Privacy support in people-centric sensing

Luca Becchetti, Luca Filipponi, Andrea Vitaletti
Diparitmento di Ingegneria Informatica Automatica e Gestionale "A. Ruberti", Università di Roma "La Sapienza", Italy
Email: {becchett, filipponi, vitale}@dis.uniroma1.it

***Abstract*— In this paper we present an approach to support privacy in people-centric sensing. In particular, we propose a technique that allows a central authority to select a subset of users whose past positions provide a good coverage of a given area of interest, without explicitly georeferencing them. To achieve this goal, we propose an efficient algorithm to solve the well known, NP-complete Set Cover problem that does not require explicit knowledge of the sets, but only uses their compact, privacy preserving representations or sketches. We perform a thorough experimental analysis to evaluate the performance of the proposed technique and its sensitivity to a few key parameters using public data from real applications. Experimental results support the effectiveness of the proposed approach to efficiently produce accurate environmental or social maps, at the same time preserving users' privacy.**

***Index Terms*— People Centric Sensing, Data Privacy, Efficient Data Representation.**

I. Introduction

In a recent report by IBM [1], the use of smart phones as mobile sensors is envisioned to be one of the top five innovations that will change our lives in the next years and Qualcomm Chairman and CEO Dr. Paul Jacobs predicts that by 2014, there will be over 400 million wearable wireless sensor devices on humans. In his key note at last Sensys, Alex Sandy Pentland [2] discussed how to "instrument humanity", namely how to exploit wearable wireless sensors carried by humans to support novel and advanced services such as mapping social networks, predicting traffic patterns or classifying human behavior.

In this context, data is necessarily "people centric", namely, they do not only refer to the environment, as in traditional sensor networks, but they also reflect personal aspects of people and their social positions and roles. As a consequence, as also observed by Pentland, the main obstacle to this vision is users' privacy and "unfortunately, current privacy law does not do so much for us". The long term goal is to grant the control of private data to individuals instead of private companies as happens nowadays. In the future, a user should be able to delete her profile from a service provider or even to force the same service provider to pass her profile to a competitor before deleting it. However, we are still far from achieving this goal. In current systems the enforcement of privacy policies is mainly based on agreements between users and their service providers; users rely on their service providers and their technological infrastructures to securely manage their personal data.

Manuscript received February 15, 2011; revised May 15, 2011; accepted June 15, 2011.
Partially supported by PRIN 2008 research project COGENT.

People-centric sensing can see an active participation of the users (participatory sensing) to "meet an application requests out of personal or financial interest" or it can employ an opportunistic approach, in which a user configures her device to allow an application to run (subject to privacy and resource usage constraints), but may not be aware of the application's activity at any given time [3]. In opportunistic sensing, a user's sensing device (e.g., a mobile phone) is used whenever its state (e.g., connection availability, geographic location, body location) matches the context requirements of an application query. In this way, applications can leverage the sensing capabilities of all system users without requiring human intervention to actively and consciously participate in the application. This approach introduces several challenges and constraints. A first and trivial consideration is that opportunistic use of a device should not noticeably impact the normal user experience of the device (e.g., significantly reduce battery lifetime or cause sluggishness in the execution of other applications), potentially limiting actual availability of resources (e.g. connection bandwidth, cpu time). In addition, due to hardware/software heterogeneity of devices and human mobility, the availability of devices with proper capabilities is not guaranteed at any time and this leads to the so called "sensor availability problem" which poses serious limitations on the service level and data quality of such systems. Finally, sensors will be carried in a manner most convenient to humans (e.g., in a pocket or purse) and not necessarily in a manner that guarantees high fidelity data gathering for an application.

While these issues may fade over time (e.g., increased battery lifetime) or be at least partially addressed, serious privacy issues remain, since both participatory and opportunistic sensing often involve the disclosure of potentially sensitive data to the application and/or the service provider. As an example, consider a mapping application which displays the magnitude of a phenomenon (e.g. noise) in an area of interest. Environmental data is autonomously collected by users, who are then expected to communicate their data to a service provider. This in turn selects the subset of users that provided data collected in the area of interest, and then computes and visualizes the map. This selection process typically allows the service provider to become aware of users' location traces, exposing them to privacy threats.

In this paper we present an approach to support privacy in people centric sensing applications. In particular we use sketches, namely compact and privacy preserving synopses of data, that allow a service provider to compute relevant statistics over the data without disclosing users' sensitive information even to the service provider itself.

Roadmap. We discuss related work in Section II. Section III describes the overall scenario we envision and the main aspects of our approach. In particular, it describes the data collection and delivery process and it gives an overview of the general technique we use to produce compact and privacy preserving sketches of users' geographical data. Section IV describes an efficient algorithmic technique that allows the service provider to select a subset of data, among those received from all mobile users, that best match a statistical query over an area of interest. Section V briefly discusses the privacy robustness of our technique and outlines a general approach that can be adopted to make communication between mobile users and the service provider secure against third party attacks. Though this aspect is not the focus of our paper, we think it might be an aspect to consider in a practical implementation of our ideas. Section VI describes in detail an extensive experimental analysis we conducted using a large collection of GPS traces. The results obtained suggest that the approach we propose is is effective and realistic. Finally, in Section VII we summarize the main findings of this work.

II. Related work

People-centric sensing (PCS) [4] leverages the use of human carried devices (such as smart-phones) to sense information directly or indirectly related to human activity or environment, in an opportunistic or participatory way [3].

In participatory PCS, users explicitly control the access to the resources of their sensing devices and actively participate to a sensing task. As an example, the Italian newspaper Corriere della Sera, asks their readers to participate in a reportage on the current status of cultural heritage in Italy, sending photos of degraded monuments. In opportunistic PCS, once users grant access to their resources, the monitoring task is performed autonomously by the monitoring application without requiring any explicit feedback from the user. As an example, users could contribute to the definition of a noise map, opportunistically using the microphone of their mobile phones to detect the environmental noise level as well as opportunistically sending the collected data once a wireless connection becomes available.

The MetroSense project [5] is working with industry and agencies to develop new applications[6], [7], classification techniques[8], privacy approaches[9], and sensing paradigms for mobile phones [10], [11] enabling a global mobile sensor network capable of societal-scale sensing.

Metrosense is based on the opportunistic sensor network approach and has been designed according to three principles: network symbiosis, i.e. new sensing infrastructure devices should leverage traditional networking infrastructure and services, asymmetric design i.e. nodes with more resources should receive more computational and energy demanding tasks, and localized interaction. Bikenet [6] is a mobile sensing system for mapping the cyclist experience, built leveraging the MetroSense architecture to provide insight into the real-world challenges of people-centric sensing. Bubble-sensing[11] is a new sensor network abstraction in which sensing tasks are opportunistically assigned to mobile phones which deliver the data a central server for retrieval by the user who initiated the task. SoundSense [8] is a framework for modeling sound events captured by the microphone of mobile phones and projects such as NoiseTube [12] and NoiseSpy [13] use mobile phones for monitoring the urban noise pollution

CenceMe [7] is a solution to inject sensing presence, namely users status in terms of his activity (e.g., sitting, walking), disposition (e.g., happy, sad), habits (e.g., at the gym, at work) and surroundings (e.g., noisy), into popular social networking applications such as Facebook, MySpace, and IM (Skype, Pidgin). Users can specify suitable privacy policies to control and limit the access to their data.

More in general, PCS systems have to be designed and implemented to protect the privacy of participants while allowing their devices to reliably contribute high-quality data to large-scale monitoring tasks. Anonysense [9] allows applications to submit sensing tasks that will be distributed across anonymous participating mobile devices, later receiving verified, yet anonymized sensor data reports back from the field. Anonysense has been designed to provide two key security properties: anonymity for the carriers and integrity for the sensed data. As far as anonymity concerns, the main goal of the authors is to avoid adversaries to de-anonymize a carrier. In our approach, also the service provider (i.e. the Report Service in [9]) is not aware of the carrier location. In other words, we could participate to a mapping service managed by Google without explicitly disclosing sensitive information to Google itself. Furthermore, the techniques adopted in Anonysense to support integrity, such as digital signature and encryption, require a significant computational effort and are thus not suitable for resource constrained devices. The sketching techniques we adopt in our approach allow us to guarantee a satisfactory accuracy in the reconstruction of the observed phenomenon using logarithmic space in the number of samples. Privacy issue in people urban sensing has also been discussed by Shi *et al.*[14], but their approach is based on a distributed scheme that, applying a sort of network coding, relies on data exchange among mobile nodes that is not assumed in our scheme.

Privacy preservation in location based services has already been addressed by [15], [16]. In [16], accurate traffic speed maps in a small campus town are build from shared GPS data of participating vehicles, where the individual vehicles are allowed to "lie" about their actual location and speed at all times. In our approach, data are always correct but represented in a compact and privacy preserving way (i.e sketches). Differently from [15], where data are available in clear to the intended receiver, in our work sketches allow a central authority to select relevant traces to reconstruct an accurate map, but without revealing to anybody (central authority included) relevant information on users' positions.

III. System overview

In the setting we consider, our goal is to leverage users' smartphones to sample environmental or social data in the surroundings of their current locations. Data collected in this way are delivered to a central authority for further processing. Using these data, the central authority can perform monitoring

tasks, compute statistics about ongoing environmental or social phenomena in an area of interest and answer complex queries about such phenomena. This can be done using either a participatory or an opportunistic approach. In a participatory scenario, users take active part in the sampling process, by agreeing to data sharing and configuring their devices to meet application requirements. While this approach is less challenging, it allows to exploit human carried devices in a way that is the most convenient to the application. A more challenging approach is the opportunistic one, in which users are not involved in any active step of the sampling and elaboration process. An opportunistic approach is motivated by the consideration that nowadays mobile phones are personal devices, primarily intended to offer telephony, messaging and other services. Additional services like sensing cannot be considered of primary importance and users are not likely to support them if this is going to negatively affect the performance of "primary" services or if it requires an excessive involvement in the process. For this reason, we believe opportunistic sensing should be transparent and exploit an opportunistic communication pattern, so that data collected by mobile users is delivered to the central authority whenever these connect for their own purposes.

