Trading Utility for Privacy in Shared Spectrum Access Systems

Matthew Clark*†, Konstantinos Psounis*
*University of Southern California, Los Angeles, CA
{clarkma,kpsounis}@usc.edu
†The Aerospace Corporation, El Segundo, CA

***Abstract*—In an effort to meet growing demands on the radio frequency spectrum, regulators are exploring methods to enable band sharing among a diverse set of user devices. Proposed spectrum access systems would dynamically assign spectrum resources to users, maintaining databases of spectrum use information. While these systems are anticipated to increase the efficiency of spectrum sharing, incumbent users have raised concerns about exposing details of their operations and have questioned whether their privacy can be protected.**

In this paper, we explore whether primary users can retain a critical level of privacy in a spectrum access system setting, where they must reveal some information to enable dynamic access to the spectrum by other users. Under a variety of operational scenarios and user models, we examine adversary techniques to exploit the spectrum access system and obfuscation strategies to protect user privacy. We develop analytical methods to quantify the resulting privacy and validate our results through simulation. To our knowledge, this is the first work that considers inference attacks on primary users in the setting of a highly dynamic spectrum access system. Privacy analysis of this kind will help to enable adoption of shared spectrum access systems by allowing incumbent users to quantify and mitigate risks to their privacy.

I. Introduction

With rapid growth of radio frequency spectrum hungry applications, regulators are exploring the feasibility of sharing spectrum bands among increasingly broad types of systems. In the United States, the Federal Communications Commission is conducting a proceeding on a new Citizens Broadband Radio Service (CBRS) in the 3550-3700 MHz band where recent rulings specify that spectrum sharing should be accomplished through advanced shared spectrum access systems (SAS) [1], [2]. The proposed SAS are intended to protect a tier of primary users (PUs) from harmful interference while dynamically assigning spectrum resources to lower tier, or secondary user (SU), devices. Assignments are determined from databases of policy and spectrum use information. In prior database based spectrum sharing efforts, such as for television white space [3], the PU systems were well defined and anticipated to change relatively slowly. Assignment policies were determined a priori based on interference analysis of the most severe potential cases. The SAS in the CBRS, however, are envisioned to be more dynamic, responsive and generally capable of supporting a diverse set of operational scenarios and heterogeneous networks.

Many of the incumbent systems in 3550-3700 MHz are operated by government entities e.g., Department of Defense radars. Between the databases of information held by the proposed sharing systems and their dynamic nature, incumbent users have raised concerns about maintaining the privacy of their operations with these systems. The information that a SAS would need to assign spectrum resources, such as locations, frequencies, time of use, and susceptibility to interference, may be considered very sensitive by the incumbents and should be protected from exposure to a potential adversary. However, typical cyber security approaches are not alone sufficient as the normal operation of the SAS may allow an adversary to deduce critical aspects of a PU operation. For example, an adversary may legitimately operate a number of cell phones within a secondary network and leverage assignments from the SAS to make inferences about the characteristics of a military radar.

In this paper, we study methods and measures for maintaining the privacy of PUs operating with a SAS. We consider a variety of operational PU models and adversary techniques to exploit the SAS. Our focus is on the inherent privacy exposure associated with the intended operations of the SAS and we refer the reader to the literature for works on other cyber security threats [4]. We examine PU obfuscation strategies to protect privacy without reducing the utility of the shared spectrum to unacceptably low levels. As a key result, we offer a lower bound for the expected time the privacy of a static PU can be maintained with a given SAS, obfuscation strategy, and threat model. This bound can be tied directly to PU operational requirements. We leverage this metric in the study of dynamic PU scenarios where we treat the PU privacy problem of selecting strategies to achieve privacy requirements while having a minimal impact on the utility of the spectrum for SUs. In this way, we identify a fundamental tradeoff between PU privacy and SU utility in the design of a SAS, and offer a method to find the set of dominant privacy strategies. While other works have studied PU privacy under relatively static sharing systems (e.g., based on television white space models) [5], [6], to our knowledge, this is the first work that provides privacy guarantees with a SAS model that is generally applicable to highly dynamic use cases and heterogeneous users. We find that the dynamic system model substantially affects the analysis as well as considerations for both PU privacy and adversary inference strategies. Privacy analysis of this kind may help to enable wider adoption of shared spectrum access systems by allowing incumbent users to quantify the risk posed to their privacy and by providing specific techniques to mitigate that risk.

The organization of the rest of this paper follows. Related work is reviewed in Section II. The system model is given in Section III with a detailed model for the SAS in III-A, the adversary model in III-B, and the PU privacy model in Section III-C. Several practical adversary estimation schemes are considered in Section IV. As a case study, we analyze location privacy in Section V, considering stationary PUs in V-A and dynamic PUs in V-B. Our conclusions and a discussion of future work are offered in Section VI.

Throughout the paper, we will use uppercase to denote vectors and their elements, and subscripts to index the elements. Lowercase will denote scalar variables, where subscripts are used to distinguish variables that are similar in nature. Similarly, we use superscripts to distinguish related vectors. We will also use superscripts to denote a time index, but will ensure the meaning is clear from context. Calligraphic font will be used to denote sets, and bold face to denote random variables.

II. Related Work

The requirements of the SAS limit the applicability of other works on the privacy of general statistical databases. Differential privacy, for example, provides a measure of privacy and useful results for statistical database design [7], [8]. By definition, a randomized database operator $K : \mathcal{X} \rightarrow \mathcal{K}$ is ϵ-differentially private if for any two databases $\mathcal{X}^1, \mathcal{X}^2 \subseteq \mathcal{X}$ that differ by one entry, the following is satisfied:

$$Pr(K(\mathcal{X}^1) \in \mathcal{S}) \leq e^{\epsilon} Pr(K(\mathcal{X}^2) \in \mathcal{S}), \forall \mathcal{S} \subseteq \mathcal{K}.$$

The competing goals of the SAS, i.e., protecting PUs from harmful interference and offering non-negligible utility to the SUs, prevents it from being differentially private. To see this, let K be the SAS operation to grant assignments to SUs. Suppose $\mathcal{S}$ is the set of assignments that would cause harmful interference to the operations of a single PU radar described by $\mathcal{X}^1$. To avoid harmful interference, the SAS must ensure $Pr(K(\mathcal{X}^1) \in S) = 0$. Suppose $\mathcal{X}^2$ describes a database where no PUs are operating. The SAS assignment operator can only be ϵ-differentially private if $Pr(K(\mathcal{X}^2) \in S) = 0$, i.e., the SAS must protect the radar in $\mathcal{X}^1$ from harmful interference in all other cases that differ by one entry, including the case that no PUs are operating. Because this must hold for any arbitrary $\mathcal{X}^1$, this requires the SAS to simultaneously protect any potential PU operation, effectively precluding SU access to the spectrum. Thus the SAS cannot be considered differentially-private and still provide SU access to the spectrum. Other statistical database methods, including relaxed versions of differential privacy [9], offer some mechanisms to treat this problem, but the meaningfulness of these measures for the SAS setting are also questionable [10]. In this paper we seek to identify measures of privacy that are meaningful for the SAS and that can be related back to PU operator requirements.

Of the many works specific to spectrum sharing, there are relatively few that address privacy issues. Several works consider SU privacy. [11]–[13] assume that SU requests can be aggregated to preserve privacy. This approach is not necessarily applicable to PU privacy, particularly if PU operations are sparse or the users share a common identity, where both are true for military PUs in CBRS. However, the use of false database entries as in [11], [12] and random perturbation of entries as in [13] are basic privacy preservation mechanisms that can also be applied to the case of PU privacy. [14] applies a secure computation technique to maintain SU privacy, but does not allow for spatial reuse of spectrum by SUs, severely limiting the potential efficiency of a SAS in practice.

The privacy of both PUs and SUs is considered in [15], where all users are assumed to want to keep their information private from the SAS itself. The proposed encryption method along with introduction of a fourth party ”key distributor” to perform a subset of the SAS operations is effective in keeping user information private from the SAS. However, this approach does not offer protection against an adversary that is able to hack both the SAS and the key distributor simultaneously, nor does it address the issue of PU privacy loss due to inference attacks from the information exposed to the SUs.

PU privacy with a SAS is also considered in [6] where the authors assess strategies for the SAS falsely denying SU resource assignments to protect PU privacy. The analysis is provided in the context of a system with binary decisions where either any given channel is available to an SU at a particular timeslot or it is not. This assumption limits the applicability of the analysis and results to PU time-of-use privacy. Both this time-of-use privacy problem and the falsification/obfuscation strategy used can be considered as special cases of a more general SAS privacy framework.

The most relevant works addressing the PU privacy problem are [5] and [16]. In both works, PU privacy preserving obfuscation methods are studied for a SAS that grants transmit power assignments based only on the distance of an SU to the nearest PU. This approach is consistent with the television white space model, where the PUs are well defined and conservative assumptions can be made without substantial loss of efficiency. Despite its merit in such settings, this approach is too limiting for more general future systems designed to be efficient for a diverse group of PU and SU devices. In a more heterogeneous system, worst case assumptions, e.g., fixed margins for aggregate interference, will quickly degrade the utility of the spectrum for SU networks. Efficient use of the spectrum requires a more dynamic SAS that accounts not just for the point-to-point distance between a single PU and SU, but for the overall topology of both the PU and SU systems, along with their operating characteristics. In this paper, we formulate a model for a dynamic SAS and find that this model significantly affects the analysis and resulting strategies in the PU privacy problem.

III. System Model

We offer a formal model for the study of PU privacy in a dynamic SAS setting. The model for the SAS includes the information exchanged between the SAS, PUs, and SUs that enable the SAS to grant spectrum access to SUs while also protecting PUs from harmful interference. We model an adversary that makes inference attacks based on the information gathered from the SAS such that PU privacy depends on the effectiveness of the adversary in learning sensitive attributes of the PU system, requiring relevant metrics to measure PU privacy as well as methods to achieve a target privacy level.

Recognizing the potential for application to a wide variety of spectrum sharing scenarios, we will keep the discussion general. However, to help make fundamental concepts clear, we will also refer to a specific working axample motivated by CBRS. In this example, we assume PUs are military radars susceptible to interference from SUs, but have a short pulse duration such that their interference to SUs is neglible. For the SUs, we assume a cellular network where the user equipment (UE) transmits to base stations (BS) in the shared spectrum, and where the BS transmissions take place in another frequency band and are not relevant to the PU privacy. Note that we could alternatively consider a SU scenario where the BSs transmit to the UEs, but the UE transmission case is actually more general in that it allows us to explore the effect of SU transmitter mobility on PU privacy.

A. The Spectrum Access System

The SAS operates on a database of policies and real-time information provided by PU and SU operators. In CBRS, there are three tiers of users with the higher tiers having priority. We only include two tiers in our model however, PUs and SUs, recognizing that loss of privacy is greatest between the first two tiers. Results based on two tiers may be applicable more generally since, e.g., a third tier would not be able to differentiate between the first and second tier users, and would treat them together as a single group.

