Privacy-Utility Trades in Wireless Data via Optimization and Learning

Lillian Clark
University of Southern California
lilliamc@usc.edu

Matthew Clark
The Aerospace Corporation
matthew.a.clark@aero.org

Konstantinos Psounis
University of Southern California
kpsounis@usc.edu

Peter Kairouz
Google Research
kairouzp@stanford.edu

Abstract—**The multi-objective optimization problem which considers the competing goals of privacy and utility, e.g. users want location-based services but wish to keep their location secret, frames the design of data obfuscation functions. A theoretical treatment of this optimization over functional spaces is possible under certain assumptions, e.g. considering the convex privacy metric of mutual information between the true state and an adversary's estimate of that state, and treating obfuscation functions as conditional probabilities specifying obfuscated data given true data. When this analytical treatment of the problem is not feasible or desirable, data-driven approaches like adversarial learning can be used to design obfuscation functions. Generative adversarial privacy, which leverages recent advancements in generative adversarial networks, considers two players in a minimax game: a privatizer that obfuscates data to minimize the risk of inference attacks while preserving the utility of the true data, and an adversary that infers sensitive information from the obfuscated data. In this work we study both theoretical and data-driven methods of approaching privacy-utility trades and apply these methods in the context of spectrum sharing and signal mapping.**

I. Introduction

The collection of user data provides obvious benefits in many contexts, e.g. personalized content, accurate predictions, and coordinated resource sharing. However, these aggregated datasets raise legitimate privacy concerns. Obfuscation mechanisms, in which the true state of users is masked in some way, may be used to protect privacy at the expense of utility. In this paper we define formal metrics for privacy and utility and investigate the fundamental trade between the two.

We consider two approaches, the first of which is formal optimization. For the applications we consider, it is often the case that there are hard constraints on either privacy or utility, therefore we can maximize one subject to a constraint on the other. In general, such optimization is over functional spaces, and a theoretical treatment of the problem is possible by considering convex privacy metrics like the mutual information [1] between the true state and an adversary's estimate of that state, and by treating obfuscation functions as conditional probabilities specifying obfuscated data given true data.

When an analytical treatment of the problem is not possible, or in the presence of rich datasets, we opt to use data-driven approaches like adversarial learning. Specifically, we use generative adversarial privacy (GAP) [2], [3] which leverages recent advancements in generative adversarial networks (GANs) [4]–[6] and allows learning obfuscation/privatization schemes from the data itself. The framework considers two players in a minimax game: a privatizer that obfuscates data to minimize the risk of inference attacks while preserving the utility of the true data, and an adversary that infers sensitive information from the obfuscated data.

We study two real-world use cases in the context of wireless data. The first is spectrum sharing. The combination of bandwidth hungry wireless networks such as cellular, Wi-Fi and the internet of things, and the shortage of unencumbered spectrum for new services has caused the government to consider sharing underutilized spectrum among incumbent users and new users. Spectrum Access Systems (SAS) [7] which maintain databases of spectrum policy and use information, are opening bands used for military radar for access to new commercial services [8], [9], however incumbent systems will retain priority access. The SAS is expected to identify suitable protections to prevent harmful interference to priority/primary users (PUs) when granting spectrum access to secondary users (SUs). This raises privacy concerns [10], as an adversary may make inference attacks on PU information by passively observing the sharing system. PU privacy may be preserved by obfuscating the information reported to the SAS, e.g. radar locations or by obfuscating the allocations made by the SAS to SUs, e.g. power assignments to phones, at the expense of operating the shared system under more severe interference constraints and thus offering SUs less additional bandwidth.

The second use case is that of (cellular) signal map generation. Signal maps consist of measurements in various locations of key performance indicators such as received signal strength (RSS). Cellular operators rely on signal maps to understand the performance and coverage of their own network and their competitors. Although cellular providers can collect measurements themselves, due to cost and liability they increasingly choose to outsource the collection to third-party mobile analytics companies, such as OpenSignal [11], RootMetrics [12], Tutela [13] and others. These companies crowdsource measurements directly from user devices, via standalone mobile apps [11], or measurement software development kits [13] integrated into popular partnering apps. While these measurements are assumed to be sparse in space and time and over thousands of users, previous work has shown that the identities are easily inferable from location data [14]. User privacy may be preserved by obfuscating the datasets prior to release, at the expense of generating less accurate signal maps.

In the next section we briefly discuss the state-of-the-art in privacy, especially as it relates to spectrum allocation and signal map generation. Section III describes the optimization and learning methodologies in detail. Section IV applies the two methodologies to the two use cases under consideration. The formal treatment of the privacy-utility problem in the context of spectrum sharing yields an optimal strategy and a practical heuristic which samples the state space and generates an allocation and reporting codebook for the PUs and SAS which achieves an almost optimal trade-off between privacy and utility. Applying GAP to cellular signal strength traces, we train a privatizer which manages to achieve significantly better levels of privacy without a loss of utility with respect to signal map generation, relative to standard obfuscation approaches. Section V concludes this work.

II. Related Work

Differential privacy (DP) [15]–[19] is a popular approach to privatize data. It is studied under the so called local or global models of privacy. The global model assumes that a trust service provider has acquired a dataset and wants to release queries computed on it in a privacy-preserving fashion. The local model assumes that users do not have trust in anyone (including a service provider they may be communicating with) and want to randomize their data before sharing it. DP assumes a worst case adversarial threat model and a worst case distribution on the data when quantifying information leakage. Thus, when applied to the local model with distortion of the data as a direct measure for utility, DP, which could be achieved via additive noise, often leads to an unacceptable reduction in utility in order to achieve significant levels of privacy [20]–[25]. When there is a context-aware, application-specific utility that does not depend directly on the distortion of the data, one may envision privacy strategies that exploit the application-specific structure of the problem to achieve better privacy-utility trades than approaches involving additive noise.

Another way around the utility limitations of DP is the collection of data using cryptographic secure aggregation, a secure multi-party computation protocol that allows the service provider to only see aggregated data [26], [27]. This approach, though appealing for a variety of applications, has several restrictions. First, it requires user-to-user communication causing a significant increase in communication cost. Second, such schemes are often complex to implement on-devices and lead to increase computational complexity. Finally, this approach restricts the service provider to learning algorithms that requires sums of data. To overcome the above limitations, we focus on schemes that satisfy information theoretic notions of privacy.

Privacy in spectrum sharing systems has recently received attention. Several works consider SU privacy [28]–[30] and assume that SU requests can be aggregated to preserve privacy. [31] applies a secure computation technique to maintain SU privacy, but does not allow for spatial reuse of spectrum by SUs, severely limiting the potential efficiency of a SAS in practice. PU privacy with a SAS is considered in [32], [33], where strategies for granting SU resource assignments to protect PU privacy are considered. The privacy of both PUs and SUs is considered in [34], where the proposed encryption method along with introduction of a fourth party "key distributor" performs a subset of the SAS operations to keep user information private from the SAS. Various perturbation strategies such as adding false PU entries or randomly reducing SU transmit power allocations are studied in [35], [36], [37]. While some of this prior work has taken steps towards a formal treatment of PU privacy, the optimization is usually over parameters of predetermined strategies rather than on the strategies themselves.

