Protecting the Primary Users' Operational Privacy in Spectrum Sharing

Behnam Bahrak*, Sudeep Bhattarai*, Abid Ullah*, Jung-Min "Jerry" Park*, Jeffery Reed*, David Gurney†
*Department of Electrical and Computer Engineering, Virginia Tech
{bahrak, sbhattar, abid, jungmin, reedjh}@vt.edu
†Motorola Solutions
David.P.Gurney@motorolasolutions.com

***Abstract*—Although using geolocation databases is a practical approach for enabling spectrum sharing, it poses a potentially serious privacy problem. Secondary users (queriers), through seemingly innocuous queries to the database, can determine the types and locations of incumbent systems operating in a given region of interest, and thus compromise the incumbents' *operational privacy*. When the incumbent systems (primary users) are commercial systems, this is typically not a critical issue. However, if the incumbents are federal government systems, including military systems, then the information revealed by the databases can lead to a serious breach of operational privacy. In this paper, we propose privacy-preserving mechanisms and techniques for an obfuscated geolocation database that can enable the coexistence of primary and secondary users while preserving the operational privacy of the primary users.**

I. Introduction

The FCC ruling on TV white spaces proposed relying on a database of the incumbents' spectrum usage information as the primary means of determining white space availability at any white space device (WSD) [1]. The database is required to house an up-to-date repository of incumbents (a.k.a. primary users (PUs)), and to use this information to determine white space availability at a white space device's location. Geolocation enabled devices have knowledge of the specific interference protection requirements of each incumbent, which allows varying levels of protection to be applied, and thus maximize utilization of the spectrum.

Although using geolocation databases (GDBs) for spectrum sharing has many advantages, it poses potentially serious privacy issues. For instance, secondary users (SUs), through seemingly innocuous queries to the database, can determine the *types* and *locations* of incumbent systems operating in a given region of interest—we refer to this as the *operational privacy* of the incumbents. When the incumbent systems are commercial systems, this is not a major issue. However, when the incumbents are federal government, possibly military, systems, then the information revealed by the databases may result in a serious threat to the PUs' operational privacy. Moreover, there is the possibility that SUs can obtain knowledge beyond that revealed directly by the database's query replies by using sophisticated inference techniques—we refer to this as a *database inference attack*.

The operational privacy of PUs is an especially critical concern in light of the recent calls in the United States for sharing of federal government, including military, spectrum in the 3.5 GHz band with non-government systems [2]. The 3.5 GHz band is currently used by the U.S. Department of Defense (DoD) for certain radar installations as well as by non-federal users. It is highly likely that spectrum sharing in the 3.5 GHz band will be enabled by GDBs aided by local spectrum sensing.

Some of the operational attributes of PU transmitters that may need to be protected include transmitter identity, geolocation, antenna parameters, transmission power, transmission protected contours, and times of operation. This problem cannot be adequately addressed by limiting access to the database, since all SUs need access to it to enable spectrum sharing. A more viable approach is to "obfuscate" the information revealed by the database in an intelligent manner such that a certain level of privacy is assured while supporting efficient use of the spectrum. In this paper, we propose privacy-preserving mechanisms and techniques for an obfuscated GDB that can enable spectrum sharing between PUs and SUs while preserving the PUs' operational privacy.

The contributions of this paper are summarized below:

- We describe database inference attacks, and demonstrate how the SUs can readily infer operational characteristics of PUs using seemly innocuous database queries. Specifically, we describe attacks for inferring PUs' location, path of movement, speed, and time of operation.
- We propose several privacy-preserving techniques for preserving the operational privacy of the PUs.
- In order to adequately evaluate the proposed solutions, we have developed metrics to quantify the operational privacy of PUs and the spectrum utilization efficiency of SUs.
- Using results from simulation experiments, we show the effectiveness of the proposed privacy-preserving techniques. Our results demonstrate the fundamental tradeoff between the PUs' location privacy and the SUs' spectrum utilization efficiency.

The rest of the paper is organized as follows. We first discuss some related work in section II, and then describe the spectrum access system and the database access protocol in section III. Section IV introduces inference attacks against operational privacy of PUs, and in section V, we propose privacy-preserving techniques to mitigate these attacks. Section VI introduces metrics for quantifying operational privacy of PUs and spectrum utilization efficiency of SUs. Issues related to the experimental evaluation are discussed in VII and finally section VIII concludes the paper.

II. Related Work

Obfuscating the contents of the query replies from the GDB is one approach for preserving the privacy of PUs in spectrum sharing. Because privacy is an important concern in many database applications, privacy-preserving data management techniques [3] is an area of active research. Although there is very little, if any, existing work on privacy-preserving databases for spectrum sharing, there is an abundance of existing work on the topic in the context of other applications.

Probably the most widely-used method for privacy-preserving databases is *perturbation* [4]. The perturbative masking method (a.k.a. randomization method) is a technique for privacy-preserving databases that uses data distortion in order to mask the attribute values of records. In this method, sufficiently large noise is added to individual record values to prevent recovery of those values by an adversary.

k-anonymity [5], *l-diversity* [6], and *t-closeness* [7] are other well-known privacy protection techniques that use methods such as generalization and suppression to reduce the granularity of data representation in order to keep the sensitive data private. The concept of k-anonymity was originally introduced in the context of relational data privacy [8] to address the following problem: "How can a data holder release its private data with guarantees that the individual subjects of the data cannot be identified while the data remain practically useful?" [5]. The l-diversity model was designed to address the weaknesses in the k-anonymity model when there is homogeneity of sensitive values within a group [6]. The t-closeness model is a further enhancement on the concept of l-diversity [7].

Differential privacy [9] is another emerging privacy-preserving paradigm that has recently gained considerable attention. Unlike the aforementioned privacy-preserving techniques that use generalization (i.e., k-anonymity, l-diversity, and t-closeness) to provide a *syntactic model*, differential privacy provides a *semantic* privacy model with strong protection guarantees. In other words, differential privacy is able to capture the amount of disclosure that occurs due to the publication of sensitive data in addition to mandating how the published data should look.

The vast majority of the existing literature on location privacy focuses on preserving the privacy of the users' location from an untrusted database (or service provider) in location-based services [10]. The location-based services rely on accurate, continuous, and real-time streams of the users' location data. However, if such information is mishandled by the database, location-based services pose a significant privacy risk to the users. Techniques for mitigating such a risk include sending a space- or time-obfuscated version of the users' actual locations [11], hiding some of the users' locations by using mix zones [12], sending fake queries, indistinguishable from real queries, issued from fake locations to the database [13], and applying k-anonymity to location privacy [14].

The Internet Engineering Task Force (IETF) is currently working on the development of a Protocol for Access to Whitespace Spectrum (PAWS) [15], that is used to request resources from the GDB. The IETF draft considers some security issues such as impersonation and man-in-the-middle attacks, that can be prevented using digital certificates and strong SU authentication, but it does not address the problem of preserving the operational privacy of PUs in the database.

In [16], a scheme called *PriSpectrum* is proposed that protects the SUs' location information in database-driven spectrum sharing. However, to the best of our knowledge, there is no existing work that addresses the problem of the PUs' operational privacy in the context of database-driven spectrum sharing.

III. Database Access Protocol

To facilitate the readers' understanding of the database inference attacks described in the next section, we provide a brief explanation of the database access protocol that serves as a reference model for the discussions henceforth.