A common issue in both paradigms is the preservation of user privacy, since the application has to collect and process data originating at users' personal devices, which might at least in part be sensitive. In particular, sampled data must typically be geo-referenced to be of any utility. As a consequence, users' movements could be easily tracked with a serious loss of privacy if these data were simply disclosed in clear. Thus, a privacy preserving representation of data is an important requirement. The above discussion implies that we have the following, general issues: i) a privacy issue, in the sense that the central authority should receive the information needed to perform its tasks, but sensitive data should be represented in a way that prevents access to sensitive information, such as geographical positions of the users; ii) a security issue, i.e., we don't want sensitive data to be intercepted by a third party during their delivery to the central authority; iii) at the same time, data should be represented in a form that allows the central authority to manipulate the data, as to extract the information that is necessary to perform its task.

A. Actors

The general scenario for people centric sensing we envision involves three main actors: mobile users, central authority and system users.

Mobile users. These are responsible for data collection. They participate to the service by running a monitoring application on their phones. This application can exploit on board sensors or other resources to sample data over environmental (Microphone, Accelerometers, Gyroscope, camera) or social (contacts, SMS, etc.) phenomena. These data are eventually delivered to a central authority, which is in charge for their processing.

Central authority. The main goal of the central authority is to elaborate data collected by mobile users, so as to provide a monitoring service to systems users.

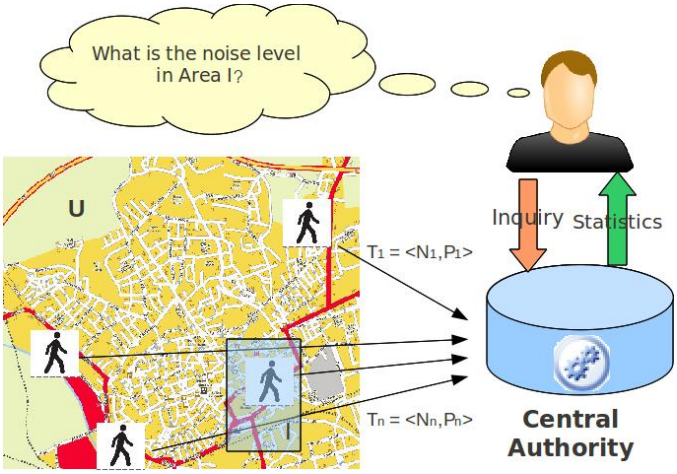

Figure 1. System Overview

System users. These submit queries that in general entail the computation of one or more statistics over the data collected by users in an area of interest. These can be for example the average noise or air pollution in the area of interest, an estimate of the number of people attending a social event etc.

B. Data collection and delivery

In the scenario we consider (see figure 1), we have n users moving in an area U and collecting data on one or more phenomena of interest. More precisely, at any point in time each mobile user maintains data about the set of the last k positions she visited. In particular, the ℓ-th *trace* generated by user i, denoted by $T_{i\ell}$ consists of k pairs $< p_i^t, n_i^t >$, with $t = 1, \ldots, k$, where p_i^t and n_i^t are respectively the t-th user position in the trace and the corresponding sampled value of the measure of interest. Let $P_{i\ell} = \{p_i^t\}_{t=1}^k$ and $N_{i\ell} = \{n_i^t\}_{t=1}^k$, be respectively the set of positions (named hereafter i's ℓ-th *position set*) occupied by the user in her ℓ-th trace and the corresponding set of samples (named the *sample set* hereafter).

We note that parameter k is very important for at least two reasons. First, it allows to exercise some control over the time scale of traces; in particular, smaller values of k allow result in traces that are more likely to refer to shorter time intervals, which can be useful in some applications. Even more importantly, the choice of k has an impact on performance, as discussed in detail in Section VI.

Also, we emphasize that n_i^t (and therefore $N_{i\ell}$) is a placeholder for any set of parameters we might want and be able to measure. In particular, n_i^t might be a tuple of values. For example, for each position we might be interested in sampling the noise and CO_2 levels, as well as record the time at which the sample was taken, so that each n_i^t would be a triple in this case.

1) Sample data delivery: As informally mentioned in the introduction to this section, in order to guarantee users' privacy, $P_{i\ell}$ should not be disclosed to third parties, including the central authority, and thus it should be represented in a suitable, privacy preserving way. The sample set can be publicly available or not, depending on the application. Without

loss of generality, we assume in the rest that sample sets can be publicly available. In this scenario, user i sends to the central authority the pair $(\mathbf{Sk}(P_{i\ell}), N_{i\ell})$, where $\mathbf{Sk}(P_{i\ell})$ is a *sketch*, i.e., a suitably generated compact summary of $P_{i\ell}$. We defer a thorough discussion about the way we generate sketches and about their mathematical properties to Subsection III-C. For the moment, suffice it so say that the sketches we use enjoy the following general properties:

- For every (i, ℓ) pair, $\mathbf{Sk}(P_{i\ell})$ represents $P_{i\ell}$ implicitly in small space (in the order of 10^4 bytes at most) and does not allow to easily infer $P_{i\ell}$;
- Considered any area I of interest, $\mathbf{Sk}(P_{i\ell})$ allows the central authority to estimate the extent to which $P_{i\ell}$ covers I. Note that this is achieved using only $\mathbf{Sk}(P_{i\ell})$, so that $P_{i\ell}$ is never explicitly disclosed to third parties.

2) Query answering: All users' traces $\mathcal{T} = T_{11}, T_{21}, .., T_{i\ell}$ are made available to the central authority according to the mechanism described above. The central authority in turn collects user traces and processes them to answer queries issued by system users. Each query involves the computation of some statistical aggregate of interest over data collected by mobile users within any area $I \subseteq U$ of interest. Denote by $\mathcal{Q}(I)$ a generic query involving area I. Upon reception of $\mathcal{Q}(I)$, the central authority i) selects the minimum number of position sets $P_{i\ell}$ ensuring the maximum possible coverage of I and ii) computes the statistical aggregate of interest over the data contained in the corresponding sample sets. We note that step i) above corresponds to the well-known, $NP-hard$ set cover problem. The central authority uses the techniques described in Section IV to perform this task. In particular, the central authority only uses the sketches $\mathbf{Sk}(P_{i\ell})$ of users' position sets and not the position sets themselves.

The general scenario described so far is summarized in Figure 1.

Remark. Note that our emphasis in this paper is on the following aspects: i) the collection, transmission and representation of position sets; ii) the way in which the central authority performs query answering by computing collections of traces that cover an area of interest, without explicitely accessing users' position sets; iii) Our main purpose is showing that monitoring tasks over georeferenced data can be efficiently performed without explicitely disclosing users' location data. The approach we propose is in fact agnostic with respect to the nature of the sampled data (i.e., the $N_{i\ell}$'s).

C. Privacy preserving data representation

As discussed before, in the application we envision, a user only sends a compact summary of her position set, from which it is hard to recover the original set. In this section we present a class of sketches [17], [18], [19] that, while compact and addressing the privacy issues mentioned above, allow the (approximate) implementation of some basic primitives on sets (such as union and intersection) that are required to implement the algorithms presented in section IV. In the rest of this subsection we present techniques used by mobile users' terminals to produce compact summaries of their respective position sets.

1) Compact representation of sets: We only briefly outline the principles underlying the technique we propose, leaving out many theoretical aspects for the sake of brevity. The interested reader can refer to [17], [18], [19]. In the remainder of this subsection, we consider without loss of generality subsets of $[n] = \{0, \ldots, n-1\}$, for a suitable integer n. In our case, this means that we are regarding position sets as subset of $[n]$, where n is the number of possible locations. We briefly note that standard techniques allow us to reduce to this situation in all practical cases[1].

Assume we have a family $\mathcal{H}$ of hash functions such that: i) every $H \in \mathcal{H}$ produces a permutation of $[n]$; ii) if H is chosen uniformly at random from $\mathcal{H}$ the following holds: for every set $S \subseteq [n]$:

$$\mathbf{P}[x = \arg\min(H(S))] = 1/|S|, \forall x \in S,$$

where $H(S)$ is the subset of $[n]$ onto which the elements of set S are hashed and where we define $\min(H(S)) = \min_{x \in S} H(x)$. Such a family is said *minwise independent* [19]. In practice, minwise independent hash functions are hard to generate, since they require a high number of truly random bits. In this paper, we use functions [20] of the form $H(x) = ((ax + b) \mod c) \mod n$, that excellently approximate minwise independent families. Here, c is a large prime (e.g., the well-known Mersenne prime $2^{31} - 1$) and n is the number of possible locations in U. Finally, $a \in \{1, \ldots, c-1\}$ and $b \in \{0, \ldots, c-1\}$.

2) Sketch generation: Considered any subset S of $[n]$, we construct its sketch as follows: for m times, we choose, independently, uniformly and at random, a hash function from a minwise independent family. Let $H_i(x)$ the i-th function chosen. Then the sketch of S is $\mathbf{Sk}(S) = \{\min(H_1(S)), \ldots, \min(H_m(S))\}$. In our case, the generic i-th hash function has the form described above, i.e., $H_i(x) = ((a_i x + b_i) \mod c) \mod n$. In practice, generating such a hash function means generating a_i and b_i uniformly at random from $\{1, \ldots, c-1\}$ and $\{0, \ldots, c-1\}$ respectively.

3) Sketch properties: The sketch generation technique we consider enjoys interesting properties that prove extremely useful to solve the problem we consider. They are briefly discussed below.

Composability with respect to set union. Given sets S_1 and S_2, the sketch of $S_1 \cup S_2$ can be immediately obtained from $\mathbf{Sk}(S_1)$ and $\mathbf{Sk}(S_2)$ using the following, obvious fact: $\mathbf{Sk}(S_1 \cup S_2) = \{M_1, \ldots, M_m\}$, where $M_i = \min\{\min(H_i(S_1)), \min(H_i(S_2))\}$.

Estimation of the Jaccard coefficient. Another interesting property of these sketches is that they allow to easily and accurately estimate the Jaccard coefficient of two sets, a standard measure of the similarity between sets, widely used in information retrieval and a key ingredient to solve the problem we are interested in. Given two subsets S_1 and S_2 of $[n]$, their

[1]In our case, the ℓ-th position set $P_{i\ell}$ of a user i is a finite set of geographical positions (e.g., GPS coordinates). As such, it can be put in correspondence with a subset of the integers using standard techniques, e.g., Rabin's fingerprinting method [19].