SU Information Requirements: An SU requests an assignment from the access system by sending information on its present location and useful parameter ranges via a connection that does not rely on spectrum from the SAS. Specifically, for SU devices indexed from 1 to n_s, the ith SU device will provide the range of transmission powers it can profitably use as elements of the vectors $P^{min} = [P_i^{min}] \in \mathbb{R}_+^{n_s}$ and $P^{max} = [P_i^{max}] \in \mathbb{R}_+^{n_s}$ that will be kept by the access system. The SU device, referred to as SU for brevity from this point on, will also provide a set of frequency tuning ranges as well as a range of useful bandwidths. To simplify the notation, we will assume these SU restrictions are captured in a set of feasible assignments $\mathcal{P}$. We also treat the SU location as a scalar index for a cell in a discretized region, although in practice, it may be given as a vector of coordinates. The SAS stores all SU locations in a set we denote $\mathcal{L}^s \subseteq \mathcal{L}$, where $\mathcal{L}$ is the set of all cell locations in the considered region. To allow the access system to take advantage of frequency dependent scheduling, an SU may also send channel state information for the links in the SU network. Assuming n_c discrete frequency channels, the SAS will store all SU reported channel gains in an array $G^s = [G_{k,i}^s]$, where k is the frequency channel index and i is the index for the ith SU. In our working example, one or more cellular network operators will send information about the UE transmissions they would like to schedule. Frequency, power and bandwidth ranges identify device hardware limitations as well as application specific requirements, while G^s specifies the estimated channel gains on the UE to BS transmission path.

PU Information Requirements: PUs will send their location information to the SAS, which stores the information in a set denoted by $\mathcal{L}^p \subseteq \mathcal{L}$. The access system will assume a propagation model with uncertainty, i.e., $\mathbf{G}^p = [\mathbf{G}_{k,i,j}^p] \in \mathbb{R}^{n_c \times n_s \times n_p}$ is the random array for the channel gains between each PU (e.g., radar) and SU device (e.g., UE) with indices k, i and j corresponding to the frequency channel, the SUs and the n_p PUs identified to the SAS respectively. The PUs must also report their interference criteria, given by vectors $I^{th} \in \mathbb{R}_+^{n_p}$ and $\Lambda^{th} \in \mathbb{R}_+^{n_p}$. I_j^{th} is the harmful received interference power threshold for the jth PU and $0 < \Lambda_j^{th} < 1$ is a reliability requirement for the PU, i.e., the maximum probability that the threshold given by I_j^{th} can be exceeded. This reliability parameter accounts for inherent uncertainty in the resulting interference due to the random $\mathbf{G}^p$. In our example, I^{th} could be the received interference power that would prevent a radar from detecting a target. Since a practical radar will have some baseline probability of missed detection even without SU interference, selecting Λ^{th} to be significantly less than this baseline probability will ensure that sharing spectrum with SUs will have a negligible effect on the overall probability of missed detection. We will denote the information provided by all PUs to the SAS with the set $\mathcal{X} = \{\mathcal{L}^p, I^{th}, \Lambda^{th}\}$ and denote the set of all possible PU information sets as $\mathbb{X}$.

SAS Assignments to SUs: The SAS will manage a set of frequency channels $\mathcal{F} = \{1, ..., n_c\}$, and will also use discrete power levels and discrete time slots. The SAS will grant spectrum assignments to SUs by specifying the frequency channel and maximum power level the SUs are permitted to use for the duration of a time slot. SU assignments cannot be changed by the SAS until the start of the next time slot. A short time slot duration will allow the SAS to rapidly react to changing conditions, but will require greater communication overhead between the SAS and SUs. Selecting a time slot duration thus is a trade between efficiency and complexity of the system. For each upcoming time slot, the SAS will need to solve a scheduling problem to maximize some utility function subject to constraints protecting the PUs from harmful interference. Deferring consideration of PU privacy for the moment, we formulate a SAS assignment function $f()$ operating on n_s SUs and n_p PUs that returns maximum transmit power assignments for each SU-channel pair as an array $P = [P_{k,i}] \in \mathbb{R}_+^{n_c \times n_s}$ as

$$\begin{aligned} P = & f(\mathcal{L}^p, \mathcal{L}^s, G^s, I^{th}, \Lambda^{th}, P^{min}, P^{max}, \mathcal{P}) \\ & \approx \arg\max_{P' \in \mathcal{P}} \quad U(P', G^s, P^{min}, P^{max}) \\ & \qquad \text{subject to} \quad Pr\left(\sum_{k=1}^{n_c}\sum_{i=1}^{n_s} P'_i \mathbf{G^p_{k,i,j}} \geq I^{th}\right) \leq \Lambda^{th} \\ & \qquad\qquad 1 \leq j \leq n_p, \end{aligned} \tag{1}$$

where $U()$ is an SU utility function. Note that a power assignment of zero is possible and corresponds to excluding a particular SU from being served in the corresponding frequency channels during this time slot. In this way, $f()$ acts as an admission control, channel assignment, and power assignment function. The basic exchange of information between the PUs, SUs and the SAS for a single frequency channel and timeslot is diagrammed in Figure 1. In our example, denying an assignment to an individual UE is an indication to the cellular operator that, relative to other UEs, the interference impact from this UE on one or more PUs is too great relative to the utility that could be achieved by granting the assignment. Granting a UE an assignment with very low permissible transmit power is a similar, although weaker indication to the cellular operator.

Identifying solutions to (1) is non-trivial. For the purpose of this paper, we will offer a general methodology, but when a specific form for $f()$ is called for in the following results, we will limit the analysis to a single frequency channel for ease of notation, and make use of the algorithm in [17]. There are many prior works in the literature that develop power assignment algorithms, see, for example, [18]–[20]. We opt to use [17] because it accounts for the network topology of the SUs, the aggregate interference due to multiple interfering SUs, the need for the algorithm to be low in complexity, management of control information overhead, and the uncertainty in estimating the interference that will result for a particular scheduling decision, all of which are critical for a dynamic SAS. The basic method in [17] is to determine with a function $I()$, the maximum mean equal interference power level that can be caused by each of the SUs and still satisfy the interference constraints. $I()$ is specific to the uncertainty model assumed and is negatively correlated with the number of SUs that will receive nonzero power assignments, which we denote by a, $0 \leq a \leq n_s$, and where a is selected to maximize $U()$. The power assignment P_i for the ith SU is given by

$$P_i = \frac{I(a, I^{th}, \Lambda^{th})}{\max_j \bar{G}^p_{ij}}, \tag{2}$$

where we drop the channel index subscript to simplify the notation in the single channel case. In order to handle multiple PU locations, the SAS computes the mean SU-to-PU channel gain, $\bar{G}^p_{ij}$, i.e., the mean of $\mathbf{G}^p_{ij}$, from the inputs $\mathcal{L}^p$ and $\mathcal{L}^s$. The maximization in the denominator of (2) clusters SUs with the PUs they will interfere with most, such that equal interference assignments are computed for each cluster of SUs separately. a therefore depends on the clustering, and we define a vector $A = [A_i] \in \mathbb{R}^{n_s}_+$ to denote, for each SU, the number of SUs that receive nonzero power assignments in the same PU cluster. We also include a small margin in the output of $I()$ to account for interference sources other than those in the specific PU cluster.

The utility $U()$ may be left general, addressing considerations including throughput, fairness and multiple access among SUs. In the remainder of this work, we assume a single SU network where fairness between SUs is assumed to be handled external to the SAS, e.g., by the cellular network operator, allowing us to focus on privacy considerations. Specifically, when a specific form for $U()$ is required, we will use the sum-rate of the SUs as our metric, i.e., $U(P, G^s, P^{min}, P^{max}) = \sum_{i=1}^{n_s} U_i(P_i, G^s_i, P^{min}_i, P^{max}_i)$, where for $P_i \geq P^{min}_i$,

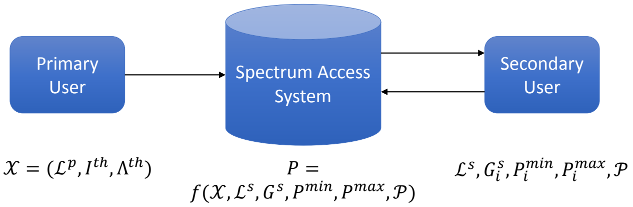

Primary User
Spectrum Access System
Secondary User
$\mathcal{X} = (\mathcal{L}^p, I^{th}, \Lambda^{th})$
$P = f(\mathcal{X}, \mathcal{L}^s, G^s, P^{min}, P^{max}, \mathcal{P})$
$\mathcal{L}^s, G^s_i, P^{min}_i, P^{max}_i, \mathcal{P}$

Fig. 1. Spectrum Access System data exchange model.

$$U_i(P_i, G^s_i, P^{min}_i, P^{max}_i) = \log_2\left(1 + \frac{\min(P_i, P^{max}_i)G^s_i}{\eta}\right) \tag{3}$$

and where η is the thermal noise in the SU receiver. For $P_i < P^{min}_i$, the utility is assumed to be zero.

To incorporate PU privacy, additional constraints may be needed in Problem (1). Further, while we have defined f as a deterministic operator, we should allow for consideration of a random operator to increase privacy. In this case, SU assignments will be realizations of a random variable, which we denote $\mathbf{\Psi} = \Psi \in \mathbb{R}^{n_c \times n_s}_+$. Before we can incorporate these concepts formally, we must first define our adversary.

B. The Adversary Model

We limit our analysis to an adversary that seeks to learn information about the PUs without being detected. Such an adversary will not attempt to disrupt the operations of the SAS, but may use any information collected from the SAS to conduct inference attacks on sensitive aspects of PU systems or operations. In reality, there may be a wide variety of inference attacks that would be a cause for PU concern. For example, a history of location information for a military PU may reveal the identity of the system, its current location, and its destination. All of this may be considered sensitive by a military operator reluctant to reveal details of their deployments. Similarly, PU interference threshold parameters, I^{th} and Λ^{th} might be sensitive in that they would be dangerous in the hands of an adversary intending to field a jammer targeted at that PU system.

Typically, a prospective government PU operator will not publicly reveal which inference attacks would be of greatest concern. Rather than speculate, we assume all the PU parameters represented directly in the SAS model are potentially sensitive. Specifically, we will denote the true state of the PUs on any particular frequency channel at time t with $\tilde{\mathcal{X}}^t$ as the tuple of $\{\tilde{\mathcal{L}^p}^t, \tilde{I^{th}}^t, \tilde{\Lambda^{th}}^t\}$. Note that the true state $\tilde{\mathcal{X}}^t$ may or may not be the same as $\mathcal{X}^t$, i.e., the information provided to the SAS by the PUs. We will discuss PU obfuscation strategies that leverage this in the next subsection.