We assume that the region that is serviced by a GDB is divided into an $m \times n$ grid of square cells, $c(i, j)$, where $1 \leq i \leq m$ is the row index and $1 \leq j \leq n$ is the column index. We assume that there are A primary users whose movements are confined to their cells (i.e., each PU can move in his/her cell, but cannot go from one cell to another one) and B secondary users, which may be either stationary or mobile users, within the region serviced by the database. Primary and secondary users are represented as PU_i and SU_j, respectively, where $1 \leq i \leq A$ and $1 \leq j \leq B$. Let P^{max} define the maximum transmission power that the database can assign a SU transmitter. There are a total of C channels, and the database may allow the SUs to access those channels if it determines that their access does not cause interference to the PUs. Before starting transmission, a secondary user, SU_i, sends a query, $Q = (ID_i, loc_i, AN_i)$, to the database, where ID_i is the unique identifier of SU_i, $loc_i = (x_i, y_i)$ are its location coordinates, and AN_i denotes antenna attribute information. The unique identifier, ID_i, of SU_i is acquired by a SU during the registration process, and it uniquely identifies each SU that accesses the database. Only registered users are serviced by the database. When a registered SU with the proper credentials queries the database, the database checks the spectrum availability of all of the C channels to determine which ones are available for the querier's use. The database's response is $R = ((ch_1, P_1, t_1), (ch_2, P_2, t_2), \cdots, (ch_k, P_k, t_k))$, where ch_i denotes the i-th channel, P_i denotes the maximum allowed transmission power in ch_i ($0 < P_i \leq P^{max}$), and t_i denotes the time interval of the channel's availability. Note that the database reply indicates the availability of one or more available channels, but not necessarily all of the non-occupied channels.

We also assume that the maximum transmission power (MTP) that is assigned to a SU by the GDB is calculated as $P = h(d)$, where P is the MTP, d is the distance between the SU and a PU receiver, and $h(\cdot)$ is a deterministic function referred to as the *MTP function*, which is prescribed by a regulatory agency such as the FCC. The $h(\cdot)$ function could utilize information such as transmission spectral mask of the SU, maximum allowed transmission power (P^{max}), terrain databases, and various incumbent interference protection ratios to compute MTP.

IV. Threats to the Incumbent Users' Operational Privacy

In this section, we describe database inference attacks. We show that SUs can employ these attacks to readily infer the PUs operational characteristics by using seemly innocuous database queries. Specifically, we describe and formulate attacks for inferring PUs' location, path of movement, speed, and times of operation.

A. Inference Attack against the Location of Stationary Incumbents

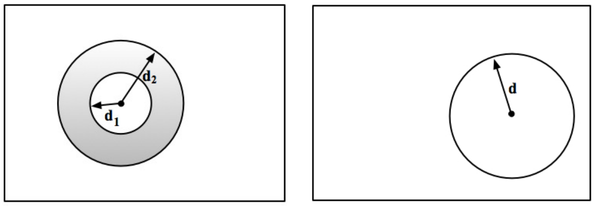
We assume that the attacker is a single mobile SU that can move throughout the region serviced by the database, and send queries to the database. Alternatively, we can assume that the attacker represents a group of colluding stationary SUs that are located throughout the region. The attacker's goal is to infer the location of the stationary PUs by analyzing the database's responses to its queries. We also assume that there is a non-negligible time gap between two queries of a SU such that the database has enough time to update its knowledge about the user before responding to another query from the same user.

Here, we describe how an attacker can infer the location of a stationary PU. The attacker merely uses a series of database query replies—which by themselves do *not* directly reveal the PU's location—to infer the PU's location within certain accuracy.

Assume that $X_{ij}^{(k)}$ is a Bernoulli random variable that is equal to 1 if a PU exists in cell $c(i,j)$ that operates on channel k, and is 0 otherwise. Let $p[X_{ij}^{(k)} = 1] = p_{ij}^{(k)}$ and $p[X_{ij}^{(k)} = 0] = 1 - p_{ij}^{(k)}$. The attacker uses the value, $p_{ij}^{(k)}$, to infer the location of the PUs. We assume that the attacker has no prior information about the presence of PUs in all the cells; in other words, $p_{ij}^{(k)} = \frac{1}{2}$ for all values of i, j and k. After each response of the database to the attacker's query, the attacker updates the value of $p_{ij}^{(k)}$ for a group of cells in the grid. If the value of $p_{ij}^{(k)}$ exceeds a threshold, δ, the attacker infers that there is a PU in cell $c(i,j)$ that operates on channel k. To update the values of $p_{ij}^{(k)}$, the attacker uses the procedure explained in the following paragraphs.

After querying the GDB and obtaining the corresponding responses, the attacker analyzes all of the C channels to extract information about the location of PUs. We assume that the database reply indicates the availability of one or more available channels, but not necessarily all of the non-occupied channels. For each channel ch_k, where $1 \leq k \leq C$, the query response indicates one of three possible constraints on its access:

- Case 1: Channel is unavailable. Channel ch_k is not in the query response's list of available channels. The unavailability of channel ch_k means that either there is a PU close to the SU's location and the channel is occupied, or it is fallow but the database did not include it in the list of available channels. In this case, the attacker does not update $p_{ij}^{(k)}$ values, because there is not enough information.



(a) Case 2: Channel is available but transmission power is limited

(b) Case 3: Channel is available and maximum possible transmission power is allowed

Fig. 1: Illustrations of Cases 2 and 3.

- Case 2: Channel is available but transmission power is limited. Access to channel ch_k is allowed under a limited transmission power constraint, i.e., $P_k < P^{max}$. Provided that the attacker has perfect knowledge of the MTP function, he knows that there is no PU within distance d_1 from the SU, and there exists at least one PU at a distance between d_1 and d_2 from the SU. This is illustrated in Figure 1(a). Therefore for all the cells between distance d_1 and d_2 from the querier, the attacker should increase the $p_{ij}^{(k)}$ values. Assuming that there are G cells between distance d_1 and d_2 from the querier, using the Bayes' rule, the attacker updates the $p_{ij}^{(k)}$ values using the equation
$$p_{ij}^{(k)} = \frac{p_{ij}^{(k)}}{1 - \frac{1}{G}(1 - p_{ij}^{(k)})}, \tag{1}$$
for these cells [17]. The attacker also sets $p_{ij}^{(k)} = 0$ for the cells within distance d_1 from the querier.
- Case 3: Channel is available. Channel ch_k is available and maximum transmission power is allowed, i.e., $P_k = P^{max}$. This type of reply indicates that there is no PU within distance d from the SU. However, this reply reveals no information about the possible presence of PUs beyond distance d. Figure 1(b) illustrates this case. The attacker sets $p_{ij}^{(k)} = 0$ for the cells within distance d from the querier and does not change the $p_{ij}^{(k)}$ values for other cells.

As described above, an attacker updates the values of $p_{ij}^{(k)}$ based on the query replies to estimate the location of PUs in the cell that are operating on a particular channel.

B. Path of Movement Inference for Mobile Incumbents

Using the same source of information (power values in the GDB's replies to the SUs' queries), SUs can infer the path of movement and the average speed of a mobile PU even when the query replies from the database do not directly reveal any location-related information. The problem of inferring a PU's path of movement can be interpreted as a *target-tracking* problem. In [17] we showed that using recursive Bayesian

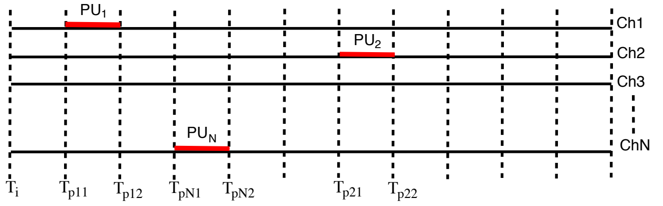


Fig. 2: An example of time of operation inference.

estimation (a.k.a. Bayes filter) [18], it is possible to track the mobile PUs using the information extracted from the GDB's query replies. Using a Bayes filter, a probability can be assigned to the attacker's belief about the current location of PUs in the grid, and that probability can be continuously updated from additional information that the attacker gathers from the database's responses to its queries.

C. *Time of Operation Inference*

An adversarial SU can send multiple queries to the database to infer a PU's time of operation. In GDB-driven spectrum sharing, an SU makes query Q_i to the database at any time T_i. If we assume the time interval between query and database response is negligible; then SU receives response R_i instantly at time T_i. Response R_i consists of (C_i, P_i, t_i), where C_i, P_i and t_i denote the available channel, maximum allowed transmission power in channel C_i and time duration for which querying SU can use channel C_i. Parameter t_i is an important component of the database response as it ensures that SU transmits only when PU is not operating in C_i. In general, t_i is the duration, starting from the query time, for which PU does not operate in channel C_i.

