Privacy-Preserving Collaborative Spectrum Sensing with Multiple Service Providers

Wei Wang, *Student Member, IEEE,* Qian Zhang, *Fellow, IEEE*

***Abstract*—In cognitive radio networks (CRNs), collaborative sensing has been considered as an attractive means to improve spectrum sensing performance. However, privacy issues arise when multiple service providers (SPs) collaborate on learning the spectrum availabilities. Specifically, sharing sensing data may enable malicious SPs or secondary users (SUs) to geo-locate an SU using existing localization techniques. These malicious entities could be untrusted SPs/SUs, or external attackers that compromise SPs/SUs. To incentivize SUs to contribute their sensing data, the privacy of each SU should be guaranteed. In this paper, we propose a privacy preservation framework called *PrimCos* for multi-SP collaborative sensing, which addresses several competing challenges not yet considered in the literature, that is, being compatible with general collaborative sensing schemes, providing privacy guarantee for each SU and ensuring worst case privacy protection under collusion. Both analytical and numerical results show that the proposed framework provides privacy protection for each SU with controllable impact on the sensing performance under different types of attacks.**

***Index Terms*—Privacy Preservation, Collaborative Spectrum Sensing, Cognitive Radio Networks (CRNs)**

I. Introduction

With the proliferation of mobile devices and the rapid growing of wireless services, cognitive radio networks (CRNs) have been recognized as a promising technology to alleviate the **spectrum scarcity problem [1]**. The CRNs allow secondary users (SUs) to utilize the idle spectrum unoccupied by primary users (PUs). A major technical challenge in the CRNs is to acquire knowledge about spectrum occupancy properties through spectrum sensing. Recent standard proposals for CRNs (e.g., IEEE 802.22 WRAN [2], CogNeA [3]) adopt *collaborative sensing* to improve spectrum sensing performance, that is, the sensing data from multiple SUs is aggregated to learn the spectrum occupancy. In a realistic CRN setting, multiple service providers (SPs) operate on the same set of frequency bands in one geographic area, where each SP serves a group of SUs. This multi-SP scenario has been intensively discussed in the CRN literature (e.g., [4], [5]). Existing collaborative sensing schemes [6]–[8] also show that the performance of spectrum sensing can be improved when more SUs are involved in the collaboration since the spatial diversity can be better exploited with larger amounts of sensing data. Thus, there is strong motivation for multiple SPs to acquire the spectrum occupancy status collaboratively.

Although the collaboration among multiple SPs results in better sensing performance, it suffers from privacy threats that compromise SU's incentives to join the collaborative sensing. **On the one hand,** the experiments conducted in [9] demonstrate that SPs or SUs who are untrusted or compromised by external attackers, referred to as *adversaries*, can geo-locate an SU based on its sensing data using localization techniques. **The untrusted SPs or SUs can benefit from learning the location of other SUs by either simply selling the information to location-based service providers/advertisers or malicious entities, or extracting contextual information attached to locations, e.g., individuals' hobbies, habits, activities, and relationships [10], which are of great value to advertisers or data analyzers.** Thus, the disclosure of sensing data compromises the SU's location privacy, which has aroused wide concern among consumers [11] and governments [12]. **On the other hand, with the knowledge of other SUs' sensing data, malicious entities can improve their own utilities by better manipulating the collaborative sensing results [13] or through unfair competition due to information asymmetry, which compromises the utilities of honest SUs and SPs, thereby making them reluctant to join the collaborative sensing.** Being aware of the potential privacy threats, SUs may not want to participate in the collaborative sensing if their privacy is not guaranteed. Therefore, it is essential to guarantee the privacy of each SU's sensing data in collaborative sensing.

However, most previous approaches on collaborative sensing have focused on performance improvements [6]–[8] or security related issues [13]–[15], while privacy issues are less discussed. To our knowledge, the only work on privacy issues in collaborative sensing is [9], which proposes two privacy preserving protocols to protect SU's sensing data from an untrusted server. Unfortunately, the privacy issues in the multi-SP collaboration context have not yet been investigated. Although there are numerous approaches that protect location privacy [16], [17] and sensing data privacy [18], [19], no prior framework can be applied in the context of multi-SP collaborative sensing when the following practical requirements are considered.

- **Compatibility with general collaborative sensing schemes**. The privacy preservation framework should be applied to most existing collaborative sensing schemes (e.g., [6]–[8]) directly or with simple modifications. Otherwise, the applications of the privacy preservation framework would be limited.
- **Privacy guarantee for each SU**. Each SU is mainly concerned about its own individual privacy rather than the overall privacy in CRNs. To incentivize SUs to participate

W. Wang, and Q. Zhang are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong. e-mail: {gswwang, qianzh}@cse.ust.hk.

in the collaborative sensing, it is an essential requirement that every SU's privacy is guaranteed.

- **The worst case guarantee under collusion**. To provide enough incentives for SUs to contribute their sensing data, the sensing data of each SU should be private in all cases, which is measured by the privacy guarantee in the worst case collusion. That is, each SU's privacy should be protected even when all potential adversaries, including all other SUs and SPs, collude together to launch attacks, in which the sensing data from all other SUs and the aggregated data are known to the adversaries.

To satisfy the above practical requirements, we propose a **Pri**vacy preservation framework for **m**ulti-SP **Co**llaborative **s**ensing, referred to as *PrimCos*. The core idea of PrimCos is that before executing the collaborative sensing algorithm, the original data is transformed into a privacy preserving form that maintains statistical information. Specifically, SPs share distribution information of overall sensing data in the form of *cloaks* with perturbed counts. A cloak is a generalized range that contains the sensing data from a group of SUs to hide the exact sensing data, and the perturbed count for each cloak is generated by adding random noise to the number of SUs in the cloak. The cloaks with noisy counts aggregated from all SPs can be taken as the input of general collaborative sensing schemes by simple modifications, e.g., using interpolation to reconstruct the sensing data.

The key challenge in designing PrimCos is to obtain an optimal cloaking strategy with minimal information distortion for multiple SPs without leaking the SU's private sensing data. To address this challenge, each SP first projects the original sensing data into a single-dimensional space, and then shares the projected data to collaboratively compute the optimal cloaking strategy. To achieve this goal, there are two major components in PrimCos. The first component aims at finding a projection that maintains as much statistical information as possible while still preserving each SU's privacy. To identify such a projection, the SPs collaboratively optimize the projection vector by providing only perturbed constraints without sharing the sensing data directly. The second component is to compute the optimal cloaking strategy based on the projected data, which is perturbed to preserve privacy against collusion attacks. The computation for the optimal cloaking strategy is formulated as a dynamic programming problem, and the analytical results show that the optimal cloaking strategy computed from the perturbed data is also optimal with respect to the original data.

The main contributions of this paper are threefold. First, we identify and formulate the privacy threats in multi-SP collaborative sensing. Second, we propose a privacy preservation framework, which can be applied in general collaborative sensing schemes. Its impact on collaborative sensing performance is confined with provable error bounds. Third, we quantify the privacy guaranteed for each individual SU, and analyze the privacy level achieved in the worst case collusion.

The rest of the paper is organized as follows. Section II describes the system model. The PrimCos framework is proposed in Section III and the ensured privacy and accuracy impact are analyzed in Section IV. Section V provides simulation evaluations and Section VI reviews the related works. Finally, Section VII concludes the paper.

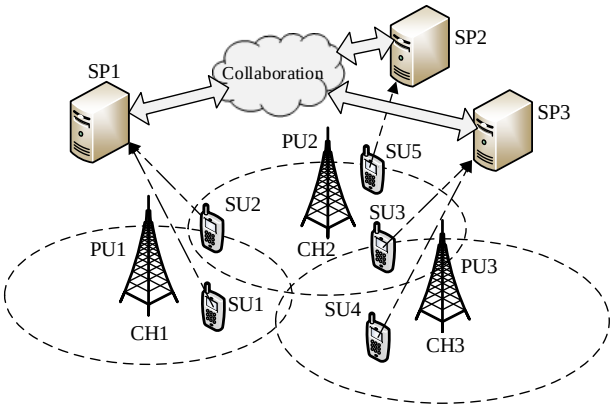

Fig. 1. System model for collaborative spectrum sensing. **In this example, five SUs are served by three collaborative SPs, and sense three channels, i.e., CH1, CH2 and CH3. Each SU sends sensing reports containing RSS values in the three channels to its own SP.**

TABLE I
NOTATIONAL CONVENTIONS

$\mathcal{U}$	the set of all SUs
$\mathcal{P}$	the set of all SPs
$\mathcal{U}^{(p)}$	the set of SUs served by SP p
$\mathcal{H}$	the set of all channels
H	the total number of channels
$\mathbf{r}_u$	the RSS vector of SU u
$\mathcal{M}$	a differential privacy mechanism
$\epsilon, \epsilon_1, \epsilon_2$	the parameters of differential privacy
$\mathbf{b}$	the projection vector
κ_{ij}	the data point distance error in projection
$\mathbf{c}_{ij}^{(p)}$	the term containing SP p's sensing data in the projection optimization problem
$\hat{\mathbf{c}}_{ij}^{(p)}$	the perturbed version of $\mathbf{c}_{ij}^{(p)}$
t_u	the projection result of $\mathbf{r}_u$
$\mathbf{x}^{(p)}$	the count vector stored by SP p
$\hat{\mathbf{x}}^{(p)}$	the perturbed version of $\mathbf{x}^{(p)}$
N	number of bins
K	number of cloaks
$\mathbf{s}$	cloaking strategy
$E(\mathbf{s}, \mathbf{x})$	the sum of squared errors in cloaking w.r.t $\mathbf{x}$
$T(i, k)$	the minimal error of a cloaking with k cloaks
$Er(j_1, j_2; \hat{\mathbf{x}})$	the squared error for an internal $[j_1, j_2]$ w.r.t $\hat{\mathbf{x}}$

II. System Model

In this section, we first describe the network architecture for collaborative sensing. Under this network architecture, we present the adversary model and discuss three possible attacks. To quantify privacy leakage, we discuss proper privacy measures. The notational conventions used in this paper are summarized in Table I.