In an inference attack, the adversary treats unknown PU parameters as random variables and creates estimates of their probability distributions based on observations from the SAS and any information known a priori. We'll denote the unknown PU state from the adversary perspective with $\hat{\mathcal{X}}$ and denote the estimated distribution with $p_{\hat{\mathcal{X}}}$. We will treat these random variables as discrete valued and assume the adversary will tolerate some quantization error. In estimating locations, for example, the adversary will divide an arbitrary region up into cells and create an estimate for the probability a PU is contained within each of the cells. Following the approach in [21] and [22], we will assume a priori information known to the adversary can be modeled with a Markov chain, where the stationary probabilities correspond to the a priori estimate of $p_{\hat{\mathcal{X}}}$ and the transition probabilities correspond to a priori knowledge of the PU dynamics, e.g., $p_{\hat{\mathcal{X}}}(\hat{\mathcal{X}}^{t+1}|\hat{\mathcal{X}}^t)$. The observations available to the adversary will depend on its role in the SAS, and we will consider three different cases.

Adversary Case #1: Compromised SAS. In the worst case, we consider a scenario where the adversary has direct access to the information provided to the SAS by the PU. This could occur if the adversary is able to hack into the SAS or otherwise eavesdrop on the communications between the PU and the SAS. In this case, the adversary observes $\mathcal{X}^t$ directly and PU privacy depends on discrepancies between $\mathcal{X}^t$ and $\tilde{\mathcal{X}}^t$. Specifically, if the PUs employ a randomized obfuscation operator $g_1 : \mathbb{X} \rightarrow \mathbb{X}$, then the adversary can compute $p_{\hat{\mathcal{X}}}(\hat{\mathcal{X}}^t|\mathcal{X}^t)$ based on a priori information about g_1. For example, suppose there is a single PU and g_1 randomly identifies false PU entries to the SAS that are indistinguishable from the true entry. Then if $\mathcal{X}^t$ contains n_p total entries, the adversary estimated probability that any entry in $\mathcal{X}^t$ corresponds to the actual PU is n_p^{-1}. This case represents the greatest privacy threat to PUs as it assumes the maximum exposure of SAS related information to the adversary.[1]

Adversary Case #2: Compromised SU networks. In this case the adversary may have hacked the SU networks, or has found a way to eavesdrop such that it can observe communication between the SAS and all SUs. At every time slot, the adversary will know the SU parameters, i.e., $\mathcal{L}^s$ and $\mathcal{P}$ and will observe the (potentially random) assignment array Ψ, e.g., UE power assignments. After T assignment observations $\{\Psi^t; t = 1, ..., T\}$, the adversary estimated probability that any arbitrary candidate sequence of PU states $\{\hat{X}^t; t = 1, ..., T\}$ corresponds to the true sequence of PU states (e.g., PU locations if location privacy is the goal) can be computed as a standard Bayesian inference problem, i.e.,

$$p_{\hat{\boldsymbol{\mathcal{X}}}^1,...,\hat{\boldsymbol{\mathcal{X}}}^T}(\hat{\mathcal{X}}^1, ..., \hat{\mathcal{X}}^T|\Psi^1, ..., \Psi^T) = \frac{p_{\boldsymbol{\Psi}^1,...,\boldsymbol{\Psi}^T}(\Psi^1, ..., \Psi^t|\hat{\mathcal{X}}^1, ..., \hat{\mathcal{X}}^T)p_{\hat{\boldsymbol{\mathcal{X}}}^1,...,\hat{\boldsymbol{\mathcal{X}}}^T}(\hat{\mathcal{X}}^1, ..., \hat{\mathcal{X}}^T)}{p_{\boldsymbol{\Psi}^1,...,\boldsymbol{\Psi}^T}(\Psi^1, ..., \Psi^T)}, \tag{4}$$

where $p_{\boldsymbol{\Psi}^1,...,\boldsymbol{\Psi}^T}(\Psi^1, ..., \Psi^T|\hat{\mathcal{X}}^1, ..., \hat{\mathcal{X}}^T)$ is the probability of the observed power assignment given the candidate PU state sequence, and the equality is a direct application of Bayes' theorem. Note that for brevity, we have assumed the dependence on the SU parameters is implicit. When the SAS assignments are memoryless, we have $p_{\boldsymbol{\Psi}^1,...,\boldsymbol{\Psi}^T}(\Psi^1, ..., \Psi^T|\hat{\mathcal{X}}^1, ..., \hat{\mathcal{X}}^T) = \prod_{t=1}^T p_{\boldsymbol{\Psi}}(\Psi^t|\hat{\mathcal{X}}^t)$. If the adversary assumes all candidates are equally likely a priori, i.e., $p_{\hat{\boldsymbol{\mathcal{X}}}^1,...,\hat{\boldsymbol{\mathcal{X}}}^T}(\hat{\mathcal{X}}^1, ..., \hat{\mathcal{X}}^T)$ is constant, then it follows from (4) that

$$p_{\hat{\boldsymbol{\mathcal{X}}}}(\hat{\mathcal{X}}^1, ..., \hat{\mathcal{X}}^T|\Psi^1, ..., \Psi^T) = \frac{\prod_{t=1}^T p_{\boldsymbol{\Psi}}(\Psi^t|\hat{\mathcal{X}}^t)}{\sum\limits_{\mathcal{X}^1,...,\mathcal{X}^T} \prod_{t=1}^T p_{\boldsymbol{\Psi}}(\Psi^t|\mathcal{X}^t)}, \tag{5}$$

where the summation in the denominator refers to the sum over all possible $\mathcal{X}^1, ..., \mathcal{X}^T \in \mathbb{X} \times ... \times \mathbb{X}$. If the adversary can compute $p_{\boldsymbol{\Psi}}(\Psi^t|\mathcal{X}^t)$, then computing (5) for every possible candidate sequence will produce the optimal estimated distribution. However, note that the size of the candidate space is likely to be large enough that such an approach is intractable. For example, if the region is divided into w discrete cells, then just the number of candidates for the location set $\hat{\mathcal{L}}^p$ at a single time slot is 2^w. The adversary could potentially use (5) to identify the maximum likelihood estimate, but we will not necessarily have a closed form to facilitate finding the maximum and the general optimization would be non-trivial. Despite these questions about tractability, (5) will be useful as an ideal adversary estimate.

Adversary Case #3: Compromised SU devices. Another form of adversary may only have access to the assignments for a subset of SUs in the secondary networks. This could be the case if the adversary owns and operates its own SU systems, e.g. its own UEs registered on the cellular SU network. In this case, the adversary only observes a subset of the SU assignments. With the assumption that the assignment Ψ is dependent on the entire topology of PUs and SUs, adversary estimation may be significantly more difficult than in case #2. The optimal adversary estimate will be similar to (5), with the addition of unknown SU parameters as random variables. Considering the likely intractability, we omit an explicit extension of (5) for this case. Later, we will offer efficient scenario-specific heuristics instead.

C. *Primary Privacy*

To formally incorporate privacy as a requirement in our model, we need metrics that measure PU privacy in the SAS setting. We may then study obfuscation strategies for satisfying PU privacy requirements.

1) Metrics: There are a variety of privacy metrics in the literature. Any measure should quantify how well an adversary can estimate the actual parameters of the PU(s). One approach is to attempt to quantify the expected error as in [21]. For a set of observations $\mathcal{Y}$ and the set of possible candidates for the estimation $\mathbb{X}$, we can define the expected error as

$$\sum_{\mathcal{X} \in \mathbb{X}} p_{\hat{\boldsymbol{\mathcal{X}}}}(\mathcal{X}|\mathcal{Y})\|\mathcal{X} - \tilde{\mathcal{X}}\|, \tag{6}$$

where $\tilde{\mathcal{X}}$ is the true state of the estimated parameters and $p_{\hat{\boldsymbol{\mathcal{X}}}}$ denotes the adversary's estimate, which may not be optimal.

[1] The adversary may reduce the privacy by other mechanisms of course, e.g., with a sensing capability, but that is beyond the scope of this paper.

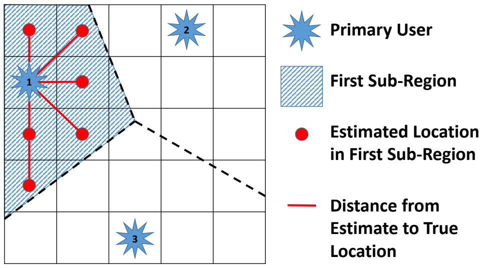

1
2
3
Primary User
First Sub-Region
Estimated Location in First Sub-Region
Distance from Estimate to True Location

Fig. 2. Average distance error.

$\|\mathcal{X} - \tilde{\mathcal{X}}\|$ denotes some measure of distance between the candidate PU state and the true state. Considering that $\mathcal{X}$ will consist of multiple tuples of unlike variables, the most generally meaningful distance measure is not obvious, and should be selected based on the precise threat model.

Specific to the issue of location privacy, the authors in [5] introduced a metric which is the weighted sum of the distance between each possible PU location and the closest real one. This metric does not quite provide a measure of the expected error since the weights, which correspond to the probability that a PU is present in the candidate location, do not sum to one. While we have used a similar metric, together with other metrics, in our prior work [22], here we choose to introduce a couple of new metrics which are more intuitive.

First, we wish to capture the concept of an average distance error, that is, the average of the distances between each estimated location and a real one. We use the actual PU locations to partition the region into n_p sub-regions using a Voronoi diagram approach, i.e., for each PU, the corresponding sub-region is the set of locations closer to that PU than any other in $\mathcal{X}$. For all estimated locations in a sub-region, we normalize their estimated probabilities to sum to one, and compute the sub-region average distance between the PU and the estimates according to the normalized probabilities. We then take the average of this computed distance over all sub-regions to compute the average distance error. More formally, denote these sub-regions $\mathcal{L}^i$ for $i \in \{1, ..., n_p\}$ corresponding to the indices for the PU entries that generate each sub-region. Let ℓ^i be the true location of the ith PU. Then we define the average distance error as

$$\frac{1}{n_p} \sum_{i=1}^{n_p} \sum_{\ell \in \mathcal{L}^i} \|\ell - \ell^i\| \frac{p_{\boldsymbol{\ell}}(\ell|\mathcal{Y})}{\sum_{\ell' \in \mathcal{L}^i} p_{\boldsymbol{\ell}}(\ell'|\mathcal{Y})}, \tag{7}$$

where $p_{\boldsymbol{\ell}}(\ell|\mathcal{Y})$ is the adversary estimated probability that any arbitrary location $\ell \in \mathcal{L}$ is contained in the PU input set $\mathcal{L}^p$ given the observations $\mathcal{Y}$. The norm is the Euclidean distance between the candidate and true locations. We illustrate the computation of this average distance error in Figure 2 for the case of three PUs operating in a rectangular region where the adversary has discretized the region into 25 square estimation cells and has formed an estimate on the probability that a PU is located in each cell. With the Voronoi diagram approach, the first sub-region, $\mathcal{L}^1$, is formed from the cells in the shaded region of the figure which are closest to the true location of the first PU, ℓ^1. The adversary estimates for each cell in the sub-region are normalized to sum to one, and then used to compute the expected distance between the adversary estimates and ℓ^1 corresponding to the inner summation in (7). This process is repeated to compute the expected distance error of the other two sub-regions, and we take the average of the three to be our average distance error metric.