Consider a system with C channels and C PUs, each operating in different channels. PUs may or may not use the assigned channel all the time. In this model, we assume that the time duration of interest is divided into multiple slots of equal length. We also assume that PUs operate from the beginning of a particular time slot to the end of the same or a different slot. A SU can query only at the beginning of a time slot. If any channel is available, the database replies with the total duration of the time slots (starting from the requested slot) for which SU can transmit in that channel. We assume the channel selection process is i.i.d., where random variable C_i has uniformly distributed over all available channels for a given timeslot T_i. In Figure 2, a SU making a query at time T_i instantly receives a database reply with any of the available C channels. Here, probability of selecting a particular channel among the available ones is $1/C$. Similarly, probability of selecting a particular channel at time T_{p11}, where there are $(C-1)$ available channels, is $1/(C-1)$.

Let us define a Bernoulli random variable ${X_i}^{(k)}$ such that:

$$
{X_i}^{(k)} = \begin{cases} 1, & \text{if PU is operating on channel } C_i \text{ in timeslot } T_k \\ 0, & \text{otherwise} \end{cases}
$$

Let $P[{X_i}^{(k)} = 1] = {p_i}^{(k)}$ and $P[{X_i}^{(k)} = 0] = (1 - {p_i}^{(k)})$. From the attacker's perspective, he does not have prior knowledge about the presence/absence of a PU in a particular timesslot. So, he begins the inference attack by setting ${p_i}^{(k)} = \frac{1}{2}$ for all values of i and k. He then uses a series of database responses (to queries made at time T_k), and updates these probabilities. When the value of ${p_i}^{(k)}$ reaches a threshold, δ, the attacker infers that PU is operating on channel C_i at timeslot T_k. The attacker uses the following procedure to update the values of ${p_i}^{(k)}$.

- Case 1: Channel is available. In the GDB, a channel is available to a SU if and only if PU is not using it. If the database responds to a query made at time T_k with a channel C_i, then the attacker is certain that there is no PU operating in that timeslot.
- Case 2: Channel is unavailable. Suppose the database responds to a query made at time T_k with a channel C_i. The unavailability of channels $C_j, j \in (1, 2, \ldots, C), j \neq i$ can be explained as follows: i) C_j is not available because it is being used by PU at that time. ii) C_j is not available because there were more than one channels available at that time, and GDB chose one of them but not C_j.

Assuming that the channel i was available at timeslot T_k for n queries, and was unavailable for m queries, the probability $p_i^{(k)}$ can be updated using the following equation:

$$
p_i^{(k)} = \frac{m+1}{m+n+2}. \tag{2}
$$

Equation 2 is the direct result of the rule of succession which states if we repeat an experiment that we know can result in a success or failure, N times independently, and get S successes, then the probability that the next repetition will succeed is $\frac{S+1}{N+2}$ [19].

The attacker uses the above procedure, and infers the presence of PUs in timeslot T_k in those channels which have high probability values. This algorithm is implemented in each of the timeslots to infer the start and end time of a PU's time of operation.

V. Countermeasures against the Privacy Threats

In this section, we describe privacy-preserving techniques to propose countermeasures that can be used to counter the inference attacks introduced in Section IV.

A. *Techniques for Preserving Location Privacy*

We propose four privacy-preserving techniques for protecting the location privacy of PUs. Note that since PUs' path of movement is computed directly from their location information, the techniques described below can also be used to protect their movement information privacy.

1) Perturbation with Additive Noise: The perturbative masking method (a.k.a randomization method) is a technique for privacy-preserving databases that uses data distortion in order to mask the attribute values of records. In this method, we add sufficiently large noise to individual record values to prevent recovery of these values by an adversary [20]. One key advantage of the randomization method is that it is relatively simple, and does not require knowledge of the distribution of other records in the data. Additive noise is the most basic

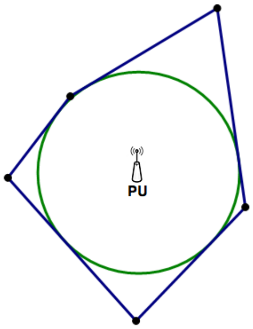


Fig. 3: An example of perturbation with transfiguration.

perturbative method that can be used for privacy-preserving databases. In this technique, the vector of observations x_j for the j-th attribute of the original dataset X_j is replaced by a vector $z_j = x_j + \epsilon_j$, where ϵ_j is the random noise vector. We can use this technique to obfuscate the information that the attacker can infer from the database's reply to its queries.

Recall that the database's response to each SU query is a set of values indicating the maximum allowed transmission power for all the channels available in a given area, i.e., $\{(ch_k, P_k, t_k)\}$, where P_k denotes the maximum transmission power for channel ch_k and t_k represents the time interval of the channel's availability. To preserve the privacy of PU's location using perturbation with additive noise, the database can provide a *false positive* response to a query. This false positive response corresponds to adding a negative random noise value ϵ_k to P_k and replacing P_k with $P'_k = P_k + \epsilon_k$. Note that we must use a negative noise to have $P'_k \leq P_k$ for each k, because replacing P_k with P'_k such that $P'_k > P_k$ (i.e., allowing the SU to transmit with a power higher than the original maximum allowed transmission power) will likely result in interference to PUs. Also note that adding negative noise to the maximum transmission power, P_k, is equivalent to increasing the radius of the PU's protected contour.

2) Perturbation with Transfiguration: Another form of perturbation is to change the shape of the protected contour. Replacing the circular or non-circular protected contours with random shapes that envelop the actual protected contour will increase the location privacy of PUs. Figure 3 illustrates an example where a circular protected contour is inscribed in a pentagon-shaped protected contour. The circular contour is the actual contour, and the pentagon-shaped contour is the transfigured contour.

We propose an algorithm for preserving the location privacy of PUs using transfiguration of the protected contour. This algorithm replaces the protected contour of a PU with an irregular N-sided polygon. The value N controls the level of privacy, i.e. a smaller value for N guarantees a higher level of privacy but reduces efficiency of spectrum utilization by the SUs. As $N \to \infty$, the irregular polygon approaches the original shape of the protected contour.

Algorithm 1 outputs coordinates for a set of N points that form a convex protected contour that envelops the original circular protected contour. The algorithm starts by dividing the circular protected contour into N arcs, and then chooses one random point on each of these arcs. The points are chosen in such a manner as to guarantee that the tangent lines at the points form an N-sided polygon that circumscribes the original circular protected contour.

Algorithm 1 An algorithm for perturbation with transfiguration

Input: Circular protected contour C_u of primary user u, number of polygon sides N.

Output: Coordinates of vertices of an N-sided polygon.

```
Divide $C_u$ to $N$ arcs of equal size, i.e., each arc is $\frac{360}{N}$ degrees.
Choose a random point $q_i$ on each arc $i$.
for each $i$ do
    Compute $l_i$, the tangent line to $C_u$ at point $q_i$.
end for
for each $i$ do
    Compute $M_i$, the point of intersection between $l_i$ and $l_{i+1}$. ($l_{N+1} = l_1$)
end for
return the set of coordinates $\{M_i\}$.
```

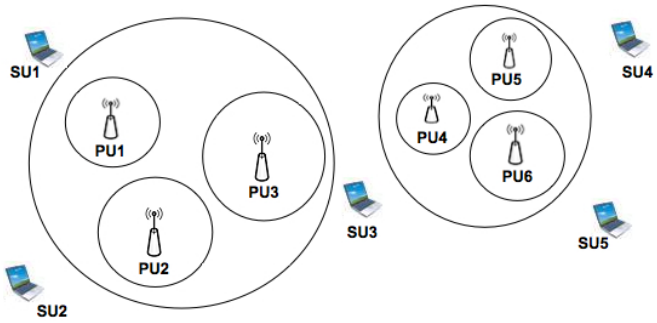


Fig. 4: 3-anonymity for PUs' location privacy.