Jaccard coefficient is defined as

$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}.$$

It can be shown [19] that for every S_1, $S_2 \subseteq [n]$, if hash function $H(\cdot)$ produces a permutation of $[n]$ and is chosen uniformly at random from a minwise independent family then the following holds:

$$\mathbf{P}[\min(H(S_1)) = \min(H(S_2))] = J(S_1, S_2).$$

This suggests a simple statistical estimator of the Jaccard coefficient of two sets and motivates the sketch we adopted for compact set representation. To estimate $J(S_1, S_2)$, we simply consider their sketches $\mathbf{Sk}(S_1)$ and $\mathbf{Sk}(S_2)$ and let $C_m = |\{i : \min(H_i(S_1)) = \min(H_i(S_2))\}|$. Then, the result above and a simple high probability argument allow to show that C_m/m is an increasingly accurate estimator of $J(S_1, S_2)$.

4) Compact representation of position sets: As already discussed in the introduction of this section, every mobile user over time submits to the central authority a sketch of her most recent trace, i.e., a sketch of the set of the last k positions she visited. The general technique adopted to produce sketches of integer subsets has been discussed in the paragraphs above. Below, we describe how this is implemented in the mobile scenario we consider.

Hash function generation. All mobile users will use the same set $H_1(\cdot), \ldots, H_m(\cdot)$ of minwise independent hash functions. These will be generated by the central authority and then sent to each mobile user once, i.e., the first time she joins the application. Note that, in practice, the linear functions we use are represented in terms of a small set of parameters. For example, if we use 100 hash functions, each mobile user will need to receive 202 integer values: for the generic i-th hash functions, the coefficients a_i and b_i, plus c and n, which are the same for all hash functions. This amounts to a total of less than 1 KByte, if integers are represented using 4 bytes.

Sketch generation and update. Mobile user i will generate a sketch of her ℓ-th position set incrementally as follows: her sketch $\mathbf{Sk}(P_{i\ell})$ is initialized $\{0, \ldots, 0\}$. Let $\{M_1, \ldots, M_m\}$ be i's sketch at some point. If she moves to a new position p (e.g., identified by the GPS coordinates of a new base station she connects to), then $\mathbf{Sk}(P_{i\ell})$ is updated as follows: $M_j = \min\{M_j, H_j(p)\}, \forall j = 1, \ldots, m$. This sketch update corresponds to updating i's position set as follows: $P_{i\ell} = P_{i\ell} \cup \{p\}$, thanks to the property of sketch composability with respect to set union discussed in Subsection III-C.3. Note that in the above paragraph we have assumed that p is an integer. In fact, positions are preprocessed, so that every GPS position is mapped onto a 32-bit integer and different coordinates are mapped onto different integers. There are several efficient ways to achieve this, the simplest of which is to regard the binary representation of a coordinate pair as an unsigned integer. In our experimental analysis, to make simulation faster, we used a dictionary to give the association between GPS coordinates and integers. Though the use of a dictionary would be unnecessary in practice, it is interesting to note that its size is in the order of 10^5 bytes and thus perfectly compatible with the storage resources available on commercial devices.

IV. Set Covering without the Sets

As discussed in Section III, the central authority serves queries submitted by system users. The generic query $\mathcal{Q}(I)$ requires the central authority to compute some statistic over the data sampled in an area I of interest. To accomplish this task, the central authority uses the sampled data received from the mobile users. We remind the reader that these are in the form $(\mathbf{Sk}(P_{i\ell}), N_{i\ell})$, where $\mathbf{Sk}(P_{i\ell})$ is a sketch of the ℓ-th position set of the i-th user and $N_{i\ell}$ is the set of values sampled in each of the positions of $P_{i\ell}$. So, the main problem is finding a collection of positions sets (i.e., a collection of $P_{i\ell}$'s) that together cover area I. For the sake of efficiency, this collection should have the smallest possible cardinality. Once such a collection has been found, the statistic can be estimated by considering, for every position set, the corresponding sampled data $N_{i\ell}$. Two issues arise here: i) finding a minimum cardinality collection of position sets that together cover I is an instance of the $NP - hard$ set cover problem; ii) the central authority does not know the $P_{i\ell}$'s explicitly, but only their sketches $\mathbf{Sk}(P_{i\ell})$. We address these issues in the following two subsections.

Before doing this, we remind the reader that in the classical minimum set cover problem [21] we are given a set I, taken from a universe U, and a collection $\mathcal{T} = T_1, T_2, .., T_n$ of subsets of U. The pair $(U, \mathcal{T})$ is called a *set system*. The goal is to compute a sub-collection $\mathcal{T}' \subseteq \mathcal{T}$ which covers I with minimum cost, namely using the smallest number of sets in $\mathcal{T}$. In our setting, each T_l is a position set.

A. Greedy Algorithm for Set Cover

The best known algorithm for the minimum set cover problem is the greedy algorithm summarized in Figure 2 [21].[2]. Without loss of generality, in the pseudo-code the set I to cover is assumed to coincide with the universe U. If this is not the case, we simply replace each set T_l of the set system with $T_l \cap I$. The algorithm is very simple: it maintains a collection C of sets that will belong to the cover. Iteratively, in each step it selects the set in $\mathcal{T} - C$ that covers the largest number of still uncovered elements of U (lines 3 and 7) and adds it to the collection (line 6).

Since our purpose is to give a sketch-based version of the greedy algorithm above and since it seems hard to compute the sketch of the difference of two sets from their respective sketches, we slightly modify Algorithm `Standard-Greedy`, by replacing $|T \cap (U - C)|$ with $|(T \cup C) \cap U|$ in steps 3, 4 and 7. Maximizing the former quantity is equivalent to maximizing the latter, as proved in the following:

Fact 1: In Algorithm 2, maximizing $|T \cap (U - C)|$ is equivalent to maximizing $|(T \cup C) \cap U|$.

[2]In order to make the pseudo-code more readable, we slightly abuse notation, since we regard C as both a set of sets (the set cover) in lines 1 and 9 and as the union of the sets that form the cover in lines 3, 4, 6 and 7. Analogous considerations hold for Algorithm 3.

Algorithm `Standard-Greedy`

Require: set system $(\mathcal{T}, U)$
1: $C = \emptyset$ (C contains identifiers of sets in set cover)
2: $\hat{\mathcal{T}} = \mathcal{T}$
3: $\hat{T} = \arg\max_{T \in \hat{\mathcal{T}}} |T \cap (U - C)|$
4: **while** $|T \cap (U - C)| > 0$ **do**
5: $\quad \hat{\mathcal{T}} = \hat{\mathcal{T}} - \{\hat{T}\}$
6: $\quad C = C \cup \{\hat{T}\}$
7: $\quad \hat{T} = \arg\max_{T \in \hat{\mathcal{T}}} |T \cap (U - C)|$
8: **end while**
9: return C

Figure 2. Greedy Algorithm for Set Cover.

Proof: Consider the generic iteration of Algorithm 2 and assume the partial cover is the same for both versions of the algorithm at the beginning of the iteration. This is clearly the case during the first iteration, if both versions of the algorithm break ties the same way. For every set $T \in \hat{\mathcal{T}}$ we have:

$$(T \cup C) \cap U = (T \cap U) \cup C =$$
$$= (T \cap (U - C)) \cup (T \cap C) \cup C =$$
$$(T \cap (U - C)) \cup C$$

where the first equality follows since $C \subseteq U$, while the third follows since $T \cap C \subseteq C$. Since C is fixed (it is the partial set cover computed at the end of the previous iteration), maximizing $|(T \cup C) \cap U|$ is equivalent to maximizing $|(T \cap (U - C))|$. ■

Another important fact is the following:

Fact 2: Maximizing $|(T \cup C) \cap U|$ is equivalent to maximizing the Jaccard coefficient between $T \cup C$ and U.

The proof of this fact is obvious and follows since $T, C \subseteq U$.

B. Sketch-based Greedy Algorithm for Set Cover

We next describe `PP-Greedy`, a sketch-based version of Algorithm `Standard-Greedy`. The main difference is that in `PP-Greedy`, set operations have been replaced by simple operations on the corresponding sketches.

In particular, in lines 5 and 11 of figure 3, following Fact 2, we choose the set T, such that the (estimated) Jaccard coefficient between $T \cup C$ and U is maximized, since this set also maximizes $|(T \cup C) \cap U|$. Recalling the discussion in Subsection III-C.3 this is, up to approximations, the set T such that $\mathbf{Sk}(T \cup C)$ and $\mathbf{Sk}(U)$ agree on the largest possible number of components. In the pseudo-code of Figure 3, $Eq(U, C \cup T) = \sum_{i=1}^{m} X_i$, where $X_i = 1$ if the i-th components of $\mathbf{Sk}(C \cup T)$ and $\mathbf{Sk}(U)$ have the same value 0 otherwise.

V. Privacy and Security

The next two subsections discuss the privacy and security aspects of the approach we propose. In particular, Subsection V-B describes and discusses a simple protocol to enforce security in the framework we consider in a cryptographic sense. Though this latter aspect is not the focus of our paper, we think it might be of interest and it also demonstrates the flexibility of the approach we propose and its compatibility with state-of-art cryptographic techniques.

Algorithm `PP-Greedy`

Require: Sketch $\mathbf{Sk}(T_i)$, for $i = 1, \ldots, |\mathcal{T}|$, $\mathbf{Sk}(U)$
1: $E = 0$
2: $C = \emptyset$ (C contains identifiers of sets in set cover)
3: $\mathbf{Sk}(C) = \{\infty\}_{i=1,\ldots,m}$
4: $\hat{\mathcal{T}} = \mathcal{T}$
5: $\hat{T} = \arg\max_{T \in \hat{\mathcal{T}}} Eq(U, C \cup T)$
6: $\hat{E} = Eq(U, C \cup \hat{T})$
7: **while** $\hat{E} > E$ **do**
8: $\quad E = \hat{E}$
9: $\quad \hat{\mathcal{T}} = \hat{\mathcal{T}} - \{\hat{T}\}$
10: $\quad C = C \cup \hat{T}$
11: $\quad \hat{T} = \arg\max_{T \in \hat{\mathcal{T}}} Eq(U, C \cup T)$
12: $\quad \hat{E} = Eq(U, C \cup \hat{T})$
13: **end while**
14: return C

Figure 3. Privacy Preserving Greedy Algorithm for Set Cover.