Even without any information from the SAS, one may argue that an adversary could simply search the whole space to find the location of PUs, but this may be prohibitively expensive in practice. With this in mind, we introduce a second metric to measure PU privacy in terms of the extent to which the search space is reduced by the acquired SAS information. Formally, define the required search radius for an estimate $\hat{\mathcal{X}}$ as

$$r_s = \max_{l^p \in \mathcal{L}^p} \min_{l \in \hat{\mathcal{X}}} \|l - l^p\|. \tag{8}$$

This search radius reflects the minimum distance around each estimated location the adversary would have to search in order to intercept the actual PU locations. Let each discrete cell in the region have area α. We define the search area as

$$\sum_{l \in \mathcal{L}} \alpha \mathbf{1}(\min_{\hat{l} \in \hat{X}} \|l - \hat{l}\| \leq r_s), \tag{9}$$

where $\mathbf{1}()$ is the indicator function. This metric can be applied to a particular candidate set $\hat{\mathcal{X}}$ and can be used in (6) to compute an expected search area. Both the average distance error in (7), and the search area in (9) offer an intuitive measure of the PU privacy loss. Either metric can potentially be tied to actual PU operator requirements, depending on the specific threat(s) of concern to that operator.

That (7) operates on a location-wise estimate, i.e., whether a particular location is in $\mathcal{L}^p$, and (9) operates on an entire candidate set $\hat{\mathcal{X}}$ is a significant distinction. There is clearly more information contained in the candidate set estimate since a location-wise estimate may be computed from the candidate set estimate, but not vice versa. We can generalize this idea beyond just location, considering an element-wise estimator as one that computes the probability that a PU with a particular set of attributes is contained in $\mathcal{X}$. Computing expected error as in (6) requires operating on the candidate sets directly, but enumerating all possible candidate sets may be intractable, as we noted in III-B, making direct computation of (6) intractable as well. For practical adversary estimates, we may only be able to sample from the set $\mathbb{X}$ or else we will need to use the element-wise approach.

2) Obfuscation Strategies: Referring back to the adversary cases in III-B, PU privacy can potentially be lost by the information provided directly to the SAS, i.e., $\mathcal{X}$, or by the assignment provided by the SAS to the SUs, i.e., Ψ. We have already suggested that obfuscating these parameters could increase PU privacy, and we now formalize that idea. Let $h : \mathbb{X} \times \mathbb{R}_+^{n_s} \to \mathbb{R}_+$ be a mapping from these parameters to a chosen privacy metric. Let h_t be a required minimum level of privacy. We formally define the PU privacy problem in the SAS setting as

$$
\begin{aligned}
\max_{g_1,g_2} \quad & \mathbb{E}\{U(\boldsymbol{\Psi^1}, ..., \boldsymbol{\Psi^T}, G^s, P^{min}, P^{max})\} && \text{(10a)}\\
\text{subject to} \quad & \mathbb{E}\{h(\boldsymbol{\mathcal{X}^1}, ..., \boldsymbol{\mathcal{X}^T}, \boldsymbol{\Psi^1}, ..., \boldsymbol{\Psi^T})\} \geq h_t && \text{(10b)}\\
& \boldsymbol{\mathcal{X}^1}, ..., \boldsymbol{\mathcal{X}^T} = g_1(\tilde{\mathcal{X}}^1, ..., \tilde{\mathcal{X}}^T) && \text{(10c)}\\
& \boldsymbol{\Psi^1}, ..., \boldsymbol{\Psi^T} = g_2(\boldsymbol{\mathcal{X}^1}, ..., \boldsymbol{\mathcal{X}^T}, \mathcal{L}^s, G^s, \mathcal{P}) && \text{(10d)}\\
& Pr\left(\sum_{k=1}^{n_c}\sum_{i=1}^{n_s} \boldsymbol{\Psi}_{\boldsymbol{i}}^{t} \mathbf{G^{P}_{k,i,j}} \geq I^{th}\right) \leq \Lambda^{th} && \text{(10e)}\\
& 1 \leq j \leq n_p,\ 1 \leq t \leq T. && \text{(10f)}
\end{aligned}
$$

In words, Problem (10) maximizes the expected utility of the SUs subject to constraints on the expected PU privacy in (10b) and on the probability of harmful interference to the PUs in (10e). Since the optimization is over all possible mappings g_1 and g_2, it may not be practical to solve Problem (10). Instead, we consider how obfuscation strategies might be applied to offer a constrained approximation to Problem (10). Applicable strategies fit into one of three categories.

Obfuscation Strategy #1: Inserting false PU entries into the database. Starting with $\tilde{\mathcal{X}}$, false tuples of location and interference threshold parameters can be randomly added to produce $\mathcal{X}$. The SAS does not need to know which entries correspond to real PUs, and will protect them all from harmful interference such that even if the adversary can infer $\mathcal{X}$ exactly, there will still be some uncertainty in whether any particular tuple corresponds to a real PU. In general, we might also consider the removal of true PU tuples, but this would prevent the SAS from protecting the removed PU from harmful interference, and we limit consideration to insertion of false entries only. False entries should be inserted randomly to prevent an adversary from learning strategies over time, and should not be generated in a way that they could be excluded by an adversary based on a priori information. An adversary looking for a car, for example, may immediately exclude any false locations that do not coincide with roads. Drawing randomly from a set of false entries that are assumed to be indistinguishable to the adversary, denoted $\mathcal{X}^f$, we can adjust the privacy-utility tradeoff with the choice of n_f, the number of additional false entries. A larger n_f will provide more PU privacy, but will decrease the SU utility since the SAS will need to make assignments that protect a larger number of PUs.

Obfuscation Strategy #2: Parameter randomization. This strategy entails adding noise to the the SU power assignment P and the PU thresholds I^{th} and Λ^{th}. To ensure the PUs are protected from harmful interference, we do not randomize location, time, or frequency use entries. Also, we only allow the parameters to be reduced by the randomization to guarantee protection from harmful interference and recognize that, for any bounded randomization, uniformly distributed results will maximize adversary uncertainty.

For example, we can specifically generate the random SU assignment $\mathbf{\Psi}$ according to $\mathbf{\Psi}_i = P_i/\mathbf{Q}_i$, where P is a deterministic power assignment as in (1), and $\mathbf{Q} \in \mathbb{R}_+^{n_s}$ is a vector of random reductions with entries that satisfy $\boldsymbol{Q}_i \leq b_P$, i.e., b_P bounds the randomization. We assume a realization $\boldsymbol{\Psi_i} = \Psi_i < P_i^{min}$ would yield a power assignment of zero, equivalent to denying the SU access to the channel. Similar bounds b_I and b_Λ can also be used for randomization of the interference parameters. Increasing the randomization bounds should increase the privacy, but as with inserting false entries, will come with a cost to the SU utility.

Obfuscation Strategy #3: Dynamic PU behavior. Some sequences of $\tilde{\mathcal{X}}^1, ..., \tilde{\mathcal{X}}^T$ may offer the PU more privacy than others. If we can increase the uncertainty in the adversary's a priori information for $p(\tilde{\mathcal{X}}^{t+1}|\tilde{\mathcal{X}}^t)$, we expect that the PU privacy may also increase. For example, a PU might employ frequency hopping such that an adversary cannot readily infer current channel use from observations of past use, or mobility strategies to obfuscate location. We can view $\tilde{\mathcal{X}}$ as an optimization variable instead of as a fixed parameter in (10), where an additional constraint would be necessary to capture any practical constraints on the PU dynamics.

We can apply these obfuscation strategies as specific forms for g_1 and g_2 in Problem (10), producing an approximate problem where we optimize over the relevant parameters, e.g., n_f and b_P, rather than over all possible random mappings. Before we can consider solving this constrained problem, we first need to understand how we might map a selection of obfuscation parameters to a privacy measure, i.e., we require a specific form for $h()$ in (10).

3) Privacy Duration: Given a large number of observations to estimate a static parameter, eventually the adversary may reduce a selected privacy metric to an arbitrarily low level. A PU working with the SAS will need to determine whether a threshold level of privacy can be maintained for some minimum duration, e.g., as long as a particular operating parameter remains static. Let τ be a random variable for the number of time slots before the privacy is reduced below a threshold level, where adversaries get new observations at each time slot. We will seek here to measure $E\{\tau\}$, i.e., the expected duration that privacy can be maintained. For a known adversary estimation scheme and PU obfuscation strategies, $E\{\tau\}$ can be estimated through Monte Carlo simulation. However, in treatment of Problem (10), running Monte Carlo simulation for each potential obfuscation strategy may be impractical, particularly if the PU is attempting to solve the problem in an on-line setting where time and computational resources may be limited. Instead, we develop a lower bound on the privacy duration based on the optimal adversary estimator.

Consider adversary case #2, where the adversary only observes Ψ^t at each time slot. With each observation, the optimal adversary estimator in (4) may be able to conclude that a particular candidate PU is not an element of $\tilde{\mathcal{X}}$, e.g., if the candidate interference constraint would be violated by an observed power allocation. In this way we consider the cardinality of the candidate PU space as a privacy metric, where an optimal estimate will always include the elements of $\tilde{\mathcal{X}}$ with positive probability. In an offline mode, the PU operator will consider possible PU and SU topologies when selecting obfuscation parameters to ensure the adversary cannot reduce the cardinality of the candidate space down to some unacceptable level. Treat $\boldsymbol{\mathcal{L}}^s$, $\boldsymbol{G}^s$ and $\boldsymbol{\mathcal{X}}$ as random variables with arbitrary distributions, and assume a perfectly symmetric region, i.e., a torus. Consider the set of PU tuples that are not in the database, i.e., $\{(l,i,\lambda) \in \mathbb{X}; (l,i,\lambda) \notin \mathcal{X}\}$, and arbitrarily index this set from 1 to v. An adversary observing the random power assignments will have some probability of determining that the kth tuple is not a PU entry in the database. Denote this probability of eliminating the kth tuple as p_k. If we consider multiple trials with i.i.d. $\boldsymbol{\mathcal{L}}^s$, $\boldsymbol{G}^s$ and $\boldsymbol{\mathcal{X}}$, then eliminating any particular entry at a particular trial can be treated as a geometric random variable with parameter p_k and cumulative distribution function $F_k(t) = 1-(1-p_k)^t$. With the approximation that eliminating entries is independent, the probability that at least w entries are eliminated after t observations is

$$s_w(t) = \sum_{r=w}^{v} \sum_{\pi \in \Pi_r} \prod_{i \in \pi} F_i(t) \prod_{j \in \bar{\pi}} 1 - F_j(t), \tag{11}$$

where Π_r contains all subsets of entries with cardinality r, i.e., $\Pi_r = \{\pi \subseteq \{1,...,v\}; |\pi| = r\}$. Then the expected time to eliminate any w of the v entries follows as

$$\mathbb{E}\{\boldsymbol{\tau_w}\} = \sum_{t=1}^{\infty} t[s_w(t) - s_w(t-1)], \tag{12}$$

where $s_w(t) - s_w(t-1)$ is simply the probability that w locations are eliminated with observation t. In the special case of location privacy, we can make an additional approximation that p_k is the same for each location due to the symmetry of the region. In this case, we can expand $F_k(t)$ and express the expected privacy duration as

$$\begin{aligned}\mathbb{E}\{\boldsymbol{\tau_w}\} = \sum_{t=1}^{\infty} t \sum_{r=w}^{v} \binom{v}{w} (1-p_k)^{(t-1)(v-r)} \\ \left[(1-p_k)^{v-r}(1-(1-p_k)^t)^r - (1-(1-p_k)^{t-1})^r\right].\end{aligned} \tag{13}$$