3) k-Anonymity: The concept of k-anonymity was originally introduced in the context of relational data privacy [21]. The motivating factor behind the k-anonymity model was the possibility of indirect identification of records from public databases via *quasi-identifiers* i.e., combination of multiple record attributes that can be used to identify individual records [22]. For instance, a medical institution may want to release a table of medical records with the names of the individuals replaced with dummy identifiers. However, some set of attributes (which are referred to as the *quasi-identifiers*) can still lead to identity breaches. In the k-anonymity method, the granularity of data representation is reduced with the use of techniques such as generalization or suppression.

In the context of location privacy of PUs in database-driven spectrum sharing, we can achieve location k-anonymity by combining protected contours of k PUs that are closest together, and creating a larger protected contour that works like the cloak box for Location-Based Services (LBSs) [23]. The SUs are not allowed to transmit in the area covered by this larger protected contour. Figure 4 illustrates this idea for $k = 3$.

Note that although 3-anonymity by spatial cloaking in Figure 4 improves the location privacy of the PUs, it results in lower spectrum utilization efficiency, because the new protected contours formed by 3-anonymity is larger than the sum of the areas of the original protected contours.

It is known that finding an optimal solution to the k-anonymity problem, even under simplifying assumptions, is NP-hard [24]. Therefore, we focus on a heuristic-based approach that provides us with a good approximation to the optimal solution.

Here, we propose an algorithm for classifying n PUs into $g = \lceil \frac{n}{k} \rceil$ groups of k PUs, such that the k PUs in each group are the k closest neighbors. After grouping the PUs, the algorithm replace each group $G = \{g_1, g_2, \cdots, g_k\}$ with a virtual primary user Q that minimizes the equation:

$$\max_{1 \leq i \leq k} (d(g_i, Q) + R_i), \tag{3}$$

where $d(A, B)$ is the Euclidean distance between A and B, and R_i is the radius of the protected contour of primary user i. Then, the algorithm sets the radius of the protected contour of Q as:

$$R_Q = \max_{1 \leq i \leq k} (d(g_i, Q) + R_i). \tag{4}$$

In other words, the algorithm replaces the group of k PUs in each group with one virtual PU such that its protected contour is the smallest possible protected contour that envelops the protected contour of all the k PUs. Algorithm 2 describes the algorithm in detail.

Algorithm 2 k-anonymity algorithm for location privacy

```
Input: Set of PUs $U = \{u_1, u_2, \cdots, u_n\}$ and the radius values of their corresponding protected contours $\{R_1, R_2, \cdots, R_n\}$.
Output: Groups of $k$ closest PUs.
while $U$ is nonempty do
    if $|U| \leq k$ then
        Put all members of $U$ in group $G$.
        Remove members of $G$ from $U$.
    else
        Choose one member $u_i$ of $U$ randomly.
        Compute its distance from all other members of $U$.
        Choose the $k-1$ members closest to $u_i$, and put them along with $u_i$ in group $G$.
        Remove the $k$ members of $G$ from $U$.
    end if
    for each group $G = \{g_1, g_2, \cdots, g_k\}$ do
        Find point $Q$ that minimizes Equation 3.
        Compute the protected contour radius $R_Q$ using Equation 4.
        return $Q$ and $R_Q$.
    end for
end while
```

The complexity of the first part of Algorithm 2 (grouping the n PUs into groups of k PUs) is:

$$O(n + (n-k) + (n-2k) + \cdots + (n - \frac{n}{k}k)) = O(\frac{n^2}{k})$$

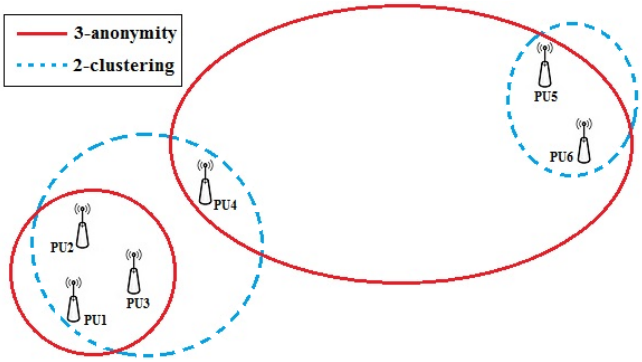


Fig. 5: Using 2-clustering to address the limitation of 3-anonymity.

The complexity of finding a point Q that minimizes equation 3 is $O(k \log k)$, so the complexity of the second part of the algorithm is $\frac{n}{k} \times O(k \log k) = O(n \log k)$. Therefore the total complexity of this algorithm is $O(\frac{n^2}{k} + n \log k)$.

Although k-anonymity can enhance location privacy, it may cause the spectrum utilization efficiency to drop significantly. Figure 5 illustrates this limitation when using 3-anonymity to improve the location privacy of six PUs. We observe that PU4 is much closer to PU1, PU2, and PU3, but since 3-anonymity forces the database to divide PUs into groups of size 3, the algorithm groups PU4 together with PU5 and PU6 into one group. The large protected contour that envelops these three PUs prohibits SU access to spectrum over a needlessly large area, and thus lowers spectrum utilization efficiency significantly. To address this limitation of k-anonymity, we propose another privacy-preserving technique in the next section.

4) k-Clustering: In this section we propose another technique for improving location-privacy that solves the aforementioned problem of k-anonymity. In this technique, instead of classifying PUs into groups of k users, we group them into k clusters. Figure 5 illustrates how applying 2-clustering algorithm will solve the problem of 3-anonymity for a specific orientation of PUs.

Instead of dividing PUs into groups of equal size, Algorithm 3 divides them into k clusters of PUs that are closest to each other. These clusters may not be of equal size. The complexity of Algorithm 3 (grouping the n PUs into k clusters of PUs) is $O(n^2)$.

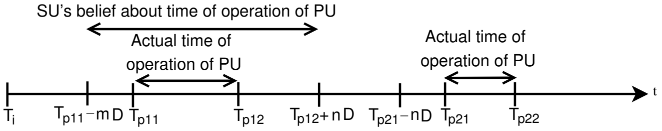
The main disadvantage of k-clustering is that some PUs that are distant from other PUs may end up in a single cluster by itself. In other words, this method may provide unequal levels of location privacy to different PUs. This shortcoming can be addressed by using a hybrid approach that combines k-clustering with other privacy-preserving techniques such as perturbation with additive noise or transfiguration of protected contour.

B. Countermeasures against Time of Operation Inference

In this section, we propose two privacy-preserving techniques for time of operation privacy.

Algorithm 3 k-clustering Algorithm for Preserving Location Privacy

```
Input: Set of Primary users $U = \{u_1, u_2, \cdots, u_n\}$ and their corresponding radius of protected contour $\{R_1, R_2, \cdots, R_n\}$.
Output: $k$ clusters of PUs.
Compute the distance $d_{ij}$ between all pairs of PUs $u_i$ and $u_j$.
Put $d_{ij}$ values into a sorted array D.
Put each $u_i$ into a single cluster.
while (number of clusters) $> k$ do
    Choose the smallest value $d_{ij}$ from array D.
    Combine clusters of $u_i$ and $u_j$.
end while
for each cluster $G = \{g_1, g_2, \cdots, g_m\}$ do
    Find point $Q$ that minimizes Equation 3.
    Compute the protected contour radius $R_Q$ using Equation 4.
    return $Q$ and $R_Q$.
end for
```

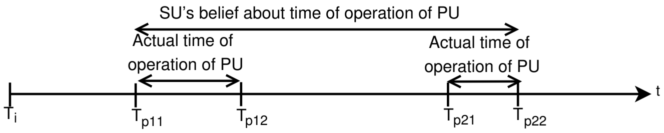


SU's belief about time of operation of PU
Actual time of operation of PU
Actual time of operation of PU
t
T_i
$T_{p11}-mD$
T_{p11}
T_{p12}
$T_{p12}+nD$
$T_{p21}-nD$
T_{p21}
T_{p22}

Fig. 6: Countering Time of Operation Inference using Buffer TimeSlots

1) Buffer Time Slots: This technique adds random buffer slots to the time of operation, t_i, to insert false positives in the attacker's query responses. In other words, this technique is a variation of perturbation with additive noise. Using this technique, the database always responds to a query with time duration that is smaller than or equal to the actual available time duration for a particular channel. In Figure 6, PU is active from time T_{p11} to T_{p12} and again from T_{p21} to T_{p22}. A query made at time T_i such that ($T_i \leq T_{p11}$) receives a response with time of operation from T_i to ($T_{p11} - mD$); $0 \leq mD \leq (T_{p11} - T_i)$, where m represents random number of time slots, and D is the time duration of each slot. Similarly, a query made at time T_{p12} receives a response with time of operation from ($T_{p12} + nD$) to ($T_{p21} - nD$) such that $0 \leq nD \leq (T_{p21} - T_{p12})$ where n represents random number of time slots. With this information, attacker believes that a PU is operating between time ($T_{p11} - mD$) and ($T_{p12} + nD$). This is a correct inference but the accuracy of the inference has been lowered due to the addition of the buffer slots. This countermeasure is applied to each channel to thwart an adversary's attempts to infer a PU's time of operation. The spectrum utilization efficiency of the system is negatively impacted by adding buffer time slots.

