DPSense: Differentially Private Crowdsourced Spectrum Sensing

Xiaocong Jin
Arizona State University
xcjin@asu.edu

Rui Zhang
University of Delaware
ruizhang@udel.edu

Yimin Chen
Arizona State University
ymchen@asu.edu

Tao Li
Arizona State University
tli@asu.edu

Yaochao Zhang
Arizona State University
yczhang@asu.edu

ABSTRACT

Dynamic spectrum access (DSA) has great potential to address worldwide spectrum shortage by enhancing spectrum efficiency. It allows unlicensed secondary users to access the underutilized licensed spectrum when the licensed primary users are not transmitting. As a key enabler for DSA systems, crowdsourced spectrum sensing (CSS) allows a spectrum sensing provider (SSP) to outsource the sensing of spectrum occupancy to distributed mobile users. In this paper, we propose DPSense, a novel framework that allows the SSP to select mobile users for executing spatiotemporal spectrum-sensing tasks without violating the location privacy of mobile users. Detailed evaluations on real location traces confirm that DPSense can provide differential location privacy to mobile users while ensuring that the SSP can accomplish spectrum-sensing tasks with overwhelming probability and also the minimal cost.

CCS Concepts

•Security and privacy → Privacy-preserving protocols; Mobile and wireless security;

Keywords

Dynamic spectrum access; differential privacy; crowdsourced spectrum sensing; location privacy

1. INTRODUCTION

Dynamic spectrum access (DSA) [28] is a key technique to address worldwide spectrum shortage by enhancing spectrum efficiency. It allows unlicensed secondary users (SUs) with cognitive radio capabilities to access the underutilized licensed spectrum when the licensed primary users (PUs) are not transmitting. Database-driven DSA [10,12] is the FCC-approved de facto paradigm. In such a system, a spectrum service provider (SSP) accepts registrations from PUs and determines spectrum availability, and SUs are all required to inquire the SSP about the availability of any interested spectrum before using it. Current SSPs estimate spectrum availability based on PUs' registered locations and transmission schedules in combination with radio propagation modeling. Recent measurement studies such as [21], however, show that such estimations are often inaccurate and tend to be overly conservative due to ignoring local environmental factors, resulting in a considerable waste of valuable spectrum resources.

Crowdsourced spectrum sensing (CSS) [8, 11] can effectively improve the spectrum-estimation accuracy in database-driven DSA systems. In this approach, the SSP deploys a small number of dedicated spectrum sensors at strategic locations and outsources the majority of spectrum-sensing tasks to ubiquitous mobile users. The feasibility of CSS lies in three main aspects. First, 497 million mobile devices were added in 2014, and global mobile devices will grow to 11.5 billion by 2019 at a CAGR of 9% [1]. Such ubiquitous penetration of mobile devices into everyday life implies sufficient geographic coverage especially in highly populated regions where DSA systems are of high demand. Second, future mobile devices are expected to be capable of spectrum sensing as DSA is mature and widely deployed. Last, most mobile users in daily life take routine or preplanned routes and may participate in CSS systems if proper incentives are provided.

Privacy and efficiency are two conflicting concerns that impede the wide deployment of CSS systems. In particular, spectrum-sensing tasks are spatiotemporal in nature and pertain to specific time and physical locations. In addition, mobile users often need travel to designated sensing locations, and they may desire rewards commensurate with the travel distance. On the one hand, the SSP seeks to maximize the system efficiency such that the spectrum-sensing task can be successfully fulfilled with the minimum cost which is equivalent to the minimum total travel distance for all participating users. To achieve the maximum system efficiency, the SSP needs to know the locations of candidate participants. On the other hand, mobile users are increasingly wary of location privacy, and disclosing their locations to the SSP may severely discourage their participation.

This paper presents DPSense, a novel framework for striking a good balance between location privacy and system efficiency in CSS systems. In DPSense, the SSP publishes spectrum-sensing tasks for specific locations and time periods in the future. Each candidate CSS participant responds to the SSP by submitting his/her predicted (either routine or preplanned) mobility trace which is perturbed to satisfy differential location privacy. We then present an optimization formulation for the SSP to assign spectrum-sensing tasks based on perturbed mobility traces and show that it is NP-hard. Finally, we propose a heuristic solution and thoroughly evaluate it via detailed trace-driven simulations based on real-world mobility traces.

CCS'16, October 24-28, 2016, Vienna, Austria
© 2016 ACM. ISBN 978-1-4503-4139-4/16/10. . . $15.00
DOI: http://dx.doi.org/10.1145/2976749.2978426

Our results confirm that DPSense can simultaneously achieve the following desirable objectives.

- *Differential location privacy*. DPSense offers differential location privacy to mobile participants under a strong adversary model by incorporating the mechanism in [27].
- *Minimal cost or travel distance*. DPSense assigns spectrum-sensing tasks to mobile participants based on their perturbed location traces while ensuring the minimal cost for the SSP or equivalently minimum total travel distance for mobile participants.
- *High task-completion rate*. DPSense guarantees that each spectrum-sensing task can be successfully conducted with overwhelming probability.

The rest of the paper is structured as follows. Section 2 introduces the system and adversary models. Section 3 motivates the requirement for location privacy in CSS. Section 4 reviews the differential privacy mechanism in [27] underlying DPSense. Section 5 presents the DPSense framework. Section 6 demonstrates the experimental evaluations. Section 7 discusses the related work. Section 8 concludes this paper.

2. SYSTEM AND ADVERSARY MODELS

In this section, we introduce the system model, the spectrum-sensing model, and the adversary model.

2.1 System Model

We consider a crowdsourced spectrum sensing (CSS) system consisting of a spectrum service provider (SSP) and N mobile participants in CSS. In addition to having the similar functionalities to traditional database-driven DSA system operators [10,12], the SSP explores mobile crowdsourcing to estimate spectrum availability in its service region and answers spectrum access requests from SUs.

Each mobile participant is a user who carries an advanced mobile device with spectrum sensing capabilities and wishes to earn rewards by participating in CSS. The participant registers with the SSP and communicates with the SSP via an app installed on his[1] mobile device. Developed by the SSP, the app is assumed to pass the strict vetting process of the trusted app store and has no unauthorized access to the user's locations.

The SSP generates a spectrum-sensing task either periodically or on demand upon receiving a spectrum-access request from an SU. Our system works in the same way for both cases. The SSP converts each sensing task into a number of subtasks to ensure that the sensing reports submitted by different mobile participants are independent of each other. In particular, let T_j denote the j-th sensing task, which includes R_j as the physical sensing region, t_j^s as the sensing time period, and div_j as the targeted diversity order to be further explained in Section 2.2. The SSP first selects n_j candidate sensing locations in R_j, denoted by $\{l_{j,k}^s\}_{k=1}^{n_j}$, such that any two locations are separated with a distance over d_0, where d_0 is a fixed system parameter. The SSP then generates n_j subtasks $\{S_{j,k}\}_{k=1}^{n_j}$, where $S_{j,k} = (l_{j,k}^s, t_j^s)$. Finally, the SSP assigns subtasks to mobile participants based on their mobility traces. A subtask can be accepted or declined by the chosen mobile participant. Task T_j is said to be *completed* if and only if at least div_j subtasks are accepted by mobile participants.

To enable spectrum-sensing task assignment, each participant i periodically predicts his mobility trace for the upcoming time period and submits it to the SSP. This can be easily done in practice, as most mobile users have target locations to go instead of wandering around. Each mobility trace can be represented as a sequence of γ location and time pairs, $L_i = \langle (t_{i,1}, l_{i,1}), \dots, (t_{i,\gamma}, l_{i,\gamma}) \rangle$, where $t_{i,u}$ and $l_{i,u}$ ($\forall u \in [1,\gamma]$) denote the u-th time and location points, respectively, and γ is a system parameter. To be more practical, $t_{i,u}$ and $l_{i,u}$ can be the indexes of a time slot and a physical cell, as specified by the SSP. The mobility traces can be either automatically obtained via popular location service APIs such as Google Map API or manually fed to the mobile app by participants. Some participants may opt to not provide their mobility traces, in which case they are considered unavailable for the entire time period.

2.2 Spectrum Sensing Model

Each mobile participant performs spectrum sensing by detecting PU transmissions on the specified channel in the time and location designated by the SSP. We adopt the following conventional spectrum-sensing model to facilitate the presentation, but our work can be easily extended to support other sensing models.

We assume that the channels between PUs and mobile participants are Rayleigh fading with additive white Gaussian noise (AWGN). The shadow fading is spatially correlated, and the correlation of the received signals for two spectrum sensors separated by distance d can be modeled as an exponential function $e^{-a\mathsf{d}}$ [20], where a refers to an environment parameter which is approximately 0.1204 and 0.002 in urban non-line-of sight and suburban environments, respectively. The *de-correlation distance* d_0 is defined as the minimum distance for two spectrum sensors when the correlation is under a desired threshold.