The term in brackets can be expanded and written $\sum_{i=0}^{r} \binom{r}{i} (1-p_k)^{i(t-1)}(-1)^i[(1-p)^{i+v-r} - 1]$. Then rearranging the summation and using the identity $\sum_{t=1}^{\infty} tr^{t-1} = (1-r)^{-2}$ for $|r| < 1$, the expected privacy duration is

$$E\{\boldsymbol{\tau_w}\} = \begin{cases} \sum_{r=w}^{v} \binom{v}{w} \sum_{i=0}^{r} \frac{\binom{r}{i}(-1)^{i+1}}{1-(1-p_k)^{i+v-r}} & \text{for } w < v \\ \sum_{i=1}^{w} \frac{\binom{w}{i}(-1)^{i+1}}{1-(1-p_k)^i} & \text{for } w = v. \end{cases} \tag{14}$$

To compute (12), we must first compute p_k, which, in turn requires the computation of $p_k|\mathcal{X}, \mathcal{L}^s, G^s$, since $p_k = \sum p_k|\mathcal{X}, \mathcal{L}^s, G^s \cdot p_{\mathcal{X},\mathcal{L}^s,G^s}$, that is, we simply treat p_k as the marginal for the probability conditioned on the realizations of $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{L}}^s$, and $\boldsymbol{G}^s$. For a single realization, if (l,i,λ) is the kth tuple, we can eliminate the kth tuple from consideration if the power assignment provided to the ith SU, Ψ_i, is greater than the maximum or less than the minimum that could result if the candidate were actually included in $\mathcal{X}$. Specifically, if $\hat{\mathcal{X}}$ is a candidate PU set with $(l,i,\lambda) \in \hat{\mathcal{X}}$, and $\hat{P}_i$ and $\check{P}_i$ are the maximum and minimum power assignments for the ith SU under $\hat{\mathcal{X}}$ respectively, then for the kth tuple to be eliminated we need $\Psi_i > \max_{\{\hat{\mathcal{X}}:(l,i,\lambda)\in\hat{\mathcal{X}}\}} \hat{P}_i$ or $\Psi_i < \min_{\{\hat{\mathcal{X}}:(l,i,\lambda)\in\hat{\mathcal{X}}\}} \check{P}_i$.

Let $B = [B_i] \in \mathbb{Z}_+^{n_s}$ be the vector whose elements are the number of realizations of Ψ_i that will allow the adversary to eliminate the kth tuple. Computing p_k is then straightforward, given the uniform distribution assumed for Ψ_i. However, computing each B_i may be complicated depending on the power assignment algorithm. With the selected equal interference power allocation algorithm (see (2)), B_i depends on the specific topology, i.e., $\mathcal{X}$, $\mathcal{L}^s$, and G^s since this determines the clustering of the SUs with each PU and the resulting power assignments. For the chosen obfuscation and SAS assignment strategies, we can write the condition $\Psi_i > \max_{\{\hat{\mathcal{X}}:(l,i,\lambda)\in\hat{\mathcal{X}}\}} \hat{P}_i$ as

$$\frac{I(A_i, I^{th}, \Lambda^{th})}{Q_i \max_j \bar{G}_{ij}^p} > \max_{\{\hat{\mathcal{X}}:(l,i,\lambda)\in\hat{\mathcal{X}}\}} \frac{I(\hat{A}_i, \hat{I}^{th}, \hat{\Lambda}^{th})}{\hat{G}_i^p}, \tag{15}$$

where Q_i is the realization of the random power assignment reduction described in Section III-C2 and $\hat{G}_i^p$ is the estimated gain between the ith SU and the nearest entry in the candidate $\hat{X}$. Let $D = [D_i] \in \mathbb{R}_+^{n_s}$ denote a vector whose elements are the euclidean distances between the SUs and the nearest PUs. Then, assuming that the mean channel gain is inversely proportional to the distance raised to a path loss exponent n, we can rewrite the condition as

$$\left(Q_i \max_{\{\hat{\mathcal{X}}:(l,i,\lambda)\in\hat{\mathcal{X}}\}} \gamma\right)^{1/n} \hat{D}_i < D_i, \tag{16}$$

where γ is defined as the ratio of the equal interference levels returned by $I()$ as in (2) for $\hat{\mathcal{X}}$ and actual PU locations $\mathcal{X}$, i.e.,

$$\gamma = \frac{I(\hat{A}_i, \hat{I}^{th}, \hat{\Lambda}^{th})}{I(A_i, I^{th}, \Lambda^{th})}. \tag{17}$$

Determining the probability of eliminating the kth tuple is complicated in the SAS setting because it requires eliminating all candidate PU configurations for which $(l,i,\lambda) \in \hat{\mathcal{X}}$, which may be a very large set. However, we can lower bound γ over all candidates such that we lower bound the PU privacy estimate. Specifically, since the entries in $\mathcal{X}$ will always be viable candidates for the adversary, eliminating each $\hat{\mathcal{X}}$ will require, at a minimum, eliminating all candidates that are subsets from $\{(l,i,\lambda)\} \cup \mathcal{X}$. With the assumption that the power allocation algorithm will group individual SUs with the nearest PU, the adversary can only hope to eliminate a candidate with location l based on assignments to SUs that are closer to l than any location in $\mathcal{X}$. Let $\mathcal{V}_k$ denote the set of SUs that satisfy this condition. If we set $\hat{A}_i = |\mathcal{V}_k|$ in (17), we can compute a bounding γ for (16) and can count the number of realizations Q_i that satisfy (16). We can analgously derive a bound for the condition on the minimum, $\check{P}_i$, but omit that derivation for brevity. These conditions offer a computationally efficient method to bound B_i and thus bound p_k.

By estimating p_k, we have a method, with (12), to compute a lower bound for the expected time until an adversary can eliminate w PU candidates from consideration. This is a useful metric for the PU optimization Problem (10). Treating $\boldsymbol{\mathcal{X}}$ as i.i.d. is applicable to the case of a PU interested in the suitability of a SAS to protect its typical operations. Alternatively, a fixed $\mathcal{X}$ corresponds to a specific PU operation in an online mode. Similarly, we can model $\mathcal{L}^s$ and G^s as initially fixed in an online mode, or with arbitrary distributions to model known characteristics of SU deployments in a particular region. If we wish to model known SU behavior, e.g., a specific adversary SU control strategy in adversary case #3, rather than i.i.d. sampling, we can draw correlated sequences of SU parameters according to the behavioral model. In these cases, the assumed symmetry of the region may no longer hold, and the expected duration will need to be computed from (12) instead of (14), accounting for location-dependent p_k. The derived bound (16) is still applicable in computing each p_k, and allows for relatively efficient computation.

In any scenario, we can numerically estimate p_k as described, and also numerically estimate the expected sum-utility. Recognizing that the utility is non-increasing for the obfuscation parameters n_f, b_P, b_I, and b_Λ, and that the privacy is non-decreasing, we can efficiently solve Problem (10) with these estimates by searching on a multi-dimensional sorted array corresponding to the obfuscation strategies.

IV. Adversary Estimation Schemes

In the prior section, we described the optimal adversary estimate, where (5) can be applied in theory. We will consider a worst-case scenario where the adversary knows the SAS assignment strategy, including obfuscation operators, g_1 and g_2, as well as the PU state dynamics, $p_{\mathcal{X}^{t+1}|\mathcal{X}^t}$, exactly. In practical scenarios, computing (5) is still potentially intractable given the very high dimensionality. For use in validating our privacy framework in practical scenarios, we now describe several approximate approaches for adversary estimation.

Sequential Monte Carlo methods, often referred to in the literature as particle filters, take a genetic approach to approximating distributions with established convergence results [23], [24]. They have been shown to offer good performance in a wide variety of settings [25], [26]. Specifically, we implement a variation on a Sequential Importance Sampling (SIS) particle filter for the adversary as provided in Algorithm 1. The basic approach is to first pick an initial set of particles $\{\hat{\mathcal{X}}_i^0 \in \mathbb{X}; i \in \{1, ..., m\}\}$, where each particle corresponds to a candidate set. The selection of particles for subsequent time steps is based on a proposal function. We take the common approach used in the literature where the proposal function is based on the state dynamics of the PUs. At the first time step, particles are selected from the adversary's a priori distribution for the PU locations. Importance weights, $q(t)$ are then computed for each particle and normalized to $\tilde{q}(t)$. The importance weights are usually computed according to $P(\mathcal{Y}|\hat{\mathcal{X}})$, where $\mathcal{Y}$ is the adversary observation. In our case, we include a penalty parameter in the weight calculation to allow consideration of particles that are infeasible, but are still potentially close to the true candidates. This helps to compensate for the sensitivity of the assignment strategy to relatively small changes in the input. In step 8, we estimate the effective number of particles in our sample, n_{eff}, where small n_{eff} indicates that some particles are significantly more likely to correspond to a correct guess than others. In this case, a new set of particles is created by sampling with replacement from the current set of particles according to the importance weights. In this way, ”good” particles are likely to be included in the new set while particles with low importance weight will tend not to be carried over into the new set. In this resampling step, we also deviate from the traditional particle filter by introducing a number of ”fresh” particles generated from the a priori distribution along with the resampled particles. This helps to combat the high dimensionality of the space without tracking an excessive number of particles at each iteration. The particle filter then proceeds iteratively for each observation, picking particles from the proposal function and the prior particle set. The resulting sequence of particles and importance weights serves as the adversary estimate. As the number of tracked particles grows large, this particle filter sample will converge to the optimal adversary estimate.