Previous work on privacy in map generation has considered strategic sampling, distribution modeling, and noise addition as obfuscation strategies. In [38], the authors exploit compressive sensing techniques to sample and compress RSS values along road segments, and define a privacy metric which is a function of the topology of a graph connecting reported road segments, which can be used to recover users' traces. The authors of [39] provide a formal privacy analysis for synthetic data generation, where a statistical model of the true data is built and subsequently sampled and apply this technique to the problem of mapping user's daily commutes. In [40], distributed algorithms for Gaussian and exponential noise addition are explored as an alternative to a trusted privatizing centralized server. To the best of our knowledge, this work is the first to apply generative adversarial privacy in the context of map generation.

III. Methodologies

A. Optimization: Information Theoretic Approach

To establish a general, rigorous framework for analyzing the privacy-utility tradeoff in mobile measurement settings, we consider a system of m networked users, e.g. handheld users in the context of signal maps or primary and secondary users in the context of spectrum sharing. n functions take as input user and other data or metadata (e.g. model parameters) and provide some utility to the users and the system but also potentially expose user privacy. Functions can be user specific, e.g. obfuscation functions for user data, or system wide residing at a central entity, e.g. functions computing power assignments in the context of spectrum sharing, and functions estimating signal maps. The general problem is how to design these functions to practically achieve a near-optimal privacy-utility tradeoff.

The general framework is illustrated in Figure 1a, and specific applications of the framework to our signal map and spectrum sharing examples are illustrated in Figures 1b and 1c respectively. Consider signal maps where users are asked to report signal coverage measurements including their location at the time of the measurement. To mitigate concerns about their privacy, users may obfuscate their reported measurements, at the expense of the utility of the signal map estimate. Consider spectrum sharing, where PUs and SUs (users) share the spectrum via a spectrum sharing system. PUs, e.g., military radar, will report their state to the system to receive protection from interference. The system will use this to grant spectrum access to SUs, e.g., cellular networks. PUs may obfuscate the information they report at the expense of SU utility.

Let $X \in \mathcal{X}$ be the set describing the relevant state of all m users, e.g. current signal strength measurements or user data to be shared, where $\mathcal{X}$ is the state space. Let $g_i \in \mathcal{G}_i, i \in \{1, \dots, n\}$ denote n functions implemented in the system, e.g. obfuscation functions, where $\mathcal{G} = \mathcal{G}_1 \times \dots \times \mathcal{G}_n$ is the set of all feasible implementations. Let $Y \in \mathcal{Y}$ be the set of outputs produced by all functions, e.g. obfuscated user data, where $\mathcal{Y}$ describes the space of all possible outputs. In general, each function will map some observable subset of the state space, which may include outputs of other functions, to some output. Thus, by letting $X_i \subseteq X \cup Y$ denote the input for the ith function and $Y_i \subseteq Y$ denote the output of that function, we have $Y_i = g_i(X_i)$. Examples of user functions include obfuscating signal strength measurements or PUs locations, and examples of system wide functions include functions to create the signal map estimate or to make spectrum assignments to SUs.

To measure privacy, we consider an adversary that seeks to learn about the true state of one or more of the users without disrupting the system. The adversary will passively observe the outputs produced by the functions, and use these observations to make an inference attack on the true state space. For example, in the spectrum sharing problem, an adversary may observe the assignments made to the SUs and attempt to infer the location of the operational PUs. Treating the true state space as a random variable, the adversary will compute an estimated conditional probability distribution $P_{X|Y}$.

The effectiveness of an adversary's inference attack can be taken as a measure of the users' privacy. In theory, the optimal adversary estimate can be computed as the solution to a standard Bayesian inference problem. In practice, the size of the candidate space is likely to be too large to allow for this. The adversary could potentially solve for the maximum likelihood estimate, but in general we will not have a closed form to facilitate finding the maximum. Instead, mutual information [41] offers strong privacy guarantees in an average sense. Specifically, mutual information is defined as: $I(X;Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} P_{X,Y}(x,y) \log \left(\frac{P_{X,Y}(x,y)}{P_X(x) P_Y(y)} \right),$ and it measures the dependence of two random variables. Larger mutual information implies a greater potential for an adversary to estimate the state X from observations Y. Mutual information may allow us to study problems in an analytically tractable fashion under a formal optimization framework.

The competing goals of user privacy and utility frame the design of obfuscation functions as a multi-objective optimization problem. One may treat utility as the objective function and account for privacy as a constraint or do the opposite. Since it is more intuitive to define thresholds for user utility than for privacy in the contexts we explore, we treat user utility as a constraint on the optimization of user privacy, where thresholds on the utility may be parametrically varied to explore the privacy-utility tradeoff. Let $(Y;X)$ be the function that returns some metric quantifying user privacy. We formally state this privacy optimization problem as

$$\max_{g_1, \dots, g_n \in \mathcal{G}} \quad (Y;X) \tag{1a}$$

$$\text{subject to} \quad U_1(Y;X) \geq u_1, \dots, U_k(Y;X) \geq u_k \tag{1b}$$

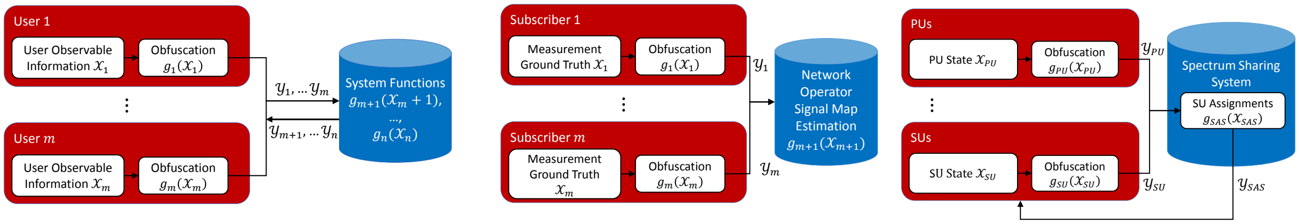
$$Y_1 = g_1(X_1), \dots, Y_n = g_n(X_n) \tag{1c}$$

$$X \in \mathcal{X}, Y \in \mathcal{Y}, X_i \subseteq X \cup Y, Y_i \subseteq Y, \tag{1d}$$

where $U_i, i \in \{1, \dots, k\}$ denotes utility functions for the system and k will depend on the application setting. In spectrum sharing, PUs and SUs each have their own utility, i.e., $k = m$, and we will specifically examine the case of one PU network and one SU network, i.e., where $k = 2$. The utility of the PUs will depend on the likelihood of experiencing interference, while the utility of the SUs will depend on the amount of spectrum they are permitted to access. For signal maps, we will assume a single utility for the quality of the resulting map, i.e., $k = 1$.