A. Network Architecture

We consider a CRN as illustrated in Fig.1, where a set of SUs $\mathcal{U} = \bigcup_{p=1}^{P} \mathcal{U}^{(p)}$ served by a set of SPs $\mathcal{P} = \{1, ..., p, ..., P\}$

collaboratively senses channels to learn the channel occupancy properties. Each SU $u \in \mathcal{U}$ senses a set of licensed channels $\mathcal{H} = \{1, ..., h, ..., H\}$, which may be dynamically used by PUs, and obtains a vector of normalized *received signal strength* (RSS) $\mathbf{r}_u = [r_{u1}, ..., r_{uh}..., r_{uH}]$, where $\forall r_{uh} \in \mathbf{r}_u, 0 \leq r_{uh} \leq 1$ with 1 for the strongest signal and 0 for no signal. Then, an SU $u \in \mathcal{U}^{(p)}$ sends $\mathbf{r}_u$ to its SP p as the sensing report.

Multiple SPs $\mathcal{P}$ collaboratively learn channel availabilities by considering the sensing reports from all SUs $\mathcal{U}$ without revealing the sensing data of any individual SU. We make no assumptions about the aggregation functions or learning techniques of the collaborative sensing. The goals of collaboration among SPs can be manifold. SPs can aim to detect the presence of PUs using certain detection algorithms (e.g., [6], [7]), or learn the spectrum occupancy properties across multiple channels using some machine learning techniques (e.g., [8]). At the end of collaborative sensing, each SP sends the detection or learning results to all its SUs as feedback.

B. Adversary Model

We assume that an SU only trusts its own SP, and does not want its sensing data to be inferred by other SPs or SUs, which are considered as adversaries. The adversaries correctly follow the collaborative sensing scheme, yet attempt to learn the sensing data of the target SU by analyzing the information received in the collaboration. Specifically, there are three kinds of attacks considered in this paper.

- **SP Attack**. An SP adversary has perfect knowledge about the sensing data of all its SUs and collects information offered by other SPs during the collaboration. The SP adversary tries to infer the sensing data of the SUs served by other SPs based on its knowledge and the information attained in the collaboration. Multiple SPs may collude by combining their knowledge and information to infer other SUs' sensing data.
- **SU Attack**. One or multiple SU adversaries try to infer the sensing data of other SUs based on the feedback results and their own sensing data.
- **SP-SU Collusion Attack**. SP adversaries and SU adversaries could further collude by combining their knowledge and information to launch an attack.

Note that for an SU $u_i^{(p)} \in \mathcal{U}^{(p)}$, even other SUs served by the same SP p are not trustworthy. Thus, in the worst case, all SPs and SUs collude except the target SU and its SP.

C. Privacy Measure

An intuitive approach to privately compute collaborative sensing results is using secure multiparty computation [20]. However, the secure multiparty computation techniques either fail to be compatible with general collaborative sensing schemes, or cannot support real-time sensing due to the substantial overhead [20], [21]. Moreover, the secure multiparty computation techniques assume that a fraction of (e.g., a two-thirds of or a half of) the parties are trustworthy, and thus cannot defend the SP-SU Collusion Attack. Hence, we turn to privacy preservation techniques to overcome these limitations. As the traditionally used anonymization model [16], [17] was reported insufficient in protecting location data [22], we adopt a more rigorous privacy model, *differential privacy* [23], which has been recently used to quantify privacy leakage in network trace [24] and smartphone applications [25]. The intuition of differential privacy is that the aggregated results preserve an individual's privacy if the influence of any individual's data on the results is bounded. A major appealing feature of differential privacy is that it makes the worst case guarantee, that is, even if the adversaries know the data of all the individuals except the target individual, the adversaries are still uncertain about the data of the target individual. This worst case guarantee is suitable for measuring the collusion attacks in collaborative sensing. In the following, we give the formal definition of ϵ-differential privacy in the context of collaborative sensing.

Definition 1 (ϵ-differential privacy) *A mechanism $\mathcal{M}$ provides ϵ-differential privacy for an SU u if for any possible sets of sensing reports $\mathbf{R} = [\mathbf{r}_1, ..., \mathbf{r}_u, ..., \mathbf{r}_U]$ and $\mathbf{R}' = [\mathbf{r}_1, ..., \mathbf{r}'_u, ..., \mathbf{r}_U]$ differing only on u's sensing data,*

$$\left| \ln \frac{\Pr[\mathcal{M}(\mathbf{R}) = \mathbf{O}]}{\Pr[\mathcal{M}(\mathbf{R}') = \mathbf{O}]} \right| \leq \epsilon, \tag{1}$$

for all $\mathbf{O} \in Range(\mathcal{M})$, where $Range(\mathcal{M})$ is the set of possible outputs of $\mathcal{M}$.

The parameter $\epsilon > 0$ specifies the level of privacy. Specifically, lower value of ϵ ensures stronger privacy. Normally, ϵ is set to be small enough (e.g., 0.1) to make sure that $\Pr[\mathcal{M}(\mathbf{R}) = O]$ and $\Pr[\mathcal{M}(\mathbf{R}') = O]$ are roughly the same, meaning that the output O is insensitive to the change of any single individual's data. On the other hand, it implies that adversaries obtain roughly no extra information about SU's sensing data by observing $\mathbf{O}$. To ensure that an output $f(\mathbf{D})$ is ϵ-differential private, a basic method [23] is to add random noise to $f(\mathbf{D})$, where the noise follows a zero-mean Laplace distribution with noise scale of λ, denoted as $\text{Lap}(\lambda)$.

Whereas, directly applying this method to sensing data requires a significant amount of noise which results in severe information distortion. In the next section, we propose a framework to achieve differential privacy with minimal information distortion.

III. Privacy Preserving Collaborative Sensing

In this section, we propose a framework to preserve the privacy of collaborative sensing, called PrimCos, which provides privacy guarantee for each SU against three types attacks described in the previous section.

A. Design Rationale

To design a privacy preservation framework for collaborative sensing schemes leveraging different techniques, such as hypothesis testing [6], [7], or collaborative filtering [8], we observe that the basic steps of collaborative sensing are the same: (i) the sensing data is stored in a vector (sensing data of single channel) or a matrix (sensing data of multiple channels),

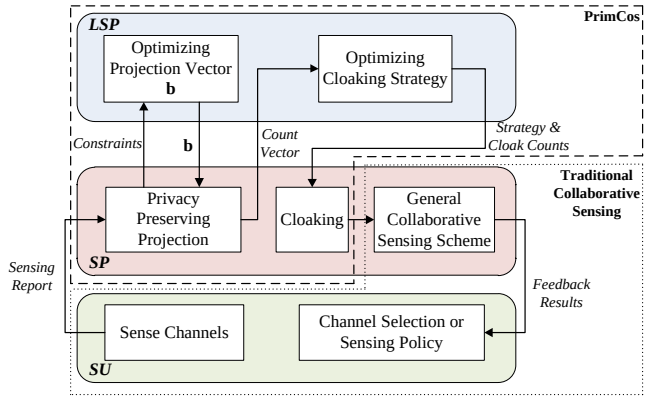

Fig. 2. The framework of PrimCos.

where each entry is an RSS value on a channel reported by an SU, (ii) the entries in the vector or the matrix are taken as input of an aggregation function to extract some statistical information, e.g., weighted summation [6], [7], or correlation coefficients [8], (iii) a central server computes the final results based on the extracted statistical information, and then sends the results as feedback to the SUs to guide their channel selection or sensing policy. According to these basic steps, we observe that the final results of collaborative sensing are based on statistical information about the aggregated sensing data rather than sensing data from any individual SU.

Based on the above observation, we design a privacy preservation framework called PrimCos, which is applicable for general collaborative sensing schemes. PrimCos transforms the original sensing data in a private form that maintains statistical information about the overall data. Specifically, PrimCos maps the original sensing data into a set of cloaks, and count the number of sensing vectors in each cloak. The benefit of cloaking is that we can do some simple modifications on cloaks to adapt to an existing collaborative sensing scheme. For example, we can use interpolation to reconstruct sensing data from cloaks by assuming that the sensing data is uniformly distributed in a cloak. The reconstructed sensing data maintains the overall statistical properties of the original data. Then, the reconstructed data is taken as input of a data aggregation function or a machine learning algorithm adopted by the collaborative sensing scheme.

B. Overview of PrimCos

Fig. 2 illustrates PrimCos, which thwarts the attacks as described in Section II. First, each SU senses channels and sends the sensing data to its own SP, which is the only trustworthy entity to the SU. To prevent attacks from other SP adversaries, each SP first projects its SUs' sensing data to a single-dimensional space, and then shares statistical information of the projected data with other SPs. The statistical information is represented by a count vector whose entry is the number of projected values in a small interval. One SP is selected as a leader, denoted as *LSP*, to gather the count vectors of projected data from all SPs to compute the optimal cloaking strategy, which is then sent back to each SP to guide the distributed cloaking algorithm. The cloaking algorithm makes sure that the feedback results leak no extra information about the original sensing data. After performing the cloaking algorithm within each SP, the cloaked data is taken as input to a general collaborative sensing scheme. The details are stated in the following parts of this section.

C. Privacy Preserving Projection

As stated in Section III-B, to preserve privacy, the SPs transform the original sensing data by a non-invertible projection before sharing it with the LSP. In this subsection, we focus on identifying an optimal matrix that projects the original H-dimensional sensing data $\mathbf{r}_u$ to a single value t_u. The benefits of applying a dimensionality-reducing projection are threefold: (i) The non-invertible transformation preserves the privacy of the original data; (ii) the communication overhead is reduced since the single-dimensional data instead of H-dimensional data is transmitted; (iii) the computations on a single-dimensional data require lower complexity.

In the following, we formulate an optimization problem to identify the projection vector $\mathbf{b}$. The projection is optimal at maintaining the distances between the data points so that similar data points are mapped to the same cloak with high probability. Thus, the objective is to minimize the overall data point distance errors introduced by the projection. The problem can be formally written as:

$$\begin{aligned} \min_{\mathbf{b},\{\kappa_{ij}\}} \quad & \sum_{\forall i,j\in\mathcal{U}} \kappa_{ij} \\ \text{s.t.} \quad & |\|\mathbf{b}^\top(\mathbf{r}_i-\mathbf{r}_j)\|_2^2 - \|(\mathbf{r}_i-\mathbf{r}_j)\|_2^2| \le \kappa_{ij}, \forall i,j\in\mathcal{U}, \end{aligned} \tag{2}$$

where $\mathbf{b}\in\mathbb{R}^{1\times H}$ is the projection vector, and $\mathbf{r}_i$, $\mathbf{r}_j$ are the sensing data from any two SUs. Unfortunately, this problem is non-convex and unlikely to be solved in polynomial time. Thus, we relax the problem as follows.

