Differential Privacy for Sets in Euclidean Space

Matthew T. Hale

Abstract

As multi-agent systems become more numerous and more data-driven, novel forms of privacy are needed in order to protect data types that are not accounted for by existing privacy frameworks. In this paper, we present a new form of privacy for set-valued data which extends the notion of differential privacy to sets which users want to protect. While differential privacy is typically defined in terms of probability distributions, we show that it is more natural here to define privacy for sets over their capacity functionals, which capture the probability of a random set intersecting some other set. In terms of sets’ capacity functionals, we provide a novel definition of differential privacy for set-valued data. Based on this definition, we introduce the Laplacian Perturbation Mechanism (so named because it applies random perturbations to sets), and show that it provides $\epsilon$-differential privacy as prescribed by our definition. These theoretical results are supported by numerical results, demonstrating the practical applicability of the developments made.

I. INTRODUCTION

Multi-agent systems are studied in diverse applications, including robotics [26], communications [17], [6], and the smart power grid [2], [30]. Agents in these systems work together by exchanging information, though in some cases the data that must be shared can be quite sensitive. For example, it has been shown that the power usage data shared in smart power grids can reveal a great deal about users’ personal lives, including when a user is at home or away, awake or asleep, and even what a user is doing inside his or her home [28], [29]. In this and other data-intensive applications, there is therefore a need for multi-agent coordination while protecting the data of individuals. In response to such privacy concerns, privacy tools have been developed that allow for sharing data while providing privacy guarantees to data-producing entities.

One privacy framework that has seen wide use is differential privacy. Differential privacy originates in the database literature [7], [8], [9], and its original formulation keeps individual database entries private whenever a database is queried. Differential privacy is an attractive means to protect sensitive data because it provides several strong privacy guarantees. In particular, it is resilient to post-processing, meaning that post-hoc analyses do not threaten the privacy protections afforded to sensitive data [10], and it is also robust to arbitrary side information, meaning that learning more about a data-producing agent does not fully defeat that agent’s privacy [15]. To provide these strong protections to other data types, differential privacy has been extended to other settings, including trajectory-valued data in [19], functional data in [14], and arbitrary metric spaces in [3].
Along these lines, in this paper we extend differential privacy to set-valued data. To consider a concrete example of sensitive set-valued data, we consider forecasting in financial markets. Given the current market prices of some collection of related assets, a financial firm can make estimates of the future prices of these assets, though each predicted price is often expressed as an interval in which the future price will lie, rather than as a single point. As a result, predictions about this collection of assets most naturally take the form of a set [16]. An analyst or government agency monitoring market behavior may wish to access these predictions from several firms, though revealing them may be undesirable. Indeed, revealing these set-valued data can not only make a firm’s future intentions predictable or known, but can also potentially reveal the manner in which these predictions are made. As a result, these set-valued data need to be kept private.

Beyond this example, set-valued data are also used in a range of diverse applications, including state estimation [24], [25], [23], optimization [4], [18], and economics [16], and similar privacy concerns exist across these domains. From these applications, there arises a need to share set-valued data in a manner which makes it private, while still allowing for meaningful analyses to be performed. Targeting all of these applications, we develop differential privacy for set-valued data.

Several existing works have proposed techniques for keeping sets private, though many of these works target different kinds of sets than we consider here. In particular, emphasis has been placed on privacy for transactional data, which are non-numerical data that can represent purchases made, and there has been focus on anonymization techniques for such datasets [13], [27], [31], including time-varying datasets [32], and scalable privacy for high-dimensional sparse datasets of this kind [11]. In addition to existing work on anonymization, the work in [5] develops differential privacy for counting queries over transactional datasets. Motivated by the aforementioned applications, our emphasis in this paper is on privacy for subsets of \( \mathbb{R}^n \), and we differ from the existing literature precisely because we develop differential privacy for subsets of Euclidean space. In general, such sets are not readily protected by existing methods for set-valued privacy. To the best of our knowledge, this is the first work to develop differential privacy for sets in \( \mathbb{R}^n \).

There are unique challenges in defining privacy for set-valued data, and there are several desirable characteristics a general theory of private sets should have. First, although the framework we develop should certainly accommodate...
sets with non-empty interior, it should also apply just as well to singletons or sets comprised by isolated points, simply because doing so significantly broadens the scope of this work. Second, the shape of the set should be preserved. In the above financial forecasting example, the exact set should be kept private, though if it is a cube, parallelepiped, sphere, or other shape, that aspect of the set should be preserved in order to keep the privatized data in a state amenable to analysis. Third, the privacy framework developed should be computationally simple so that it can be seamlessly integrated into a range of applications with minimal overhead and without the need for specialized computing hardware.

To develop differential privacy for set-valued data, we will draw from existing work on the theory random sets. A general theory of random sets has been developed over time, driven by a range of applications from geology [20] to microscopy [21]. The work in [22] appears to provide the most comprehensive reference on the subject, and we therefore use it as our starting point for random sets. The notion of a random set is distinct from that of a fuzzy set because a fuzzy set allows for degrees of inclusion in a set, while a random set is simply a random variable whose outcomes are ordinary sets, and we favor this framework because privatized sets should still be ordinary sets in order to be useful in applications dependent upon set-valued data. The work in [22] develops a theory of random closed sets, and we build upon this work to provide privacy for sets in a manner that meets the three criteria described above.