A. Quantifying Location Privacy

Since Mobile users disclose their locations to possibly untrusted entities, or may unwillingly expose private information to malicious eavesdropping entities over the wireless channel, it's necessary to define an attacker model to quantify the privacy achieved by the proposed approach. We take [22] as a reference framework for privacy quantification of our approach, framing it into the family of Location Privacy Preserving Methods (LPPM) based on obfuscation by means of precision lowering. In order to evaluate our approach as an LPPM, we must model the adversary against whom the protection is placed. In our reference scenario we consider two kind of adversaries: an external attacker eavesdropping the obfuscated traces sent from the user to the central authority, and the central authority itself; recall that a unique characteristic of our system is that the central authority should be able to compute relevant statistics on data without knowing the users' positions.

a) Type of attack: Since the most powerful attacker is clearly the central authority (it knows the hash functions and all the system parameters), in the following we consider the case in which an untrusted central authority performs an attack aimed at recovering users' traces from their sketches. We next discuss the amount of location information an attacker can recover from a sketch. We assume in the remainder of this paragraph that the attacker knows the hash functions used to produce sketches of the position sets and the sketch $\mathbf{Sk}(P_{i\ell})$ and size of a position set $P_{i\ell}$ she intends to recover as accurately as possible. The basic operation an attacker can perform is the following: given a position p and $\mathbf{Sk}(P_{i\ell})$, check whether or not $p \in P_{i\ell}$. To this purpose, the attacker

computes $H_j(p)$, for $j = 1, \ldots, m$. Setting in the remainder of this paragraph $S = P_{i\ell}$ and $|S| = s$ for the sake of brevity, the following cases can occur (see Subsection III-C): i) $H_j(p) < \min(H_j(S))$ for at least a j. In this case, the attacker can conclude that $p \notin S$; ii) $H_j(p) = \min(H_j(S))$ for at least a j. In this case, the attacker can conclude that $p \in S$; iii) $H_j(p) > \min(H_j(S))$ for all j's. In this case, p is a *false positive* (i.e., p does not belong to S but it behaves as if it did, since $H_j(p) > \min(H_j(S))$ for all j's) with some probability that depends on s and m, as we see further.

Note the important fact that from case ii) above, the adversary can claim membership in S for at most m positions in U. For all other positions, it can either conclude that they don't belong to S (case i)) or nothing (case iii)).

In the remainder, we make the worst case assumption that resources are not an issue for the attacker and it can perform the above described basic operation for every potential candidate position in the universe U. We discuss below how to make the probability that case iii) occurs large enough that, even under a brute force attack using the approach described above, a large fraction of positions in $U-S$ are false positives. As a result, if $|S|$ is sufficiently larger than m, the attacker may be able to recover at most m of the user's locations with certainty, but a large fraction of all possible positions will be potential candidate members of S, thus making the data recovered by the adversary extremely noisy and of little use.

To this end, we next study the probability that, given a position $p \notin S$, $H_j(p) < \min(H_j(S))$ for at least a j. To this purpose, we assume that the hash functions are minwise independent, though this in practice is not the case. The paper [20] discusses how the functions we use closely approximate the ideal minwise behaviour. From the definition of a minwise independent family (see Subsection III-C) we have:

$$\mathbf{P}[H_j(p) < \min(H_j(S))] = \frac{1}{s+1},$$

where the probability is taken with respect to the initial, random choice of the $H_j(\cdot)$'s and where we remind the reader that we are assuming $p \notin S$. Next, for every $p \notin S$, we define the binary variable $X_p = 1$ if p is a false positive, $X_p = 0$ otherwise. We also define $X = \sum_{p \notin S} X_p$. Notice that:

$$\begin{aligned}\mathbf{P}[X_p = 1] &= \mathbf{P}\left[\bigcap_{j \in [m]} H_j(p) > \min(H_j(S))\right] \\ &= \left(1 - \frac{1}{s+1}\right)^m,\end{aligned}$$

where the first equality follows from the definition of case iii) above. As a consequence, we can conclude:

$$\mathbf{E}[X] = \sum_{p \notin S} \mathbf{P}[X_p = 1] = |U - S| \left(1 - \frac{1}{s+1}\right)^m,$$

where we remind the reader that U denotes the universe of all possible positions. Now, for a fixed *constant* $0 < \delta < 1$, note that we have $\mathbf{E}[X] \geq \delta |U - S|$, whenever

$$\left(1 - \frac{1}{s+1}\right)^m \geq \delta.$$

This happens as soon as

$$s \geq \frac{2m}{\ln \frac{1}{\delta}} - 1.$$

This result follows from simple calculus after observing that:

$$\left(1 - \frac{1}{s+1}\right)^m \geq e^{-\frac{2m}{s+1}}$$

and imposing that the right-hand side be at least δ.[3] This result tells us that, as soon as the size of position sets is large enough (a constant number of times the number of hash functions used), the expected number of false positives can be extremely high.

For example, if we set $\delta = 1/e$, we obtain for s the condition $s \geq 2m - 1$. Under this condition, the expected number of false positives is at least $\frac{1}{e}|U - S|$. Considering that $|U| >> |S|$ in practice, it follows that a large fraction of all possible positions are false positives. Assuming that the attacker is able to recover the at most m positions of S that achieve the minima of the hash functions used to generate $\mathbf{Sk}(S)$, recovering the missing $s-m$ ones may be a non trivial task, given that in expectation, at least $\frac{1}{e}|U-S|$ positions will be perfectly equivalent candidates. It is clear that knowing a subset of the positions in a trace may help prune many unlikely positions based on geographical proximity to known positions, but this strategy becomes less and less effective as the size of the position set grows and in any case, it is not possible to distinguish false positives from positions that actually belong to S.

B. Enforcing security

In previous sections we have described a privacy preserving, sketch-based representation of position sets. As discussed in the previous Subsection V-A, this technique prevents an external attacker or the central authority itself from recovering significant portions of the original set. Still, we have seen that an attacker (or the central authority) could still recover a portion of a user's position set using essentially a brute force approach, at least under the assumption that the attacker knows all hash functions used to produce positions set sketches (this assumption holds for the central authority but it might be unrealistic for an external attacker). Furthermore, as discussed in section III, we outlined how another general issue regards security, namely the risk for sensitive data to be intercepted by a third party during their delivery to/from the central authority (for example, the parameters of the hash functions to be used). Sketches offer some degree of privacy preservation but still some minor privacy leaks in this technique can be exploited to grasp some information about the set. For this reason, we discuss here the main aspects of an additional protocol that can be transparently applied to the sketch technique, to enforce privacy and ensure data security. Let us consider two sets A and B, each holding a vector of length m. For the sake of simplicity, let's denote each party with the name of the set she owns (A and B). In our application to the computation of

[3] The inequality above follows since $e^{-2x} \leq 1 - x \leq e^{-x}$ for $0 \leq x \leq \frac{1}{2}$. We choose $x = 1/(s+1) \leq \frac{1}{2}$ in our case.

the Jaccard coefficient, the vectors will be the sketches of the respective position sets. Assume that A and B wish to compute the number of positions i for which $A[i] = B[i]$ without revealing any additional information on the vectors. Coming back to the application context, we have two parties A (Mobile User) and B (Central authority), each with a private set of positions. Namely, A holds a user's position set $P_{i\ell}$, while B holds an area I of interest. A and B wish to compute the Jaccard coefficient $J(P_{i\ell}, I))$ (the key step of the `PP-Greedy` algorithm in Figure 3) in order to perform the set cover. We will describe a protocol that uses an additively homomorphic encryption scheme $(E; D; K)$ like Paillier cryptosystem (see [23] for further information).

Homomorphic encryption scheme. Let $(E; D; K)$ be a homomorphic encryption scheme and assume that the message space for a public key pk returned by the key generator algorithm K on input security parameter m is $\mathbb{Z}_p$ for some integer p of length m. The following additive homomorphic properties hold:

1) the product of two ciphertexts is a ciphertext for the sum of the plaintexts; that is, for all messages $a; b \in \mathbb{Z}_p$ and public keys pk, we have $D(E(pk, a)\cdot E(pk, b), sk) = a + b$;
2) raising a ciphertext for message a to power r gives a ciphertext for $r\cdot a$; that is, for all $r \in \mathbb{Z}_p$ we have that $D(E(pk, a)^r, sk) = r\cdot a$.

The protocol. The protocol can be described as follows:

1) A picks a pair of public and secret key (pk, sk) for encryption scheme (E, D, K) by running the key generator algorithm K on input 1^m; for $i \in [n]$, A computes encryption $a_i = E(pk, A[i])$ of $A[i]$; A sends pk and $(a_i)_{i\in[n]}$ to B;
2) for $i \in [n]$, B computes encryption $b_i = E(pk, -B[i])$ of $-B[i]$, picks random $r_i \in \mathbb{Z}_p$ and sets $c_i = (a_i + b_i)^{r_i}$. Notice that by the homomorphic properties of (E, D, K), c_i is a ciphertext for $r_i\cdot(A[i] - B[i])$. Therefore if $A[i] = B[i]$, then c_i is an encryption of 0; otherwise c_i is an encryption of a random element of $\mathbb{Z}_p$. B randomly permutes the c_i's and sends them to A.
3) A decrypts the m ciphertexts received from A, counts the number s of ciphertexts that are an encryption of 0 and sends s to B.

Properties of the protocol. We make the following simple observations: *Correctness.*The value s computed by the protocol is the number of indices i for which $A[i] = B[i]$, with probability exponentially close to 1. *Privacy of the input.* Each of A and B gets no information on the other party's vector, besides what can be obtained from the output of the protocol. For A, this can be easily seen by exhibiting a probabilistic polynomial-time simulator S that, for all vectors A and B, on input vector A and the number s of positions in which A and B coincide (but not vector B) outputs A's view of the protocol. Similarly, we can construct a simulator for B.

Obviously, the Jaccard coefficient can be computed by applying the above protocol to the characteristic vector of the two sets. The protocol will then run in time linear in the size of the underlying universe set. A much more efficient protocol is instead obtained by running the above protocol with each party holding as an input the sketch of PositionSets computed using the same sequence of random (or min-wise independent) permutations. Depending on security requirements this protocol can be used as a black box, since it doesn't have impact on the performances of the sketching techniques we discussed.