We assume that the SSP uses the Neyman-Pearson (NP) detector to combine multiple sensing reports from mobile participants to reliably determine spectrum occupancy. Specifically, for a target average decision error probability P^* that accounts for both false positives and false negatives, the number of independent spectrum-sensing reports needs to be no less than the *diversity order* [6,24],

$$\mathsf{div}^* = -\lim_{\mathsf{SNR}\to+\infty} \frac{\log P^*}{\log \mathsf{SNR}}, \qquad (1)$$

where SNR is the average signal-to-noise ratio at the sensing participants. We subsequently assume that the SSP can determine proper div^* for each spectrum sensing task.

Once the diversity order is concretely defined, the following theorem can be similarly derived according to [6].

THEOREM 1. *For multiuser sensing with soft information fusion, when the sensing threshold is chosen to minimize the average error probability, the diversity order of the NP detector equals the number of cooperative users.*

Similar conclusions can be drawn for hard decision fusion as well. For details, please refer to [6].

2.3 Adversary Model

We assume that the SSP is *honest but curious*, which is commonly used to characterize a reasonable service provider. In particular, the SSP is trusted to faithfully follow the protocol execution but is also interested in learning mobile participants' locations. We assume that the SSP can have arbitrary prior knowledge for attempting to breach the participants' location privacy. In particular, it may infer target mobile participant's location by exploiting the temporal correlation among the submitted mobility traces.

3. LOCATION INFERENCE IN CSS

In the original CSS system, the SSP needs to know the locations of mobile participants for assigning sensing tasks. This require-

[1] No gender implication.

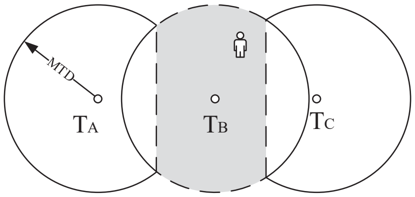


Figure 1: An exemplary location-inference attack, where the participant chooses T_B over T_A and T_C.

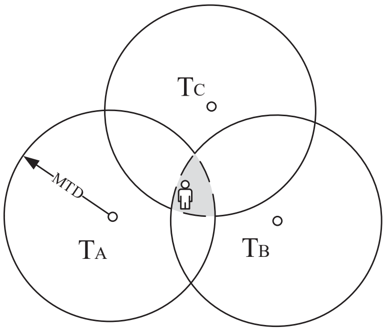


Figure 2: Another exemplary location-inference attack, where triangulation is used to locate the possible region of the victim.

ment obviously violates the location privacy of mobile participants in the desired sensing period. It is worth emphasizing that location privacy here refers to the secrecy of each participant's original mobility trace when he is not involved in CSS. In this section, we illustrate several location-inference attacks against several plausible attempts to improve the location privacy in CSS.

One plausible solution to protecting location privacy in CSS is to let the SSP broadcast spectrum-sensing tasks to all mobile participants who then claim tasks without disclosing their locations to the SSP. Unfortunately, since mobile participants tend to select sensing tasks close to their locations, the SSP could still infer their locations based on the tasks they choose. The reason is that mobile participants generally are only willing to travel up to a certain distance (e.g., slightly deviating from their scheduled routes), which is commonly referred to as the maximum travel distance (MTD) and can be learned from publicly available data [14]. The sensing task chosen by a mobile participant simply indicates that his location is most likely within the circle centered at the chosen task's center location with a radius of MTD, and such information is what the participants may not want to disclose. The SSP can go one step further to shrink the area a participant resides from his sensing preference. Consider Fig. 1 as an example. Assume that the SSP broadcasted three tasks T_A, T_B, and T_C, where three circles represent their corresponding maximum travel regions. Suppose that a target participant chose task T_B. Under the reasonable assumption that the participant always chooses the closest task, the SSP can easily confine the participant's location within the shaded area.

A more subtle attack against the above plausible solution is to use trilateration. Assume that the SSP broadcasts one sensing task in one round around the target area but with slight modification of the sensing region, as shown in Fig. 2. The three rounds can be carefully scheduled so that during the three rounds, the participant could be very likely located in the same location. For example, the three rounds can be scheduled simply at the same time of the day. In the figure, the participant sequentially chooses the sensing tasks T_A, T_B, and T_C. The SSP could simply use triangulation to find out the intersection of the three regions so that the victim is very likely in the highlighted region. As is shown, the area of the intersection could be very small. Therefore, the participant's location privacy is further compromised.

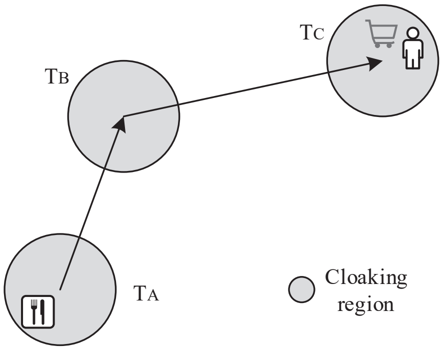


Figure 3: Another exemplary location-inference attack that explores the temporal correlation of adjacent reported locations.

Another possible solution is to let the participants submit perturbed locations to the SSP which in turn assigns sensing tasks based on perturbed locations. Unfortunately, based on a recent study [27], the SSP can still infer participant locations by exploiting the temporal correlation among multiple perturbed locations submitted within a short time period. Consider Fig. 3 as an example. Suppose that one participant moved from a restaurant in area 1 to a supermarket in area 3 and submitted three circular cloaking areas generated from some spatial cloaking mechanism. Although the individual locations were cloaked at each time, the order of the three cloaking areas along with some side information such as road constraints may reveal his exact location at the supermarket.

The three exemplary attacks discussed above highlight the risk of location privacy breach in CSS and call for an advanced solution to protect mobile participants' location privacy.

4. DIFFERENTIAL PRIVACY WITH TEMPORAL CORRELATION CONSIDERATION

In this section, we briefly review the differential privacy mechanism in [27], which DPSense relies on for generating differentially private mobility traces.

4.1 Inference Model

We first discuss the Markov chain to model the temporal correlations among the submitted locations of a particular CSS participant. From the SSP's point of view, since it can only observe the perturbed mobility trace instead of the original one, the inference process is a Hidden Markov Model (HMM).

Assume that the sensing region is divided into disjoint cells, indexed from 1 to m. Let $p_t = (p_t[1], \ldots, p_t[m])$ denote the probability distribution of a certain participant at time t. For example, if a participant at time t is likely to reside in cell 1, 2, 3, and 4 with probability 0.15, 0.25, 0.35, and 0.25, respectively, we have $p_t = \{0.15, 0.25, 0.35, 0.25, 0, ..., 0\}$. Let $M^t = [m_{ij}]$ denote the transition matrix, where m_{ij} is the probability that the participant moves form cell i to cell j for all $1 \leq i, j \leq m$ between consecutive timestamps. Given a probability vector p_{t-1}, the probability at time t can be computed as $p_t = p_{t-1} M^t$. We assume that the transition matrix M^t is given as a priori knowledge, which can be generated either from public transportation data or from personal transportation data[2] using existing methods [22]. Since M^t can be constructed based on some public anonymized mobility datasets that are totally unrelated to the participants in our system, it does not negatively affect the location privacy of our system participants.

We further define the prior and posterior probabilities of a user's location before and after observing the perturbed location at time t as p_t^- and p_t^+, respectively. It is obvious that $p_t^- = p_{t-1}^+ M^t$.

4.2 Differential Location Privacy

Differential location privacy is defined over a δ-location set [27].

Definition 1. (δ-Location Set). Let p_t^- be the prior probability of a user's location at time t. The δ-location set is a set containing the minimum number of locations that have the prior probability sum no less than $1 - \delta$:

$$\Delta \mathrm{X}_t = \min\{z | \sum_z p_t^-[z] \geq 1 - \delta\}. \quad (2)$$

Definition 2. At any time t, a randomized mechanism $\mathcal{A}$ satisfies ϵ-differential privacy on the δ-location set $\Delta \mathrm{X}_t$ if, for any output $\hat{u}_t$ and any two locations u_1 and u_2 in $\Delta \mathrm{X}_t$, the following holds:

$$\frac{\mathbf{Pr}(\mathcal{A}(u_1) = \hat{u}_t)}{\mathbf{Pr}(\mathcal{A}(u_2) = \hat{u}_t)} \leq e^{\epsilon}. \quad (3)$$

To satisfy the differential privacy requirement defined above, a location release algorithm that relies on Markov inference and the planar isotropic mechanism is proposed in [27]. The output of the algorithm is a differentially private version of the input mobility trace. We defer the algorithm outline to Section 5.2 for clarity.

5. DPSENSE FRAMEWORK

In this section, we present the DPSense framework.

5.1 Overview

DPSense is intended to strike a balance between the spectrum-sensing quality, the overall spectrum-sensing cost, and the location privacy. The DPSense framework is illustrated in Fig. 4.

Assume that the SSP has M sensing tasks, denoted by $\mathcal{T} = \{T_j\}_{j=1}^{M}$, to fulfil in a future time period, e.g., starting one hour later. Each mobile participant i submits his predicated mobility trace either periodically or in response to the SSP's request. Recall that the mobility trace of participant i in the target sensing period is defined as $L_i = \langle (t_{i,1}, l_{i,1}), \ldots, (t_{i,\gamma}, l_{i,\gamma}) \rangle$. Instead of submitting L_i to the SSP, participant i submits a perturbed version L_i^o based on the algorithm in [27]. Subsequently, the SSP smooths the perturbed traces according to the procedure in Section 5.3 and finally runs our proposed algorithm in Section 5.6 on the smoothed mobility traces to assign the M sensing tasks. As discussed, each task can be divided into a number of subtasks at different locations in the desired sensing region. Each mobile participant receives either zero or one subtask assignment, and he may accept or decline the assignment (e.g., when the subtask location is too far from his original route). A sensing task is completed if the number of mobile participants accepting the subtask assignment is no less than the predefined diversity order. Each participant could be granted some monetary rewards or reputation points that are proportional to the distance he has to travel to perform the sensing task. How the participants are actually rewarded is orthogonal to this paper.