Beyond our own requirements, there are several challenges unique to privacy of set-valued data, including substantial difficulties in even defining probability distributions over rich spaces of sets. We address these difficulties in our definition of privacy for sets, and one contribution of this paper is the privacy definition itself. Our definition deviates from the usual definition of differential privacy because it is not in terms of probability distributions. Instead, our definition is in terms of the capacity functionals of random sets, which give the probability of a random set intersecting another set, precluding the need for using explicit probability distributions. This definition in turn lets us consider the events of interest in a significantly simpler setting; it will be shown that the definition here is more appropriate for set-valued privacy and does not incur any meaningful losses in privacy or applicability in exchange for its simplicity. The definition used here is an adaptation of existing work on differential privacy on metric spaces, and differential privacy as we define it is a direct generalization of the work in [3] to the case of set-valued data. In terms of this definition, we define the Laplacian Perturbation Mechanism for privacy of sets, so named because it applies a random perturbation to a set in order to make it private.

The rest of the paper is organized as follows. Section II reviews the necessary background on the theory of random sets and spaces of sets. Then Section III provides exposition on differential privacy, including its generalization to metric spaces. Next, Section IV provides our definition for set-valued privacy, and defines a privacy mechanism that satisfies this definition. After that, Section V provides numerical results showing the applicability of the theoretical developments we make. Finally, Section VI concludes the paper.
II. BACKGROUND ON RANDOM SETS AND SPACES OF SETS

In this section we provide the necessary background on random sets and the associated definitions needed to study them. Unless otherwise indicated, the exposition in this section draws from Chapter 1 of [22], and we refer the reader to that reference for an extensive treatment of this subject. We begin by reviewing the necessary background on the theory of random sets, and then we discuss the metric which will be used in Section III to define differential privacy for sets.

A. Definition of a Random Set

As described above, we work exclusively with sets in $\mathbb{R}^n$ because the applications of interest generate such sets. In this setting, we use $\mathcal{G}$ to denote all open subsets of $\mathbb{R}^n$, $\mathcal{F}$ for all of its closed subsets, and $\mathcal{K}$ for all of its compact subsets. We also fix a complete probability space $(\Omega, \mathfrak{F}, \mathbb{P})$, which will be used throughout when defining random objects. As noted in the introduction, we are interested in creating a general theory of privacy for sets which incorporates singleton sets (and unions thereof), and one way to do so is to develop privacy for arbitrary closed sets. Along these lines, the work in [22] provides a rich theory of random closed sets, and we therefore use it as our starting point for developing set-valued privacy. Toward doing so, we now present the definition of a random closed set.

**Definition 1: (Random closed set)** A random closed set is a map $X : \Omega \to \mathcal{F}$ satisfying

$$\{\omega \in \Omega \mid X(\omega) \cap K \neq \emptyset\} \in \mathfrak{F}$$

for all $K \in \mathcal{K}$.

Intuitively, Definition I says that $X$ is a random set if one can observe $X$ and always determine if $X$ does or does not intersect any compact set $K \in \mathcal{K}$. An equivalent characterization that highlights key probability-theoretic points is as follows. For all $K \in \mathcal{K}$, define the families of sets

$$\mathcal{F}_K := \{F \in \mathcal{F} \mid F \cap K \neq \emptyset\}$$

and for all $G \in \mathcal{G}$ define

$$\mathcal{F}_G := \{F \in \mathcal{F} \mid F \cap G \neq \emptyset\}.$$

The Fell topology on $\mathcal{F}$ is then generated by the open sets $\mathcal{F}_G$ and $\mathcal{F}_K$ for all $G \in \mathcal{G}$ and $K \in \mathcal{K}$. Using $\mathcal{B}(\mathcal{F})$ to denote the Borel $\sigma$-algebra generated by the Fell topology on $\mathcal{F}$, we note that $\mathcal{B}(\mathcal{F})$ is also generated by the open sets $\mathcal{F}_K$ for all $K \in \mathcal{K}$. This characterization of $\mathcal{B}(\mathcal{F})$ leads to the alternative definition of a random closed set as a map $X : \Omega \to \mathcal{F}$ where

$$X^{-1}(\mathcal{X}) = \{\omega \in \Omega \mid X(\omega) \in \mathcal{X}\} \in \mathfrak{F}$$

for all $\mathcal{X} \in \mathcal{B}(\mathcal{F})$. In words, this definition specifies the precise sense in which a random closed set is measurable with respect to the Borel $\sigma$-algebra $\mathcal{B}(\mathcal{F})$. 