In spectrum sharing, PUs will have a function to report their requirements for interference protection and share (obfuscated) PU locations, SUs have a function to request access to the spectrum, and the sharing system has a function to grant SU access by assigning allowable transmission power levels. We will examine a case where the access system is assumed to be trusted and SUs are assumed to be truthful such that a single function for the access system to make assignments to SUs will be optimized to trade utility and privacy, i.e., $n = 1$. In signal maps, we will consider the optimization of a collaborative user privatization function and a function to generate the signal map from the privatized user data, i.e., $n = 2$. Given that the optimization is over functional spaces, solving Problem (1) may not be tractable. Regardless, this provides a general framework that can be applied to rigorous comparison of heuristics in specific problem settings without further limitations or assumptions.

Suppose the state spaces $\mathcal{X}$ and $\mathcal{Y}$ are discrete and finite. In this case, g_i can be modeled as $P_{Y_i|X_i}(y|x)$, i.e., the probability that the output y is returned given that the observation is x, where $y \in \mathcal{Y}, x \in \mathcal{X}$. To simplify exposition we can combine the conditional probabilities corresponding to each function into a single conditional probability for the observation of an adversary given the true state space, i.e., $P_{Y|X}$. In the spectrum sharing setting, Y contains the power assignments to SU cellular customers observed by the adversary while X contains the user states, e.g. PU (radar) locations. In most practical settings, it will be intractable to write down these probability distributions explicitly owing to the large number of possible states, but we can theoretically map (1) to an


User 1
User Observable Information $\mathcal{X}_1$
Obfuscation $g_1(\mathcal{X}_1)$
⋮
User m
User Observable Information $\mathcal{X}_m$
Obfuscation $g_m(\mathcal{X}_m)$
$y_1, \dots y_m$
$y_{m+1}, \dots y_n$
System Functions $g_{m+1}(\mathcal{X}_m + 1)$, ..., $g_n(\mathcal{X}_n)$
Subscriber 1
Measurement Ground Truth $\mathcal{X}_1$
Obfuscation $g_1(\mathcal{X}_1)$
y_1
⋮
Subscriber m
Measurement Ground Truth $\mathcal{X}_m$
Obfuscation $g_m(\mathcal{X}_m)$
y_m
Network Operator Signal Map Estimation $g_{m+1}(\mathcal{X}_{m+1})$
PUs
PU State $\mathcal{X}_{PU}$
Obfuscation $g_{PU}(\mathcal{X}_{PU})$
y_{PU}
⋮
SUs
SU State $\mathcal{X}_{SU}$
Obfuscation $g_{SU}(\mathcal{X}_{SU})$
y_{SU}
Spectrum Sharing System
SU Assignments $g_{SAS}(\mathcal{X}_{SAS})$
y_{SAS}

(a) General Model (b) Signal Maps (c) Spectrum Sharing

Fig. 1: Examples of Privacy System Models.

equivalent problem:

$$\max_{P_{YjX}} \quad (Y;X) \tag{2a}$$

$$\text{subject to} \quad \sum_{x2X}\sum_{y2Y} P_{YjX}(yjx)U_1(x;y) \quad u_1;$$

$$:::; \sum_{x2X}\sum_{y2Y} P_{YjX}(yjx)U_k(x;y) \quad u_k \tag{2b}$$

$$P_{YjX}(yjx) \quad 0; 8y \ 2 \ Y; x \ 2 \ X; \tag{2c}$$

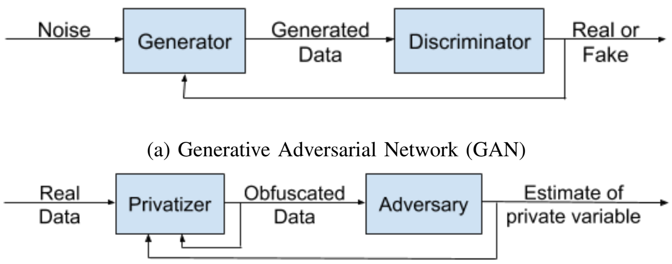
$$\sum_{y2Y} P_{YjX}(yjx) = 1; 8x \ 2 \ X; P_{YjX} \ 2 \ P_c: \tag{2d}$$

Here, the constraints in (2b) correspond to the constraints in (1b), where we have made the assumption that for some practical settings, given a specific realization $x \ 2 \ X$ and $y \ 2 \ Y$, we can pre-compute the utility $U_i(x;y)$, which can be treated as a constant parameterizing the optimization. For example, in spectrum sharing, given a user state x and a set of SU assignments y, we can directly compute the SU utility as the sum rate of cellular customers of the SUs. (2d) simply ensures that P_{YjX} yields a valid probability distribution. We also include $P_{YjX} \ 2 \ P_c$ to allow for other general constraints on the randomized operation, such as causality of the individual functions, which we assume can be described by a convex set P_c.

Since all of the constraints in Problem (2) are convex, if the privacy cost metric $(Y;X)$ is convex with respect to P_{YjX}, e.g. mutual information [42], Problem (2) is convex, and we can derive conditions on the optimal (obfuscation) functions. In practical cases, such optimal conditions may not be directly applied in practice, but we can use the derived conditions to inform the development of more practical strategies suited to the specific application. In cases where computing utilities may be impractical, the most appropriate privacy metrics may be non-convex, other non-convex constraints may be necessary, or we may lack the functional relationships needed to write down Problem (1) in the first place, and will instead rely on data traces and data-driven obfuscation techniques.

B. *Learning: Data-Driven Approach*

Leveraging recent advancements in generative adversarial networks (GANs) [4]–[6], our generative adversarial privacy (GAP) system learns how to discern private features from the dataset, and then how to cleverly obfuscate the data such that these private features are difficult to discern [2]. Like traditional GANs, which learn to create synthetic data that could pass as real data by positioning a generative model and a discriminative model against each other and training each until the generator is sufficiently good at tricking the discriminator, our GAP system positions a privatizer and an adversary against each other and trains each until the privatizer is sufficiently good at tricking the adversary. In either case, "sufficiently good" is reached when any change in the model parameters does not result in better performance, and it has converged to optimal. Unlike traditional GANs, our privatizer has a complex objective wherein it minimizes the success of our adversary while also minimizing the distortion of our original dataset.