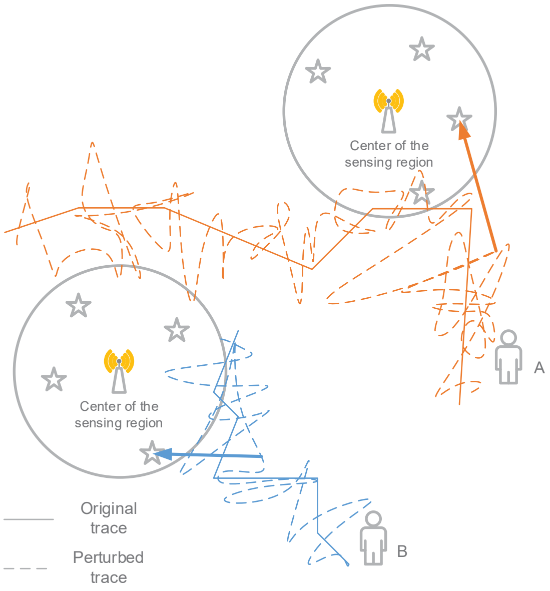


Figure 4: The DPSense framework.

5.2 Generating Differentially Private Mobility Traces

We use the location release algorithm in [27] which is based on Markov inference and the planar isotropic mechanism (PIM). The algorithm accepts a true mobility trace as input and outputs a perturbed mobile trace that satisfies differential privacy on the δ-location set. Specifically, the algorithm sequentially perturbs each location in the mobility trace through the following steps.

- First, prior probabilities are derived using posterior probabilities and the matrix M^t based on the Markov model.
- Second, a δ-location set is generated to identify the set containing the minimum number of locations that have prior probability sum no less than $1 - \delta$.
- Third, the location at the current timestamp is perturbed by adding a noise generated using PIM based on the K-norm mechanism.
- Fourth, location inference is conducted based on the output perturbed location to update the posterior probability of the user in each location of the δ-location set.

It is proved in [27] that the algorithm guarantees differential privacy. We subsequently call the perturbed mobility trace as the PIM trace and refer interested readers to [27] for detailed illustrations.

5.3 Smoothing Perturbed Mobile Traces

Since the SSP can only use the perturbed mobility trace for task assignment, it is intuitive that the closer the perturbed trace is to the original location trace, the more accurate the SSP can estimate

[2] For example, Google Now continuously tracks users' locations and display relevant information to users in the form of "cards" [26].

Table 1: Summary of Notation

Symbol	Definition
N	Total number of participants
M	Total number of sensing tasks
T_j	The jth sensing task
t_j^s	Sensing timestamp for task T_j
R_j	Sensing region for task T_j
div_j^*	Desired sensing diversity order for T_j
$S_{j,k}$	Sensing sub-task set
n_j	Number of subtasks for task T_j
L_i	True mobility trace of participant i
γ	Number of timestamps for each mobility trace
$l_{i,\kappa}$	The κth location in L_i
$t_{i,\kappa}$	The κth timestamp in L_i
L_i^o	PIM trace of participant i
L_i^h	Smoothed PIM trace of participant i
$l_{i,\kappa}^o$	The κth location in L_i^o
$l_{i,\kappa}^h$	The κth location in L_i^h
μ	Size of the sliding window
α	Distance weight ratio
β	Diversity order multiplicator

Algorithm 1 PIM Traces Smoothing

```
Input: A set of PIM traces $\{L_i^o\}_{i=1}^N$ and sliding window size $\mu$.
Output: A set of smoothed traces $\{L_i^h\}_{i=1}^N$.
for all $i \in \{1, \dots, N\}$ do
    $L_i^h \leftarrow \emptyset$.
    for all $\kappa \in \{\lfloor \mu/2 \rfloor + 1, \dots, \gamma - \lfloor \mu/2 \rfloor\}$ do
        $l_{i,\kappa}^h = \frac{1}{\mu} \sum_{x=\kappa-\lfloor \mu/2 \rfloor}^{\kappa+\lfloor \mu/2 \rfloor} l_{i,x}^o$
        $L_i^h \leftarrow L_i^h \bigcup \{(l_{i,\kappa}^h, t_{i,\kappa})\}$
    end for
end for
return $\{L_i^h\}_{i=1}^N$.
```

each participant's travel cost, and the higher probability that the sensing task can be completed while ensuring differential location privacy to mobile participants. It is therefore essential to reduce the negative impact the noise added to the mobility trace. Recall that for the original location at each timestamp, noise is generated in the isotropic space using the K-norm mechanism. The probability of generating noise of a certain value and the probability of generating noise of the exact inverse value are the same. By averaging multiple consecutive locations, the deviation of the averaged location to the original true location could be smaller in contrast to the difference between the disturbed location to the original true location. When the noise amplitude is large, the average could reduce the negative impact introduced due to the noise.

Based on the above intuition, we propose to smooth each user's differentially private mobility trace using a sliding window and assign tasks based on smoothed location traces. Specifically, we define the size of the sliding window as μ, where μ is an odd integer and system parameter. For each timestamp, we generate a smoothed location as the average of the previous consecutive $\lfloor \mu/2 \rfloor$ PIM locations, the current PIM location, and the next consecutive $\lfloor \mu/2 \rfloor$ PIM locations. The details of the smoothing algorithm are summarized in Algorithm 1. We will show in our simulations the effectiveness of the sliding window and the impact of μ.

5.4 Accepting/Declining Task Assignments

Participants may accept or decline an assigned sensing task for various reasons. We now introduce a model to characterize the probability that an assigned task is accepted, which takes into account of both the physical travel distance and potential wait time.

We first consider the impact of physical travel distance. According to our system model in Section 2.2, each task T_j includes R_j as the physical sensing region, t_j^s as the sensing time period, and div_j^* as the targeted diversity order. The SSP further divides T_j into n_j subtasks $\{S_{j,k}\}_{k=1}^{n_j}$ at locations $\{l_{j,k}^s\}_{k=1}^{n_j}$, respectively. Consider subtask $S_{j,k}$ and participant i as an example. Let L_i be participant i's true mobility trace and v be the average speed. For participant i to travel from location $l_{i,\kappa}$ at time $t_{i,\kappa}$ to perform subtask $S_{j,k}$ at sensing location $l_{j,k}^s$, the time of arrival at the sensing location is subject to the following condition,

$$\mathsf{dist}(l_{i,\kappa}, l_{j,k}^s) \leq v(t_j^s - t_{i,\kappa}), \tag{4}$$

where $\mathsf{dist}(\cdot,\cdot)$ denotes the Euclidian distance.

We then consider the participant's potential waiting time. In particular, participant i may arrive at the sensing location $l_{j,k}^s$ early. If he needs to wait for a long time period to perform the task, he may reject the task at the very beginning. We therefore define *synthetic distance* to jointly consider the travel distance and the waiting time for a given sensing task, which is computed as

$$\begin{aligned} &\mathsf{dist}^*(l_{i,\kappa}, l_{j,k}^s) \\ &= \begin{cases} \begin{array}{l} \mathsf{dist}(l_{i,\kappa}, l_{j,k}^s)+ \\ \alpha v(t_j^s - t_{i,\kappa})- \\ \alpha \mathsf{dist}(l_{i,\kappa}, l_{j,k}^s) \end{array} & \text{if } \mathsf{dist}(l_{i,\kappa}, l_{j,k}^s) \leq v(t_j^s - t_{i,\kappa}), \\ \infty & \text{otherwise.} \end{cases} \end{aligned} \tag{5}$$

Synthetic distance defined above essentially converts the waiting time into additional travel distance. The system parameter α indicates the weight of the waiting-time equivalent distance versus that of the true travel distance. Since simply waiting generally involves less effort in comparison with the actual travel, it is reasonable to require that $\alpha \leq 1$.

We use a simple linear distribution model to characterize the probability that participant i will accept subtask $S_{j,k}$. Let the MTD be the maximal travel distance within which a participant is willing to travel to perform a sensing task, which can be obtained from historical data [14]. Similar to [23], we calculate the probability that participant i will accept subtask $S_{j,k}$ at sensing location $l_{j,k}^s$ by departing from $l_{i,\kappa}$ at time $t_{i,\kappa}$ as

$$\begin{aligned} &\mathbf{Pr}[P_i \leftarrow S_{j,k} | t_{i,\kappa}] \\ &= \begin{cases} 1 - \frac{\mathsf{dist}^*(l_{i,\kappa}, l_{j,k}^s)}{\mathrm{MTD}} & \text{if } \mathsf{dist}^*(l_{i,\kappa}, l_{j,k}^s) < \mathrm{MTD}, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{6}$$

In other words, $\mathbf{Pr}[P_i \leftarrow S_{j,k} | t_{i,\kappa}]$ is one when the corresponding synthetic distance is zero and zero when the synthetic distance exceeds MTD.

