonymity: A Model for Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 10, 5 (Oct. 2002), 557–570.

[31] Isabel Wagner and David Eckhoff. 2018. Technical Privacy Metrics: A Systematic Survey. *ACM Comput. Surv.* 51, 3 (June 2018).

[32] Changqing Zhou, Dan Frankowski, Pamela Ludford, Shashi Shekhar, and Loren Terveen. 2004. Discovering Personal Gazetteers: An Interactive Clustering Approach. In *GIS*. 266–273.