2) k-Anonymity: False positives can be added to a query response in another way by using k-anonymity. When this technique is employed, the database groups a PU's k continuous time intervals of operation into a single interval. In the example shown in Figure 7, a PU is active from time T_{p11} to T_{p12} and again from T_{p21} to T_{p22} in the same channel. The database implementing k-anonymity, k = 2, groups these two intervals into one interval, from T_{p11} to T_{p22}, and responds to queries accordingly.



SU's belief about time of operation of PU
Actual time of operation of PU
Actual time of operation of PU
t
T_i
T_{p11}
T_{p12}
T_{p21}
T_{p22}

Fig. 7: Countering Time of Operation Inference Using k-Anonymity

VI. Metrics

All of our proposed privacy-preserving techniques improve the operational privacy of PUs at the cost of reducing the spectrum utilization efficiency of the secondary network. In this section we introduce a number of metrics for quantifying operational privacy and spectrum utilization efficiency. These metrics can be used to evaluate the impact of the inference attacks and the effectiveness of the proposed countermeasures.

A. Metrics for Location Privacy

In [25], the authors discussed three general metrics for evaluating location inference attacks, namely *uncertainty*, *inaccuracy*, and *incorrectness*. In this section, we redefine these metrics in the context of our problem so that they are appropriate for evaluating the PUs' location privacy.

Assume that $Y_{ij}^{(k)}$ denotes the database's knowledge about the presence of a PU that operates on channel k in cell $c(i, j)$. $Y_{ij}^{(k)}$ is a deterministic function that is equal to 1 if a PU exists in cell $c(i, j)$ that operates on channel k, and is 0 otherwise.

Recall that $X_{ij}^{(k)}$ is a Bernoulli random variable that is equal to 1 if a PU exists in cell $c(i, j)$ that operates on channel k, and is 0 otherwise. This random variable represents the attacker's estimation of $Y_{ij}^{(k)}$. In section IV, we described how an attacker can use $p_{ij}^{(k)}$ to infer the location of the PUs. Using the aforementioned notations, we define three different metrics for location privacy.

1) Uncertainty: Suppose o denotes the observed sensory information (i.e., database's reply to a query), and assume that the information that an attacker extracts from a query response is in the form of $p(X_{ij}^{(k)} | o)$, which is a probability distribution for the possible values of the PU's location given the observed information. *Uncertainty* is the ambiguity of this posterior distribution with respect to finding a unique answer (note that a unique answer need not be the correct one). We employ the concept of entropy to define $h_{ij}^{(k)}$, which is the attacker's uncertainty about the presence of a PU that operates on channel k in cell $c(i, j)$ as follows:

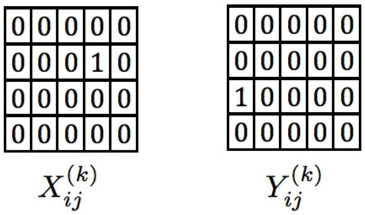
$$h_{ij}^{(k)} = -p_{ij}^{(k)} \log(p_{ij}^{(k)}) - (1 - p_{ij}^{(k)}) \log(1 - p_{ij}^{(k)}). \quad (5)$$

Therefore, the total uncertainty about the location of PUs that are operating on channel k is:

$$H^{(k)} = \sum_{i=1}^{M} \sum_{j=1}^{N} h_{ij}^{(k)}, \quad (6)$$

The attacker's uncertainty is maximized when the attacker conjectures that the aforementioned posterior distribution is a uniform distribution.

We contend that uncertainty, which is a widely used metric for measuring privacy, is not suitable for location privacy of PUs. The justification for this conclusion is that the metric does not accurately quantify the privacy level when the attacker has a high level of certainty about an incorrect distribution. Figure 8 illustrates such an example. The right figure shows the true distribution of the PUs, and the left figure shows the attackers estimation. Note that the attacker's uncertainty is zero, but this does not mean that the location privacy of the PU is threatened, because the attacker's estimation is wrong.



0	0	0	0	0
0	0	0	1	0
0	0	0	0	0
0	0	0	0	0

$X_{ij}^{(k)}$

0	0	0	0	0
0	0	0	0	0
1	0	0	0	0
0	0	0	0	0

$Y_{ij}^{(k)}$

Fig. 8: An example showing a disadvantage of using the uncertainty metric.

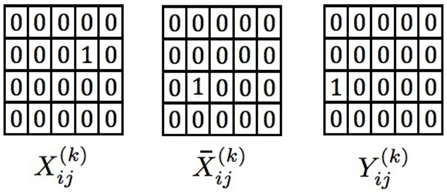
2) Inaccuracy: Because the attacker does not have infinite resources, the result of a location inference attack is only an estimate, $p(X_{ij}^{(k)}|o)$, of the PUs' location distribution, $Y_{ij}^{(k)}$. *Inaccuracy* is the discrepancy between the distributions $p(X_{ij}^{(k)}|o)$ and $Y_{ij}^{(k)}$.

Here we define inaccuracy as:

$$IA^{(k)} = \sum_{i=1}^{M}\sum_{j=1}^{N}(p_{ij}^{(k)} - Y_{ij}^{(k)})^2. \tag{7}$$

Inaccuracy quantifies the difference between the attacker's estimation of the PUs' location distribution from the PUs' real location distribution.

However, inaccuracy has a limitation in measuring the location privacy of PUs. Figure 9 illustrates this limitation. Assume that the $Y_{ij}^{(k)}$ is the real distribution of PUs, and that $X_{ij}^{(k)}$, and $\bar{X}_{ij}^{(k)}$ are two different estimations of $Y_{ij}^{(k)}$. The inaccuracy for both of these estimations is the same, but it is obvious that $\bar{X}_{ij}^{(k)}$ is a much better estimation because the estimated location of the PU is closer to the actual location.



0	0	0	0	0
0	0	0	1	0
0	0	0	0	0
0	0	0	0	0

$X_{ij}^{(k)}$

0	0	0	0	0
0	0	0	0	0
0	1	0	0	0
0	0	0	0	0

$\bar{X}_{ij}^{(k)}$

0	0	0	0	0
0	0	0	0	0
1	0	0	0	0
0	0	0	0	0

$Y_{ij}^{(k)}$

Fig. 9: An example showing a disadvantage of the inaccuracy metric.

3) Incorrectness: We showed that uncertainty and inaccuracy have limitations, and can serve as only indirect measures for location privacy. We define a different metric, referred to as *incorrectness*, which is more appropriate for the problem at hand. We define incorrectness as the expected distance between the location inferred by the attacker and the PU's true location, and is expressed as follows:

$$IC^{(k)} = \sum_{i=1}^{M}\sum_{j=1}^{N}p_{ij}^{(k)}d_{(i,j)}^{k}, \tag{8}$$

where $d_{(i,j)}^{(k)}$ is the distance between the cell $c(i,j)$ and the closest PU to it. Henceforth, we will use incorrectness as a metric for quantifying the location privacy of PUs.