As the first step, we derive a semi-definite relaxation (SDR) [26] of (2). Let $\mathbf{A}=\mathbf{b}\mathbf{b}^\top$, we have

$$\begin{aligned} & \|\mathbf{b}^\top(\mathbf{r}_i-\mathbf{r}_j)\|_2^2 - \|(\mathbf{r}_i-\mathbf{r}_j)\|_2^2 \\ =\ & (\mathbf{r}_i-\mathbf{r}_j)^\top(\mathbf{b}\mathbf{b}^\top-\mathbf{I})(\mathbf{r}_i-\mathbf{r}_j) \\ =\ & \mathbf{r}_i^\top(\mathbf{A}-\mathbf{I})\mathbf{r}_i - 2\mathbf{r}_i^\top(\mathbf{A}-\mathbf{I})\mathbf{r}_j + \mathbf{r}_j^\top(\mathbf{A}-\mathbf{I})\mathbf{r}_j. \end{aligned} \tag{3}$$

To find an optimal $\mathbf{b}$, we can first obtain an optimal $\mathbf{A}$. The condition $\mathbf{A}=\mathbf{b}\mathbf{b}^\top$ is equivalent to $\mathbf{A}$ being a rank one symmetric positive semi-definite matrix, while the rank one constraint is dropped in SDR. Thus, according to (2) (3), the SDR version of identifying an optimal $\mathbf{A}$ is given by:

$$\min_{\mathbf{A},\{\kappa_{ij}\}} \quad \sum_{\forall i,j\in\mathcal{U}} \kappa_{ij} \tag{4a}$$

$$\text{s.t.} \quad |\mathbf{r}_i^\top(\mathbf{A}-\mathbf{I})\mathbf{r}_i - 2\mathbf{r}_i^\top(\mathbf{A}-\mathbf{I})\mathbf{r}_j + \mathbf{r}_j^\top(\mathbf{A}-\mathbf{I})\mathbf{r}_j| \le \kappa_{ij}, \quad \forall i,j\in\mathcal{U}, \tag{4b}$$

$$\mathbf{A}\succeq\mathbf{0}. \tag{4c}$$

The above problem is convex and can be solved via semi-definite programming (SDP). However, the complexity for solving this problem is still high since the number of constraints increase quadratically with the cardinality of $\mathcal{U}$ and the SDP does not scale well with the problem size.

Algorithm 1 Privacy Preserving Projection

(1) *At SP p*
for each $(i, j) \in \mathcal{U}$ **do**
  Compute $c_{ij}^{(p)}$;
  Add independent random noise Lap($\frac{3H-1}{\epsilon_1}$) to each entry of $\mathbf{c}_{ij}^{(p)}$ to obtain a perturbed version $\hat{\mathbf{c}}_{ij}^{(p)}$;
end for
Put all $\hat{\mathbf{c}}_{ij}^{(p)}$ in a set $C^{(p)}$, and send $C^{(p)}$ to the LSP;
(2) *At the LSP:*
if $C^{(p)}$ from all SPs are received **then**
  Solve the LP problem (8) and obtain $\mathbf{A}$;
  Compute $\mathbf{b}$ by applying rank-one approximation on $\mathbf{A}$;
  Send $\mathbf{b}$ to each SP;
end if
(3) *At SP p*
Project original sensing vector $\mathbf{r}_u$ into a single value using the formula $t_u = \mathbf{b}^\top \mathbf{r}_u$;
Construct a length-N count vector $\mathbf{x}^{(p)}$ based on $\left\{t_u : u \in \mathcal{U}^{(p)}\right\}$;
Injecting independent noise Lap($\frac{1}{\epsilon_2}$) to each entry of $\mathbf{x}^{(p)}$ to get $\hat{\mathbf{x}}^{(p)}$;
Send $\hat{\mathbf{x}}^{(p)}$ to the LSP;

To provide better scalability, we further approximate (4) to a linear programming (LP) problem. Specifically, we approximate the positive semi-definite constraint by tightening it into diagonal dominance constraints [27], which introduces an auxiliary variable $\mathbf{S} \in \mathbb{R}^{H \times H}$. According to [27], the positive semi-definite constraint on $\mathbf{A}$ is replaced by

$$\begin{aligned} &\mathbf{A} = \mathbf{A}^\top, \\ &-s_{lm} \le a_{lm} \le s_{lm}, \forall l, m, m \ne l, \\ &a_{ll} \ge \sum_{m=1, m\ne l}^{H} s_{lm}, \forall l, \end{aligned} \tag{5}$$

where s_{lm} (or a_{lm}) denotes the entry in the lth row and mth column of $\mathbf{S}$ (or $\mathbf{A}$). It is obvious that the diagonal dominance constraints (5) are linear, and thus the problem (4) is approximated to an LP problem.

However, (4b) contains the sensing data of SUs which cannot be shared directly. In the following, a privacy preserving projection protocol (Algorithm 1) is proposed, in which each SP contributes a set of perturbed constraints to compute $\mathbf{A}$.

We denote $\tilde{\mathbf{r}}_{ij} = \text{vec}(\mathbf{r}_i \mathbf{r}_j^\top)$ and $\hat{\mathbf{a}} = \text{vec}(\mathbf{A} - \mathbf{I})$, where vec is the vectorization function that aligns all of the matrix entries in a column vector. Then, (4b) can be rewritten as:

$$|\hat{\mathbf{a}}^\top [\tilde{\mathbf{r}}_{ii} + \tilde{\mathbf{r}}_{jj} - 2\tilde{\mathbf{r}}_{ij}]| \le \kappa_{ij}, \forall i, j \in \mathcal{U}. \tag{6}$$

The term containing SU's sensing data is denoted as $\mathbf{c}_{ij} = \tilde{\mathbf{r}}_{ii} + \tilde{\mathbf{r}}_{jj} - 2\tilde{\mathbf{r}}_{ij}$. To avoid privacy breach, $\mathbf{c}_{ij}$ is computed in a distributed way. Each SP p computes the following term:

$$\mathbf{c}_{ij}^{(p)} = \tilde{\mathbf{r}}_{ii}^{(p)} + \tilde{\mathbf{r}}_{jj}^{(p)} - 2\tilde{\mathbf{r}}_{ij}^{(p)}, \forall i, j \in \mathcal{U}^{(p)}, \tag{7}$$

where i, j are any two SUs served by the SP p.

However, sharing $\mathbf{c}_{ij}^{(p)}$ with the LSP is still not safe if the LSP colludes with one of the SUs i, j to attack the other SU. To defend against such collusion, a perturbed version $\hat{\mathbf{c}}_{ij}^{(p)}$ is sent to the LSP. Then, the LSP computes $\mathbf{A}$ by solving the following LP problem:

$$\min_{\mathbf{A}, \mathbf{S}, \{\kappa_{ijp}\}} \quad \sum_{\forall i, j \in \mathcal{U}^{(p)}, p \in \mathcal{P}} \kappa_{ijp} \tag{8a}$$

$$\text{s.t.} \quad -\kappa_{ijp} \le \hat{\mathbf{a}}^\top \hat{\mathbf{c}}_{ij}^{(p)} \le \kappa_{ijp}, \forall i, j \in \mathcal{U}^{(p)}, p \in \mathcal{P}, \tag{8b}$$

$$\mathbf{A} = \mathbf{A}^\top, \tag{8c}$$

$$-s_{lm} \le a_{lm} \le s_{lm}, \forall l, m; m \ne l, \tag{8d}$$

$$a_{ll} \ge \sum_{m=1, m\ne l}^{H} s_{lm}, \forall l. \tag{8e}$$

To preserve privacy, we add random noise Lap($\frac{3H-1}{\epsilon_1}$) to each entry in $\mathbf{c}_{ij}^{(p)}$ to obtain the perturbed version $\hat{\mathbf{c}}_{ij}^{(p)}$.

After obtaining an optimal $\mathbf{A}$, the last issue is to find an optimal $\mathbf{b}$ based on the condition $\mathbf{A} = \mathbf{b}\mathbf{b}^\top$. If $\mathbf{A}$ is of rank one, $\mathbf{b}$ can be derived directly from $\mathbf{A}$. On the other hand, if the rank of $\mathbf{A}$ is larger than one, we derive $\mathbf{b}$ using the approximation method in [28]. The intuitive idea is to apply rank-one approximation to $\mathbf{A}$ based on eigenvalue decomposition.

Then, the LSP sends $\mathbf{b}$ to each SP. An SP p projects the original sensing data $\{\mathbf{r}_u : u \in \mathcal{U}^{(p)}\}$ to a set of single values $\{t_u : u \in \mathcal{U}^{(p)}\}$ using the formula $t_u = \mathbf{b}^\top \mathbf{r}_u$. Since $\mathbf{b}$ is publicly known, t_u needs to be perturbed before sharing with other SPs to preserve privacy. However, the sensitivity of $\mathbf{r}_u \to t_u$ is given by $\max_k |b_k|$, where b_k is the kth entry of $\mathbf{b}$, which can be very large, rendering large scale noise that makes the perturbed data meaningless. To address this issue, we apply generalization on the projected values instead of adding noise directly. Since all entries of a sensing vector $\mathbf{r_u}$ fall into $[0, 1]$, the range of the projected values is bounded by $[\sum_k \min\{0, b_k\}, \sum_k \max\{0, b_k\}]$. Thus, we can divide the range into N basic bins with even lengths of $\frac{\sum_k \max\{0,b_k\} - \sum_k \min\{0,b_k\}}{N}$. Then, for any SU u, its projected value t_u falls into one of the bins. The number of values in each bin is counted and stored in a length-N vector. This count vector stored by an SP p is denoted as $\mathbf{x}^{(p)}$. Note that $\mathbf{x}^{(p)}$ implies the distribution of the projected values, and larger N preserves more information about the projected values. To make sure the aggregated data at the LSP are consistent, N is pre-defined and shared by all SPs. Then, the SP p perturbs $\mathbf{x}^{(p)}$ by adding random noise Lap($\frac{1}{\epsilon_2}$) and sends the perturbed version $\hat{\mathbf{x}}^{(p)}$. Algorithm 1 concludes the projection procedure. We can see that the data shared with the LSP are $\hat{\mathbf{c}}_{ij}^{(p)}$ and $\hat{\mathbf{x}}^{(p)}$, and both are perturbed to preserve privacy. The achieved privacy is quantified in Section IV.