Algorithm 1 Adversary SIS Particle Filter

```
1:  Given: $\mathcal{Y}^1, ..., \mathcal{Y}^T, m, p_{\hat{\mathcal{X}}}, p_{\mathcal{Y}|\hat{\mathcal{X}}}, p_{\hat{\mathcal{X}}^{t+1}|\hat{\mathcal{X}}^t}, n_{th}$
2:  $t = 0$, draw $\hat{\mathcal{X}}_1^0, ..., \hat{\mathcal{X}}_m^0$ from $p_{\hat{\mathcal{X}}}$
3:  Set importance weights $\tilde{q}_i(-1) = 1/m$, $i \in \{1, ..., m\}$
4:  for $i = 1, ..., m$ do
5:      $q_i(t) = p_{\mathcal{Y}|\hat{\mathcal{X}}}(\mathcal{Y}^t|\hat{\mathcal{X}}_i^t)\tilde{q}_i(t-1)$
6:      $\tilde{q}_i(t) = q_i(t)/\sum_{j=1}^m q_j(t)$
7:  end for
8:  $n_{eff} = 1/\sum_{i=1}^m (\tilde{q}_i(t))^2$
9:  if $n_{eff} < n_{th}$ then
10:     Resample $\hat{\mathcal{X}}_1^t, ..., \hat{\mathcal{X}}_m^t$ with replacement from $\tilde{q}(t)$
11: end if
12: for $i = 1, ..., m$ do
13:     Draw $\hat{\mathcal{X}}_i^{t+1}$ from $p_{\hat{\mathcal{X}}^{t+1}|\hat{\mathcal{X}}^t}(\hat{\boldsymbol{\mathcal{X}}}_i^{t+1}|\hat{\mathcal{X}}^t)$
14: end for
15: if $t = T$ then
16:     return  $\hat{\mathcal{X}}_i^t$, $\tilde{q}_i(t)$ $\forall$ $i \in \{1, ..., m\}$, $t \in \{1, ..., T\}$
17: else
18:     $t = t + 1$, go to step 4
19: end if
```

In addition to the particle filter, we include several heuristics. At each observation, the adversary can exclude (l, i, λ) from further consideration if the harmful interference constraint is violated. In addition, if (l, i, λ) is excluded, then all $\{(l, i^{'}, \lambda^{'}); i^{'} \leq i, \lambda^{'} \leq \lambda\}$ can also be excluded. The first and simplest approach then is to start with a set of all possible tuples in $\mathbb{X}$ and prune those that are found to receive harmful interference at one or more time slots. If enough locations are eliminated in this way, it may then be feasible to compute (5) directly. We will refer to this approach of pruning in the first phase and then computing the optimal estimates directly in a second phase as the *composite method*.

For candidates that have not been eliminated, we argue that those candidates which have come closest to receiving harmful interference are more likely to be the source of limitation to the assigned SU transmit powers, i.e., they are more likely to correspond to actual PUs. With this in mind we calculate the harmful interference constraint margin for each candidate and use the inverse as a weight, where the normalized weights can be used as an approximated adversary estimate. We'll refer to this approach as the *interference constraint margin (ICM)*.

We also recognize that a power assignment resulting from a set of multiple PUs should have some similarity to an assignment that was based on a single PU within that set due to the clustering in the power assignment per (2). We therefore consider an approach where we compute the maximum power assignment $\hat{P}$ that would result for each candidate as if it were the only PU protected. By comparing the squared error between $\hat{P}$ and the observed power assignment Ψ, we have a measure of the similarity and consider the normalized inverses as another basis for an adversary estimate. We'll refer to this approach as *single PU error (SPE)*. More detail and performance assessments are provided for these approaches with the context of a specific case study in Section V.

With our heuristics and optimal estimates, we will also apply a thresholding approach to identify an adversary guess for the actual PU candidates. This corresponds to scenarios where the adversary must take an action based on the estimates. We specifically apply a window to each estimate and identify local maxima. The largest local maxima provide the adversary guess, from which we will directly compute error metrics between the guess and the actual PU entries as a measure of adversary performance and PU privacy.

V. Location Privacy Case Study

In this section we conduct a case study via simulation and validate the proposed PU privacy framework. We will limit our study to two-dimensional location privacy in a single SAS channel to simplify the simulation and corresponding discussion. In doing so, we assume I^{th} and Λ^{th} are included in the adversary a priori information.

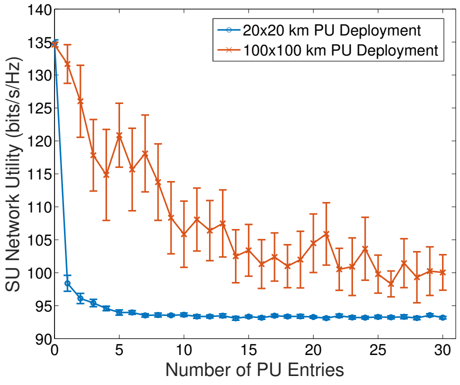


Fig. 3. Sample average SU sum-rate utility versus number of PU locations (error bars denote standard deviation).

We model PUs as military radars and SUs as cellular networks, consistent with the parameters and assumptions in CBRS [27]. We assume a 10 MHz channel with center frequency 3645 MHz. We deploy SUs in a 20 km by 20 km region, laying down a grid of SU BSs with a 2.5 km inter-site spacing, and randomly placing UEs then connecting them with the nearest BS. We assume UE uplink transmissions in the SU network, with transmission powers in the range $P^{max} = 23$ dBm and $P^{min} = -40$ dBm. BSs run a proportional fair scheduler with parameters $P_0 = -90$ dBm and path loss correction coefficient 0.8. The SAS grants SU assignments for 30 second time slots. PU locations are protected with $I^{th} = -114$ dBm and $\Lambda^{th} = 0.5\%$. All systems are assumed to have omnidirectional antenna gains for simplicity. For propagation between transmitters and receivers, we use a two-ray model ($n = 4$ after a breakpoint) with PU heights of 15 m and UE heights of 2 m. For the channel gain on the UE to PU paths, $\boldsymbol{G}^p$, we use a log-normal distribution with standard deviation $\sigma = 10$ dB. Finally, for the adversary's a priori information, we assume all candidate solutions are equally likely, resulting in a 50% probability of a PU in any specific location. In V-A, we first evaluate scenarios where the PU locations remain fixed for the duration of the simulation, and then consider mobile PUs in V-B.

A. Stationary PU Location Privacy

We assess a PU system that dwells in one location for many time slots, studying privacy for the three adversary cases.

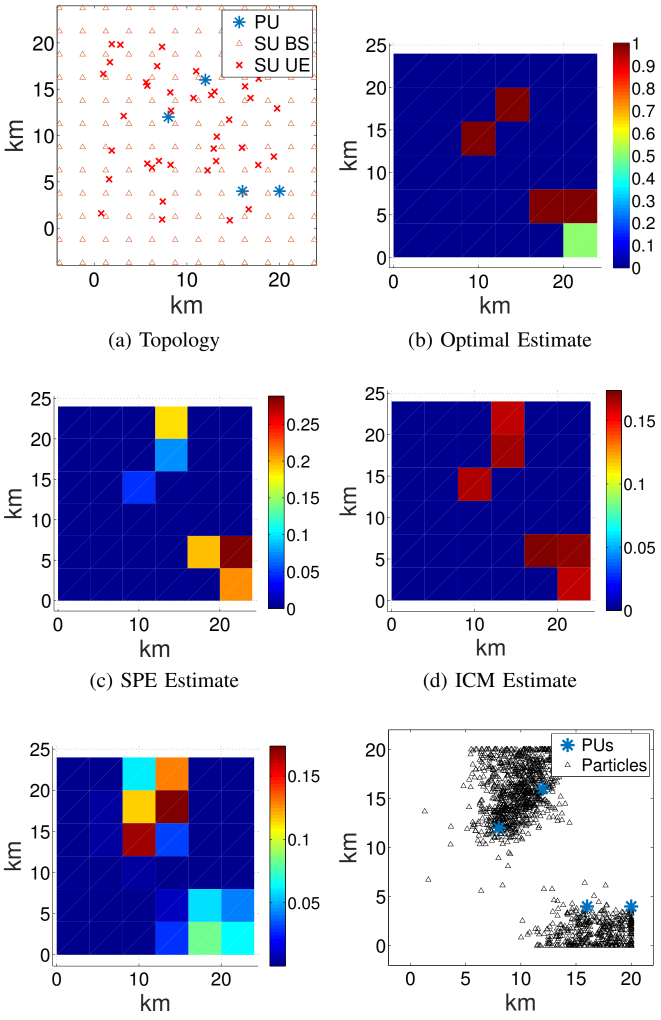
Adversary Case #1: We first consider the PU privacy level for the case where the adversary observes $\mathcal{X}$ directly. We consider two deployments for the PUs. In the first, the PUs are uniformly placed in the same 20 km by 20 km region as the SUs. In the second, the PUs are uniformly placed in a 100 km by 100 km region and the SUs are uniformly placed in a 20 km by 20 km corner of the region. The latter deployment is obviously more favorable for SU utility, and it is also more representative of real-world sharing scenarios. For example, in 3.5 GHz, we would expect to find large concentrations of SUs in population centers, whereas many military operations would be more likely to take place in somewhat remote areas. In this case, the PU privacy corresponds to the number of false entires in $\mathcal{X}$. For an SU network of 1,000 UEs, Figure 3 plots the sample average SU sum-rate versus n_p, averaged over 40 randomly generated topologies. We see from this figure a direct way to trade SU utility for PU privacy, though the SU utility is, not surprisingly, much more limited when the PUs and SUs are concentrated in the same region.

Adversary Case #2: We now consider an adversary that does not have direct access to $\mathcal{X}$, but does have access to all SU assignments. We use a random direction mobility model for movement of 40 SU devices with a constant speed of 30 km per hour. The adversary divides the region into 4 km by 4 km cells, resulting in a small enough number of candidate solutions that the optimal estimate may be computed directly from (4). There are four PU locations ($n_p = 4$) and uniform randomization is bounded with $b_P = 20$ dB. Figure 4 provides a visualization of the adversary estimate in this small scale scenario with the given topology and based on observation of a single time slot. The optimal estimate and the heuristics introduced in Section IV, namely the particle filter (PF), SPE, and ICM, are able to identify the actual PU locations to varying degrees. Figure 4b shows the optimal estimate after just a single observed time slot. The adversary is able to immediately pinpoint all four locations, with some uncertainty about the possibility of a fifth location. Figure 4c shows the SPE estimate after 20 minutes of observation. SPE assigns significant weight to two locations that do not contain a PU and almost fails to assign any weight to the westernmost PU. ICM in Figure 4d assigns similar weights to the actual PU locations as it assigns to two other locations, also after 20 minutes of observation. Figure 4e shows the estimate resulting from the particle filter and Figure 4f illustrates the surviving particles tracked by the particle filter. The actual PU locations are tracked by the particle filter, but exact locations are not precisely identified.

Figure 5 plots the progression of the adversary estimates over time with respect to our two metrics. In Figure 5a, average distance error is plotted over time for the four estimators. The optimal estimator is able to reduce the average distance error to about 200 m after a single observation, and then reduces it to zero, identifying the PU locations exactly, after a few minutes of observations. The other three estimators are able to steadily reduce the error during the first 10 minutes of observation, reaching an error of about 5 km, until they are able to pinpoint the PU locations exactly from an observation about 50 minutes into the simulation.

We plot the search area error from (9) versus time for the estimators in Figure 5b in terms of the remaining search area as a percentage of the total region area. For SPE and ICM, since these are location specific estimators and search area applies to a set of candidate locations, we apply windowed local maxima thresholding to these estimators to compute an adversary guess of the most likely PU locations. We plot these thresholded versions as SPE-W and ICM-W in Figure 5b and also include unthresholded ICM/SPE as a reference, i.e., where all locations in ICM or SPE with nonzero weight are included in the guess. The relative performance of the estimators is similar to that in terms of average distance error. The optimal estimator is able to rapidly reduce the search area to the actual PU entries, which corresponds to a little over 10% of the region area. The other estimators reduce the search area to about 17% of the region area until they are finally able to identify the PU locations exactly.