B. Metrics for Time of Operation Privacy

An appropriate metric for measuring time of operation privacy is the one that can measure the correctness of inference results. Based on this, we define time of operation privacy of PU operating in channel C_i as follows:

$$\sum_{n=1}^{S}[p_i^{(n)}.(|\hat{t_s}^{(n)} - t_s^{(n)}| + |\hat{t_e}^{(n)} - t_e^{(n)}|)]$$

where $p_i^{(n)}$ is the probability that attacker uses to infer PU's operation in n^{th} session, $\hat{t_s}$ and $\hat{t_e}$ represent the estimated start and end times for PU's n^{th} active session in channel C_i, t_s and t_e represent the actual start and end times of the PU's operation in channel C_i, and S is the total number of active sessions of PU in channel C_i for a considered duration of time. A large value of this metric corresponds to better times of operation privacy.

C. Metric for Spectrum Utilization Efficiency

All the techniques that are described in section V-A for preserving location privacy of PUs enlarge the protected contour of PUs. Perturbation with additive noise decreases the available transmission power for SUs which is equivalent to increasing the radius of protected contour for PUs. Perturbation using transfiguration changes the shape of protected contour of PUs such that the new protected contour envelops the original protected contour, in other words this method increases the coverage area of a protected contour. Finally k-anonymity and k-clustering, combine the protected contour of multiple PUs to a large protected contour that not only envelops the protected contour of the PUs, but also includes part of area that is not covered by protected contour of any PUs.

Increasing the PUs protected contour (i.e. decreasing SUs network's coverage area) cause a reduction in SU network capacity which is also our metric for spectral utilization efficiency. SU capacity is an appropriate metric for quantifying the spectrum utilization efficiency with and without the countermeasures against the PU parameter inference attacks. Both SU capacity and PU privacy depend on the transmit power allocated by the database to the SUs. The area averaged SU capacity increases when higher transmit powers are allocated to SU by the database. While the PU privacy is enhanced when the SU assigned transmit power is randomized in a known way from actual values by the privacy enhancing techniques.

Consider a scenario as mentioned in the section III. The goal of the analysis is to estimate the relationship between spatial spectrum utilization and PU privacy. We define spectrum utilization as the capacity that the SUs can achieve by reusing the PU bands outside its protection contour zones. The SU's capacity depends on its transmit power and the signal to interference and noise ratio (SINR) that it can achieve with it.

A distance dependent maximum transmit power (MTP) allocation function [2] [26] [27] is used to calculate the maximum allowable transmit power of the SU.

$$P_{ts} = \begin{cases} 0 \quad \text{mWatt} & \text{for } d < 8km \\ 1/2P_{max} \ \text{mWatt} & \text{for } 8km < d < 14km \\ 3/4P_{max} \ \text{mWatt} & \text{for } 14km < d < 25km \\ \ P_{max} \ \text{mWatt} & \text{for } 25km < d, \end{cases}$$

where, P_{ts} is the SU transmit power, d is the distance between the primary and SUs and the value of P_{max} is the maximum SU transmit power in the channel.

From the MTP function, the SU's SINR ρ_s is calculated using the following equation [28], which determines the secondary achievable data rates.

$$\rho_s = \frac{P_{ts}/L_s(r_{cell})}{n_s W_s + I_{P2S} + I_{S2S}} \tag{9}$$

where, $L_s(r_{cell})$ is the path loss between the SU transmitter and receiver in the cell, P_{ts} is the SU transmit power, n_s is the background noise power spectral density at the secondary receiver, W_s is the secondary bandwidth per user, I_{P2S} is the primary to secondary interference, and I_{S2S} is the secondary to secondary interference.

The capacity achieved by a transmitter and receiver averaged over the area [28] in a single frequency is given by

$$C_{cell} = Pr(ch)W_s \int_A \frac{log_2(1+\rho_s(loc_i))}{A} dloc_i \tag{10}$$

where, $loc_i = (x_i, y_i)$ is the location of the SU and ρ_s is the SU SINR given by equation 9 and $Pr(ch)$ is the channel availability probability, depending on co-channel and adjacent channel protection area. The capacity can be calculated for each channel separately and the overall capacity of a cell that is exploiting all available channels is obtained by summing over all the channels as indicated by the following equation.

$$C_{celltotal} = \sum_{ch_{1,2,..,k}} Pr(ch)W_s \int_A \frac{log_2(1+\rho_s(loc_i))}{A} dloc_i \tag{11}$$

where $C_{celltotal}$ is the total capacity achieved with all frequencies available at the SU location loc_i.

Equation 11, gives the area averaged spectrum utilization of the SUs. The SU spectrum utilization improves when the protection area of the PU is reduced or the SU can transmit with higher transmit power. In this metric, only spatial aspect is considered, the times of operation is not taken into account.

VII. Simulation Results

In this section, we present simulation results showing the performance of the proposed privacy preserving techniques. In the first half of this section, we compare, for all countermeasures, the effect of number of query responses on the privacy of PUs. In the second half, we show the trade-off that exists between privacy of PUs and spectrum utilization efficiency.

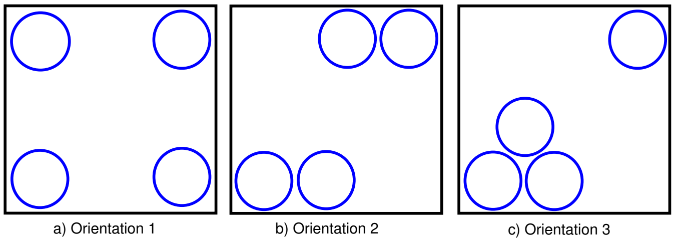


Fig. 10: Different Orientations of PUs

To study the effect of number of queries on the privacy of PUs for different countermeasures, let us define the database coverage area as a 300 km by 300 km square which is divided into 100 by 100 square grids, where the side of each square grid is 0.5 km long. There are 3 channels in the system, and 12 non-mobile PUs are assumed to be operating in these channels: 4 PUs in each channel. All PUs are assumed to have the same MTP function, and have a circular protected contour with an outermost radius of 25 km. Outside the protected contour, the SUs can transmit with maximum power of P_{max} = 1 Watt. PUs operating in the same channel are sufficiently separated from each other such that their protected contours do not overlap, and they do not interfere each other. We consider the scenario, as shown in Figure 10(c), where 3 PUs operating in the same channel form a cluster, and the remaining PU is separated from the cluster by a large distance. In the figure, circles represent the outermost protection contour of a PU located at its center.

We assume that an attacker performs a location inference attack by querying the GDB from the centers of multiple random grids. It is assumed that the attacker keeps track of its query locations and, for non-mobile PUs, does not make multiple queries from the same grid. In response to each query, the database replies with one channel and the maximum available transmit power in that channel. Based on the database response, the attacker updates its inference probabilities after each query response. After collecting Q query responses, the attacker makes an inference (based on its updated probability values for each grid) about the location of PUs in each channel. The privacy of PUs, i.e., the incorrectness of attacker's estimate of PUs location, is calculated and averaged over 3 channels. These experiments are performed for different values of Q from 0 to 500. For each value of Q, 20 attacks are made from different location sets, and average privacy is calculated. The final privacy values are normalized before plotting.

For perturbation with transfiguration, we assume that the circular protected contours of PUs are transfigured to equilateral triangles. For perturbation with noise, it is assumed that database replies 75% of the queries with false positives.