D. Identify Optimal Cloaking Strategy

After the projection procedure (Algorithm 1), the LSP identifies the optimal cloaking strategy, as described in Algorithm 2. The aim of cloaking is to provide privacy preservation for the original sensing data by merging N bins into K groups (i.e.,

Algorithm 2 Identifying Optimal Cloaking Strategy

Input: a set of perturbed count vectors $\{\hat{\mathbf{x}}^{(p)} : p \in \mathcal{P}\}$; bin number N; cloak number K

1: **if** $\hat{\mathbf{x}}^{(p)}$ from all SPs are received **then**
2: Aggregate all count vectors $\hat{\mathbf{x}} = \sum_p \hat{\mathbf{x}}^{(p)}$;
3: Identify the optimal cloaking strategy $\mathbf{s}^*$ via dynamic programming following the recursive rule described by (10);
4: Merge the counts of all bins in a cloak to get the cloak count vector $\hat{\mathbf{z}}$;
5: Send $\mathbf{s}^*$ and $\hat{\mathbf{z}}$ to each SP;
6: **end if**

cloaks), while still maintaining the statistical properties for the general collaborative sensing scheme. The K cloaks cover all of the N bins and there is no intersection between cloaks. The kth cloak containing a set of bins $\{x_{s_k}, ..., x_{s_{k+1}-1}\}$ is denoted by an interval $[s_k, s_{k+1} - 1]$. Note that $s_1 = 1$ and $s_{K+1} - 1 = N$. In this way, the cloaking strategy $\mathbf{s}$ for a count vector can be presented by a length-K vector with each entry representing a border between two neighbor cloaks, i.e., $\mathbf{s} = [1, ..., s_k, ..., s_K]$.

A cloaking strategy is optimal in the sense of minimal information distortion. We measure the information distortion by the sum of squared errors between the estimated count and the true count for each bin. We assume that the count for any bin in a cloak is estimated by the average bin count in the cloak, i.e., $y = \frac{\sum_{i=1}^{n} x_i}{n}$, where y denotes the estimated count for an n-bin cloak and x_i is the true count for the ith bins in the cloak. Then, the sum of squared errors is given by

$$E(\mathbf{s}, \mathbf{x}) \triangleq \sum_{k=1}^{K} \sum_{i=s_k}^{s_{k+1}-1} (y_k - x_i)^2, \tag{9}$$

where $y_k = \left(\sum_{i=s_k}^{s_{k+1}-1} x_i\right)/(s_{k+1} - s_k)$, $s_k \in \mathbf{s}$ for all $k = 1, ..., K$, and s_{K+1} is set to be $(N + 1)$. Note that y_k is the estimated count for any bin in the kth cloak, i.e., the count for the ith bin where $\forall i \in [s_k, s_{k+1} - 1]$ is estimated by y_k.

Accordingly, the problem of finding the optimal cloaking strategy is to identify a strategy vector $\mathbf{s}$ so that $E(\mathbf{s}, \mathbf{x})$ is minimized, where $\mathbf{x}$ is the aggregated count vector given by $\mathbf{x} = \sum_p \mathbf{x}^{(p)}$. However, only a perturbed version $\hat{\mathbf{x}} = \sum_p \hat{\mathbf{x}}^{(p)}$ is available to the LSP. To solve this problem, the LSP first identifies an optimal cloaking strategy with respect to the perturbed count vectors $\hat{\mathbf{x}}$ rather than $\mathbf{x}$ by running Algorithm 2. Then, we show that the optimal cloaking strategy with the minimal expected $E(\mathbf{s}, \hat{\mathbf{x}})$ is equivalent to the optimal strategy with the minimal expected $E(\mathbf{s}, \mathbf{x})$.

We formulate the problem as follows. $T(i, k)$ is denoted as the minimal error of any cloaking with exactly k cloaks covering a perturbed partial count vector $\hat{\mathbf{x}} = [\hat{x}_1, ..., \hat{x}_i]$, and $Er(j_1, j_2; \hat{\mathbf{x}})$ is denoted as the squared error for an internal $[j_1, j_2]$ with respect to $\hat{\mathbf{x}}$, i.e., $Er(j_1, j_2; \hat{\mathbf{x}}) = \sum_{i=j_1}^{j_2} (y_j - \hat{x}_i)^2$ where $y_j = \frac{\sum_{i=j_1}^{j_2} \hat{x}_i}{j_2 - j_1 + 1}$. Therefore, the problem of identifying a cloaking strategy $\mathbf{s}$ with minimal $E(\mathbf{s}, \hat{\mathbf{x}})$ can be solved by the dynamic programming with the recursive rule described as:

$$T(N, K) = \min_{K-1 \le i \le N-1} \{T(i, K - 1) + Er(i + 1, N; \hat{\mathbf{x}})\}. \tag{10}$$

Solving this dynamic programming problem requires time complexity of $O(N^2K)$ and space complexity of $O(NK)$. After identifying the optimal cloaking strategy $\mathbf{s}^*$ via (10), the LSP sends $\mathbf{s}$ to each SP (Line 5 in Algorithm 2).

E. Cloaking and Collaborative Sensing

As depicted in Fig. 2, after receiving the optimal cloaking strategy $\mathbf{s}^*$ and corresponding count vector $\hat{\mathbf{z}}$ (Line 5 in Algorithm 2), each SP performs cloaking in a distributed way. For an SP p, its projected values $\{t_u : u \in \mathcal{U}^{(p)}\}$ are cloaked according to the optimal cloaking strategy $\mathbf{s}^*$. Then, a set of the original sensing data $\{\mathbf{r}_u : u \in \mathcal{U}_i^{(p)}\}$ is cloaked together if their corresponding projected values $\{t_u : u \in \mathcal{U}_i^{(p)}\}$ are cloaked, where $\mathcal{U}_i^{(p)} \subseteq \mathcal{U}^{(p)}$. For the original sensing vector, a cloak is presented by a vector of intervals where each interval corresponds to a channel and contains all the RSS values of that channel in the cloak. The counts for cloaks are obtained directly from $\hat{\mathbf{z}}$.

Note that $\mathbf{s}^*$ is optimized considering the sensing data from all SPs, and $\hat{\mathbf{z}}$ is the count statistics from all SPs. Thus, the cloaking result contains statistical information from all SPs. Based on the statistical information contained in the cloaking results, each SP can perform general collaborative sensing schemes (e.g., [6]–[8], [29]). **Our framework can be applied to general centralized collaborative sensing schemes as categorized in [29]. The centralized collaborative sensing schemes leverage different techniques, including hypothesis testing [6], [7], collaborative filtering [8], equal gain combination [9], and so on.** The cloaking results can be easily applied to these techniques. Recall that the basic steps of these collaborative sensing techniques include: (i) collecting sensing data from multiple SUs and storing the data in a vector or a matrix, in which each entry is an RSS value on a channel reported by an SU, (ii) making the final sensing decision based on a certain aggregation function, which extracts statistical information from the sensing data. As such, the final decision of collaborative sensing is based on statistical information about the aggregated sensing data rather than sensing data from any individual SU. As the cloaking results maintain statistical information from the sensing data, the final decision can be made by posting statistical queries over the cloaking result. **For example, we can request a histogram of the aggregated sensing data and then leverage curve fitting to derive the probability density function (PDF), based on which hypothesis testing or collaborative filtering can be smoothly performed for collaborative sensing.** Another way to derive the final decision based cloaking results is to first reconstruct the sensing data using interpolation, and then directly apply traditional collaborative sensing schemes using the reconstructed data. As both statistical queries and data reconstruction can provide the statistical information for collaborative sensing, the traditional collaborative sensing schemes are still applicable. **We illustrate the above methods with an example where the aggregated sensing data is**

partitioned into three cloaks whose ranges are [-60dBm,-80dBm], [-80dBm,-90dBm], [-90dBm,-104dBm], with corresponding counts being 5, 2, 2. Collaborative sensing is performed by matching the aggregated sensing data to either "available" hypothesis or "occupied" hypothesis as described in [6], [7]. We can derive the PDF of the cloaking data using curve fitting, and then generate synthetic sensing data, which is fed into the hypothesis testing scheme to determine whether the channel is occupied. Or we can directly generate linearly interpolated sensing data for each cloak as [-62dBm, -66dBm, -70dBm, -74dBm, -78dBm], [-82.5dBm, -87.5dBm], [-93.5dBm, -100.5dBm], and treat these vectors as inputs for hypothesis testing.

IV. Privacy and Accuracy Analysis

This section proves the privacy guarantees achieved by the proposed framework, analyzes its impact of on collaborative sensing performance.

A. Differential Privacy Analysis

Possible privacy leakages are involved with privacy preserving projection (sharing constraints and projection vector) and cloaking strategy sharing.

First, we quantify the privacy guarantee offered by privacy preserving projection (Algorithm 1), in which there are two steps involving information sharing among SPs. The first step is sending constraints $\hat{\mathbf{c}}_{ij}^{(p)}$ to the LSP, and its privacy level is quantified by the following lemma.

Lemma 1 *The mapping* $\{\mathbf{r}_i,\ \mathbf{r}_j\} \rightarrow \hat{\mathbf{c}}_{ij}^{(p)}, \forall i, j$ *ensures* ϵ_1*-differential privacy.*

Proof: We first derive the sensitivity of the mapping $\{\mathbf{r}_i, \mathbf{r}_j\} \rightarrow \hat{\mathbf{c}}_{ij}^{(p)}, \forall i, j$. w.l.o.g, we assume that r_{ih} changes and all other entries in $\mathbf{r}_i$, $\mathbf{r}_j$ stay the same. For the entry in $\mathbf{c}_{ij}^{(p)}$ that involves r_{ih} can be written as $c_{g(h-1)} = r_{ih}r_{ig} + r_{jh}r_{jg} - 2r_{ih}r_{jg}$, or $c_{h(g-1)} = r_{ig}r_{ih} + r_{jg}r_{jh} - 2r_{ig}r_{jh}$, where $1 \le h, g \le H$, and $c_{g(h-1)}$, $c_{h(g-1)}$ are the $g(h-1)$th, $h(g-1)$th entries in $\mathbf{c}_{ij}^{(p)}$, respectively. We denote the sensitivity of $c_{g(h-1)}$, $c_{h(g-1)}$ as $\Delta c_{g(h-1)}$, $\Delta c_{h(g-1)}$. Since $0 \le r_{ih} \le 1$, we have $\Delta c_{h(g-1)} = |r_{ig}| \le 1$, $\Delta c_{g(h-1)} = |r_{ig} - 2r_{jg}| \le 2$ when $h \ne g$, and $\Delta c_{h(h-1)} = \max\{r_{jh}^2, (1-r_{jh})^2\} \le 1$. Thus, $\Delta \mathbf{c}_{ij}^{(p)} = \sum_g (\Delta c_{h(g-1)} + \Delta c_{g(h-1)}) = 3H - 1$.