VI. Experimental Analysis

In this section, we present and discuss the results of extensive experimental analysis aimed at assessing the performance of our approach, in particular the performance of the `PP-Greedy` algorithm and how it compares to the `Standard-Greedy` algorithm, whose behaviour it tries to imitate without explicitly knowing the position sets of the user, but only their respective sketches. We first investigate the effect of increasing the number of hash functions used to calculate sketches. We then evaluate the performance of the algorithms when varying the sizes of the position sets (i.e. $P_{i\ell}$) and of the area of interest I. This section is organized as follows: Subsection VI-A briefly recalls the behaviour of the two algorithms we consider. Subsection VI-B describes the metrics we define for the comparison of the algorithms, Subsection VI-C describes the dataset we used for our experiments, and finally Subsections VI-D and VI-D.2 analyse the performance of the algorithms.

Notation. For the sake of brevity, in the rest of this section we denote by Gr and $PP - Gr$ respectively the `Standard-Greedy` and the `PP-Greedy` algorithm. Assume the overall number of position sets is s. For ease of notation, we assume an arbitrary order of the position sets (for example, the order in which their sketches were received by the central authority) and we denote them by $P_1, .., P_s$. We also let $P = \{P_1, .., P_s\}$.

A. Algorithms

The Greedy algorithm Gr receives in input the area of interest I and the position sets of the users. Note that every user may provide a different number of position sets. Gr provides as output a *set system* $P_{Gr} \subseteq P$ that approximates the minimum cardinality set cover of I. The PP-Greedy algorithm $PP - Gr$, instead of the set P, receives in input the set of sketches $Sk_P = (Sk(P_1), .., Sk(P_s))$, and it also provides as output a set $P_{PP-Gr} \subseteq Sk_P$ that approximates the minimum cardinality set cover of I.

Note the following: i) in both cases the output of the algorithm is simply a collection of identifiers of the position sets that provide a cover for I and not the sets themselves. This information will allow the identification of the traces to obtain in order to compute the statistic of interest over I and of the mobile users that possess the data. The difference is that, while Gr needs the position sets to perform this task, $PP - Gr$ only uses their sketches; ii) Gr provides the best coverage possible of area I, though its output might not be a cover of minimum cardinality, since Gr is an approximation algorithm for an NP-hard problem. The only reason why the coverage of Gr may not be 100% is that some positions in

I might not be covered by any position set, as indeed is the case with the dataset we consider.

Remark. Note that we only compare our approach with the standard greedy heuristic. We believe this is not a limitation, since the standard greedy heuristic provides the best possible approximation of the optimal solution that can be achieved by a polynomial time algorithm. Improving over the greedy heuristic entails considering approaches (such as Linear Integer Programming) that can have exponential computational times and are in our opinion not very realistic in the scenario we consider, in which the size of the input can be very large.

B. Metrics

We evaluate the performance of Gr and $PP-Gr$ with respect to three metrics.

- The *cardinality* of output i.e., the number of position sets used to cover the area of interest,
- The *coverage* of the output, intended as the fraction of positions in the area of interest that are covered by the output.[4]
- The *error*, defined as the fraction of positions in the output which are not in I.

As an example consider the following sets $I = 1, 3$, $P_{Gr} = 1, 2, 3, 4$. In this case the cardinality is 2, the coverage is 100% and the error is 50%.

C. Dataset

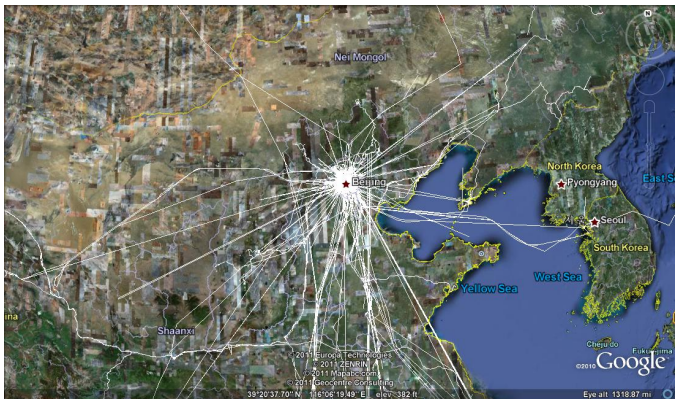

Figure 4. Google Earth's representation of GeoLife traces

The input for our experiments has been obtained from a real public dataset published by the Microsoft GeoLife project [24], [25]. The dataset contains GPS trajectories collected by 165 users in a period of over two years (from April 2007 to August 2009). To the best of our knowledge, this is the largest publicly available dataset that summarizes a broad range of users' outdoor movements, including routine movements such as driving to work or back home, but also entertainment and sports activities, such as shopping, sightseeing, dining, hiking, and cycling. Therefore, the dataset can be used in many research fields, such as mobility pattern mining, user activity recognition, location-based social networks, and location recommendation. This dataset contains approximately 22 milions of position records, mostly concentrated in the area of Beijing (China), as results from figure 4.

[4]We calculate the coverage of P_{PP-Gr} considering the set of corresponding positions.

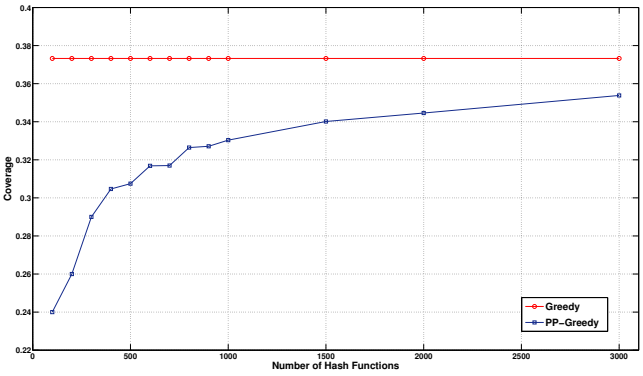

Figure 5. Coverage achieved by both algorithms with varying number of hash functions for PP-Greedy

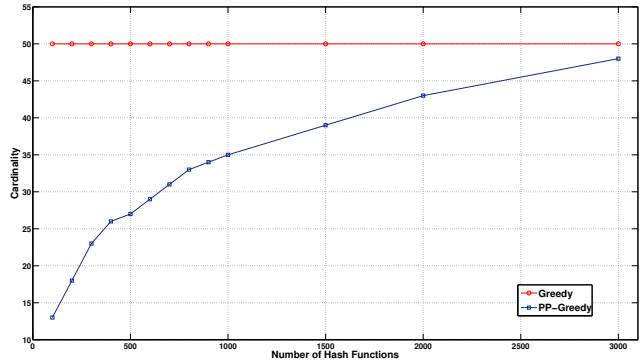

Figure 6. Cardinality achieved by both algorithms with varying number of hash functions for PP-Greedy

D. Experimental Results

1) Role of Hash function number on the PP-Greedy performance: Recall that, as observed at the end of Subsection III-C.3, it is easy to prove that the fraction of positions on which the sketches of two sets agree is an increasingly (wrt the number of hash functions) accurate estimator of the Jaccard coefficient. In this section we evaluate to which extent the accuracy of the estimator increases with the number of hash functions. To this purpose, we consider a reference area of interest I to be covered and we compare the results of both algorithms on multiple runs and with an increasing number of hash functions used by $PP-Gr$. The results are averaged over 10 runs of both algorithms, for each of the reported number of hash functions, keeping the same I for all experiments. Since the coverage achieved by the Gr algorithm is the highest possible, this algorithm is used as a baseline to evaluate the coverage of `PP-Greedy`. Figure 5 describes the coverage achieved by both algorithms with a number of hash functions used by $PP-Gr$ that goes from 100 to 3500.

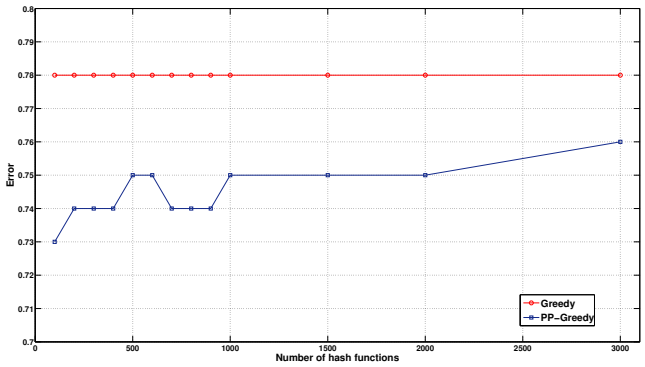

Figure 7. Error achieved by both algorithms with varying number of hash functions for PP-Greedy

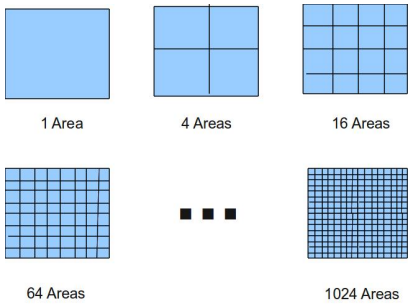

Figure 8. Different areas I, with varying size, used for the experiments

As expected, the performance of `PP-Greedy` in coverage increases with the number of hash functions used. In particular the `PP-Greedy` line in figure 5 shows an asymptotic behaviour approximating the `Standard-Greedy` one. The plot can be divided into 2 parts: when the number of hash functions $m < 1000$, the coverage significantly increases from a very low coverage with 100 Hash (~ 0.23 compared to the 0.38 of the Greedy), to a value of 0.34 with 1000 Hash functions. In the second part, the increase of coverage is smaller, with less than 0.1 every 500 additional Hash functions.
On the other side, figures 7 and 6 respectively show the values of error and cardinality for the same experiments. While cardinality follows an asymptotic behaviour as well as coverage, the error tends to remain high irrespectively of the number hash functions.

This can be explained as follows: when the number of hash functions is small, as observed before, the resolution of sketches and the ability to estimate similarity between sets is small, which results in a higher error rate. As the number of hash function increases, the accuracy in set similarity estimation improves and this brings to an increase in coverage and cardinality. However, improving accuracy in similarity estimation implies that a number of sets with a marginal overlap with the area of interest are included in the final solution, thus increasing the error. Furthermore, we note that the number of hash functions is an upper bound on the cardinality of the solution. This is trivially due to the fact that in each iteration, if the `PP-Greedy` algorithm adds a new position set $P_{i\ell}$ to the partial solution, than it has to be the case that it strictly increases the number of entries of the partial solution's sketch that agree with the corresponding entries of $\mathbf{Sk}(I)$ (see the description of algorithm `PP-Greedy` in Figure 3 and Subsection IV-B). Of course, this can happen at most m times.