(a) Generative Adversarial Network (GAN)

(b) Generative Adversarial Privacy (GAP)

Fig. 2: GAN and GAP

1) GAP Theoretical Model: Consider a dataset D which consists of random variables denoted by $(X;Z)$ where $X \ 2 \ X$ represents the public user state and $Z \ 2 \ Z$ represents the private user state. Each $(X_i;Z_i)$ pair corresponding to the i^{th} entry of D is distributed according to the joint probability $P(X_i;Z_i)$ and is independent from other entry pairs in the dataset, meaning our privatizer can perform a mapping $Y = g(X)$ on each pair independently, where g represents the privatization policy. We restrict our privatizer to mappings on the public variable only, as the private variable would not be released even after obfuscation due to its sensitive nature. However a general privatizer could take both X and Z as input.

We denote $\hat{Z} = h(Y) = h(g(X))$ to be the adversary's estimate of private variable Z, where h represents the adversary's estimation policy. We define the adversary's loss function to be $l(h(g(X = x); Z = z)$, where x and z are instances of sets X and Z. This loss function captures how well the adversary estimates $\hat{Z}$ from Y and depends on both g and h. The expected loss of the adversary is

$$L_a(g;h) = E[l(h(g(X));Z)]; \tag{3}$$

where the expectation is taken over the joint probability distribution $P(X;Z)$.

We define the privatizer's loss to be $\quad l(h(g(X=x));Z=z) + \quad d(g(X=x);X=x)$ where l is the adversary's loss, d captures the distortion, or loss in utility, between Y and X, and $\quad$ is a parameter quantifying the relative importance of minimizing loss in utility versus maximizing privacy. Given that privacy and utility are opposing metrics, $\quad$ may be parametrically varied to explore the privacy-utility trade, analogous to the distortion threshold under the theoretical treatment. The total expected loss of the privatizer is

$$\begin{aligned} L_p(g;h) &= E[\quad l(h(g(X));Z) + \quad d(g(X);X)] \\ &= \quad L_a(g;h) + E[\quad d(g(X);X)]; \end{aligned} \tag{4}$$

where the expectation is similarly taken over $P(X;Z)$.

Equations (3) and (4) lead to the following minimax game between the privatizer and adversary

$$\min_g \max_h L_p(g;h): \tag{5}$$

2) GAP Data-Driven Model: With complete knowledge of $P(X;Z)$, we can derive h and g directly, however without knowledge of $P(X;Z)$ we must take an iterative approach to converge on the optimal h and g for the above minimax game, which will depend on the dataset. Analagous to GAN, we restrict g and h to be modeled as multi-layer neural networks - the privatizer and adversary. The privatizer $g(X;\ _p)$ is characterized by its weights $\ _p$. The adversary $h(Y;\ _a)$ is characterized by its weights $\ _a$. For the dataset $D = f(x_i;z_i)g_{i=1}^n$, Equations (3)-(5) become the following empirical loss functions

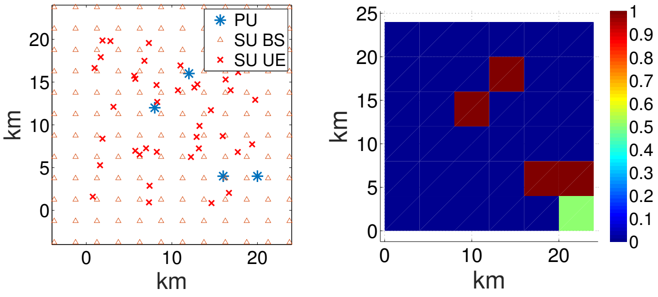
$$L_a(\ _p;\ _a) = \frac{1}{n}\sum_{i=1}^{n} l(h(g(x_i;\ _p);\ _a);z_i) \tag{6}$$

$$\begin{aligned} L_p(\ _p;\ _a) = \frac{1}{n}\sum_{i=1}^{n}[\quad & l(h(g(x_i;\ _p);\ _a);z_i) \\ & + \quad (d(g(x_i;\ _p);x_i)))] \end{aligned} \tag{7}$$

$$\min_g \max_h L_p(\ _p;\ _a): \tag{8}$$

Our iterative approach involves fixing the weights $\ _p$ and perturbing the weights $\ _a$ along the negative gradient of L_a until convergence. At this point we have found the worst case adversary for this instantiation of the privatizer. We then perturb the weights $\ _p$ along the negative gradient of L_p, repeatedly retraining $\ _a$ until convergence at each iterative step. When L_p converges, we have found the equilibrium of our minimax game and the weights which characterize h, our data-driven privatization policy.

It is shown in [2] that GAP can recover mutual information privacy for a log-loss adversary loss, and therefore presents a reasonable alternative to the information-theoretic approach.



(a) Topology

(b) Optimal Estimate

Fig. 3: Topology and Adversary Estimates.

IV. Applications

A. Spectrum Sharing

A spectrum access system (SAS) collects location and channel information from the SU and PU networks and then assigns spectrum to the SUs. Denote the reported PU and SU parameters with Y_{PU} and Y_{SU} respectively, the SAS assignments to the SUs with Y_{SAS}, and the true state of the PU with $\mathcal{X}_{PU}$. The SAS makes assignments to the SUs according to $Y_{SAS} = g_{SAS}(Y_{PU};Y_{SU})$. We consider the special case of a SAS that is trusted and an adversary that observes Y_{SAS}, i.e., the adversary can hack the SU network, or operate its own SU devices legitimately, and infer the PU state from the SU assignments. The design of the function g_{SAS} is an instance of the privacy-utility problem.

To simulate an adversary inference attack, consider 4 PU radars in a 20 km by 20 km area, where SU base stations (BSs) provide service to nearby user equipment (UE). Fig. 3 provides a visualization of an optimal adversary estimate for each location based on selection of g_{SAS} to maximize utility without consideration for privacy. Fig. 3b shows the estimate after just a single observed time slot (single assignment to all SU transmitters). The adversary is able to immediately identify all 4 locations, with some uncertainty about a fifth location.

We analyze the spectrum sharing privacy-utility problem using the framework given by Problem (2) where the applicable utility functions are the probability of harmful interference to the PUs, and the achievable data rates with the spectrum granted to the SUs. X is the state (e.g. location) of all PUs, and Y contains the observed power assignments granted by the system to the SUs at known locations. Applying mutual information as a privacy metric, we can define the privacy optimization problem as

$$\min_{P_{Yjx}} I(Y;X) = \sum_{y2Y}\sum_{x2X} P_{X;Y}(x;y)\log\left(\frac{P_{X;Y}(x;y)}{P_X(x)P_Y(y)}\right); \tag{9}$$

$$s.t. \sum_{y2Y} P_Y(y)U_{SU}(y) \quad u_{SU};$$

$$\sum_{y2Y} P_{Yjx}(yjx)U_{PU}(x;y) \quad u_{PU}; 8x\ 2\ X$$

$$P_{Y|X}(y|x) \geq 0, \forall y \in \mathcal{Y}, x \in \mathcal{X}; \sum_{y \in \mathcal{Y}} P_{Y|X}(y|x) = 1, \forall x \in \mathcal{X},$$