For brevity, we use $\mathbf{c}_{ij}$ to denote $\mathbf{c}_{ij}^{(p)}$. Denote $\mathbf{c}'_{ij}$ as the constraint after one entry of $\mathbf{r}_i$ or $\mathbf{r}_i$ changes. $c_{ij}(k)$, $c'_{ij}(k)$ are the entries of $c_{ij}^{(p)}$ and $\mathbf{c}'_{ij}$, respectively. Note that $\hat{c}_{ij}(k)$ is derived by injecting Laplace noise to $c_{ij}(k)$, that is, $\left(\hat{c}_{ij}(k) - c_{ij}(k)\right) \sim \text{Lap}(\frac{3H-1}{\epsilon_1})$. Denote the noisy version of $c'_{ij}(k)$ as $\hat{c}'_{ij}(k)$, we have $\left(\hat{c}'_{ij}(k) - c'_{ij}(k)\right) \sim \text{Lap}(\frac{3H-1}{\epsilon_1})$. Note that the laplace random noise has the following property:

$$\frac{\Pr[\hat{c}_{ij}(k) - c_{ij}(k) = O_1]}{\Pr[\hat{c}'_{ij}(k) - c'_{ij}(k) = O_2]} \le \exp\left(\frac{\epsilon_1 |O_1 - O_2|}{3H-1}\right), \tag{11}$$

where O_1,O_2 are any possible values of the random variables $\left(\hat{c}_{ij}(k) - c_{ij}(k)\right)$, $\left(\hat{c}'_{ij}(k) - c'_{ij}(k)\right)$.

Thus, for any possible value of random variables $\hat{c}_{ij}(k)$, $c_{ij}(k)$, denoted as O_k we have

$$\begin{aligned}\frac{\Pr[\hat{c}_{ij}(k) = O_k]}{\Pr[\hat{c}'_{ij}(k) = O_k]} &= \frac{\Pr[\hat{c}_{ij}(k) - c_{ij}(k) = O_k + c_{ij}(k)]}{\Pr[\hat{c}'_{ij}(k) - c'_{ij}(k) = O_k + c'_{ij}(k)]} \\ &\le \exp\left(\frac{\epsilon_1 |c'_{ij}(k) - c_{ij}(k)|}{3H-1}\right).\end{aligned} \tag{12}$$

Denote any possible value of $\hat{\mathbf{c}}_{ij}(k)$ as $\mathbf{O}$. Since noises are independently injected to each entry of $\mathbf{c}_{ij}$, thus

$$\begin{aligned}\frac{\Pr[\hat{\mathbf{c}}_{ij} = \mathbf{O}]}{\Pr[\hat{\mathbf{c}}'_{ij} = \mathbf{O}]} &= \prod_k \frac{\Pr[\hat{c}_{ij}(k) = O_k]}{\Pr[\hat{c}'_{ij}(k) = O_k]} \\ &\le \exp\left(\frac{\epsilon_1 \sum_k |c'_{ij}(k) - c_{ij}(k)|}{3H-1}\right) \\ &\le \exp\left(\frac{\epsilon_1 \parallel \hat{\mathbf{c}}_{ij} - \hat{\mathbf{c}}'_{ij} \parallel_1}{3H-1}\right) = \exp(\epsilon_1).\end{aligned} \tag{13}$$

The last inequality stands since the sensitivity of $\mathbf{c}_{ij}$ is at most $3H-1$ according to [23]. According to Definition 1, the mapping $\mathbf{r}_i$, $\mathbf{r}_j \rightarrow \hat{\mathbf{c}}_{ij}^{(p)}$ provides ϵ_1-differential privacy. ■

Another information sharing step in privacy preserving projection is sending count vector $\hat{\mathbf{x}}^{(p)}$ to the LSP.

Lemma 2 *The mapping* $\{\mathbf{r}_u : u \in \mathcal{U}^{(p)}\} \rightarrow \hat{\mathbf{x}}^{(p)}$ *preserves* ϵ_2*-differential privacy.*

Proof: This is a standard Laplace mechanism application for count query. For the detailed proof please refer to [23]. ■

Based on Lemma 1 and Lemma 2, we derive the privacy level guaranteed by the projection in the following theorem.

Theorem 1 *Algorithm 1 ensures* $(\epsilon_1 + \epsilon_2)$*-differential privacy.*

Proof: In Algorithm 1, , $\mathbf{r}_i, \mathbf{r}_j \rightarrow \hat{\mathbf{c}}_{ij}^{(p)}$ and $\mathbf{r}_u \rightarrow \hat{l}_u$ are the only two computations on the sensing data, and each provides differential privacy. Thus, according to sequential composition [30], the overall algorithm achieves $(\epsilon_1 + \epsilon_2)$-differential privacy. ■

Next, we analyze the privacy level of sharing cloaking strategy and cloak counts. Since Algorithm 2 runs within the LSP and requires no extra information from SPs (collecting $\hat{\mathbf{x}}^{(p)}$ is executed in Algorithm 1), no extra information about SUs' sensing data leaks in the execution of Algorithm 2. Thus, $\mathbf{s}^*$ and $\hat{\mathbf{z}}$ contain no more private information than $\hat{\mathbf{x}}$, which is formally concluded in the following lemma.

Lemma 3 *The mapping* $\hat{\mathbf{x}} \rightarrow \{\mathbf{s}^*, \hat{\mathbf{z}}\}$ *leaks no private information about* $\{\mathbf{r}_u\}$.

Based on the above analyses, we can use the *sequential composition* [30] to derive an overall privacy measure of the proposed framework by the following theorem.

Theorem 2 *The proposed framework is* $(\epsilon_1 + \epsilon_2)$*-differentially private.*

Proof: Similar to the proof of Theorem 1, we can use the *sequential composition* [30] in differential privacy. In the framework, three computations are carried out on the original

sensing data, that is, $\mathbf{r}_i, \mathbf{r}_j \rightarrow \hat{\mathbf{c}}_{ij}^{(p)}$ and $\mathbf{r}_u \rightarrow \hat{t}_u$ in Algorithm 1, and performing cloaking on $\{\mathbf{r}_u\}$ (the cloaking block in Fig. 2, described in Section III-E). The two computations in Algorithm 1 are proved to be $(\epsilon_1+\epsilon_2)$-differentially private by Theorem 1. Cloaking on $\{\mathbf{r}_u\}$ consists of two parts: cloaking counts and cloak intervals. The cloaking counts are obtained directly from $\hat{\mathbf{z}}\}$ without accessing the original sensing data. Only the cloak intervals is computed from the original sensing data. Since each SP performs cloaking according to the same cloaking strategy $\mathbf{s}^*$, the cloak intervals are the same from all SPs, and thus, leaks no private information. Therefore, according to sequential composition, the overall algorithm achieves $(\epsilon_1+\epsilon_2)$-differential privacy. ■

B. Privacy Analysis Under Different Types of Attacks

We analyze the privacy level achieved by the proposed framework under three different types of attacks described in Section II-B: SP Attack, SU Attack, and SP-SU Collusion Attack. In the SP Attack, multiple SPs and the LSP may collude to compromise the privacy of SUs served by a target SP. Note that SPs only share information with the LSP, i.e., the constraints and count vector. Thus, by simply extending Theorem 1, we derive the following corollary.

Corollary 1 *Assume that the target SP that SP adversaries try to attack is $p \in \mathcal{P}$. Denote $\Pr[\mathbf{r}_u]$ as the priori belief of adversaries on SU u' sensing data, and denote $\Pr[\mathbf{r}_u|A_{SP}]$ as the posterior belief on SU u's sensing data after launching the SP Attack, where A_{SP} denotes the event of the SP Attack. For any SU $u \in \mathcal{U}^{(p)}$, the difference in adversaries' beliefs on $\mathbf{r}_u$ rendered by the SP Attack is bounded by*

$$\left|\ln \frac{\Pr[\mathbf{r}_u = \mathbf{v}|A_{SP}]}{\Pr[\mathbf{r}_u = \mathbf{v}'|A_{SP}]} - \ln \frac{\Pr[\mathbf{r}_u = \mathbf{v}]}{\Pr[\mathbf{r}_u = \mathbf{v}']}\right| \leq \epsilon_1 + \epsilon_2, \tag{14}$$

where $\mathbf{v}$ and $\mathbf{v}'$ are any possible values of $\mathbf{r}_u$.

Proof: First, we interpret differential privacy from the viewpoint of an adversary. (1) can be rewritten as

$$\left|\ln \frac{\Pr[\mathbf{O}|\mathbf{R}, \mathcal{M}]}{\Pr[\mathbf{O}|\mathbf{R}', \mathcal{M}]}\right| \leq \epsilon. \tag{15}$$

We denote $\hat{\mathbf{R}} = [\mathbf{r}_1, ..., \mathbf{r}_{u-1}, \mathbf{r}_{u+1} ..., \mathbf{r}_U]$, and assume that each SU's sensing data is independent of each other [6], [7]. Applying *Bayesian rule* on the LHS of (15), we have

$$\begin{aligned}\left|\ln \frac{\Pr[\mathbf{O}|\mathbf{R}, \mathcal{M}]}{\Pr[\mathbf{O}|\mathbf{R}', \mathcal{M}]}\right| &= \left|\ln \frac{\Pr[\mathbf{r}_u|\mathbf{O}, \hat{\mathbf{R}}, \mathcal{M}] \Pr[\mathbf{r}'_u|\hat{\mathbf{R}}, \mathcal{M}]}{\Pr[\mathbf{r}'_u|\mathbf{O}, \hat{\mathbf{R}}, \mathcal{M}] \Pr[\mathbf{r}_u|\hat{\mathbf{R}}, \mathcal{M}]}\right| \\ &= \left|\ln \frac{\Pr[\mathbf{r}_u|\mathbf{O}] \Pr[\mathbf{r}'_u]}{\Pr[\mathbf{r}'_u|\mathbf{O}] \Pr[\mathbf{r}_u]}\right|. \end{aligned} \tag{16}$$

Combining (15) and (16), we derive

$$e^{-\epsilon} \cdot \frac{\Pr[\mathbf{r}_u]}{\Pr[\mathbf{r}'_u]} \leq \frac{\Pr[\mathbf{r}_u|\mathbf{O}]}{\Pr[\mathbf{r}'_u|\mathbf{O}]} \leq e^{\epsilon} \cdot \frac{\Pr[\mathbf{r}_u]}{\Pr[\mathbf{r}'_u]}. \tag{17}$$

$e^{-\epsilon}$ and e^{ϵ} approach 1 as ϵ decreases, which implies that adversaries obtain roughly no extra information about SU's sensing data by observing $\mathbf{O}$, given the condition that ϵ is small enough.