The results described above suggest that the larger the number of hash functions we use, the better the performance of `PP-Greedy`. On the other hand, we also point out that a number m of hash functions generates a sketch consisting of m integers, for an overall size of $4m$ bytes. Recalling that the application scenario we refer to can involve mobile devices that communicate opportunistically, available bandwidth has to be considered a scarce resource. Under these circumstances, larger sketches can provide solutions of better quality, but they can be in contrast with these network constraints.

Observing the figures, beyond 1000 hash functions, the marginal increase in performance is less significant. For this reason, in the following experiments we consider 1000 hash functions as a good trade-off between the size of the sketches, which is in this case limited to 4KB, and the quality of the approximation of the `Standard-Greedy` algorithm given by `PP-greedy`.

2) Role of the P_i's and I on $PP-Greedy$ performance: As described at the end of the previous section, we consider the `PP-Greedy` algorithm using 1000 hash functions and we analyze its performance as the size of the area of interest I and of the position sets P_i vary. The objective of this analysis is to investigate the relations that exist between these two variables in order to maximize the performance of `PP-Greedy`.

The reference area, namely the world U considered for our experiments is the city of Beijing. This area has been splitted into smaller areas I by dividing at each step the reference area by two as shown in figure 8. The resulting areas used to evaluate the impact of the size of I on the performance of the algorithms are summarized in the following:

- 1 Area with size $\sim$ 71000 discrete GPS locations ($I \equiv U$),
- 4 areas I of size $\sim$ 17900 discrete GPS locations,
- 16 areas I of size $\sim$ 4480 discrete GPS locations,
- 64 areas I of size $\sim$ 1180 discrete GPS locations,
- 256 areas I of size $\sim$ 300 discrete GPS locations,
- 1024 areas I of size $\sim$ 90 discrete GPS locations.

On the other side we set the size of the position sets used to perform the Set-cover of I, namely the size of the P_i's,to the following values:10, 50, 100, 150, 200, 500, 1000, 1500, 2000, 2500, 3000, 3500 samples.

Remark. As noted at the beginning of this subsection, we used 1000 hash functions in the experiments that we present below. On the other hand, we considered positions set sizes above but also well below this value. It is clear that using sketches larger than the size of the position sets they represent has little sense in terms of efficiency. The only reason why we

also considered sizes less than 1000 for position sets was to analyze the interplay between this parameter and the size of I, as summarized for example in Figure 9.

The system has to be designed considering a complex trade-off among accuracy, efficiency and privacy: while increasing the number of hash functions used clearly improves accuracy, the ratio between the size of position sets and this parameter plays a crucial role. The larger this ratio, the more efficient the system of course, since we are representing positions sets using smaller and smaller numbers of bits. Furthermore, a higher ratio between these two parameters improves privacy, as discussed earlier. On the other hand, increasing the size of positions sets reduces the granularity of the users' traces (for a given area I of interest, larger position sets are more likely to include a larger number of irrelevant positions) and thus negatively affects accuracy.

The experimental results discussed in the following, are obtained averaging the results of 10 runs for each possible combination of I's and P_is' sizes.

a) Coverage.: Figure 9 shows the values of coverage obtained in the experiments as a function of the size of the P_i's for each area possible size of the area of interest I. As already pointed out, the coverage of Gr is the highest possible and it does not depend on these parameters. Its value is 0.38. On the contrary, the value achieved by $PP-Gr$, significantly changes as the sizes of the P_i's and/or I change.
Worst performances are obtained when the sizes of I and the $P_i's$ differ significantly, due to the fact that the resulting Jaccard coefficient is fairly low. Indeed, if $|P_i| \ll |I|$, then $J(P_i, U) = \frac{|I \cap P_i|}{|I \cup P_i|} \simeq \frac{P_i}{I} \ll 1$. This implies that the number of minima that are common to both the sketches $\mathbf{Sk}(P_i)$ and $\mathbf{Sk}(I)$ is low, and thus the number of sets not included in the solution is high, resulting in a low value of coverage. On the contrary, when $|P_i| \simeq |I|$, we have $J(P_i, I) = \Theta(1)$ and consequently more sets will be included in the final solution. This is also confirmed in Figure 9, in which position sets of smaller size are more suitable to cover small areas of interest and viceversa. Apparently, the best performance are obtained when $|I| = |U| = 71000$ and $|P_i| = 3500$. However this is the limit case when the area I of interest is the whole world U we consider, namely the whole city of Beijing in the experiments. In this case, the error is zero, because all the elements in any P_i are necessarily part of the world we are considering. For this reason, we do not consider this particular case in the discussion that follows. The best coverage is obtained at the boundaries of the graph in Figure 9, when $|P_i| = 10$ and $|I| = 300$ or when $|I| = 17920$ and $|P_i| = 3500$. In those regions, the best performer is about 50% better than the worst one. On the contrary, the region in the middle of the graph, for sizes of $|P_i|$ between 100 and 200, is characterized by differences between best and worst performers of only 15%, but it is also the region where the coverage is lower and is about half the best coverage.

The performance of the system in terms of coverage is optimized when is possible to tune the sizes of the P_i's according to the size of I or viceversa. However, the size of the P_i's has an impact on the security (see section V-A) and on the amount of required resources (i.e. memory and bandwidth) of the mobile devices. Since the size of a sketch is independent of the size of P_i from which the sketch has been generated, it turns out that smaller P_i's generate a greater number of sketches that have to be stored and sent either establishing a new connection or opportunistically exploiting available ones. As an example, consider a mobile device collecting samples every minute, if the the size of the P_i's is 10, this device will send 6 sketches every hour, while if the limit is 60, it will send only a sketch every hour.

b) Cardinality and Error.: We now focus on the other two metrics to evaluate the performance of our system, namely cardinality and error. In this case, also the performance of the `Standard-Greedy` algorithm depends on the size of the P_i's and I. Figures 10, 12, 11 and 13 depict the cardinality and error values for both Gr and $PP-Gr$ algorithms. These four plots show a common behaviour for both algorithms. As expected, the bigger the size of the P_i's, the lower the number of sets selected by the algorithm, namely the cardinality. Furthermore, areas of interest I of bigger size are always characterized by solutions of higher cardinality (number of position sets in the *set system*).

Figure 14 shows two lines obtained as the average of the values shown in Figures 10 and 11. In this figure, the trend of the two algorithms is confirmed to be similar, but it is now evident that $PP-Gr$ always selects a lower number of sets in its solution.

Similar considerations apply to the error metric, as shown in Figures 12 and 13. The larger the size of the P_i's, the larger the error for both algorithms. Indeed, position sets of big size and only marginally overlapping with the area of interest, marginally contribute to improve coverage, but they cause substantial error increases, because most of their positions are outside the area of interest.

Thus, it follows that P_i 's of small size allow to perform the set cover with a finer granularity and thus decrease error; on the other hand, it is worth to analyze how an excessive small granularity impacts on the error for the `PP-Greedy` algorithm.

Analogously to what we have done for the cardinality, Figure 15 depicts the average error. The error of $PP-Gr$ is constantly lower than the error of the `Standard-Greedy` algorithm, except for the first point of the plot. The reason for this discrepancy, has to be found in the intrinsic limit of the sketch-based approach to estimate similarity between small sets (i.e. position sets) and sets of much larger size (the area of interest). In fact, when the size of the P_i's is 10 (i.e. the smallest size of the position sets), $PP-Gr$ has a higher average error. Comparing Figures 12 and 13, it appears that the higher average error in this case is essentially due to input instances in which $|I|$ is relatively small and comparable to 10, so that $PP-Gr$ has a non-negligible probability of detecting position sets that have small, but non empty intersection with I.

Discussion. From the above experiments we have learnt some important lessons about the performance of the `PP-Greedy`

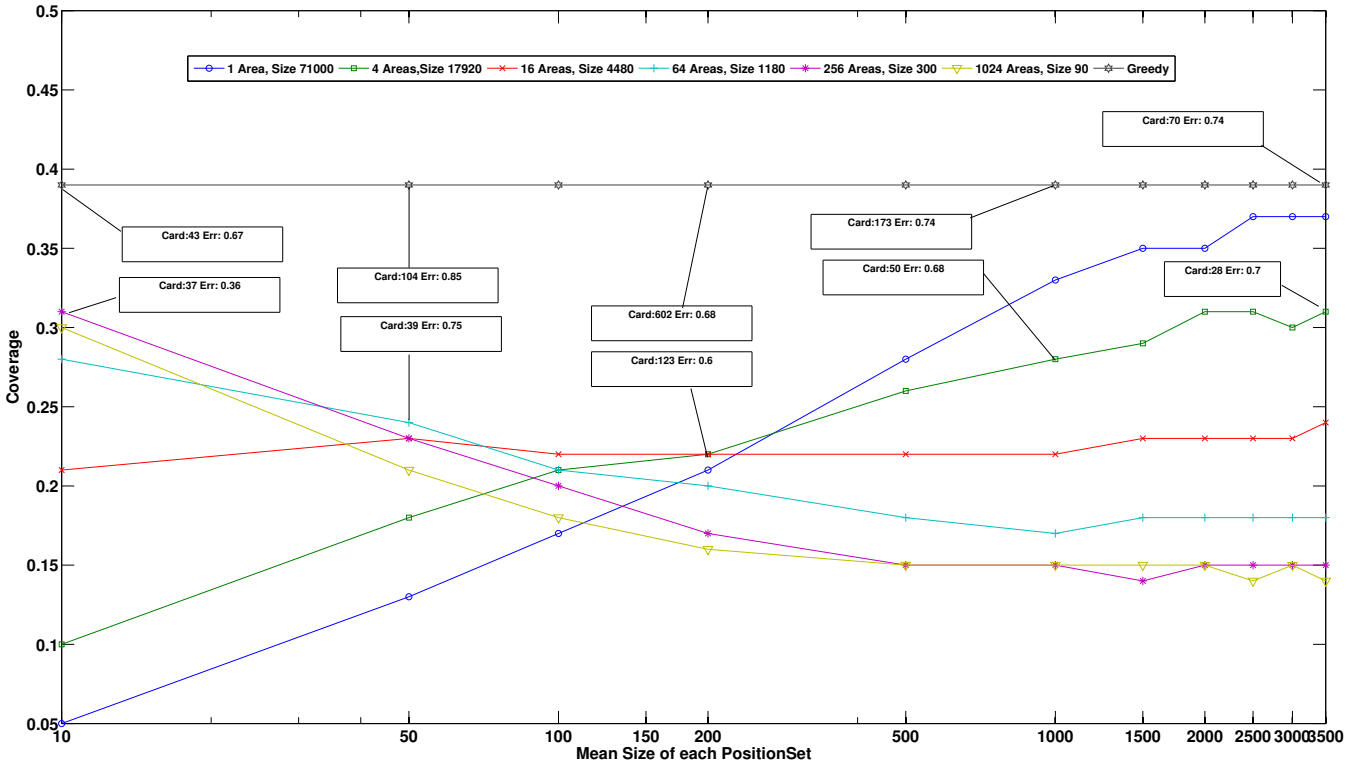

Figure 9. Coverage achieved by PP-Greedy algorithm, as a function of the average size of P_i and I