where the SU utility for a given realization is denoted $U_{SU}(y)$, the threshold on SU utility is u_{SU}, the upper limit on the probability of interference is u_{PU}, and $U_{PU}(x, y)$ is the probability of interference. Note that since less mutual information means more privacy we minimize rather than maximize it. This is a convex formulation and we can derive conditions on the optimal solution using standard Lagrange multiplier techniques and a little algebra. The first-order optimality condition yields $P_{Y|X}(y|x) = P_Y(y) \exp(\lambda_u U_{SU}(y) - U_{PU}(x, y) \lambda_x / P_X(x)) / N_C$ where N_C is a normalization constant and λ_u, λ_x are Lagrange multipliers. We may interpret each term in the numerator as follows: (i) the probability that y is reported, $P_{Y|X}(y|x)$, should exponentially increase with $U_{SU}(y)$, i.e., the SU utility offered, (ii) for a given true PU state x, the probability that y is reported should exponentially decrease with $U_{PU}(x, y)$, the probability that reporting this state will cause harmful interference to the PUs, and (iii) the probability that y is reported for a given x should linearly increase with $P_Y(y)$, the probability that y is reported for all other states, i.e., we should reuse the same reported states to the extent practical.

In practice, we may do this by sampling the space of adversary observations $\mathcal{Y}$ and constructing a "codebook" of possible observations $\mathcal{C} \subseteq \mathcal{Y}$. At each time slot, for the given true PU and SU topologies, we can compute the SU utility and probability of interference to PUs that would result for any selected codeword, which we use to weight the codewords exponentially. We then randomly select a codeword according to the weights. We will refer to this as the Allocation and Reporting Codebook (ARC) method. Algorithm 1 provides pseudocode for ARC. To compute the codeword weights, we first recognize that the multiplier λ_u corresponds to the SU utility constraint and thus can be viewed as a design parameter for trading between SU utility and PU privacy. Codebook size is also a design parameter which potentially offers improved performance at the cost of complexity. We also observe that λ_x corresponds to the harmful interference constraint. We can solve for the λ_x that ensures the constraint is satisfied. To construct the codebook, we assume that there is at least one allocation that will not result in harmful interference, e.g., when no assignments are granted to SUs. When SU utility, U_{SU}, and interference, U_{PU}, can be computed in polynomial time, which is the case for sum-rate utility and the probabilistic channel uncertainty models we will consider, ARC runs in polynomial time with respect to the size of the codebook, s, and the number of PUs and SUs, offering a generally efficient solution that can be applied to large problems.

Consider a small scale spectrum sharing scenario with 4 PU locations and 2 SUs operating in a 10x10 km region, where we can enumerate all potential PU states and adversary observations. This allows for computing the optimal privacy strategy and assessing how close to the optimal other practical strategies perform. Let SU utility be the sum-rate throughput, computed as the sum Shannon capacity of all SU assignments. We implement the optimal privacy strategy for minimizing mutual information by applying a convex solver to (9), precomputing parameters $U_{SU}(y)$ and $U_{PU}(x, y)$ for every possible PU state and SU allocation.

```
Input: $\lambda_u$, Codebook Size = $s$
Output: Codebook $\{y_i, i \in \{1, \dots, s\}\}$
for i=1:s-1 do
    Randomly generate $y_i \in \mathcal{Y}$ and compute $U_{SU}(y_i)$
end for
for True PU state $x \in \mathcal{X}$ at each time slot do
    $y_s \in \mathcal{Y}$ such that $U_{PU}(x, y_s) \leq u_{PU}$
    Compute $U_{PU}(x, y_i)$ for $i \in \{1, \dots, s\}$
    Find $\lambda_x$ such that $\sum_{i=1}^{s} w_i U_{PU}(x, y_i) \leq u_{PU}$, $w_i = \frac{\exp(\lambda_u U_{SU}(y_i) - U_{PU}(x,y_i)\lambda_x)}{\sum_{j=1}^{s} \exp(\lambda_u U_{SU}(y_j) - U_{PU}(x,y_j)\lambda_x)}$
    Randomly sample $i_r$ from $i \in \{1, \dots, s\}$ according to weights $w_i$
    return $y_{i_r}$
end for
```

Algorithm 1: Allocation and Reporting Codebook (ARC)

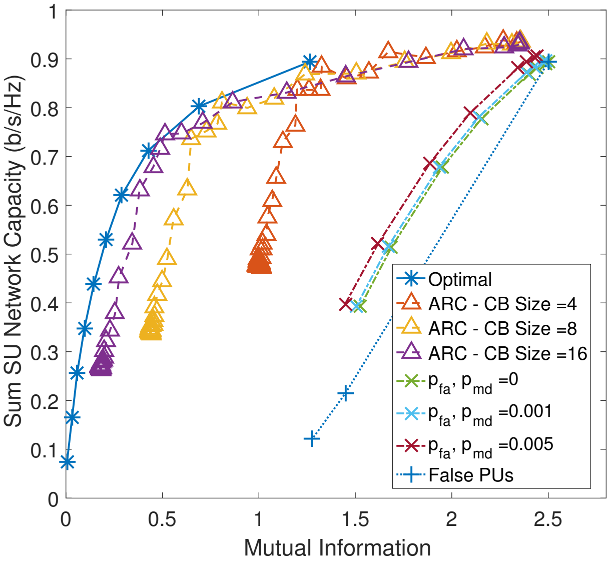


Fig. 4: Privacy-Utility Tradeoffs in Spectrum Sharing

We can quantitatively compare the privacy-utility tradeoff of the optimal approach with heuristic obfuscation strategies, which we plot in Figure 4. Recall that zero mutual information corresponds to maximum privacy, such that operating in the top left corner of the plot is ideal. The optimal strategies are plotted parametrically for different values of the required SU utility, u_{SU}. ARC is plotted with three different codebook sizes, 4, 8 and 16, and for each codebook size, is plotted parametrically for different values of λ_u. For the perturbation strategies, we consider the approach of adding either 0, 1 or 2 false PU entries to randomly select a reported state from the true state. Note an interpolation line is included in the figures for visualization, but only the operating points with markers are achievable for this strategy. We also plot strategies where false alarms (report a PU entry when there isn't any) and missed detections (don't report an existing PU entry) are applied to each possible PU location in the state. We separately plot missed detection rates of 0, 0.1% and 0.5%, and for each missed detection rate, we parametrically plot false alarm rates from 0 to 30%.

As the codebook size increases, ARC offers an increasingly close approximation of the optimal privacy strategy, with $_u$ offering an effective way to trade between SU utility and PU privacy. The false entry strategy rapidly degrades SU utility as additional false entries are included. The missed detections and false alarms offer a somewhat better tradeoff than the false entries. Both are significantly outperformed by the optimal and ARC, even with the smallest codebook size. Note that the top right marker for the false PU strategy corresponds to no obfuscation, i.e., the utility maximizing strategy. By comparison, ARC can offer a nearly 50% improvement in privacy with negligible reduction of SU utility.