5.5 Spectrum-Sensing Task Assignment Formulation

We formulate the spectrum-sensing task assignment as an optimization problem as follows.

We define the indicator variable $b_{j,k}^{i,\kappa}$ such that $b_{j,k}^{i,\kappa} = 1$ if participant i is assigned to depart from his location $l_{i,\kappa}$ at time $t_{i,\kappa}$ to perform sensing subtask $S_{j,k}$ at sensing location $l_{j,k}^s$ and 0 otherwise. If $b_{j,k}^{i,\kappa} = 1$, the expected diversity order contributed by participant i can be computed as

$$\mathsf{div}_{j,k}^{i,\kappa} = \mathbf{Pr}[P_i \leftarrow S_{i,j} | t_{i,j}], \tag{7}$$

where $\mathbf{Pr}[P_i \leftarrow S_{i,j} | t_{i,j}]$ is given in Eq. (6).

Given the set of N participants with smoothed PIM traces $\{L_i^h\}_{i=1}^N$ and the set of sensing subtasks $\{S_{j,k} | 1 \leq j \leq M, 1 \leq k \leq n_j\}$, we formulate the task assignment as an integer programming problem as follows.

$$
\begin{aligned}
\textbf{minimize} \quad & \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{n_j}\sum_{\kappa=1}^{\gamma} b_{j,k}^{i,\kappa} \cdot \mathsf{dist}^*(l_{i,\kappa}^h, l_{j,k}^s) \\
\textbf{subject to} \quad & \sum_{i=1}^{N}\sum_{k=1}^{n_j}\sum_{\kappa=1}^{\gamma} b_{j,k}^{i,\kappa} \cdot \mathsf{div}_{j,k}^{i,\kappa} \geq \beta \cdot \mathsf{div}_j^*, \forall 1 \leq j \leq M, \\
& \sum_{i=1}^{N}\sum_{\kappa=1}^{\gamma} b_{j,k}^{i,\kappa} \leq 1, \forall 1 \leq j \leq M, 1 \leq k \leq n_j, \\
& \sum_{j=1}^{M}\sum_{k=1}^{n_j}\sum_{\kappa=1}^{\gamma} b_{j,k}^{i,\kappa} \leq 1, \forall 1 \leq i \leq N, \\
& b_{j,k}^{i,\kappa} \cdot \mathsf{dist}(l_{i,\kappa}^h, l_{j,k}^s) \leq v(t_j^s - t_{i,\kappa}), \\
& \forall 1 \leq i \leq N, 1 \leq j \leq M, 1 \leq k \leq n_j, 1 \leq \kappa \leq \gamma, \\
& b_{j,k}^{i,\kappa} \in \{0, 1\}, \\
& \forall 1 \leq i \leq N, 1 \leq j \leq M, 1 \leq k \leq n_j, 1 \leq \kappa \leq \gamma.
\end{aligned}
\tag{8}
$$

where $\mathsf{dist}^*(\cdot, \cdot)$ denotes the synthetic travel distance. β is a ratio equal to or larger than 1, indicating the importance that the SSP can achieve the desired diversity order for every sensing task. A larger β value generally guarantees that the desired diversity order can be achieved, but some tasks can be over-fulfilled, leading to a higher cost. In contrast, with a smaller β value such as 1, it is most likely that only some sensing tasks can achieve the desired diversity order while others are under-fulfilled. This may not be desired because the lack of diversity in spectrum sensing can lead to false decisions on spectrum availability, leading to potential interference with PU transmissions. We will fully evaluate the impact of parameter β in our paper. In the above formulation, the first constraint means that the sum of expected diversity order should be no less than the the diversity order required for each sensing task. The second constraint indicates that every subtask can be assigned to at most one participant. The third constraint means that every participant can be assigned at most one subtask. The fourth constraint means that if participant i is assigned to leave at time $t_{i,\kappa}$ from location $l_{i,\kappa}$ at speed v to fulfill subtask $S_{j,k}$, he must arrive no later than t_j^s. Here we assume that each participant can at most complete one sensing subtask for the specified time period. How to assign multiple sensing subtasks to the same participant for sequential completion is left as our future work.

The above problem can be proved to be NP-hard by reducing it to the k-partial set cover problem, which is a generalization of the well studied set cover problem.

Definition 3. (k-partial Set Cover [9]) Given a set $\mathcal{B} = \{b_1, b_2, ..., b_n\}$, a collection $\mathcal{S}$ of subsets of $\mathcal{B}$, $\mathcal{S} = \{S_1, S_2, ..., S_m\}$, a cost function c: $\mathcal{S} \rightarrow \mathcal{Q}^+$, and an integer k, find a minimum cost sub-collection of $\mathcal{S}$ that covers at least k elements of $\mathcal{B}$.

THEOREM 2. *The integer programming problem defined in E-q. 8 is NP-hard.*

PROOF. We first take a look at a special problem derived from the formulation in Eq. 8, where β is 1, and $\mathsf{div}_{j,k}^{i,\kappa}$ is 1 for all valid i, κ, j, k values. The problem derived from Eq. 8 involves a series of timestamps to consider. To simplify the analysis, we first focus on a single timestamp $\kappa \in [1, \gamma]$. For this single timestamp, participants can only contribute a sensing diversity gain of value 1 if they meet the fourth constraint in sensing time. Hence, we can incorporate the time constraint into participant i's new travel distance $\widetilde{\mathsf{dist}}(l_{i,\kappa}^h, l_{j,k}^s)$. When participants satisfy the fourth constraint, we let $\widetilde{\mathsf{dist}}(l_{i,\kappa}^h, l_{j,k}^s)$ equal $\mathsf{dist}^*(l_{i,\kappa}^h, l_{j,k}^s)$. Otherwise, $\widetilde{\mathsf{dist}}(l_{i,\kappa}^h, l_{j,k}^s)$ is ∞. Then we set to obtain the minimum $\widetilde{\mathsf{dist}}(l_{i,\kappa}^h, l_{j,k}^s)$ for each participant among all the timestamps in $[1, \gamma]$, and we denote this value by $\widetilde{\mathsf{dist}}(l_{i,\kappa_{o_i}}^h, l_{j,k}^s)$, where κ_{o_i} is the best timestamp for participant i to leave for sensing location $l_{j,k}^s$ to achieve the lowest travel cost. Hence, the optimization objective can be changed to the minimization of $\sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^{n_j} b_{j,k}^{i,\kappa_{o_i}} \cdot \widetilde{\mathsf{dist}}(l_{i,\kappa_{o_i}}^h, l_{j,k}^s)$. So now if we focus on a single sensing task $j \in [1, M]$, the problem has already been reduced to the k-partial set cover problem as defined above. In the definition, $\mathcal{B}$ corresponds to the subtask set $\{S_{j,k} | 1 \leq k \leq n_j\}$. The collection of $\mathcal{S}$ corresponds to the task assignment: participant i to fulfill subtask $S_{j,k}$ for all i and k. The cost function c maps $\mathcal{S}$ to $\mathcal{B}$ by the cost we defined using $\widetilde{\mathsf{dist}}(l_{i,\kappa_{o_i}}^h, l_{j,k}^s)$. k is the diversity order constraint div_j^*. So the optimization problem is now a k-partial set cover problem. We then need to solve this problem for all $j \in [1, M]$.

Algorithm 2 Sensing Task Assignment

```
Input: Task set $\mathcal{T}$, subtask sets $\{S_{j,k}\}_{1\leq j\leq M, 1\leq k\leq n_j}$, participant
    set $\mathcal{P}$, PIM trace set $\{L_i^o\}_{i=1}^N$.
Output: $\{b_{j,k}^{i,\kappa}\}_{1\leq i\leq N, 1\leq j\leq M, 1\leq k\leq n_j, \lfloor\mu/2\rfloor+1\leq\kappa\leq\gamma-\lfloor\mu/2\rfloor}$.
Smooth $\{L_i^o\}_{i=1}^N$ using Algorithm 1 to obtain $\{L_i^h\}_{i=1}^N$.
for all $j \in \mathcal{T}$ do
    for all $k \in \{1, \ldots, n_j\}$ do
        for all $i \in \mathcal{P}$ do
            for all $\kappa \in \{\lfloor\mu/2\rfloor + 1, \ldots, \gamma - \lfloor\mu/2\rfloor\}$ do
                $b_{j,k}^{i,\kappa} \leftarrow 0$;
                Compute $\mathsf{dist}^*(l_{i,\kappa}, l_{j,k}^s)$ as in Eq. (5).
            end for
        end for
    end for
    $\mathrm{div_j} \leftarrow \beta \cdot \mathrm{div_j^*}$
    while $\mathrm{div_j} > 0$ do
        $\mathsf{dist}^*(l_{i^*,\kappa^*}, l_{j,k^*}^s) = \min\{\mathsf{dist}^*(l_{i,\kappa}, l_{j,k}^s)\}_{\kappa=\lfloor\mu/2\rfloor+1, i\in\mathcal{P}, 1\leq k\leq n_j}^{\gamma-\lfloor\mu/2\rfloor}$;
        $b_{j,k^*}^{i^*,\kappa^*} \leftarrow 1$;.
        Compute $\mathsf{div}_{j,k^*}^{i^*,\kappa^*}$ as in Eq. (6).
        $\mathrm{div_j} = \mathrm{div_j} - \mathsf{div}_{j,k^*}^{i^*,\kappa^*}$;
        $\mathcal{P} \leftarrow \mathcal{P} \setminus \{i^*\}$;
        $\mathcal{S}_j \leftarrow \mathcal{S}_j \setminus \{S_{j,k^*}\}$;
    end while
end for
return $\{b_{j,k}^{i,\kappa}\}_{1\leq i\leq N, 1\leq j\leq M, 1\leq k\leq n_j, \lfloor\mu/2\rfloor+1\leq\kappa\leq\gamma-\lfloor\mu/2\rfloor}$.
```

Since the special problem is NP-hard, we now conclude that the original problem defined in Eq. 8 is NP-hard. □

Note that there are alternative ways to formulate the optimization. For example, it is possible to minimize the expected total synthetic travel distance or the maximum synthetic travel distance. These alternatives are left as future work.