Based on the above analysis, we prove this corollary as follows.

The SP Attack is based on the information extracted from constraints and count vectors. According to Theorem 1, sending constraints and count vectors to the LSP preserves $(\epsilon_1+\epsilon_2)$-differential privacy. Then, based on (17), we can easily derive the formula in this corollary. ■

As shown in the above corollary, the level of privacy preserved under the SP Attack is given by $(\epsilon_1+\epsilon_2)$. The smaller $(\epsilon_1+\epsilon_2)$ ensures a tighter bound on the difference of adversaries beliefs on $\mathbf{r}_u$ rendered by the SP Attack, which means that less private information is leaked under the SP Attack.

Similar to Corollary 1, we measure the privacy level achieved under the SU Attack using the following corollary.

Corollary 2 *Assume that the target SU that SU adversaries try to attack is $u \in \mathcal{U}$. Denote $\Pr[\mathbf{r}_u]$ as the priori belief of adversaries on SU u' sensing data, and denote $\Pr[\mathbf{r}_u|A_{SU}]$ as the posterior belief on SU u's sensing data after launching the SU Attack, where A_{SU} denotes the event of the SU Attack. The difference in adversaries beliefs on $\mathbf{r}_u$ rendered by the SP Attack is bounded by*

$$\left|\ln \frac{\Pr[\mathbf{r}_u = \mathbf{v}|A_{SU}]}{\Pr[\mathbf{r}_u = \mathbf{v}'|A_{SU}]} - \ln \frac{\Pr[\mathbf{r}_u = \mathbf{v}]}{\Pr[\mathbf{r}_u = \mathbf{v}']}\right| \leq \epsilon_2, \tag{18}$$

where $\mathbf{v}$ and $\mathbf{v}'$ are any possible values of $\mathbf{r}_u$.

Proof: SU Adversaries launch an SU Attack based on the feedback results from SPs. The feedback results are output by the spectrum learning algorithm which takes the cloaked data as input. As stated in the proof of Theorem 2, the cloaked data contains no extra private information other than the information contained in $\mathbf{s}^\star$ and $\hat{\mathbf{z}}$. Then, based on Proposition 3 and Lemma 5, we derive that the mapping from the original sensing data to the feedback results is ϵ_2-differential private. Thus, similar to the proof of Corollary 1, we derive the formula in this corollary according to (17). ■

As for SP-SU Collusion Attack, we derive the following corollary by simply extending Theorem 2.

Corollary 3 *Assume that the target SU that SP and SU adversaries collude to attack is $u \in \mathcal{U}$. Note that the SP that serves u is assumed to be trustworthy and is not in the collusion. Denote $\Pr[\mathbf{r}_u]$ as the priori belief of adversaries on SU u' sensing data, and denote $\Pr[\mathbf{r}_u|A_{SP}]$ as the posterior belief on SU u's sensing data after launching the SP Attack, where $A_{SP,SU}$ denotes the event of the SP-SU Collusion Attack. The difference in adversaries beliefs about $\mathbf{r}_u$ rendered by the SP Attack is bounded by*

$$\left|\ln \frac{\Pr[\mathbf{r}_u = \mathbf{v}|A_{SP,SU}]}{\Pr[\mathbf{r}_u = \mathbf{v}'|A_{SP,SU}]} - \ln \frac{\Pr[\mathbf{r}_u = \mathbf{v}]}{\Pr[\mathbf{r}_u = \mathbf{v}']}\right| \leq \epsilon_1 + \epsilon_2, \tag{19}$$

where $\mathbf{v}$ and $\mathbf{v}'$ are any possible values of $\mathbf{r}_u$.

Proof: Theorem 2 proves that the framework is $(\epsilon_1+\epsilon_2)$-differential private, which measures the privacy level achieved when all outgoing messages from u's serving SP are considered together. Thus, even all other SUs and SPs (except u's

serving SP) collude, the information about u's sensing data is no more than the information contained in all outgoing messages from u's serving SP. Therefore, the formula in this corollary can be proved according to (17). ∎

Note that the LSP is untrusted and can launch the SP Attack and the SP-SU Collusion Attack, which is already considered in Corollary 1 and Corollary 3. Thus, our framework still ensures differential privacy for the SUs served by other SPs even when the LSP is compromised. However, we make the assumption that the LSP is honest to its own SUs and the SUs' data is safe to be kept at their own SPs. Therefore, if the LSP is compromised by another malicious party, the privacy of the SUs served by the LSP is compromised while the privacy of other SUs is still preserved. In this case, we can employ certain single-SP privacy preservation mechanisms, e.g., [9], to prevent the privacy breach of the SUs served by the compromised LSP.

C. Impact on Sensing Performance

In the following, we provide some theoretical analysis on the expected error incurred by cloaking. To validate the cloaking strategy derived by Algorithm 2, we need to compare (i) the error of the cloaking using $\mathbf{s}$ on the perturbed count vector $\hat{\mathbf{x}}$, and (ii) the error of the same cloaking on the true count vector $\mathbf{x}$. First, we analyze the error introduced by merging consecutive noisy counts into a single cloak. The expected error of a cloak on the perturbed count vector is derived by the following lemma.

Lemma 4 *Given a cloak that is constructed by merging all counts in a perturbed partial count vector $[\hat{x}_{j_1}, ..., \hat{x}_{j_2}]$, the corresponding true count vector is $[x_{j_1}, ..., x_{j_2}]$. The expected sum of squared error with respect to $[\hat{x}_{j_1}, ..., \hat{x}_{j_2}]$ is*

$$\mathbb{E}[Er(j_1, j_2; \hat{\mathbf{x}})] = \sum_{i=j_1}^{j_2} x_i^2 - \frac{\left(\sum_{i=j_1}^{j_2} x_i\right)^2}{j_2 - j_1 + 1} + \frac{2P(j_2 - j_1)}{\epsilon_2^2}. \quad (20)$$

Proof: Note that $\hat{\mathbf{x}} = \sum_p \hat{\mathbf{x}}^{(p)}, p \in \mathcal{P}$, and each entry in $\hat{\mathbf{x}}^{(p)}$ is perturbed by an independent random noise $\text{Lap}(\frac{1}{\epsilon_2})$ according to Line 16 in Algorithm 1.

$Er(j_1, j_2; \hat{\mathbf{x}})$ is defined by $Er(j_1, j_2; \hat{\mathbf{x}}) = \sum_{i=j_1}^{j_2}(y_j - \hat{x}_i)^2$ where $y_j = \frac{\sum_{i=j_1}^{j_2} \hat{x}_i}{j_2 - j_1 + 1}$.

$$\begin{aligned}
&\mathbb{E}[Er(j_1, j_2; \hat{\mathbf{x}})] \\
&= \mathbb{E}\left[\sum_{i=j_1}^{j_2}\left(y_j - \hat{x}_i\right)^2\right] = \mathbb{E}\left[\sum_{i=j_1}^{j_2}\left(\sum_p \hat{x}_i^{(p)}\right)^2 - \frac{\left(\sum_{i=j_1}^{j_2}\sum_p \hat{x}_i^{(p)}\right)^2}{j_2 - j_1 + 1}\right] \\
&= \sum_{i=j_1}^{j_2} x_i^2 - \frac{\left(\sum_{i=j_1}^{j_2} x_i\right)^2}{j_2 - j_1 + 1} + \mathbb{E}\left[\sum_{i=j_1}^{j_2}\left(\sum_p \left(\hat{x}_i^{(p)} - x_i^{(p)}\right)\right)^2\right] \\
&\quad - \frac{1}{j_2 - j_1 + 1}\mathbb{E}\left[\left(\sum_{i=j_1}^{j_2}\sum_p (\hat{x}_i^{(p)} - x_i^{(p)})\right)^2\right]. \quad (21)
\end{aligned}$$

Note that $\hat{x}_i^{(p)}$ is derived by injecting random Laplace noise to $x_i^{(p)}$, i.e., $\left(\hat{x}_i^{(p)} - x_i^{(p)}\right) \sim \text{Lap}(\frac{1}{\epsilon_2})$. We have $\mathbb{E}\left[\left(\hat{x}_i^{(p)} - x_i^{(p)}\right)\left(\hat{x}_l^{(p)} - x_l^{(p)}\right)\right] = 0, \forall i \neq l$, $\mathbb{E}\left[\left(\hat{x}_i^{(p)} - x_i^{(p)}\right)^2\right] = \frac{2}{\epsilon_2^2}$, and $\mathbb{E}[\hat{x}_i^{(p)} - x_i^{(p)}] = 0$. Thus, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=j_1}^{j_2}\left(\sum_p \left(\hat{x}_i^{(p)} - x_i^{(p)}\right)\right)^2\right] - \frac{1}{j_2 - j_1}\mathbb{E}\left[\left(\sum_{i=j_1}^{j_2}\sum_p (\hat{x}_i^{(p)} - x_i^{(p)})\right)^2\right] \\
&= \mathbb{E}\left[\sum_{i=j_1}^{j_2}\sum_p \left(\hat{x}_i^{(p)} - x_i^{(p)}\right)^2 - \frac{\sum_{i=j_1}^{j_2}\sum_p \left(\hat{x}_i^{(p)} - x_i^{(p)}\right)^2}{j_2 - j_1 + 1}\right] \\
&= \frac{2P(j_2 - j_1 - 1)}{\epsilon_2^2}. \quad (22)
\end{aligned}$$

Based on (21) and (22), we derive

$$\mathbb{E}[Er(j_1, j_2; \hat{\mathbf{x}})] = \sum_{i=j_1}^{j_2} x_i^2 - \frac{\left(\sum_{i=j_1}^{j_2} x_i\right)^2}{j_2 - j_1 + 1} + \frac{2P(j_2 - j_1)}{\epsilon_2^2}. \quad (23)$$

∎

Note that Lemma 4 estimates the sum of squared error of a cloak with respect to the perturbed count vector. The following lemma derives the expected sum of squared error of a cloak with respect to the true count vector.