We also consider a scenario with an adversary resolution of 1 km by 1 km cells and $n_p = 2$. All other parameters are the same as with the prior lower resolution scenario. Figure 6a plots the topology for this scenario. With the finer adversary resolution, direct computation of the optimal estimator is impractical and we instead provide results for the composite approach of eliminating locations until few enough remain to use direct computation. In this case we set the threshold to 40 locations or fewer. Since the PUs are static, the adversary can take multiple observations to conduct the pruning phase and progressively reduce the candidate set down to a manageable size for computation of the optimal estimate. Figures 6b, 6c and 6d provide plots visualizing the estimates at a single time slot. ICM and the particle filter identify the area around the PU locations, though SPE estimates the eastern PU to be farther south than it is, and almost misses the western PU entirely. The composite estimate plot is omitted for brevity.



(a) Topology

(b) Optimal Estimate

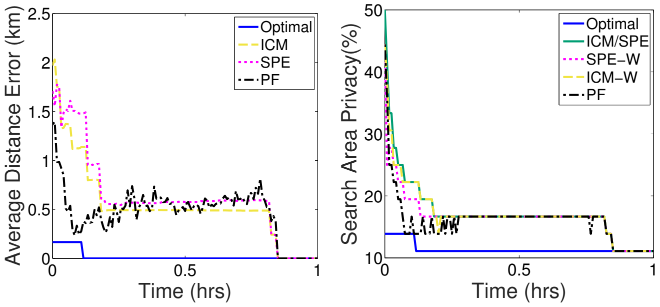
(c) SPE Estimate

(d) ICM Estimate

(e) PF Estimate

(f) Tracked Particles

Fig. 4. Topology and Estimates for Fixed PU - Low Resolution Adversary.

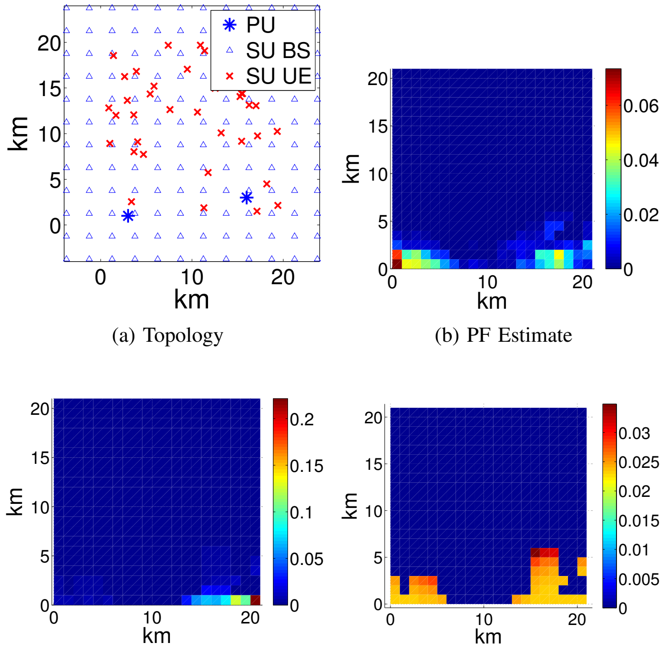


(a) Average Distance Error

(b) Search Area

Fig. 5. Adversary Estimates Over Time - Low Resolution.

Figure 7 plots the performance of the estimators over time. Once the number of candidate locations in the composite approach are reduced to 40, the direct computation of the optimal estimator identifies the exact PU locations quickly.



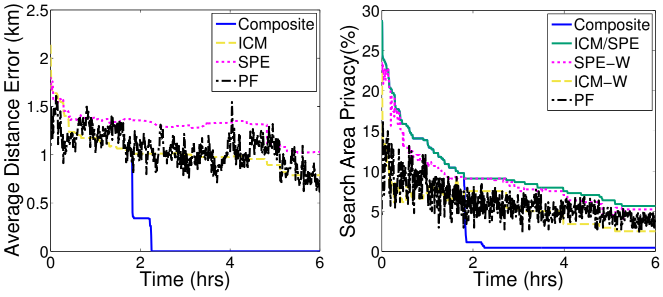
(a) Topology

(b) PF Estimate

(c) SPE Estimate

(d) ICM Estimate

Fig. 6. Topology and Estimates for Fixed PU - Higher Resolution Adversary.



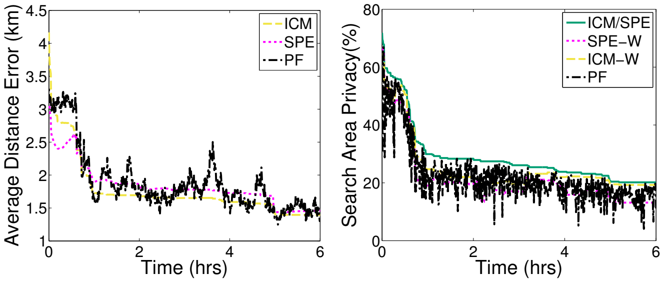
(a) Average Distance Error

(b) Search Area

Fig. 7. Adversary Estimates Over Time - Higher Resolution.

The other estimators make significant progress in the first minutes of observations, and then only improve gradually. These estimators are unable to pinpoint the PU locations as they did in the low resolution case, despite a much longer simulation time (6 hours). The higher resolution in this case does allow the adversary to achieve a reduced search area, but the greater granularity also reveals a larger average distance error owing to uncertainty in the adversary estimate that was obscured in the low resolution scenario.

Adversary Case #3: In Figure 8 we provide the progression for the adversary estimate with the topology and higher resolution scenario considered in the prior example, but here assume the adversary only has access to five of the 100 SUs. We are prevented from running the optimal or composite estimator in this case as the additional uncertainty of the unknown SU parameters makes the candidate space much too large. However, with the particle filter, ICM and SPE, we observe a substantial degradation in the quality of the adversary estimate



(a) Average Distance Error

(b) Search Area

Fig. 8. Adversary Estimates Over Time - Adversary Case #3.

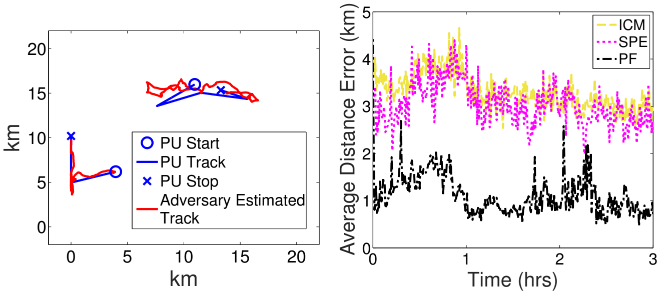
(note the scale of the y-axis). Clearly, limiting adversary access to SU information produces a harder adversary problem, stressing the importance of security measures for the access system and connected devices.

The prior plots illustrate the progression of the adversary estimate over time and the corresponding loss of PU privacy. A minimum level of privacy can be maintained by providing false PU entries to the SAS, where this privacy is maintained even against a powerful adversary that can hack into the SAS directly, or for an adversary that can observe SU assignments indefinitely. The assignment process of the SAS and the addition of randomization also offers some privacy, but we can see that this effect is only temporary, even against suboptimal estimators, and will not benefit a PU that remains static indefinitely. However, we will find that quantifying the progression of an adversary against a static PU will be a useful tool in evaluating the privacy of a dynamic PU.

B. Mobile PU Location Privacy

Movement of the PUs introduces some additional potentially sensitive PU aspects. Recording a history of the PU movement path may allow an adversary to infer identity based on locations visited. The trajectory of the movement may also allow an adversary to infer the intended destination of the PUs. The trajectory history, or track, introduces the opportunity for new metrics, e.g., how to measure the distance between an estimated track and a ground truth track. The subject of multiple target tracking has been well studied in the literature [28] and we will not go into great detail on the subject here. We will illustrate issues of PU mobility for PU privacy in the SAS setting with a few examples.

First, consider a scenario where $n_p = 2$, $b_P = 10$, and the PUs move at 10 kph with a random waypoint model. Figure 9a plots the PU movement tracks along with the estimated tracks of an adversary using a particle filter to produce estimates. We can see qualitatively that the adversary is able to well approximate the PU tracks. In Figure 9b, we plot the average distance error over time and see that the particle filter is able to track the PU typically to about 1 km accuracy. We omit a plot of the search area error for brevity. While the mobility of the PUs prevents the adversary from steadily reducing privacy as in the fixed PU case, this does not prevent the adversary from achieving a good estimate.



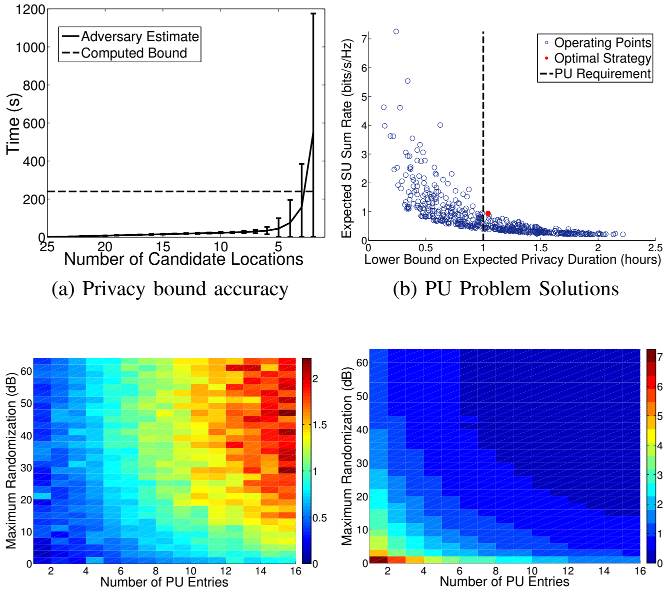
(a) Track History

(b) Average Distance Error

Fig. 9. Adversary Estimates of Mobile PUs (10 kph)

Identifying an effective obfuscation strategy for this case corresponds to an instance of the PU privacy problem. Consider our derived privacy duration bound as computed in (14). In Figure 10a, for $n_p = 2$ and $b_P = 10$, we plot the expected time for a simulated optimal adversary estimator to reduce the number of candidate locations down to a specified number, averaged over 20 topologies and where the error bars correspond to the sample standard deviation. With the horizontal line at 240 seconds, we plot the computed lower bound on expected time for the optimal estimator to identify the two PU locations exactly. We see that the bound is a reasonably close approximation to the simulated optimal estimator. The large standard deviation indicates that pinpointing the exact locations is heavily dependent on the topology. Suppose the PU requires a guarantee that the adversary will not be able to pinpoint their locations before at least an hour of observation on average. We approximately solve the PU problem (10) by brute force for a PU in this scenario. In Figure 10b, we plot all feasible solutions to the problem, illustrating the privacy-utility tradeoff, and highlighting the optimal solution. We also plot the expected privacy duration and SU utility in Figures 10c and 10d as a function of the obfuscation strategy parameters. For this scenario, we see that a little randomization will increase privacy without too dramatic of an impact on SU utility, but more than about 20 dB of randomization offers little privacy benefit. Increasing n_p on the other hand steadily increases privacy as long as b_P is about 10 dB or greater. Note that the figures are somewhat noisy, i.e., the monotonic relationship between the obfuscation parameters and the achieved privacy does not appear to hold exactly. This is an artifact resulting from variations in the sampling of random topologies which can be mitigated by increasing the number of samples.