This substantial improvement over existing privacy strategies provides clear evidence of the benefit of designing privacy strategies based on derived conditions on the optimal solution. The data-driven GAP approach to privacy could also be applied to the spectrum sharing setting, with the SU and PU utility functions used in a weighted objective function to train a privatizer against an adversary trained to explicitly estimate private PU information. For brevity, and considering we lack a real spectrum sharing data set to implement a meaningful data-driven approach, we defer applying GAP to the spectrum sharing setting. Instead we will study the effectiveness of GAP in the next subsection.

B. Signal Maps

We demonstrate generative adversarial privacy in the context of signal map creation. Our dataset contains mobile data from five users over 10 days in April, 2013 in a geographically contained region in Chania, Greece [43]. We limit our concern to five features, detailed in Table I. This dataset, representative of what third-party data collection companies such as Tutela might collect, would then be sold or publicly released. If released without obfuscation, an adversary might be able to infer sensitive information about the users. For this purpose we propose strategically obfuscating the data prior to its release.

TABLE I: Dataset Features

Feature	Sensitivity	Variable
psuedoID	private	u
timestamp	public	t
latitude	public	x
longitude	public	y
RSS	public	s

Our objective is to obscure the sensitive correlation between $(t;x;y;s)$ and (u), see Table I, without affecting the useful relationship between $(x;y)$ and (s). We implement our GAP framework with the two multi-layer neural networks shown in Figure 5 (for clarity, not all connecting weights are shown).

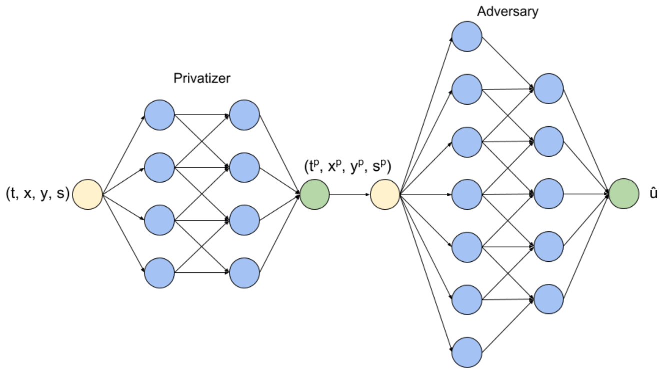


Fig. 5: Privatizer and Adversary Neural Nets

1) Implementation: Our privatizer neural net is fed as input $(t;x;y;s)$, and outputs obfuscated data $(t^p;x^p;y^p;s^p)$. To avoid overfitting, our privatizer model is kept to two fully connected layers, with four hidden units. These hyperparameters were experimentally determined to perform well on our dataset. We use gradient descent optimization with a learning rate of $_p = 0{:}005$ to ensure perturbations in $_p$ are small enough to allow adversary convergence.

Our adversary neural net model is fed as input $(t^p;x^p;y^p;s^p)$, and outputs a probability distribution $\hat{u}$ over the five possible user psuedoIDs in the set $C = f0;1;2;3;4g$. Our adversary model is kept to two fully connected layers, with seven hidden units, again experimentally determined to perform well on our dataset. We use gradient descent optimization with a learning rate of $_a = 0{:}01$.

Our adversary's training loss is the cross entropy which compares the output distribution with the true pseudoID u. The adversary updates $_a$ according to the loss function

$$L_a(\ _p;\ _a) = \frac{1}{n}\sum_{i=1}^{n}\sum_{c2C} 1_{u_i;c}\ log(h(g(t_i;x_i;y_i;s_i;\ _p);\ _a)_c) \tag{10}$$

where $1_{u_i;c}$ is an indicator function equal to 1 when u_i belongs to class c and 0 otherwise, and $h(\ )_c$ is the probability of class c estimated by the adversary.

Our privatizer's training loss is comprised of two parts (1) the loss in privacy and (2) the loss in utility. The privacy component is simply the negative of the adversary's loss $L_a(\ _p;\ _a)$ since better adversary performance implies less privacy. To define the utility component we first define a function $m(\ )$ which takes as input $(x;y;s)$ points and gives as output a set of parameters which characterize a mapping between $(x;y)$ and s. This function represents our signal map.

For this application in which we have prior knowledge that there is a single base station in the geographic region represented by our dataset, $m(x;y;s)$ performs ordinary least squares regression to estimate the parameters of a pathloss model relating RSS to distance from the base station, where RSS falls off inversely proportional to distance squared. Equation (11) states the relationship between distance, r, and RSS, s, which depends on scaling constants K, c, and s_0. Equation

(12) defines distance as a function of latitude, x, and longitude, y, given that the base station is located at $(a; b)$.

$$Kr\ ^{2} = cs + s_0; \qquad (11)$$

$$r = \sqrt{(x \quad a)^2 + (y \quad b)^2}: \qquad (12)$$

Given that five constants $(K; c; s_0; a; b)$ completely characterize this relationship, we can define a new vector of constants $[\ _1\ \ _2\ \ _3\ \ _4\ \ _5]$ which when multiplied by D, the $n \quad 5$ matrix $[1\ x\ x^2\ y\ y^2]$ give $D \quad = \frac{1}{s}$. We then directly calculate these constants

$$= (D^T D)\ ^1 D \frac{1}{s}: \qquad (13)$$

We can now define $m(\ )$ and write the pathloss-model utility component of our privatizer's loss, $l_{pathloss}$ as the mean squared error between the parameters fitted on our input data and the parameters fitted on our obfuscated data

$$m(x; y; s) = \quad ; \; m(x^p; y^p; s^p) = \quad ^p; \qquad (14)$$

$$l_{pathloss} = \frac{1}{5}\sum_{i=1}^{5}(\ _i \quad \ _i^p)^2: \qquad (15)$$

In implementation, this loss causes the privatized data to converge to points which will describe the same pathloss model as that described by the input data, however we need one additional component to restrict the obfuscated data to the geographic region specified by the input data. This is done by including another term in the utility loss which quantifies the mean squared error between the vectors $M = [\ _x\ \ _x^2\ \ _y\ \ _y^2\ \ _s\ \ _s^2]$ and $M^p = [\ _x^p\ (\ _x^p)^2\ \ _y^p\ (\ _y^p)^2\ \ _s^p\ (\ _s^p)^2]$ which capture the first and second moments of the input data and obfuscated data. Now our final privatizer loss is given by

$$L_a(\ _p;\ _a) = \quad U(\ _p) \quad L_a(\ _p;\ _a) \qquad (16)$$

$$U(\ _p) = \max f\frac{1}{5}\sum_{i=1}^{5}(\ _i \quad \ _i^p)^2; \frac{1}{6}\sum_{i=1}^{6}(M_i \quad M_i^p)^2 g; \qquad (17)$$

where $U(\ _p)$ explicitly defines our utility metric. In simplest terms, our iterative algorithm is described in Algorithm 2.

```
Input: dataset $D$
Output: obfuscated dataset $D^p$
while $_p$ has not converged do
    $D^p \quad g(D;\ _p)$
    while $_a$ has not converged do
        $\hat{0} \quad h(D_p;\ _a)$
        $_a \quad _a \quad _a \quad r\ _a L_a(\ _p;\ _a)$
    end while
    $_p \quad _p \quad _p \quad r\ _p L_p(\ _p;\ _a)$
end while
$D^p \quad g(D;\ _p)$
return $D^p$
```