Lemma 5 *Given a cloak that is constructed by merging all counts in a perturbed partial count vector $[\hat{x}_{j_1}, ..., \hat{x}_{j_2}]$, the corresponding true count vector is $[x_{j_1}, ..., x_{j_2}]$. The expected sum of squared error with respect to $[x_{j_1}, ..., x_{j_2}]$ is*

$$\mathbb{E}[Er(j_1, j_2; \mathbf{x})] = \sum_{i=j_1}^{j_2} (x_i)^2 - \frac{\left(\sum_{i=j_1}^{j_2} x_i\right)^2}{j_2 - j_1 + 1} + \frac{2P}{\epsilon_2^2}. \quad (24)$$

Proof: Note that the estimated count y_j for each bin in the cloak is given by $y_j = (\sum_{i=j_1}^{j_2} \hat{x}_i)/(j_2 - j_1 + 1)$. Similar to the proof of Lemma 4, we derive $\mathbb{E}[Er(j_1, j_2; \mathbf{x})]$ as follows.

$$\begin{aligned}
\mathbb{E}[Er(j_1, j_2; \mathbf{x})] &= \sum_{i=j_1}^{j_2} x_i^2 - \frac{2\left(\sum_{i=j_1}^{j_2} x_i\right)^2}{j_2 - j_1 + 1} + \frac{\mathbb{E}\left[\left(\sum_{i=j_1}^{j_2}\sum_p \hat{x}_i^{(p)}\right)^2\right]}{j_2 - j_1 + 1} \\
&= \sum_{i=j_1}^{j_2} x_i^2 - \frac{\left(\sum_{i=j_1}^{j_2} x_i\right)^2}{j_2 - j_1 + 1} + \frac{2P}{\epsilon_2^2}. \quad (25)
\end{aligned}$$

∎

Lemma 4 and Lemma 5 show the expected sum of squared errors of a single cloak. Based on these two lemmas, we derive the following theorem.

Theorem 3 *Finding an optimal $\mathbf{s}$ by minimizing $\mathbb{E}[E(\mathbf{s}, \hat{\mathbf{x}})]$ leads to the solution that minimizes $\mathbb{E}[E(\mathbf{s}, \mathbf{x})]$, i.e.,*

$$\arg\min_{\mathbf{s}} \mathbb{E}[E(\mathbf{s}, \hat{\mathbf{x}})] = \arg\min_{\mathbf{s}} \mathbb{E}[E(\mathbf{s}, \mathbf{x})]. \quad (26)$$

Proof: Denote that $e(j_1, j_2) = \sum_{i=j_1}^{j_2}(x_i)^2 - \frac{(\sum_{i=j_1}^{j_2} x_i)^2}{(j_2 - j_1 + 1)}$. We can simply extend the analysis from one cloak to multiple cloaks based on Lemma 4 and Lemma 5 as follows.

$$\begin{aligned}
\mathbb{E}[E(\mathbf{s}, \hat{\mathbf{x}})] &= \mathbb{E}[\sum_{k=1}^{K} Er(s_k, s_{k+1} - 1; \hat{\mathbf{x}})] \\
&= \sum_{k=1}^{K} e(s_k, s_{k+1} - 1) + \frac{2PK}{\epsilon_2^2}. \quad (27)
\end{aligned}$$

Similarly, we derive

$$\mathbb{E}[E(\mathbf{s},\mathbf{x})] = \mathbb{E}[E(\mathbf{s},\hat{\mathbf{x}})] + \frac{2P(N-2K)}{\epsilon_2^2}. \quad (28)$$

Since $\frac{2P(N-2K)}{\epsilon_2^2}$ is a constant when $\mathbf{s}$ varies, we have

$$\arg\min_{\mathbf{s}} \mathbb{E}[E(\mathbf{s},\mathbf{x})] = \arg\min_{\mathbf{s}} \mathbb{E}[E(\mathbf{s},\hat{\mathbf{x}})]. \quad (29)$$

■

Theorem 3 validates Algorithm 2, which aims to identify the optimal cloaking $\mathbf{s}^*$ with respect to $\mathbf{x}$ by minimizing the sum of squared error with respect to $\hat{\mathbf{x}}$.

Note that before cloaking, the expected sum of squared error of the perturbed count vector

$$\hat{\mathbf{x}} = \sum_{i=1}^{N}\sum_{p=1}^{P} 2/\epsilon_2^2 = 2PN/\epsilon_2^2. \quad (30)$$

Then, we get the following observation.

Corollary 4 *The cloaking* $\mathbf{s} = \{x_{s_k}, ..., x_{s_{k+1}-1}\}$ *reduces the expected sum of squared error in the perturbed count vector if* $\epsilon_2 < \sqrt{\frac{2PK}{\sum_{k=1}^{K} e(s_k, s_{k+1}-1)}}$.

Corollary 4 shows that if ϵ_2 is small enough, Algorithm 2 can reduce the overall noises by merging neighbor noisy counts.

Based on the above analyses, we observe that Algorithm 2 obtains the optimal cloaking that minimizes the sum of squared error that is introduced into the sensing results. In addition, the overall relative error is proportional to the number of SPs and is inversely proportional to the privacy parameter ϵ_2^2. In the next section, we conduct simulations to show that our framework has little impact on the sensing results.

V. Evaluations

In this section, we evaluate the privacy levels achieved by PrimCos and its impacts on collaborative sensing performance.

A. Simulation Setup

We consider a realistic CRN where 100 SUs and 10 PUs are randomly distributed within a 5km × 5km square. There are 5 SPs in the CRN and each SU randomly selects a serving SP with equal probability. The licensed spectrum is divided into 20 channels. We assume that the probability of each PU being active is uniformly distributed within [0, 1]. For simplicity, it is assumed that each active PU uses one channel and each channel is used by at most one PU. An SU u's normalized RSS on channel h is based on the model described in [6]:

$$r_{uh} = \frac{P_u}{P_o} = \left(\frac{d_o}{d_{uh}}\right)^a e^{X_{uh}}, \quad (31)$$

where P_u, P_o are the received primary signal strengths at u and at a reference distance $d_o = 1$, respectively, d_{uh} the distance between u and the PU using channel h, and $a = 4$ the path loss exponent. $e^{X_{uh}}$ is the shadowing fading factor following a Gaussian distribution described by $X_{uh} \sim \mathcal{N}(0, \sigma)$, where $\sigma_{dB} = 5.5$ dB, $\sigma_{dB} = \frac{10\sigma}{\ln(10)}$. The total number of bins $N = 50$ and the number of cloaks $K = 10$. Unless explicitly stated otherwise, we use the above configurations in the simulations.

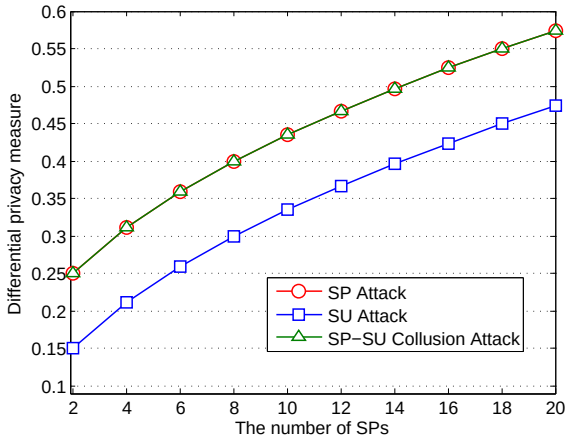

Fig. 3. **Differential privacy levels with different number of SPs.**

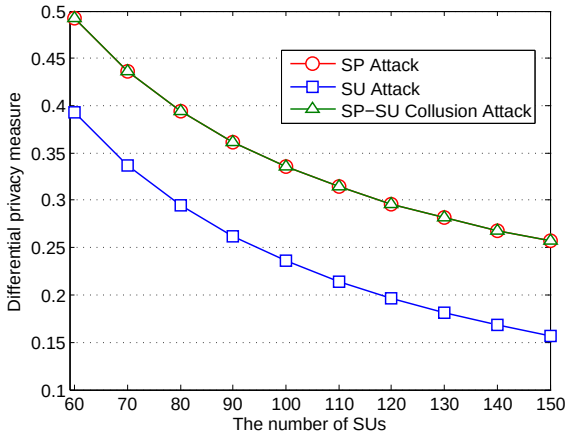

Fig. 4. **Differential privacy levels with different number of SUs.**

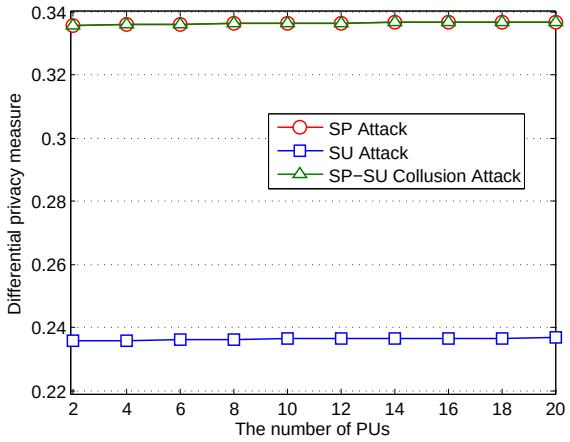

Fig. 5. **Differential privacy levels with different number of PUs.**

B. Simulation Results

Fig. 3-5 shows the achieved privacy level under different types of attacks. In Fig. 3-5, ϵ_1 is set to 0.1, and we keep the overall relative error as a constant, i.e., -5 dB, where the overall

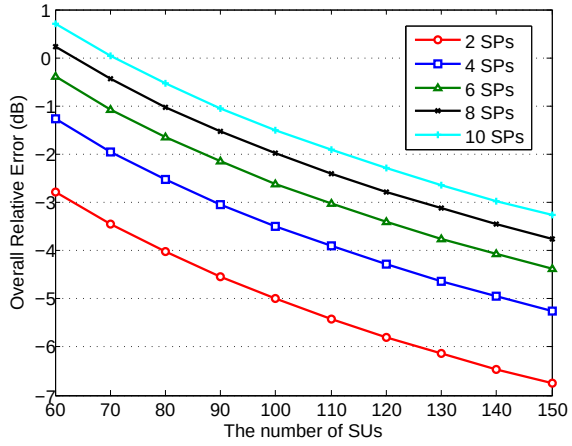

Fig. 6. Overall relative error of PrimCos.

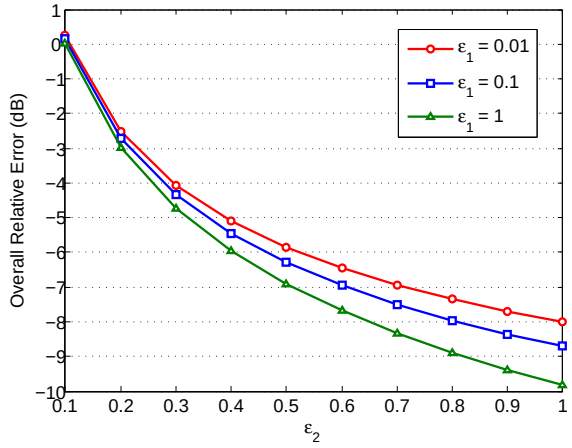

Fig. 7. Error-Privacy tradeoff of PrimCos

relative error is defined as the ratio of sum of errors on cloak counts to the overall count. The privacy level is measured by ϵ as defined in Definition 1, where a smaller ϵ implies higher privacy level. ϵ is usually set to be less than 1 to preserve privacy [24], [25]. We see that ϵ is smaller than 0.6 in all cases demonstrated in Fig. 3-5, which shows that our framework provides strong privacy protection for each SU.