5.6 A Heuristic Solution

We now introduce a heuristic approach to assign subtasks to participants based on their smoothed location traces.

The overall assignment process is summarized in Algorithm 2.

The intuition is to sequentially assign every subtask of one sensing task to each participant with the smallest synthetic travel distance until the total expected diversity order exceeds the required threshold. Specifically, the algorithm takes the sensing tasks $\mathcal{T}$, subtask set $\{S_{j,k}\}_{1\leq j\leq N, 1\leq k\leq n_j}$, participant set $\mathcal{P}$, and PIM trace set $\{L_i^o\}_{i=1}^N$ as input and outputs all subtask assignments. Line 1 smooths all the PIM traces using Algorithm 1. Lines 2 to 10 compute the synthetic travel distance for every participant with every possible departing location and every subtask $\{S_{j,k}\}_{k=1}^{n_j}$. The WHILE loop in Lines 12 to 19 assigns one subtask to one participant, whose synthetic travel distance is the smallest among all. The WHILE loop terminates when the accumulative expected diversity order exceeds the diversity order required for the sensing task T_j.

5.7 Participant Response

The SSP informs every selected participant about the subtask he is assigned to. On receiving the subtask assignment, each participant calculates the true physical and synthetic travel distance using his true predicted locations and then informs the SSP whether he accepts the assignment based on the task acceptance model in Section 5.4. If the participant agrees to fulfill a certain task, he will need to be at the sensing location in the specified time to perform spectrum sensing. Since the participants win the opportunity to perform the task based on the expected mobility traces, the payments or rewards made by the SSP to the participants should be proportional to the travel distances calculated using the expected mobility traces as well. It is possible that the expected mobility traces provided by the participants differ from the real mobility traces. In such cases, participants still need to make sure that they can perform spectrum sensing in a timely manner. The SSP can set up various types of mechanisms to handle the cases when participants fail to fulfill the sensing tasks they previously agreed to fulfill. For example, a reputation system can be constructed to model the reliability of each participant. When participants fail to perform certain tasks, their reputations in the system decrease, and so do their payments received for performing the sensing tasks. In addition, since participants' failure to perform sensing tasks could possibly lead to unsatisfied diversity requirement, the SSP could assign a discounted diversity gain when certain participants with bad history are selected. How to design a fully workable reputation system remains as our future work.

6. SIMULATION RESULTS

In this section, we present the experimental evaluation results of DPSense. We adopt the knowledge construction module in [22] to build the Markov transition matrix, which is implemented in C++. All other modules are implemented in MATLAB on a PC with 2.67 GHz Intel i7 CPU and 9 GB memory.

6.1 Mobility Trace Dataset

We use the CRAWDAD dataset roma/taxi [2, 3] for our simulations. The dataset contains the mobility traces of approximately 320 taxis collected over 30 days in Rome, Italy. Each mobility trace consists of a sequence of GPS coordinates collected roughly every seven seconds along with corresponding timestamps. In addition, the taxis in the dataset are not always moving at a high speed. Those idling at one location or moving within a small region can be used to simulate the static participants or the participants with very limited moving regions. In our simulations, the time difference between two consecutive timestamp is 20 seconds.

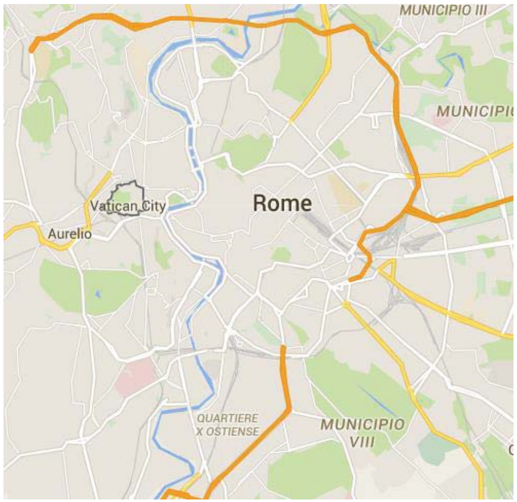


Figure 5: The city area where the mobility traces are extracted.

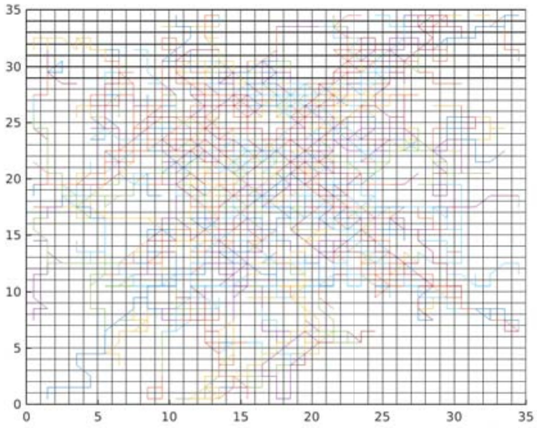
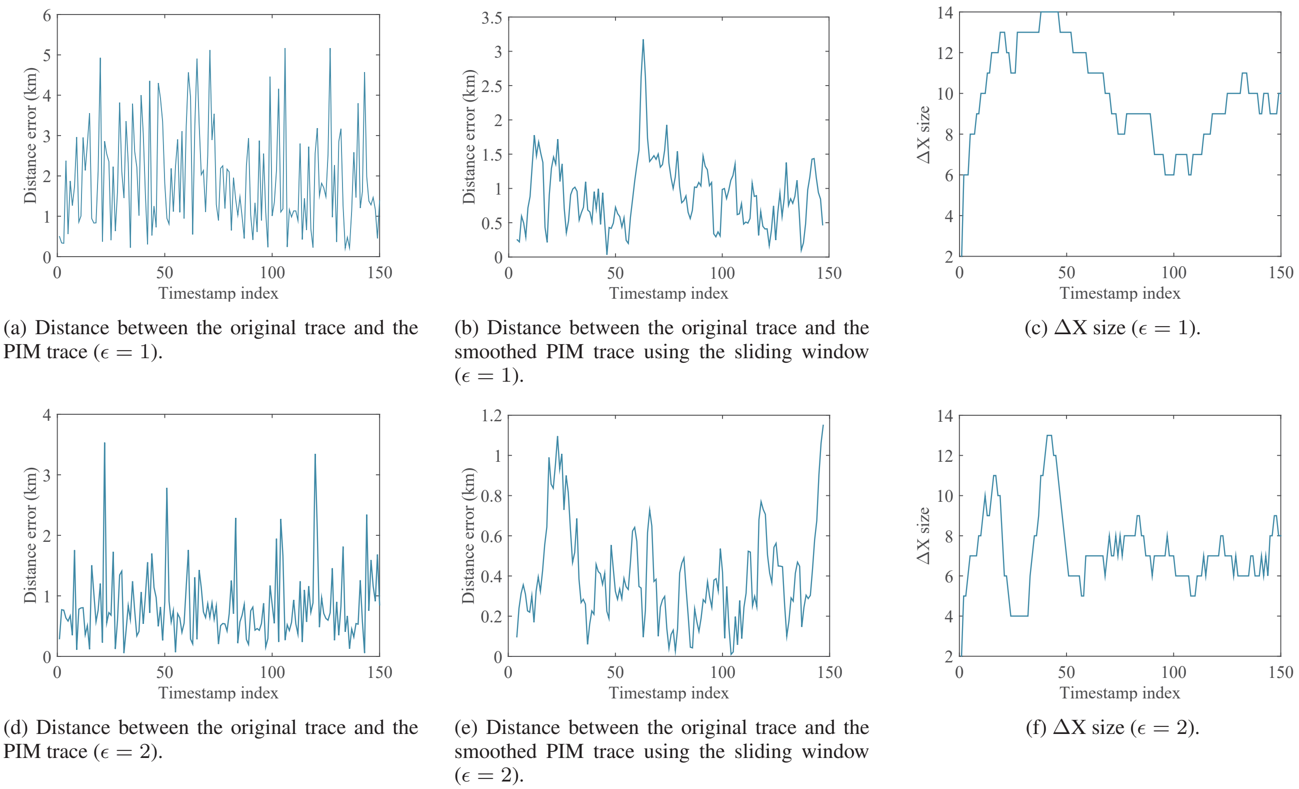


Figure 6: Sampled taxi mobility traces from dataset [2, 3].