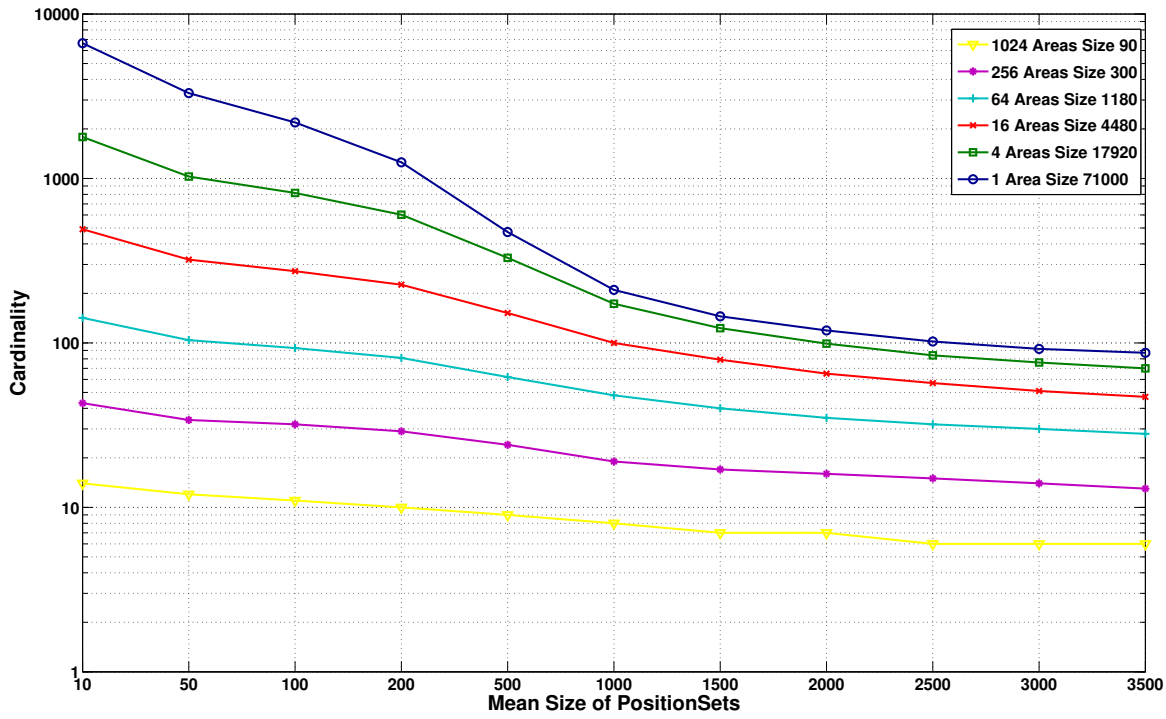

Figure 10. Greedy Cardinality

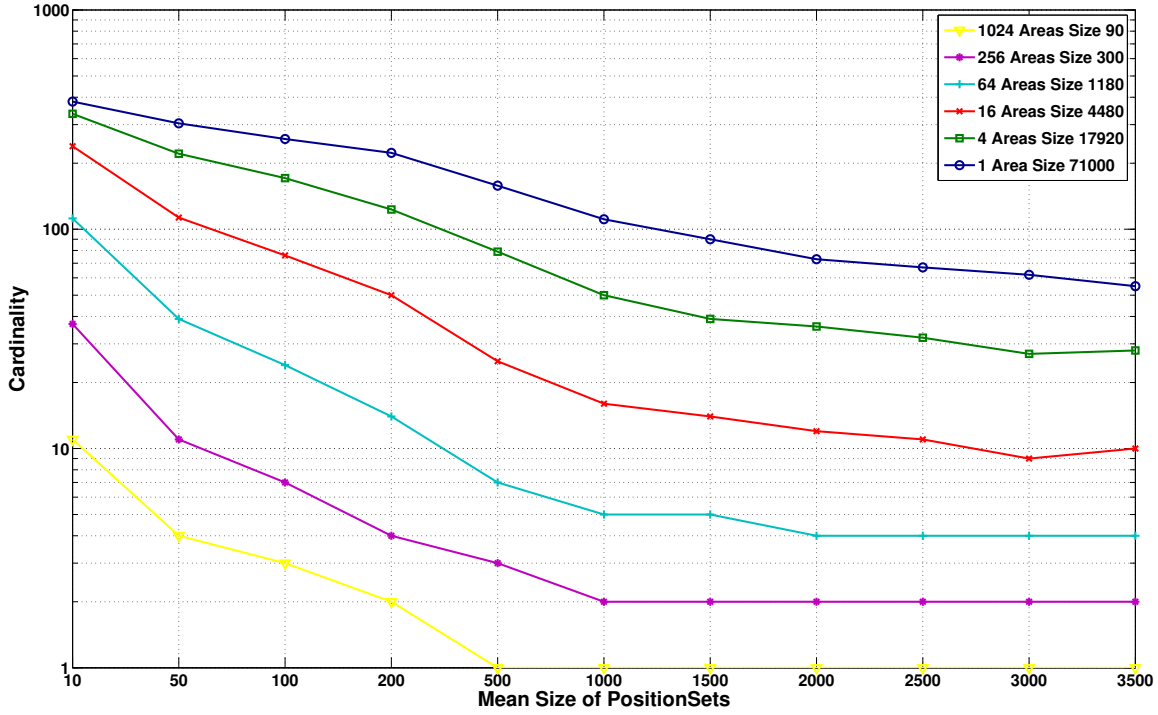

Figure 11. PP-Greedy Cardinality

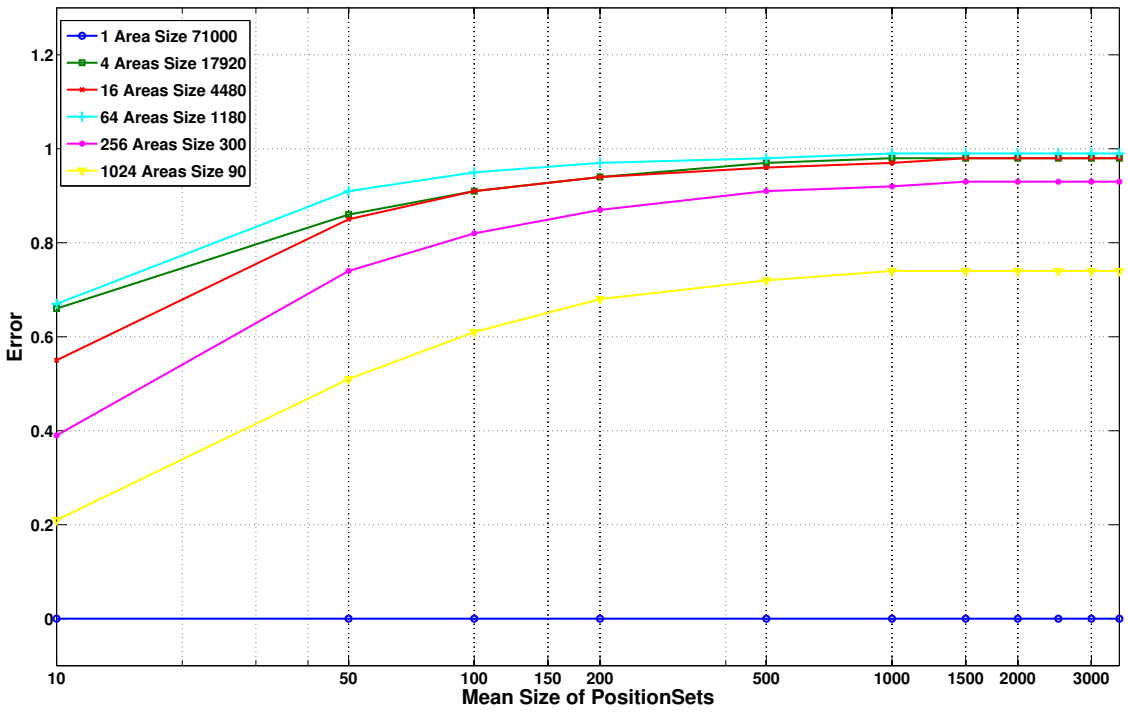

Figure 12. Greedy Error

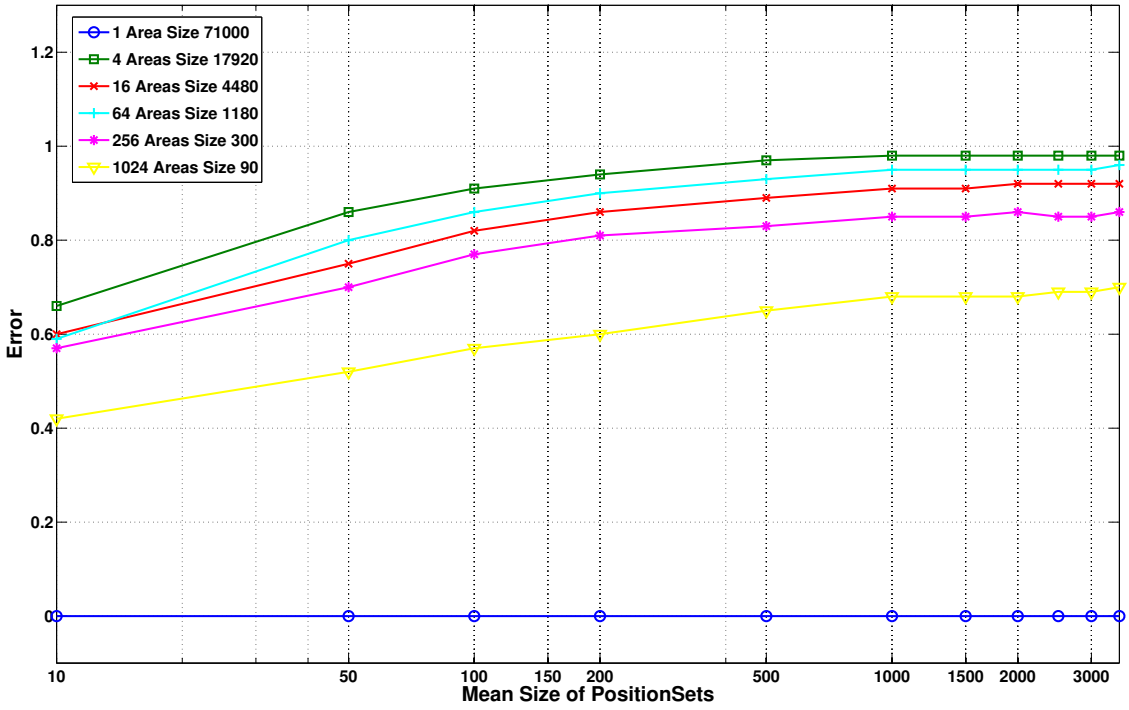

Figure 13. PP-Greedy error

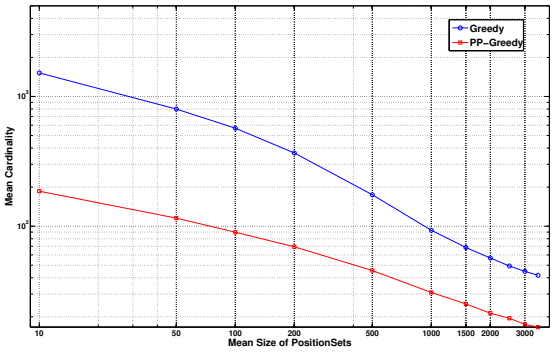

Figure 14. Average cardinality for all the I areas considered.

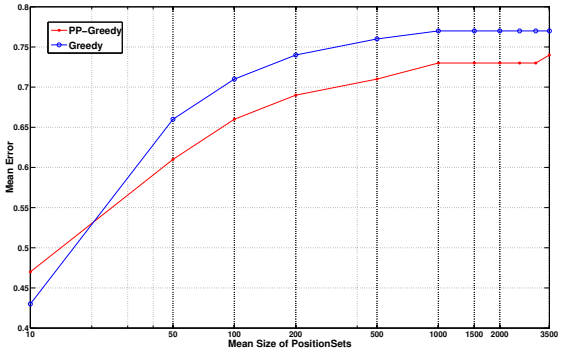

Figure 15. Average error for all the I areas considered.

algorithm. First of all, as a general consideration, the performance strongly depends on the number of hash functions used: the higher this number, the higher is the quality of the solution returned by `PP-Greedy`. At the same time, the amount of data to be transferred increases and has to be set accordingly with the application constraints. Secondly, performances can be maximized by tuning the granularity of the sets used for the set cover and/or size of the area to be covered. The less accurate is this tuning, the bigger is the distance from `Standard-Greedy`'s covering ability. Furthermore, coverage achieved by the `PP-Greedy` algorithm is always slightly lower than standard Greedy, but this lower coverage is fairly compensated by lower error and cardinality. Recalling what discussed in section IV we can state that despite a lower coverage, `PP-Greedy` produces a *set system* whose cost is much lower than the one given by the Greedy algorithm. As a conclusion, the slightly lower accuracy in coverage given by `PP-Greedy` is like the "price to pay" for privacy preservation, compared to the `Standard-Greedy`. Despite of this, privacy preservation and efficiency in terms of *set system*'s cost, are definitely the strengths of this approach. Finally, our experiments confirm that the systems parameters, namely, the number of hash functions to compute the sketches (m), the size of the position sets ($|P_i|$) and the size of the area of interest ($|I|$), have to be carefully tuned taking into consideration the tradeoff among privacy, efficency and accuracy.

VII. Conclusions and future work

This paper presented a novel approach to support privacy in people centric sensing applications, based on the use of compacy, privacy preserving synopses of user traces. In our system, the selection process is based on sketches of the location traces rather than the location traces themselves. Based on this representation, an efficient algorithm to perform query matching over the set of collected data was discussed. This algorithm solves the well known, NP-complete Set Cover problem without requiring explicit knowledge of the sets, but only using their compact, privacy preserving sketches. Furthermore, the amount of information mobile devices need to send to the service provider to allow the selection process is modest, in the order of 10^4 bytes for maximum accuracy in realistic cases, as our experimental analysis suggests. The proposed technique can be exploited in several application domains such as environmental monitoring, analysis of social patterns, traffic maps etc. The results of an extensive experimental analysis suggest that the approach we propose is feasible on state-of-art, commercial devices.

A first direction of future research is proposing a theoretical characterization of the cases in which the sketch-based heuristic for Set Cover we proposed provides approximation guarantees. A further direction is extending our techniques to other application domains. An example is the scenario in which user profiles are described by feature vectors and the goal is estimating similarities between profiles without explicitely using the profiles themselves. This would for example be useful to design efficient, privacy preserving recommender systems.

References

[1] "Ibm reveals five innovations that will change our lives in the next five years," http://www-03.ibm.com/press/us/en/pressrelease/33304.wss.

[2] A. S. Pentland, "Building a nervous system for humanity," http://www.multimedia.ethz.ch/conferences/2010/sensys.

[3] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell, "Urban sensing systems: Opportunistic or participatory," in *In Proc. ACM 9th Workshop on Mobile Computing Systems and Applications (HOTMOBILE '08)*, 2008.

[4] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G.-S. Ahn, "The rise of people-centric sensing," *IEEE Internet Computing*, vol. 12, pp. 12–21, 2008.

[5] "Metrosense - secure people-centric sensing at scale," http://metrosense.cs.dartmouth.edu/.

[6] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "Bikenet: A mobile sensing system for cyclist experience mapping," *ACM Trans. Sen. Netw.*, vol. 6, pp. 6:1–6:39, January 2010. [Online]. Available: http://doi.acm.org/10.1145/1653760.1653766