Instead of adjusting the obfuscation strategy, we might consider whether faster moving PUs might maintain higher levels of privacy. Again let $n_p = 2$ and $b_P = 10$, but this time the PUs will move at 100 kph. Figure 11 plots the PU tracks, the estimated tracks, and the performance of the estimates with respect to average distance error. We see that the faster PU mobility clearly increases the PU privacy even though the obfuscation strategy was unchanged.



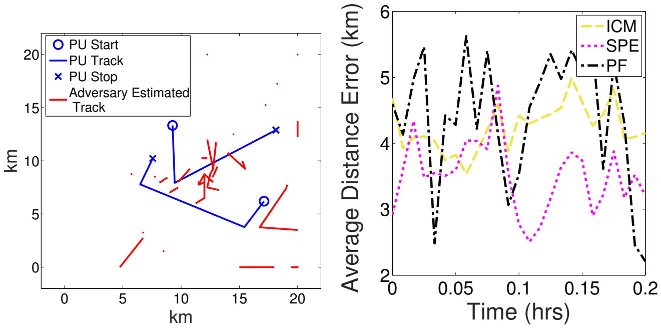
(a) Privacy bound accuracy

(b) PU Problem Solutions

(c) Expected Privacy Duration (hours)

(d) Expected SU Utility (b/s/Hz)

Fig. 10. Application of the PU privacy bound.



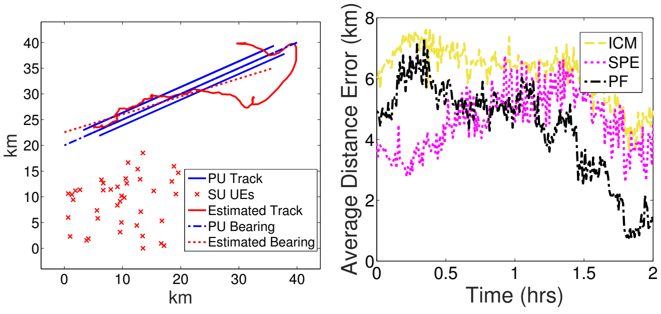
(a) Track History

(b) Average Distance Error

Fig. 11. Adversary Estimates of Mobile PUs (100 kph)

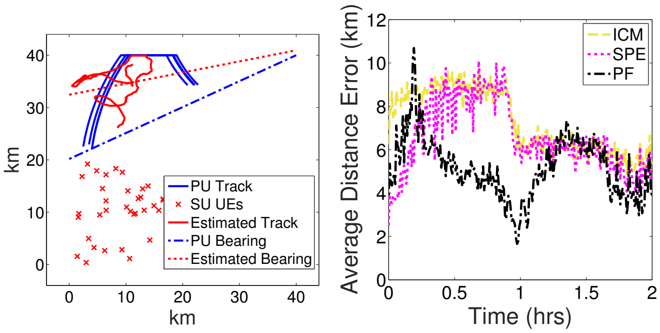
PU privacy strategies may need to account for their own mobility limitations. Specifically, false location strategies should account for potential adversary a priori information on the PU movement model. The trajectory of any false locations should be consistent with the model so that they cannot be quickly eliminated from consideration by the adversary.

Consider a scenario of a ship leaving port, where the operator of the ship wants to prevent an adversary from inferring the precise heading of the ship based on observations from the SAS. False locations cannot be created arbitrarily as the adversary could readily exclude those that are clearly not associated with a ship leaving port. However, the PU can potentially increase its privacy by inserting false locations in a random region around the current PU location and then propagate these locations with the actual PU movement. This results in a formation of false locations following the movement of the actual PU. Let the SU network be deployed in a 20 km by 20 km region, and let the ship leave port in the northwest corner of this region, heading in a northeasterly direction at 20 kph. Figure 12 plots the tracks and adversary performance when the PU adds three additional PU entries



(a) Topology and Tracks (b) Average Distance Error

Fig. 12. Formation Movement of PUs



(a) Topology and Tracks (b) Average Distance Error

Fig. 13. Zig-zag Movement of PUs

moving in formation with the ship. Figure 12a also includes the adversary estimate of the PU bearing, based on a least-squares fit to the estimated track history, where we see the adversary is able to accurately estimate the ship's course.

While adding false entries should have increased the PU location privacy in the prior example, it did not prevent the adversary from accurately inferring the bearing of the PU. To combat this, suppose in addition to the obfuscation strategies, we adjust the PU dynamic behavior to increase privacy. Suppose the PU will now randomly zig-zag as it moves towards its destination. The PU will move with a fixed speed of 20 km but will randomly select a direction within 90 degrees of the direction towards the intended destination. We plot the tracks and estimates for this case in Figure 13. The adversary's estimated bearing is far from the actual direction of the PU destination. While the estimator is certainly not optimal, this illustrates the potential for PU dynamic behavior to play an important role in the feasibility of maintaining PU privacy while operating with a SAS.

VI. Conclusions

Dynamic spectrum access systems offer potential gains in efficient spectrum use but also pose risks for user privacy. We have examined strategies for PU privacy, observing that the potential dependency of an adversary's observation on the entire user topology potentially presents a harder problem than has been considered in other location privacy settings. Still, loss of PU privacy is not a matter of if, but of when, such that an effective obfuscation strategy is one that ensures the convergence of an adversary's estimate is slower than the dynamics of the PU.

We presented a general privacy framework for assessing the privacy of a PU in the SAS setting, posing a PU privacy problem to select an obfuscation strategy that maximizes the utility of the spectrum while achieving a PU privacy requirement. With a derived lower bound on privacy duration, we showed that a constrained version of the PU privacy problem could be solved efficiently, and verified our results by simulation.

Our constrained solution to the PU privacy problem is based on specific obfuscation strategies. More general strategies might offer improved privacy or SU utility if efficient solutions can be found. Further, while we have focused here on the privacy aspects of a database-based SAS, sharing systems may also include a sensing component. How sensing and databases together affect PU privacy and obfuscation strategies remains for future work.

References

[1] *Report and Order and Second Further Notice of Proposed Rulemaking*, Number 15-47, GN Docket No. 12-354. Federal Communications Commission, Apr. 2015.

[2] *Order on Reconsideration and Second Report and Order*, 16-55, GN Docket No. 12-354. Federal Communications Commission, May 2016.

[3] S. Deb, V. Srinivasan, and R. Maheshwari, "Dynamic spectrum access in DTV whitespaces: Design rules, architecture and algorithms," in *Proc. 15th Annu. Int. Conference on Mobile Computing and Networking*, New York, NY, USA, 2009, MobiCom '09, pp. 1–12, ACM.

[4] J. M. Park, J. H. Reed, A. A. Beex, T. C. Clancy, V. Kumar, and B. Bahrak, "Security and enforcement in spectrum sharing," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 270–281, March 2014.

[5] B. Bahrak, S. Bhattarai, A. Ullah, J. M. J. Park, J. Reed, and D. Gurney, "Protecting the primary users' operational privacy in spectrum sharing," in *2014 IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*, April 2014, pp. 236–247.

[6] A. Robertson, J. Molnar, and J. Boksiner, "Spectrum database poisoning for operational security in policy-based spectrum operations," in *2013 IEEE Military Communications Conference*, Nov 2013, pp. 382–387.

[7] C. Dwork, "Differential privacy," in *Proc. Int. Colloq. Automata, Languages and Programming*, 2006, pp. 1–12.

[8] C. Dwork and A. Smith, "Differential privacy for statistics: What we know and what we want to learn," *J. Privacy and Confidentiality*, vol. 1, no. 2, pp. 135–154, 2009.

[9] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, *Our Data, Ourselves: Privacy Via Distributed Noise Generation*, pp. 486–503, Springer, Berlin, Heidelberg, 2006.

[10] S. Kasiviswanathan and A. Smith, "On the 'Semantics' of Differential Privacy: A Bayesian Formulation," *Journal of Privacy and Confidentiality*, vol. 6, no. 1, Aug. 2014.

[11] S. Li, H. Zhu, Z. Gao, X. Guan, Kai Xing, and X. Shen, "Location privacy preservation in collaborative spectrum sensing," in *2012 Proceedings IEEE INFOCOM*, March 2012, pp. 729–737.

[12] 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*, vol. 19, no. 6, pp. 106–112, December 2012.

[13] W. Wang and Q. Zhang, "Privacy-preserving collaborative spectrum sensing with multiple service providers," *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 1011–1019, Feb 2015.

[14] M. Grissa, A. Yavuz, and B. Hamdaoui, "LPOS: Location privacy for optimal sensing in cognitive radio networks," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–6.

[15] Y. Dou et al., "p^2-SAS: Preserving users privacy in centralized dynamic spectrum access systems," in *ACM MobiHoc*. IEEE, 2016.

[16] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: Attacks and countermeasures," in *Proc. INFOCOM*. IEEE, 2013, pp. 2751–2759.

[17] M. A. Clark and K. Psounis, "Equal interference power allocation for efficient shared spectrum resource scheduling," *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 58–72, Jan 2017.

[18] X. Gong, A. Ispas, and G. Ascheid, "Outage-constrained power control in spectrum sharing systems with partial primary CSI," in *Global Communications Conference*. IEEE, 2012, pp. 3879–3885.

[19] L. Zheng and C. W. Tan, "Maximizing sum rates in cognitive radio networks: Convex relaxation and global optimization algorithms," *IEEE J. Selected Areas in Communications*, vol. 32, no. 3, pp. 667–680, 2014.

[20] M. H. Islam and Z. Dziong, "Joint link scheduling, beamforming and power control for maximizing the sum-rate of cognitive wireless mesh networks," in *Veh. Technology Conference (VTC)*. IEEE, 2011, pp. 1–5.

[21] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *Symposium on Security and Privacy*. IEEE, 2011, pp. 247–262.

[22] M. Clark and K. Psounis, "Can the privacy of primary networks in shared spectrum be protected?," in *IEEE INFOCOM*, Apr. 2016.

[23] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Transactions on signal processing*, vol. 50, no. 3, pp. 736–746, 2002.

[24] O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential monte carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

[25] J. M. Pak, C. K. Ahn, Y. S. Shmaliy, and M. T. Lim, "Improving reliability of particle filter-based localization in wireless sensor networks via hybrid particle/fir filtering," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 5, pp. 1089–1098, 2015.

[26] H. T. Niknejad, A. Takeuchi, S. Mita, and D. McAllester, "On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 748–758, 2012.

[27] E. Drocella et al., "3.5 GHz exclusion zone analyses and methodology," *US Dept. of Commerce, NTIA Report 15-517*, June 2015.

[28] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell, *Bayesian multiple target tracking*, Artech House, 2013.