The mobility traces within the center of Rome city are extracted. We consider an area of 11.66×11.66 $[\mathrm{km} \times \mathrm{km}]$ as illustrated in Fig. 5. We divide the area into a 35×35 grids of equal size. We then extract 2700 mobility traces in total, each of which contains 150 timestamps. The 2700 mobility traces are shown in Fig. 6. We quantize each GPS coordinate by mapping them into one of the 35×35 cells. As we can see, most of the traces are clustered in the center area, resulting in a very dynamic and diverse transition. The density of mobility traces in the four corners is much lower than that in the center area, making it challenging to correctly track the true locations using the PIM scheme. Out of the 2700 mobility traces, 2000 are used to build the Markov transition matrix, and the remaining 700 are used to represent the participants' input traces in our system. The division of the mobility trace dataset is to emulate the practical application scenarios where the SSP can only obtain the historical mobility data based on some large-scale generic location traces which can be totally unrelated to the participants of our system. Therefore, the construction of the transition matrix does not adversely affect participants' location privacy.

6.2 Simulation Setting

We consider a time period of 50 minutes. The sensing tasks are all scheduled at the later half of the 50 minutes because it takes time for participants to arrive at the designated sensing locations. Since the timestamps are in the granularity of 20 seconds, each sensing task is scheduled at a random one of the last 75 timestamps.



(a) Distance between the original trace and the PIM trace ($\epsilon = 1$).

(b) Distance between the original trace and the smoothed PIM trace using the sliding window ($\epsilon = 1$).

(c) ΔX size ($\epsilon = 1$).

(d) Distance between the original trace and the PIM trace ($\epsilon = 2$).

(e) Distance between the original trace and the smoothed PIM trace using the sliding window ($\epsilon = 2$).

(f) ΔX size ($\epsilon = 2$).

Figure 9: Performance comparison using a single trace.

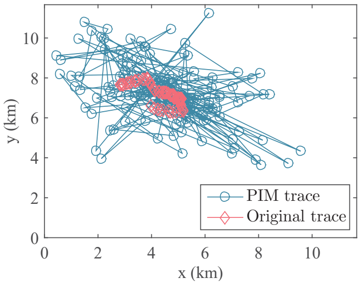


Figure 7: The original trace and the PIM trace ($\epsilon = 1$).

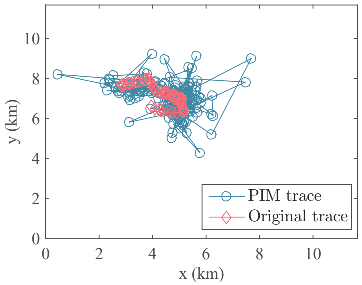


Figure 8: The original trace and the PIM trace ($\epsilon = 2$).

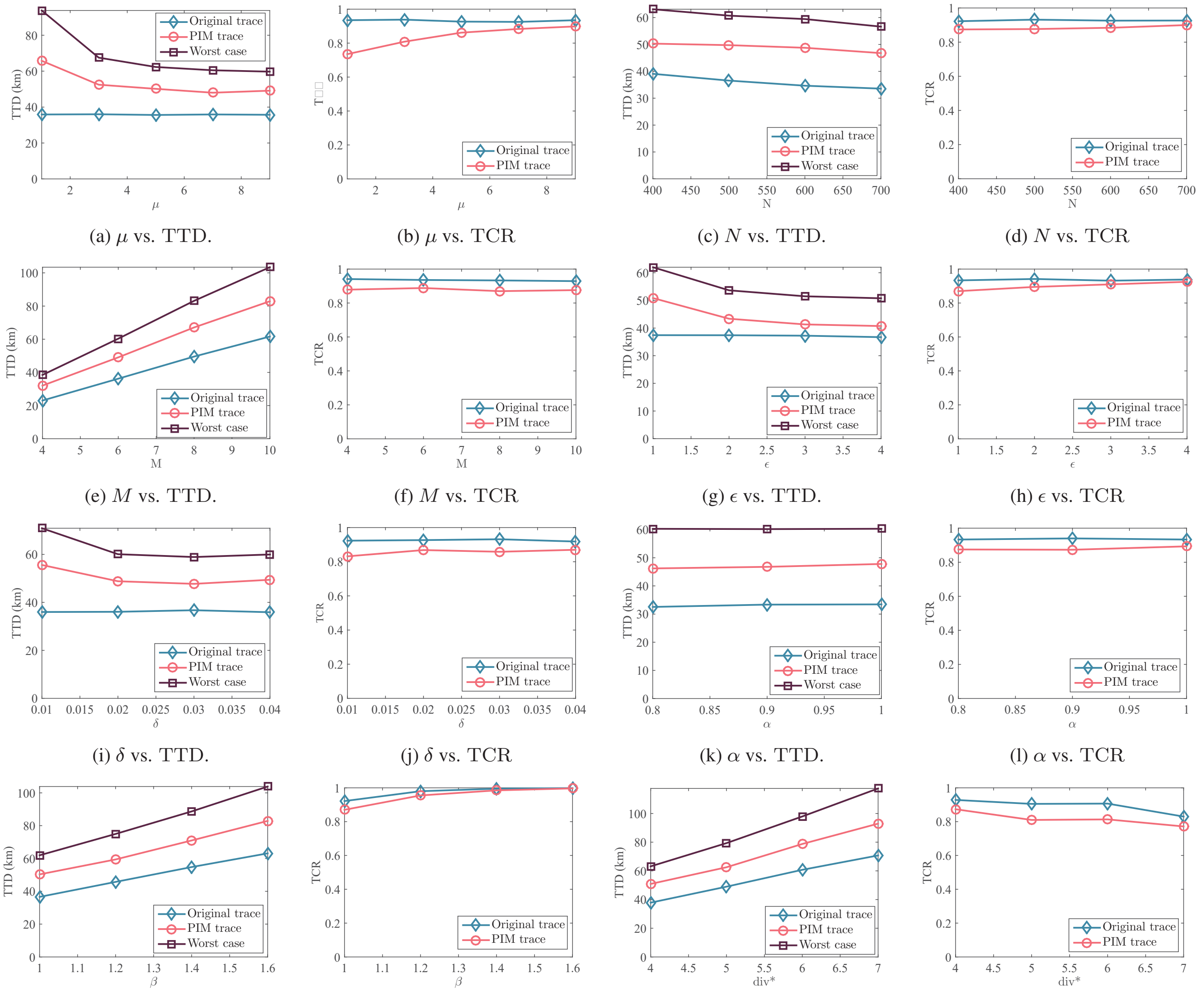
We set the simulation parameters as follows. The numbers in bold are the default values if not mentioned otherwise. For the generation of PIM traces, ϵ is chosen among [**1**,2,3,4], and δ is in the range of [0.01,**0.02**,0.03,0.04]. The number of participants N is [400,**500**,600,700]. For the spectrum-sensing task assignment, the size of the sliding window μ is chosen from [1,3,5,**7**,9], where $\mu = 1$ corresponds to the case where no sliding window is used. In addition, we assume that all the participants have the same traveling speed v=30 km/h. We expect that higher moving speed will deliver better results because participants can travel a longer distance. Other parameters are set as follows. The maximum travel distance MTD is 15 km. The number of sensing tasks M is chosen from [4,**6**,8,10] with the number of subtasks n_j=10 for every sensing tasks. The minimum separation distance between sensing locations d_0 is 20 m. The sensing region for every sensing task is a circle with radius R=300 m. The sensing tasks are randomly generated with the diversity order requirement div* chosen from [**4**,5,6,7]. The system parameter α is in the range of [0.8, 0.9, **1**], and the parameter β is chosen from [**1**, 1.2, 1.4, 1.6].

In our results, each data point represents the average of 100 runs. We use ΔX to represent the δ-location set. We also compare DPSense with the baseline scheme which does not consider location privacy and use raw mobility traces.

6.3 Performance Metrics

For the generation of PIM traces, we compare the *distance error* (i.e., the Euclidean distance between the original trace and the PIM trace for every timestamp) and $|\Delta X|$ (the size of the δ-location set). We also use the following metrics for performance evaluation.

- *Total travel distance (TTD).* This refers to the sum of the expected synthetic distances of the participants who accept assigned subtasks. Specifically, let $c^i_{j,k}$ be the indicator variable such that $c^i_{j,k} = 1$ if participant i accepts the assigned subtask $S_{j,k}$ and zero otherwise. To achieve the minimum synthetic cost, participant i needs to leave at κ^*th timestamp



(a) μ vs. TTD.

(b) μ vs. TCR

(c) N vs. TTD.

(d) N vs. TCR

(e) M vs. TTD.

(f) M vs. TCR

(g) ϵ vs. TTD.

(h) ϵ vs. TCR

(i) δ vs. TTD.

(j) δ vs. TCR

(k) α vs. TTD.

(l) α vs. TCR

(m) β vs. TTD.

(n) β vs. TCR

(o) div* vs. TTD.