Algorithm 2: GAP

2) *Performance:* Figures 6 illustrates the output of our data-driven GAP privatizer on a sample of 100 data points, with a utility penalty = 5. It can immediately be seen that while the input data is in five obviously identifiable clusters indicating users, the obfuscated data forms line-like clusters along the apparent gradient in signal strength. This learned privatization scheme intuitively makes sense, because it is difficult for an adversary to estimate five users from a line and it is the simplest shape that can capture the pathloss model which best fits the input data.

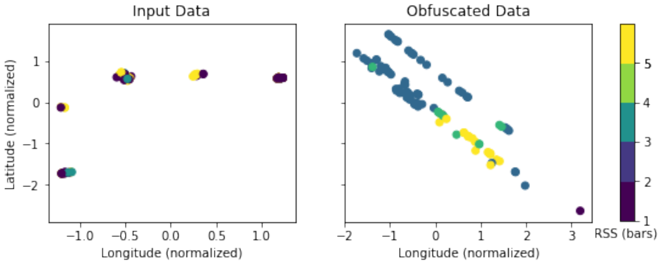
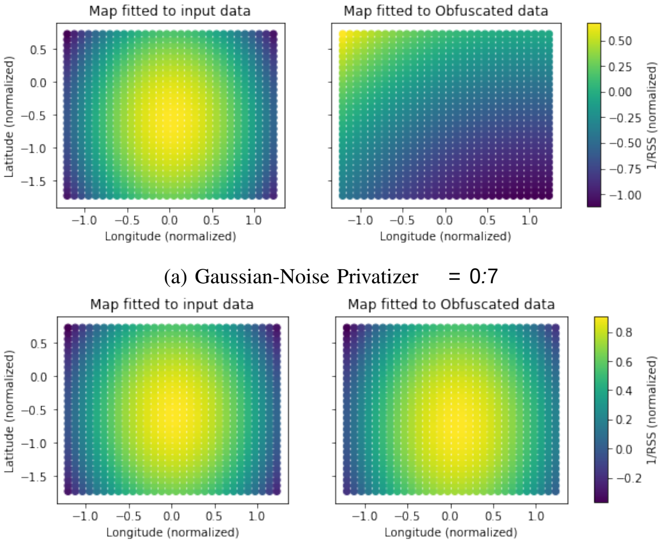


Fig. 6: GAP Obfuscated Data (= 5)



(a) Gaussian-Noise Privatizer = 0:7

(b) GAP = 5

Fig. 7: Generated Signal Maps for Privacy=0.64

Figure 7 compares the performance of our data-driven GAP privatizer with the performance of a standard gaussian noise-adding privatizer with no knowledge of the dataset characteristics. For the same level of achieved privacy, there is clearly more utility in the GAP-obfuscated data as seen by both rightside maps.

As we vary , we observe changes in the outcome of the privatizer which correspond to high utility but low privacy for high values of and high privacy but low utility for low values of , allowing us to directly outline the privacy-utility curve (Figure 8). We can observe a similar privacy-utility curve depending on the standard deviation of the noise of the gaussian noise-adding privatizer, where low noise-levels result in high utility but low privacy and high noise-levels result in high privacy but low utility. However for the same utility values our GAP privatizer achieves up to a 210% increase in privacy and for the same privacy values our GAP privatizer achieves up to a 540% increase in utility. Notably = 0 does not, as one would expect, result in the best privacy. We suspect this occurs because of the iterative nature of the algorithm - including utility causes the privatizer to train for longer, allowing the adversary to train for longer as well.

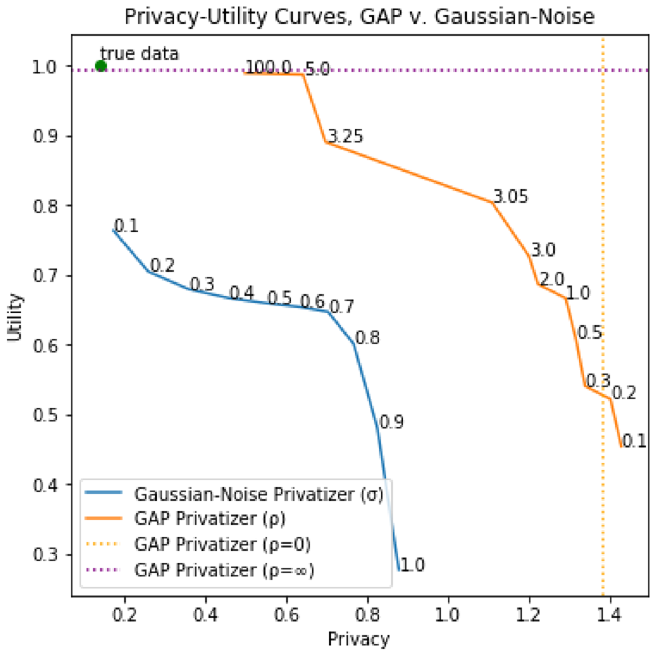


Fig. 8: Privacy-Utility Curve and Relative Performance

This significant improvement relative to standard obfuscation strategies demonstrates the advantages of learning data-driven obfuscation strategies. To apply the optimization methodology discussed in Section III-A to this context, we could replace our empirical cross-entropy privacy metric with $I(X;Y)$, the mutual information between the true data and obfuscated data. For the same utility metric used in Equation (17), and considering obfuscation strategies specified by P_{YjX}, we can minimize the mutual information subject to a minimum allowable utility and follow the codebook approach applied in section IV-A, though for brevity we do not present results here.

V. Conclusions

In this work we have explored the privacy-utility trade in the context of wireless networks. We have defined context-specific privacy and utility metrics for the applications of spectrum sharing and signal map generation, and have used these metrics to aid in the design of obfuscation schemes which add privacy without a significant loss in utility. We have demonstrated the success of our heuristic algorithm based on derived conditions on the optimal solution and the success of our adversarial learning framework relative to standard perturbation strategies.