Figure 11 compares the privacy of PUs versus number of query responses for different countermeasures. It can be seen that the privacy of PUs decreases monotonically with the increase in number of query responses. Perturbation with

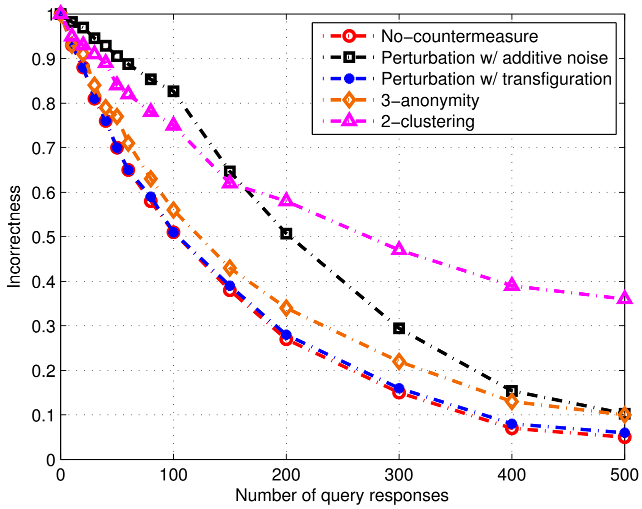


Fig. 11: Privacy versus number of query responses.

transfiguration exhibits very little privacy improvement because of the fact that database includes small number of false positives compared to the other countermeasures. For the first few queries, the rate at which privacy decreases for perturbation with transfiguration is almost same as the case without any countermeasure but a difference is seen when the number of query responses increase. For perturbation with additive noise, privacy decreases at the slowest rate among all countermeasures for the first few queries. It is obvious because the number of false positives increase with the increase in query responses, and it contributes to creating confusion to the attacker in correctly inferring PUs' location. However, when the number of query responses is increased beyond a certain number, this rate declines sharply. It is because increasing the number of query responses also increases the number of true replies, which in turn counteract the effect of false positives. Note that for perturbation with noise, increasing the number of false positives means losing more spectrum, but that doesn't help to preserve the privacy of PUs if the number of query response exceeds a certain limit.

From Figure 11, it can be seen that 3-anonymity and 2-clustering perform better than transfiguration for any number of queries. They also perform better than perturbation with noise when the number of query responses is large. From the slope of the privacy plots, it can be noted that all countermeasures except perturbation with additive noise exhibit better privacy performance for any number of queries. Perturbation with additive noise performs similar to the case without any countermeasure if the number of query responses is very large– i.e., when the true positive replies dominate and counteract the effect of all false positive replies.

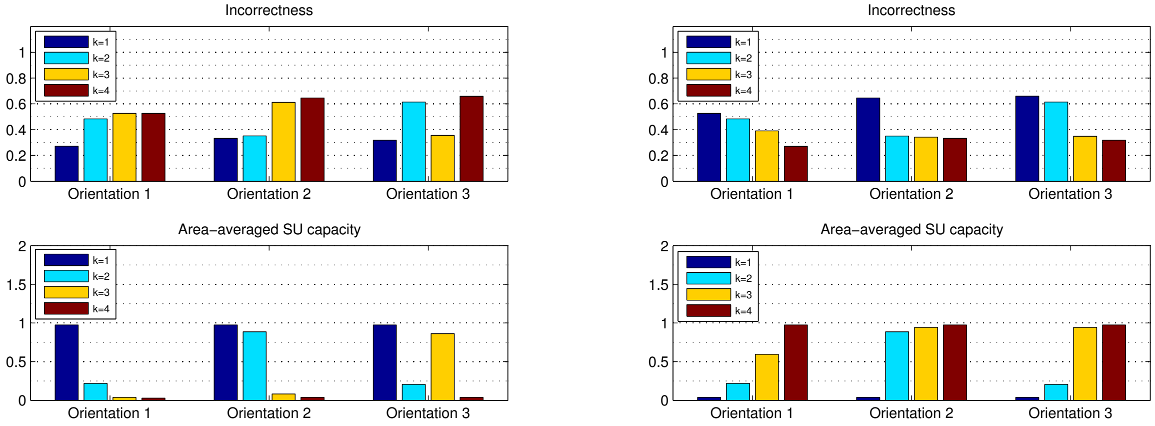
For the second half of this section, we divide our experiments into two sets to study the trade-off between privacy of PUs and spectrum utilization efficiency.

In the first set of experiments, we are interested in comparing the performance of the two privacy-preserving techniques that use grouping of PUs to protect their privacy – i.e., k-anonymity and k-clustering. We evaluated the two approaches for different values of k and various *PU orientations*. Here, PU orientation refers to the positions of the PUs within the confines of a square database coverage area (see Figure 10). All assumptions are same as that made in the first half except the following. The database coverage area is defined as a 300 km by 300 km square, and is divided into 100 by 100 square grids, where the side of each square grid is 3 km long. We consider 3 scenarios as illustrated in Figure 10, where a circle represents the outermost protected contour of a PU located at its center. In the first scenario, all PUs in same channel are located far apart from each other at the four corners of the square area. Second orientation considers the case where the 4 PUs operating in the same channel form two clusters such that each cluster is composed of two PUs located adjacent to each other and the distance between the 2 clusters is large. In the third scenario, 3 PUs operating in the same channel form a cluster, and the remaining PU is separated from the cluster by a large distance. The attacker, after collecting 100 query responses, makes an inference (based on its updated probability values for each grid) about the location of PUs in each channel.

We study the effect of k-anonymity and k-clustering for each orientation separately. For each orientation, we apply these two countermeasures with four different values of k ranging from 1 to 4. For each value of k, 20 attacks are made from different location sets. After each attack, the privacy of PUs in each channel and area-averaged SU capacity is calculated. These values are averaged over 20 such attacks. Finally, the privacy and the spectrum utilization values are averaged over 3 channels and normalized before plotting.

Figures 12(a) and 12(b) show the performance of k-anonymity and k-clustering for various values of k and the three PU orientations, respectively. In the figures, k=1 denotes a scenario in which no privacy-preserving technique is employed. Note that in the case of PU orientation 1, k-anonymity monotonically increases the level of privacy as k is increased from 1 to 2, 3 and 4. This is due to the fact that there is sufficient separation between the protection contours of the PUs operating in the same channel, and as a result, the number of false positives in the GDB's responses increases monotonically as k is increased. In contrast, for PU orientation 1, k-anonymity causes the spectrum utilization to degrade sharply as k is increased. Figures 12(a) and 12(b) also show that k-clustering performs exactly opposite to k-anonymity: k-clustering monotonically decreases the level of privacy as k is increased from 1 to 2, 3 and 4, but spectrum utilization increases rapidly. None of the techniques seem to exhibit a good trade-off between the privacy of PUs and spectrum utilization efficiency when PUs locations are similar to orientation 1 because of sharp decline in either privacy or spectrum utilization.

Orientation 2 inherently favors privacy of PUs. This can be seen by comparing the privacy performance of 1-anonymity of orientation 2 with that of orientation 1. For orientation 2, the performance of k-anonymity for k = 1, 3, or 4 is almost similar to that in orientation 1. For these values of k, privacy is improved by heavily penalizing the spectrum



(a) k-anonymity.

(b) k-clustering.

Fig. 12: Performance results for k-anonymity and k-clustering.

utilization and vice-versa. So, k-anonymity with k = 1, 3 and 4 are unacceptable approaches for PU orientation 2. However, the performance of 2-anonymity in orientation 2 is drastically different from that of orientation 1 – it exhibits improved privacy of PUs by degrading of spectrum utilization by a small amount. Similarly, for k-clustering, k = 2 and 3 show similar performance. Thus, we can conclude that for PU orientation 2, k-anonymity with k = 2, and k-clustering with k = 2 and 3 improve the privacy of PUs without sacrificing significant spectrum utilization.

For PU orientation 3, k-anonymity when k = 3 causes a dramatic increase in spectrum utilization efficiency compared to k = 2 and k = 4. Moreover, the opposite is true when looking at the privacy levels – i.e., privacy level when k = 3 is much lower than those when k = 2 or k = 4. From plots for orientations 1 and 2, we can see that k-anonymity with k = 3 has very small spectrum utilization values which is not the case in orientation 3. Also, the difference between k-clustering and k-anonymity is more noticeable for this orientation. We observe that 3-clustering outperforms k-anonymity with all k values in terms of making an optimal balance between privacy and spectrum utilization. k-clustering with k = 3 exhibits better performance in terms of spectrum utilization for almost same level of privacy as that in 3-anonymity.

From the first set of experiments, we can conclude that the performance of k-anonymity and k-clustering depends on the orientation of PUs. GDB should carefully choose the k values in order to obtain a balanced trade-off between privacy of PUs and spectrum utilization efficiency.