(p) div* vs. TCR

Figure 10: The impact of various parameters on TTD and TCR.

for the sensing location $l^s_{j,k}$. TTD is then computed as

$$\text{TTD} = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{n_j} c^{i,\kappa^*}_{j,k} \cdot \mathsf{dist}^*(l_{i,\kappa}, l^s_{j,k}). \tag{9}$$

TTD is commensurate with the total cost of the SSP for performing all the sensing tasks.

- *Task completion rate (TCR).* TCR is the ratio between the number of tasks that meets the specified diversity order requirements and N (the total number of sensing tasks).

We consider the performance comparison of TTD for three cases: the baseline scheme using the original trace $L_i, \forall i \in [1, N]$; the smoothed PIM trace L^h_i, $\forall i \in [1, N]$; the worst case. The worst case assumes that no chosen participant rejects the assigned subtask. Mathematically, the worst-case TTD is defined as

$$\text{TTD}_w = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{n_j}\sum_{\kappa=1}^{\gamma} b^{i,\kappa}_{j,k} \cdot \mathsf{dist}^*(l_{i,\kappa^*_i}, l^s_{j,k}), \tag{10}$$

where κ^*_i is the timestamp when participant i leaves for subtask $S_{j,k}$ with minimum synthetic distance.

6.4 PIM Trace Generation

We evaluate the impact on PIM traces generation by adapting ϵ values. Since the scheme in [27] is adopted to generate the PIM traces, we refer interested readers to [27] for detailed analysis and evaluations. Per our simulations, we find that δ cannot be too large or too small. With a large δ, locations with small prior probabilities are likely to be excluded in the δ-location set, ΔX. This is good in keeping a reasonable size of ΔX. However, it might fail to track location updates as well. On the other hand, with a small δ, more locations with small prior probabilities are likely to be included in ΔX. This might lead to a large ΔX (over 40 or more) and result in failure of correctly tracking the true location. Here, we choose $\delta = 0.02$ to achieve a good trade-off. We will present the impact of δ on our system performance later in Section 6.9.

We first extract one random participant and visually examine the PIM trace generated in Fig. 7 and Fig. 8 with different ϵ. It is obvious that $\epsilon = 2$ generates a PIM trace closer to the original trace,

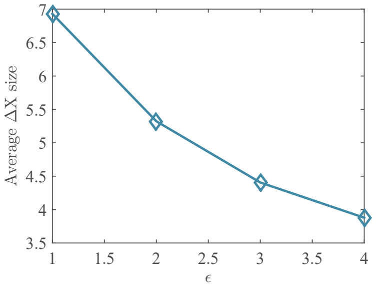


Figure 11: ΔX size for different ϵ.

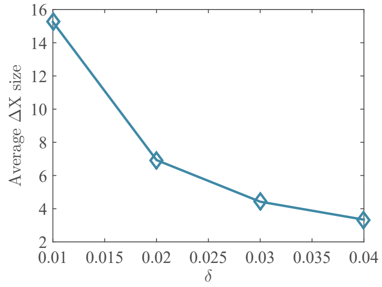


Figure 12: ΔX size for different δ.

which indicates small location errors. This is further confirmed in Fig. 9a and Fig. 9d. In these two figures, we compare the distance between the original trace and the PIM trace for every timestamp. It is clear that when ϵ is 1, the distance sometimes can be very high (e.g., over 5 km). When ϵ increases to 2, the max distance error is around 3.5 km, which is a huge improvement. We further apply the sliding window in Fig. 9b and Fig. 9e. We can see that the sliding window is very effective in reducing the distance errors. For example, when ϵ is 1, the max distance error is reduced from 5.1 km to 3.2 km. The average distance error is greatly reduced as well. Reducing the distance error can greatly benefit TCR because the SSP has more accurate knowledge about the participants' locations. Lastly, we compare the size of ΔX in Fig. 9c and Fig. 9f. We can see that the ΔX size when ϵ is 1 is larger than that when ϵ is 2. This is expected since with a larger ϵ, the mechanism is supposed to track the true location better. We will further evaluate the relationship between ΔX size and ϵ later.

6.5 Effectiveness of Sliding Window

Fig. 10a shows the impact of μ on the total travel distance (TTD). We can see that a larger μ generally results in a smaller TTD. We note that when μ is 1, which corresponds to the case where no smoothing is conducted, with the sliding window in place, TTD has been reduced from 67 km to 48 km, a 28.3% reduction. On the other hand, TTD slows down the reduction when μ is over 5. However, there is still a gap between our scheme and the baseline scheme. This is the expected utility trade-off when incorporating privacy protection. We also show the worst case for comparison. Recall that the worst case is defined as the case that no participant declines the assigned subtask. In other words, SSP simply assigns the sensing task to participants who are the closest to the sensing locations. We see that there is a gap between the worst case and the PIM trace curve. Fig. 10b shows the TCR performance. Here, we see that the TCR is dramatically improved from 0.7 to 0.9. Still, TCR slows increasing when μ passes 5. Generally, the larger μ is, the better performance we can achieve in both TTD and TCR. On the other hand, larger μ means higher computation complexity.

6.6 Impact of Number of Participants

Fig. 10c shows the impact of N on TTD. Generally, the larger N is, the smaller TTD is. This is true for both schemes. Fig. 10d shows the TCR performance. Clearly, more participants can lead to higher task-completion rates. We show the comparison with the baseline scheme using the original trace as well. We see that even without the noise added to the traces, TCR is close to 0.93. It means that some tasks, though with very small probability, might still fail to meet the diversity requirement because the sensing locations are too remote to the majority of participants.

6.7 Impact of Number of Sensing Tasks

Fig. 10e shows the impact of M on TTD. As expected, the distance increases with the number of sensing tasks although the increase is limited. Fig. 10f shows that TCR decreases with M. Since more tasks are generated, SSP might have to select participants far away to perform the tasks, which leads to a higher assignment decline rate. In addition, since some tasks might be generated in areas with low population of participants, these tasks might fail as well.

6.8 Impact of ϵ

We change the value of ϵ in our simulations and evaluate the impact on the performance. Fig. 10g shows the results of TTD. We see that with ϵ increasing, TTD decreases. This indicates that a larger ϵ can generate more precise mobility traces that are close to the original ones. We also observe from Fig. 10h that TCR in our scheme is almost identical to that in the baseline approach when ϵ is 4. In addition, we show the average ΔX size in Fig. 11. We can see that the number of candidate locations drops from approximately 6.9 to approximately 3.9 with the increase of ϵ. This matches our expectation well. When ϵ is small, the scheme generates larger noise to the participants' mobility traces, hence a larger ΔX. This indicates a better location privacy protection as well due to more candidate locations. Correspondingly, it will be more difficult for the attacker to infer the participants' true locations.

6.9 Impact of δ

δ also has a similar impact on the system performance to ϵ, though we find the system is more sensitive to it. Fig. 10i shows the TTD results with different δ values. We can see that there is a relatively big decrement when δ increases from 0.01 to 0.02, and the curve is becoming flat when δ is over 0.02. So this could indicate 0.02 is a good choice of δ for our system. Correspondingly, TCR also generally increases when δ increases but the gain is most observable when we increase δ from 0.01 to 0.02. Recall that a larger δ means a smaller δ-location set and hence worse location privacy. On the other hand, a larger δ can generate mobility traces closer to the original traces, leading to smaller distance errors. We then show the average ΔX size in Fig. 12. It can be observed that the ΔX size is very sensitive to the δ value. The size drops dramatically from 15.3 to 4.4 when δ increases from 0.01 to 0.03. To ensure sufficient location privacy, we find that $\delta = 0.02$ can be a good choice for our system.

6.10 Impact of α

We vary the value of α, the distance weight ratio between the waiting-time equivalent distance and the true travel distance in our proposed synthetic travel distance model as in Eq. 5. Fig. 10k and Fig. 10l show the results of TTD and TCR, respectively. We can observe that the increase in α only contributes to a small increase in both TTD and TCR. This indicates that our system is consistent with all the distance models that practical systems might have. In addition, the value of α might directly relate to the payment to each participant. A smaller value of α could indicate smaller payments because less weight is assigned to the waiting-time equivalent distance. Since we do not propose additional payment schemes in our framework, we omit further analysis in this regard.

6.11 Impact of β

In some cases, it is strictly required that the desired sensing diversity be achieved to completely avoid any potential transmission interference to PUs. β is such a system parameter that enables the system to adjust the priority. Fig. 10m and Fig. 10n show the simulation results of TTD and TCR, respectively. Specifically, we see that a larger β leads to an increased TTD value. This is intuitive because a larger β value usually indicates that some sensing tasks are over-fulfilled, i.e., more than a desired number of participants are selected to perform the sensing tasks. The benefit of a larger β is also clearly shown in Fig. 10n where we can see that TCR is almost 1 when β is 1.4 and 1 when β is 1.6. In other words, a larger β ensures that all sensing tasks can be fulfilled with enough participants to guarantee sufficient spatial sensing diversity. It is also worth noting that when β is 1, those tasks that are not fulfilled are usually located remotely to most participants. In practice, the SSP can thus dynamically determine the value of β for each sensing task based on the practical demographic properties of the areas where the sensing tasks are located. This strategy ensures that the majority of sensing tasks can be fulfilled and at the same time manages to reduce the unnecessary TTD (or cost) that could be incurred.