Fig. 3 depicts the differential privacy levels in the CRNs with different numbers of SPs, and shows that the privacy levels achieved under all three types of attacks are lower (i.e., larger ϵ) with more SPs. This is because ϵ measures the privacy level in the worst case, where all SPs collude excluding the target SP, leading to the result that more sensing data is known to adversaries when there are more SPs. Fig. 4 shows the differential privacy levels in the CRNs with different numbers of SUs, in which the privacy levels under the three types of attacks increase (i.e., ϵ decreases) as the number of SUs increases. This is because ϵ measures individual privacy, and more SUs in the CRNs results in more SUs in a cloak, which makes it harder for adversaries to compromise SU's privacy. Fig. 5 depicts that the differential privacy levels stay roughly the same when varying the number of PUs since the number of PUs has no impact on the cloaking algorithm. We observe that the privacy levels provided by PrimCos are the same under the SP Attack and the SP-SU Collusion Attack. **This is because all outgoing messages from an SU's serving SP is protected by $(\epsilon_1+\epsilon_2)$-differential privacy in consideration of the SP-SU collusion, which is consistent with the results of Corollary 1 and Corollary 3.**

Fig. 6 illustrates the overall relative error of PrimCos, which indicates PrimCos's impact on the system performance of collaborative sensing. We set $\epsilon_1 = 0.1$ and $\epsilon_2 = 0.2$, which provide strong privacy protection according to [24], [25]. It can be seen that in all cases demonstrated, the overall relative error is less than 1 dB, which can be neglected considering shadowing fading effect (with noise scale of 5.5 dB). **Therefore, we conclude that when the number of SPs is smaller than 10, and the number of SUs is larger than 60 – which are common cases in CRNs – PrimCos has little impact on collaborative sensing.**

Fig. 7 shows the tradeoff between the collaborative sensing performance and the SUs' privacy. Specifically, the overall error is lower when ϵ_1 or ϵ_2 increases. We also see that ϵ_2 has much stronger impact on the overall error than ϵ_1. This is because ϵ_2 controls the noise scale of the perturbed count vector $\hat{\mathbf{x}}$, which directly affects the cloak count noise, while ϵ_1 only has impact on the selection of the projection space.

VI. Related Work

Numerous techniques have been proposed for preserving privacy by modifying or transforming the original data. Basically, these techniques can be divided into four main categories: random perturbation, differential privacy, anonymization, and cryptographic methods.

First, random perturbation transforms the original data by replacing a subset of the data points with randomly selected values [9], [18], [19]. However, none of them can achieve the same individual privacy strength provided in this paper. The only work that studies the privacy issue in the context of collaborative sensing is by Li et al. [9]. Li et al. [9] identify a location privacy leakage problem in which an untrusted fusion center attempts to geo-locate a user by comparing the differences in aggregated results, and proposes a homomorphic encryption protocol and a distributed dummy report protocol to handle different types of attacks. However, the approach proposed in [9] only considers a single service provider, while we study the scenario of multiple service providers. Moreover, it only measures the adversary's overall uncertainty on users' location, which cannot provide the privacy guarantee at individual level as described in this paper.

In addition, spatial cloaking and anonymization are widely adopted to preserve privacy in location-based services [16] and participatory sensing [17], where a value provided by a user is indistinguishable from those of $k-1$ other users, known as *k-anonymity*. However, a recent measurement study [22] has reported that sharing anonymized location data may still lead to privacy risks. Moreover, none of them has considered multiple service providers, or user collusion.

Recently, differential privacy has gained popularity in privacy analysis. McSherry et al. [24] devise a collection of practical tools for network trace analyses with differential privacy guarantee. Lee et al. [25] propose a smartphone application platform that combines support for applications' functional needs with differential privacy protection for the smartphone users. In this paper, we adopt the notion of differential privacy to quantify privacy leakage, while the privacy preserving framework is quite different from these works.

The fourth category preserves privacy via cryptographic techniques. Girao et al. [31] aggregate data based on homomorphic encryption, which preserves privacy by performing certain computations on ciphertext. The limitation of homomorphic encryption is that a server must know all the users that have reported data to compute the final aggregated results. Secure information aggregation frameworks are proposed in [32]. Nonetheless, all these methods fall short under the collusion attacks described in this paper.

Privacy issues in CRNs have been studies recently [9], [33]–[35]. Liu et al. [33] apply cryptographic tools to thwart location privacy leakage in dynamic spectrum auctions. Huang et al. [34] propose a truthful spectrum auction mechanism by introducing a trusted third party. Location privacy issues in collaborative sensing and database-driven CRNs are studied in [9], [35]. Different from these studies, we provide collusion-resistent differential privacy protection for each individual in the context of multiple SPs.

VII. Conclusion

This paper proposed a privacy preservation framework called PrimCos for collaborative sensing with multiple SPs. In PrimCos, each SP transforms the original sensing data into cloaks that hide individual sensing data yet maintains overall statistical information. The transformation consists of a projection that maps the original sensing data within an SP to a single-dimensional space with minimal data distortion, and a cloaking algorithm that contains the statistical information about the overall sensing data across all SPs. The privacy guarantee and the effectiveness of PrimCos are validated by both analytical and numerical results.

References

[1] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Elsevier Computer Networks*, 2006.

[2] IEEE 802.22 WRAN WG on Broadband Wireless Acess Standards. http://www.ieee802.org/22.

[3] CogNeA: Cognitive Networking Alliance. http://www.cognea.org.

[4] D. Niyato and E. Hossain, "Competitive pricing for spectrum sharing in cognitive radio networks: Dynamic game, inefficiency of nash equilibrium, and collusion," *IEEE J. Sel. Areas Commun.*, 2008.

[5] C. Cordeiro, K. Challapali, D. Birru, and N. Sai Shankar, "Ieee 802.22: the first worldwide wireless standard based on cognitive radios," in *Proc. IEEE DySPAN*, 2005.

[6] A. Min, X. Zhang, and K. Shin, "Detection of small-scale primary users in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, 2011.

[7] R. Chen, J. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *Proc.IEEE INFOCOM*, 2008.

[8] H. Li, "Learning the spectrum via collaborative filtering in cognitive radio networks," in *Proc. IEEE DySPAN*, 2010.

[9] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen, "Location privacy preservation in collaborative spectrum sensing," in *Proc. IEEE INFOCOM*, 2012.

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

[11] Microsoft, "Location based services usage and perceptions survey," Apr. 2011.

[12] D. Weitzner, "Obama administration calls for a consumer privacy bill of rights for the digital age," Feb. 2012. [Online]. Available: http://www.whitehouse.gov/blog/2012/02/23/we-can-t-wait-obama-administration-calls-consumer-privacy-bill-rights-digital-age

[13] R. Zhang, J. Zhang, Y. Zhang, and C. Zhang, "Secure crowdsourcing-based cooperative spectrum sensing," in *Proc. IEEE INFOCOM*, 2013.

[14] S. Li, H. Zhu, Z. Gao, X. Guan, and K. Xing, "Yousense: Mitigating entropy selshness indistributed collaborative spectrum sensing," in *Proc. IEEE INFOCOM*, 2013.

[15] X. O. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "Artsense: Anonymous reputation and trust in participatory sensing," in *Proc.IEEE INFOCOM*, 2013.

[16] X. Liu, K. Liu, L. Guo, X. Li, and Y. Fang, "A game-theoretic approach for achieving k-anonymity in location based services," in *Proc. IEEE INFOCOM*, 2013.

[17] K. Vu, R. Zheng, and J. Gao, "Efficient algorithms for k-anonymous location privacy in participatory sensing," in *Proc. IEEE INFOCOM*, 2012.

[18] B. Liu, Y. Jiang, F. Sha, and R. Govindan, "Cloud-enabled privacy-preserving collaborative learning for mobile sensing," in *Proc. ACM SenSys*, 2012.

[19] H. Ahmadi, N. Pham, R. Ganti, T. Abdelzaher, S. Nath, and J. Han, "Privacy-aware regression modeling of participatory sensing data," in *Proc. ACM SenSys*, 2010.

[20] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Process. Mag*, 2013.

[21] "Is multiparty computation any good in practice?" in *Proc. IEEE ICASSP*, 2011.

[22] H. Zang and J. Bolot, "Anonymization of location data does not work: A large-scale measurement study," in *Proc ACM MobiCom*, 2011.

[23] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," *Theory of Cryptography*, 2006.

[24] F. McSherry and R. Mahajan, "Differentially-private network trace analysis," in *Proc. ACM SIGCOMM*, 2010.

[25] S. Lee, E. L. Wong, D. Goel, M. Dahlin, and V. Shmatikov, "πbox: a platform for privacy-preserving apps," in *Proc. NSDI*, 2013.

[26] P. Tan and L. Rasmussen, "The application of semidefinite programming for detection in cdma," *IEEE J. Sel. Areas Commun.*, 2001.

[27] R. Rosales and G. Fung, "Learning sparse metrics via linear programming," in *Proc. ACM SIGKDD*, 2006.

[28] Z. Luo, W. Ma, A. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, 2010.

[29] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Commun. Surveys Tuts.*, 2009.

[30] F. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proc. ACM SIGMOD*, 2009.

[31] J. Girao, D. Westhoff, and M. Schneider, "Cda: Concealed data aggregation for reverse multicast traffic in wireless sensor networks," in *Proc. IEEE ICC*, 2005.

[32] B. Przydatek, D. Song, and A. Perrig, "Sia: Secure information aggregation in sensor networks," in *Proc. ACM SenSys*, 2003.

[33] S. Liu, H. Zhu, R. Du, C. Chen, and X. Guan, "Location privacy preserving dynamic spectrum auction in cognitive radio network," in *Proc. IEEE ICDCS*, 2013.

[34] Q. Huang, Y. Tao, and F. Wu, "Spring: A strategy-proof and privacy preserving spectrum auction mechanism," in *Proc. IEEE INFOCOM*, 2013.

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