In the second set of experiments, we compared the performance of perturbation with additive noise and perturbation with transfiguration techniques. These two privacy-preserving techniques share a common trait – the technique is applied to the protection contour of an individual PU, not multiple PUs as was the case with k-anonymity and k-clustering. The database coverage area is defined as a 100 km by 100 km square area, and is divided into 200 x 200 square grids where the length of each grid's side is 0.5 km. There are 3 channels in the system. Since perturbation techniques can be applied to each PU individually, it is reasonable to analyze the performance by considering 1 PU in each channel; performance will be similar when there are multiple PUs in a channel. All PUs are assumed to have the same MTP function, and have a circular protected contour with an outermost radius of 25 km. SUs that are outside the outermost protection contour are allowed to transmit with maximum power of P_{max} = 1 Watt. To ensure that all vertices of the transfigured polygon lie within the database coverage area, we assume that PUs are located at the center of the database coverage area. For the attacker, we make the same assumptions as we did for performing the first set of simulations (grouping techniques).

Figure 13 compares the performance of the two perturbation based privacy-preserving techniques. For both these countermeasures, a key parameter is varied to study its effect on privacy and spectrum utilization efficiency. For perturbation with transfiguration, this parameter is the number of sides, N, of the polygon after transfiguring the protection contour of a PU. For our simulations, N values from 3 to 9 are used to transfigure the circular contour to a regular polygon. We can see from the figure that as we increase N (right to left in steps of 1), privacy decreases but spectrum utilization efficiency increases. This is because increasing N results in a closer approximation to a circular protection contour due to which fewer false positives are injected in the database responses. On the other hand, increasing N provides a greater area outside the transfigured protected contour and allows better usage of the spectrum by the SUs. It can also be seen that after N = 7, increasing N has no significant change in privacy and spectrum utilization values.

For plotting the curve of perturbation with additive noise, we varied the value of average noise level (NL) in database responses by varying the percentage of queries for which it replies with false positives. NL is an indirect measure of average capacity loss that SUs suffer on average, when noise is added in the database response. For our simulations, we vary NL such that the average SU capacity loss (CL)

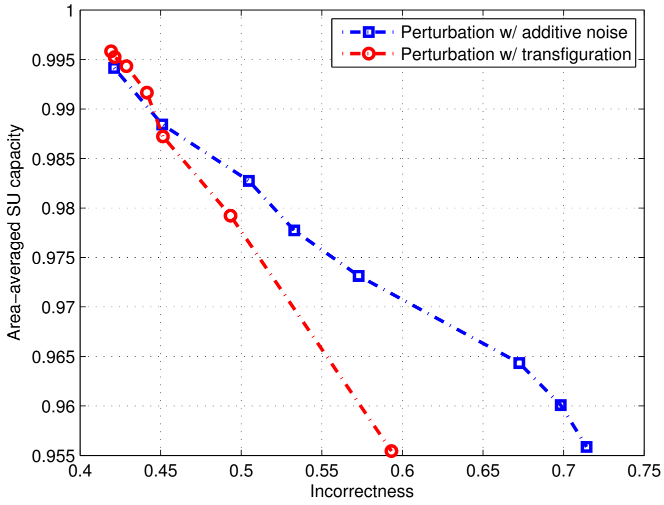


Fig. 13: Trade-off between spectrum utilization efficiency and privacy for two perturbation techniques. For perturbation with transfiguration, the red circles denote the performance for different values of N (number of sides of polygon) ranging from 3 to 9 (right to left). For perturbation with additive noise, the blue squares denote the performance for different values of area-averaged capacity loss ranging from 0.5% to 4.5% (left to right).

ranges from 0.5% i.e. less false positives to 4.5% i.e. more false positives (right to left in the figure). From Figure 13, we can see that the privacy is improved with an increase in NL because it directly results in increased number of false positives injected in the database responses. On the other hand, spectrum utilization efficiency decreases. We can also notice that perturbation with additive noise outperforms perturbation with transfiguration. From Figure 13, we can clearly see the negative correlation between the privacy level and the spectrum utilization efficiency which is inherent to both perturbation techniques.

VIII. Conclusions

In this paper, we introduced inference attacks against operational privacy of the primary users and proposed privacy-preserving techniques that the geolocation database can use to mitigate these attacks. The introduced inference attacks threaten the privacy of primary users' location, movement, and time of operation. To measure the effectiveness of privacy-preserving techniques, we developed metrics for quantifying the operational privacy of primary users. We showed that there exists a fundamental tradeoff between the operational privacy of PUs and the efficiency of the SUs' utilization of fallow spectrum—i.e., one cannot increase the former without sacrificing the latter and vice versa.

References

[1] FCC, "Third order and memorandum opinion and order, in the matter of unlicensed operation in the TV broadcast bands, additional spectrum for unlicensed devices below 900 MHz and in the 3 GHz band," Apr. 2012.

[2] FCC, "Enabling innovative small cell use in 3.5 GHZ band NPRM & order (FCC 12-148)," Dec. 2012.

[3] E. Bertino and S. Ravi, "Database security-concepts, approaches, and challenges," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 1, pp. 2–19, 2005.

[4] L. Li, M. Kantarcioglu, and B. Thuraisingham, "The applicability of the perturbation based privacy preserving data mining for real-world data," *Data & Knowledge Engineering*, vol. 65, no. 1, pp. 5–21, 2008.

[5] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[6] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," in *International Conference on Data Engineering (ICDE)*, 2006.

[7] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *Int. Conf. on Data Engineering (ICDE)*, 2007.

[8] P. Samarati, "Protecting respondents' identities in microdata release," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[9] C. Dwork, "Differential privacy," in *International Conference on Automata, Languages and Programming*, pp. 1–12, 2006.

[10] G. Ghinita, *Privacy for Location-based Services*. Morgan & Claypool Publishers, 2013.

[11] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *ACM MobiSys*, 2003.

[12] J. Freudiger, R. Shokri, and J. Hubaux, "On the optimal placement of mix zones," in *Int. Symp. on Privacy Enhancing Technologies*, 2009.

[13] R. Chow and P. Golle, "Faking contextual data for fun, profit, and privacy," in *WPES '09: Proceedings of the 8th ACM workshop on Privacy in the electronic society*, 2009.

[14] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Trans. Mobile Computing*, vol. 7, no. 1, pp. 1–18, 2008.

[15] B. Patil, "Protocol to access white space database: problem statement, use cases and requirements." [Online]. Available: http://tools.ietf.org/html/draft-ietf-paws-problem-stmt-usecases-rqmts-06, July 2012.

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

[17] B. Bahrak, *Ex Ante Approaches for Security, Privacy, and Enforcement in Spectrum Sharing*. PhD thesis, Virginia Tech, 2013.

[18] N. Bergman, *Recursive Bayesian Estimation*. PhD thesis, Department of Electrical Engineering, Linköping University, 1999.

[19] E. Jaynes, *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

[20] C. Aggarwal and S. Phillip, *Privacy-Preserving Data Mining: Models and Algorithms*, ch. A general survey of privacy-preserving data mining models and algorithms. Springer, 2008.

[21] P. Samarati, "Protecting respondents' identities in microdata release," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[22] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[23] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE transaction on mobile computing*, vol. 7, no. 1, 2008.

[24] G. Aggrawal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu, "Approximation algorithms for k-anonymity," *Journal of Privacy Technology*, 2005.

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

[26] "Evaluation of the 5350-5470 mhz and 5850-5925 mhz bands." http://www.ntia.doc.gov/report/2013/evaluation-5350-5470-mhz-and-5850-5925-mhz-bands. Online, accessed 06-25-2013.

[27] J. E. Carroll, F. H. Sanders, R. L. Sole, and G. A. Sanders, "Case study: Investigation of interference into 5 ghz weather radars from unlicensed national information infrastructure devices, part i," tech. rep., 2010.

[28] F. Hessar and S. Roy, "Capacity considerations for secondary networks in tv white space," *CoRR*, vol. abs/1304.1785, 2013.