6.12 Impact of div^*

Lastly, we evaluate the impact of the diversity requirement div^*. Fig. 10o shows the results of TTD. Clearly, with a larger div^*, more participants are likely to be selected to fulfill the sensing tasks, thus resulting in a dramatic increase in TTD. Fig. 10p shows the change of TCR. We see that a higher diversity order is very demanding, generating a larger negative impact on TCR in contrast to the results from varying M in Fig. 10e. Specifically, the decrease of TCR is found in both the baseline scheme and our scheme.

7. RELATED WORK

In this section, we discuss some prior work that is most germane to DPSense.

There is a rich literature on location privacy in general frameworks, for which a nice review for location privacy-preserving mechanisms (LPPMs) can be found in [16]. In addition, a formal framework for the analysis of LPPMs is proposed in [22].

Significant efforts have also been made to protect location privacy in CSS systems [10, 15, 17, 25]. This line of work aims at preventing location privacy leakage from sensing reports submitted by crowdsourced sensing users. We proposed a privacy-preserving crowdsourced spectrum sensing framework in [15] that can simultaneously achieve three design objectives: differential location privacy, approximate social cost minimization, and truthfulness. The spectrum service provider is assumed to be fully trusted in [15]. In contrast, here we assume that the SSP is honest-but-curious and we address a very different problem in this paper.

Another line of work aims to address location privacy leakage in general crowdsourced mobile sensing systems [19, 23]. To *et al.* [23] proposed a framework to protect location privacy of workers during the task assignment phase. Different from [23], DPSense does not need the trusted service provider as in their work to perform the sanitized database release and geocast of spatial crowdsourcing tasks. In addition, DPSense targets a totally different application scenario in which spectrum-sensing tasks have strict sensing time requirements. Pournajaf *et al.* [19] considerd spatial task assignment for crowd sensing with cloaked locations. Different from this work, DPSense considers completely different system models and involves the time constraint of the sensing tasks. The task scheduling and the probabilistic model for participants to accept/decline sensing tasks makes it challenging or impossible to adapt the scheme [19] to our application scenario.

In addition, there is a surge of interest on task assignment in spatial crowdsourcing [4, 5, 13]. He *et al.* [13] seek to maximize the rewards of the platform with consideration of geographic locations and time budgets of mobile users. But they did not consider maximizing the task fulfillment ratio, which is a critical design objective in the context of CSS. Cheng *et al.* [4] aim to maximize both the spatial and temporal diversity of spatial crowdsourcing tasks but do not consider the minimization of travel distances. Deng *et al.* are the first to study the combination of task assignment and scheduling in spatial crowdsourcing [5], but their work differs from DPSense in two main aspects. First, the task assignment in [5] is based on known participants' locations and does not provide any location privacy guarantee. Second, the tasks in their model have deadlines such that participants can perform the tasks any time before the deadline. This is different in our scenario where spectrum-sensing tasks have strict requirement on the sensing time. Hence, it is nontrivial to directly extend these existing efforts to the context of CSS.

Differential privacy [7, 18] has emerged as a powerful tool to provide statistical guarantee of the data privacy with the trade-off of the data utility. Xiao *et al.* in [27] found that the well known l_1 norm sensitivity fails to capture the geometric sensitivity in the two dimensional space and proposed a planar isotropic mechanism for the location perturbation, which is the first to achieve the lower bound of differential privacy in the specific application scenario.

8. CONCLUSIONS

Dynamic spectrum access (DSA) has great potential to address worldwide spectrum shortage by enhancing spectrum efficiency. As a key enabler for DSA systems, crowdsourced spectrum sensing (CSS) allows a spectrum sensing provider (SSP) to outsource the sensing of spectrum occupancy to distributed mobile users. In this paper, we proposed DPSense, a novel framework that allows the SSP to select mobile users for executing spatiotemporal spectrum-sensing tasks without violating the location privacy of mobile users. Detailed evaluations on real location traces confirmed that DPSense can provide differential location privacy to mobile users while ensuring that the SSP can accomplish spectrum-sensing tasks with overwhelming probability and also the minimal cost.

9. ACKNOWLEDGMENTS

We thank our shepherd and anonymous reviewers for their comments and help in preparing the final version of the paper. This work was partially supported by the US National Science Foundation under grants CNS-1320906, CNS-1421999, CNS-1514381, CNS-1422301, and CNS-1514014.

10. REFERENCES

[1] Cisco visual networking index global mobile data traffic forecast update 2014-2019.

[2] R. Amici, M. Bonola, L. Bracciale, A. Rabuffi, P. Loreti, and G. Bianchi. Performance assessment of an epidemic protocol in vanet using real traces. In *MoWNeT'14*, Rome, Italy, Sept. 2014.

[3] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi. CRAWDAD dataset roma/taxi (v. 2014-07-17).

Downloaded from http://crawdad.org/roma/taxi/20140717, Jul. 2014.

[4] P. Cheng, X. Lian, Z. Chen, R. Fu, L. Chen, J. Han, and J. Zhao. Reliable diversity-based spatial crowdsourcing by moving workers. In *VLDB'15*, Kohala Coast, HI, June 2015.

[5] D. Deng, C. Shahabi, and L. Zhu. Task matching and scheduling for multiple workers in spatial crowdsourcing. In *SIGSPATIAL'15*, Bellevue, WA, Nov. 2015.

[6] D. Duan, L. Yang, and J. Principe. Cooperative diversity of spectrum sensing for cognitive radio systems. *IEEE Transactions on Signal Processing*, 58(6):3218–3227, June 2010.

[7] C. Dwork. Differential privacy. In *ICALP'06*, Venice, Italy, Jul. 2006.

[8] O. Fatemieh, R. Chandra, and C. Gunter. Secure collaborative sensing for crowdsourcing spectrum data in white space networks. In *DySPAN'10*, Singapore, Apr. 2010.

[9] R. Gandhi, S. Khuller, and A. Srinivasan. Approximation algorithms for partial covering problems. *Journal of Algorithms*, 53(1):55–84, 2004.

[10] Z. Gao, H. Zhu, S. Li, S. Du, and X. Li. Security and privacy of collaborative spectrum sensing in cognitive radio networks. *IEEE Wireless Communications*, 19(6):106–112, Dec. 2012.

[11] A. Ghasemi and E. Sousa. Collaborative spectrum sensing for opportunistic access in fading environments. In *DySPAN'05*, Baltimore, MD, Nov. 2005.

[12] D. Gurney, G. Buchwald, L. Ecklund, S. Kuffner, and J. Grosspietsch. Geo-location database techniques for incumbent protection in the tv white space. In *DySPAN'08*, Chicago, IL, Oct. 2008.

[13] S. He, D. Shin, J. Zhang, and J. Chen. Toward optimal allocation of location dependent tasks in crowdsensing. In *INFOCOM'14*, Toronto, Canada, Apr. 2014.

[14] B. Hecht and D. Gergle. On the "localness" of user-generated content. In *CSCW '10*, Savannah, GA, Feb. 2010.

[15] X. Jin and Y. Zhang. Privacy-preserving crowdsourced spectrum sensing. In *INFOCOM'16*, San Francisco, CA, Apr. 2016.

[16] J. Krumm. A survey of computational location privacy. *Personal Ubiquitous Comput.*, 13(6):391–399, Aug. 2009.

[17] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen. Location privacy preservation in collaborative spectrum sensing. In *INFOCOM'12*, Orlando, FL, Apr. 2012.

[18] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *FOCS'07*, Providence, RI, Oct. 2007.

[19] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka. Spatial task assignment for crowd sensing with cloaked locations. In *MDM'14*, Brisbane, Australia, Jul. 2014.

[20] Y. Selen, H. Tullberg, and J. Kronander. Sensor selection for cooperative spectrum sensing. In *DySPAN'08*, Chicago, IL, Oct. 2008.

[21] S. Shellhammer, S. Shankar, R. Tandra, and J. Tomcik. Performance of power detector sensors of dtv signals in ieee 802.22 wrans. In *TAPAS'06*, Boston, MA, Aug. 2006.

[22] R. Shokri, G. Theodorakopoulos, J. Boudec, and J. Hubaux. Quantifying location privacy. In *S&P'11*, Oakland, CA, May 2011.

[23] H. To, G. Ghinita, and C. Shahabi. A framework for protecting worker location privacy in spatial crowdsourcing. In *VLDB'14*, Hangzhou, China, Sept. 2014.

[24] D. Tse and P. Viswanath. *Fundamentals of Wireless Communications*. Cambridge University Press, 2004.

[25] W. Wang and Q. Zhang. Privacy-preserving collaborative spectrum sensing with multiple service providers. *IEEE Transactions on Wireless Communications*, 14(2):1011–1019, Feb. 2015.

[26] Wikipedia. https://en.wikipedia.org/wiki/Google_Now.

[27] Y. Xiao and L. Xiong. Protecting locations with differential privacy under temporal correlations. In *CCS'15*, Denver, CO, Oct. 2015.

[28] Q. Zhao and B. Sadler. A survey of dynamic spectrum access. *IEEE Signal Processing Magazine*, 24(3):79–89, May 2007.