[7] E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell, "Cenceme: injecting sensing presence into social networking applications," in *Proceedings of the 2nd European conference on Smart sensing and context*, ser. EuroSSC'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 1–28. [Online]. Available: http://portal.acm.org/citation.cfm?id=1775377.1775379

[8] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, "Soundsense: scalable sound sensing for people-centric applications on mobile phones," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, ser. MobiSys '09. New York, NY, USA: ACM, 2009, pp. 165–178. [Online]. Available: http://doi.acm.org/10.1145/1555816.1555834

[9] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, "Anonysense: privacy-aware people-centric sensing," in *Proceeding of the 6th international conference on Mobile systems, applications, and services*, ser. MobiSys '08. New York, NY, USA: ACM, 2008, pp. 211–224. [Online]. Available: http://doi.acm.org/10.1145/1378600.1378624

[10] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson, "People-centric urban sensing," in *Proceedings of the 2nd annual international workshop on Wireless internet*, ser. WICON '06. New York, NY, USA: ACM, 2006. [Online]. Available: http://doi.acm.org/10.1145/1234161.1234179

[11] H. Lu, N. D. Lane, S. B. Eisenman, and A. T. Campbell, "Bubble-sensing: Binding sensing tasks to the physical world," *Pervasive and Mobile Computing*, vol. 6, no. 1, pp. 58 – 71, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/B7MF1-4XMD5SV-1/2/d207de79bfa2733ab04751950e22d357

[12] N. Maisonneuve, M. Stevens, M. E. Niessen, P. Hanappe, and L. Steels, "Citizen noise pollution monitoring," in *dg.o '09: Proceedings of the 10th Annual International Conference on Digital Government Research*. Digital Government Society of North America, 2009, pp. 96–103.

[13] E. Kanjo, "Noisespy: A real-time mobile phone platform for urban noise monitoring and mapping," *Mobile Networks and Applications*. [Online]. Available: http://dx.doi.org/10.1007/s11036-009-0217-y

[14] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "PriSense: Privacy-Preserving Data Aggregation in People-Centric Urban Sensing Systems," in *Proceedings of IEEE INFOCOM 2010*. IEEE, Mar. 2010, pp. 1–9. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2010.5462147

[15] S. Zhong, L. (erran Li, Y. G. Liu, and Y. R. Yang, "Privacy-preserving locationbased services for mobile users in wireless networks," Tech. Rep., 2004.

[16] N. Pham, R. Ganti, Y. Uddin, S. Nath, and T. Abdelzaher, "Privacy-preserving reconstruction of multidimensional data maps in vehicular participatory sensing," in *Wireless Sensor Networks*, ser. Lecture Notes in Computer Science, J. Silva, B. Krishnamachari, and F. Boavida, Eds. Springer Berlin / Heidelberg, 2010, vol. 5970, pp. 114–130.

[17] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig, "Syntactic clustering of the web," in *Proc. of the World Wide Web Conference*, 1997.

[18] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher, "Min-wise independent permutations," in *ACM Symposium on the Theory of Computing*, New York, NY, USA, 1998.

[19] A. Z. Broder, "Identifying and filtering near-duplicate documents," in *Combinatorial Pattern Matching, 11th Annual Symposium, CPM 2000, Montreal, Canada, June 21-23, 2000, Proceedings*, ser. Lecture Notes in Computer Science, vol. 1848. Springer, 2000, pp. 1–10.

[20] T. Bohman, C. Cooper, and A. M. Frieze, "Min-wise independent linear permutations," *The Electronic Journal of Combinatorics,*, vol. 7, 2000.

[21] V. V. Vazirani, *Approximation algorithms*. New York, NY, USA: Springer-Verlag New York, Inc., 2001.

[22] R. Shokri, G. Theodorakopoulos, J.-Y. L. Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *Proceedings of the IEEE Computer Society 2011 Security and Privacy Symposium*. IEEE, 2011, pp. 247–262.

[23] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes." Springer-Verlag, 1999, pp. 223–238.

[24] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on gps data," in *Proceedings of the 10th international conference on Ubiquitous computing*, ser. UbiComp '08. New York, NY, USA: ACM, 2008, pp. 312–321. [Online]. Available: http://doi.acm.org/10.1145/1409635.1409677

[25] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in *Proceedings of the 18th international conference on World wide web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 791–800. [Online]. Available: http://doi.acm.org/10.1145/1526709.1526816