VI. Acknowledgements

We would like to thank Athina Markopoulou and Emmanouil Alimpertis from UCI for access to the user signal strength trace discussed in Section IV-B.

References

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. New York, NY, USA: Wiley-Interscience, 2006.

[2] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, “Context-aware generative adversarial privacy,” *CoRR*, vol. abs/1710.09549, 2017. [Online]. Available: http://arxiv.org/abs/1710.09549

[3] ——, “Generative adversarial privacy,” *CoRR*, vol. abs/1807.05306, 2018. [Online]. Available: http://arxiv.org/abs/1807.05306

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[5] R. Camino, C. Hammerschmidt, and R. State, “Generating multi-categorical samples with generative adversarial networks,” *arXiv preprint arXiv:1807.01202*, 2018.

[6] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *CoRR*, vol. abs/1411.1784, 2014. [Online]. Available: http://arxiv.org/abs/1411.1784

[7] J. H. et al., “Realizing the full potential of government held spectrum to spur economic growth,” *President’s Council of Advisors on Science and Technology Report to the President*, Jul. 2012.

[8] “Report and order and second further notice of proposed rulemaking,” *FCC*, no. 15-47, GN Docket No. 12-354, Apr. 2015.

[9] “Order on reconsideration and second report and order,” *FCC*, no. 16-55, GN Docket No. 12-354, May 2016.

[10] P. Atkins, “Re: Commercial operations in the 3550-3650 MHz band (GN Docket No. 12-354),” *National Telecommunications and Information Administration Letter to the FCC*, Mar. 2015.

[11] O. S. Inc., “3g and 4g lte cell coverage map.”

[12] R. M. Inc., “Metro rootscore reports.”

[13] T. Technologies, “Manage your mobile experience.”

[14] E. Agapie, G. Chen, D. Houston, E. Howard, J. Kim, M. Y. Mun, A. Mondschein, S. Reddy, R. Rosario, J. Ryder, A. Steiner, J. Burke, E. Estrin, M. Hansen, and M. Rahimi, “Seeing our signals: Combining location traces and web-based models for personal discovery,” in *Proceedings of the 9th Workshop on Mobile Computing Systems and Applications*, ser. HotMobile ’08. New York, NY, USA: ACM, 2008, pp. 6–10. [Online]. Available: http://doi.acm.org/10.1145/1411759.1411762

[15] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[16] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, “Our data, ourselves: Privacy via distributed noise generation,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2006, pp. 486–503.

[17] C. Dwork, “Differential privacy: A survey of results,” in *International Conference on Theory and Applications of Models of Computation*. Springer, 2008, pp. 1–19.

[18] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, “Differential privacy under continual observation,” in *Proceedings of the forty-second ACM symposium on Theory of computing*. ACM, 2010, pp. 715–724.

[19] C. Dwork and A. Roth, “The algorithmic foundations of differential privacy,” *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[20] T. Wang, J. Blocki, N. Li, and S. Jha, “Locally differentially private protocols for frequency estimation,” in *Proc. of the 26th USENIX Security Symposium*, 2017, pp. 729–745.

[21] J. Acharya, Z. Sun, and H. Zhang, “Communication efficient, sample optimal, linear time locally private discrete distribution estimation,” *arXiv preprint arXiv:1802.04705*, 2018.

[22] R. Bassily, U. Stemmer, A. G. Thakurta *et al.*, “Practical locally private heavy hitters,” in *Advances in Neural Information Processing Systems*, 2017, pp. 2288–2296.

[23] P. Kairouz, K. Bonawitz, and D. Ramage, “Discrete distribution estimation under local privacy,” in *International Conference on Machine Learning*, 2016, pp. 2436–2444.

[24] M. Ye and A. Barg, “Optimal schemes for discrete distribution estimation under locally differential privacy,” *IEEE Transactions on Information Theory*, 2018.

[25] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, “Local privacy and statistical minimax rates,” in *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*. IEEE, 2013, pp. 429–438.

[26] A. Shamir, “How to share a secret,” *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[27] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, “Practical secure aggregation for federated learning on user-held data,” *arXiv preprint arXiv:1611.04482*, 2016.

[28] S. L. et al., “Location privacy preservation in collaborative spectrum sensing,” in *Proceedings IEEE INFOCOM*, March 2012, pp. 729–737.

[29] Z. G. et al., “Security and privacy of collaborative spectrum sensing in cognitive radio networks,” *IEEE Wireless Commun.*, vol. 19, no. 6, pp. 106–112, 2012.

[30] W. Wang and Q. Zhang, “Privacy-Preserving Collaborative Spectrum Sensing With Multiple Service Providers,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 1011–1019, Feb. 2015.

[31] M. G. et al., “LPOS: Location privacy for optimal sensing in cognitive radio networks,” in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–6.

[32] A. R. et al., “Spectrum database poisoning for operational security in policy-based spectrum operations,” in *MILCOM - IEEE Military Communications Conference*, Nov 2013, pp. 382–387.

[33] A. B. Mosbah, T. A. Hall, M. Souryal, and H. Afifi, “An analytical model for inference attacks on the incumbent’s frequency in spectrum sharing,” in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, March 2017, pp. 1–2.

[34] Y. D. et al., “P^2-SAS: Preserving users privacy in centralized dynamic spectrum access systems,” in *ACM MobiHoc*. IEEE, 2016.

[35] B. B. et al., “Protecting the primary users’ operational privacy in spectrum sharing,” in *2014 IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*, April 2014, pp. 236–247.

[36] Z. G. et al., “Location privacy in database-driven cognitive radio networks: Attacks and countermeasures,” in *INFOCOM*. IEEE, 2013, pp. 2751–2759.

[37] M. Clark and K. Psounis, “Achievable privacy-performance tradeoffs for spectrum sharing with a sensing infrastructure,” in *2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, Feb 2018, pp. 103–110.

[38] X. Wu, P. Yang, S. Tang, X. Zheng, and Y. Xiong, “Privacy preserving rss map generation for a crowdsensing network,” *IEEE Wireless Communications*, vol. 22, no. 4, pp. 42–48, August 2015.

[39] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber, “Privacy: Theory meets practice on the map,” in *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ser. ICDE ’08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 277–286. [Online]. Available: https://doi.org/10.1109/ICDE.2008.4497436

[40] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, “Our data, ourselves: Privacy via distributed noise generation,” in *Advances in Cryptology - EUROCRYPT 2006*, S. Vaudenay, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 486–503.

[41] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 2006.

[42] S. S. et al., “Managing your private and public data: Bringing down inference attacks against your privacy,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 7, pp. 1240–1255, Oct 2015.

[43] E. Alimpertis, N. Fasarakis-Hilliard, and A. Bletsas, “Community rf sensing for source localization,” *IEEE Wireless Communications Letters*, vol. 3, no. 4, pp. 393–396, Aug 2014.