Differential privacy is typically defined with respect to a $\sigma$-algebra, and frequently the Borel $\sigma$-algebra is favored in such definitions because it is rich enough to capture all events of interest. While $\mathcal{B}(\mathcal{F})$ is the Borel $\sigma$-algebra over the Fell topology in this setting, it is large enough that it is difficult to explicitly define a probability measure over it, and, as a result, probability distributions are often not directly used in the study of random sets [22, Page 10]. Instead, using the aforementioned fact that the collection $\{\mathcal{F}_K\}_{K \in \mathcal{K}}$ also generates $\mathcal{B}(\mathcal{F})$, attention is given specifically to the events captured by the sets $K \in \mathcal{K}$. In terms of this collection, we now define the capacity functional of a random set, which is significantly simpler than the probability distribution of a random set.

**Definition 2:** (Capacity functional) The capacity functional of a random set $X$ is denoted $T_X : \mathcal{K} \rightarrow [0, 1]$ and is defined by

$$ T_X(K) := \mathbb{P}[X \cap K \neq \emptyset] $$

for all $K \in \mathcal{K}$. △

In words, the capacity functional gives the probability that the random set $X$ intersects the compact set $K \in \mathcal{K}$.

As noted above, we choose to work with capacity functionals rather than probability distributions because doing so is significantly simpler. However, this simplicity does not result in any significant loss of precision in terms of observable events; Definition 2 lets us examine the probability of $X \in \mathcal{F}_K$ for any $K \in \mathcal{K}$, and, because these sets in turn generate $\mathcal{B}(\mathcal{F})$, we see that most events that would be captured by a set’s probability distribution over $\mathcal{B}(\mathcal{F})$ are also captured by its capacity functional.

Furthermore, Choquet’s Theorem [22, Theorem 1.13] provides that, under mild conditions, a capacity functional of the form above uniquely determines a random set, demonstrating that no specificity is lost in exchange for the simplicity of the capacity functional. With this idea in mind, we formally define differential privacy for sets in Section III in terms of the capacity functionals of the sets of interest. Before doing so, we next define the metric which we use in order to formally define differential privacy.

**B. The Hausdorff Metric**

The introduction discussed the use of metric differential privacy [3] in this setting, and one naturally must choose which metric to use based upon the needs of privacy in this context. Differential privacy on metric spaces is designed to make nearby objects in the space approximately indistinguishable, and the choice of metric provides an opportunity to define “nearby” in the correct way. In the applications of interest, two sets are typically regarded as nearby if the points they contain are all nearby. Therefore, while there are other choices for metrics on spaces of sets, e.g., the Fréchet-Nikodym metric [1, Section 1.12(iii)] which gives the distance between two sets as the volume of their mutually non-overlapping regions, we choose to use the Hausdorff metric, as it accurately captures the notion of closeness required here. We formally define it now in the form in which we use it below.

**Definition 3:** ([12]; Hausdorff Metric) Let $\| \cdot \|_1$ denote the 1-norm on $\mathbb{R}^n$. For two sets $S_1, S_2 \subset \mathbb{R}^n$, the Hausdorff distance between them is defined as

$$ d_H(S_1, S_2) := \max \left\{ \sup_{s_1 \in S_1} \inf_{s_2 \in S_2} \|s_1 - s_2\|_1, \sup_{s_2 \in S_2} \inf_{s_1 \in S_1} \|s_1 - s_2\|_1 \right\}, $$

October 4, 2017 DRAFT
where $d_H$ is called the *Hausdorff metric*.

Below we use $d_H$ to define privacy in a way which modulates the amount of noise added as a function of the distance between sets. We will also use the related notion of the diameter of a set, which is given by

$$\text{diam}(S) = \sup_{s_1, s_2 \in S} \|s_1 - s_2\|_1$$

for a set $S \subset \mathbb{R}^n$. Having defined the necessary aspects of spaces of sets, we proceed to formally define differential privacy in the next section.

### III. Differential Privacy

In this section, we review the necessary background on the theory of differential privacy. We first briefly review differential privacy in its usual form, and then elaborate on the extension of this definition to arbitrary metric spaces. We then state our definition of differential privacy for set-valued data.

#### A. Background on Differential Privacy

Differential privacy was originally defined for database queries in [7], [8], [9], and its underlying principle is simple: for a database query $q$, noise is added so that nearby databases produce query responses that are statistically close. More concretely, let $D$ be the set of databases of interest (containing ages, salaries, etc.). For any choice of privacy parameter $\epsilon \geq 0$, privacy is enforced by a *mechanism*, which is a randomized function meant to approximate exact queries by privatized queries, with the level of privacy specified by $\epsilon$. Formally, a mechanism $M$ on $D$ is $\epsilon$-differentially private if, for all databases $D_1, D_2 \in D$ differing by a single entry, we have

$$\mathbb{P}[M(D_1) \in A] \leq e^\epsilon \mathbb{P}[M(D_2) \in A]$$

for all measurable sets $A$ in the range of $M$. The purpose of protecting single changes in database values is to protect individuals’ data while still providing accurate responses to queries of groups of users. The privacy parameter $\epsilon$ determines how private data must be kept, with smaller values of $\epsilon$ giving stronger privacy protections, and typical values range from $0.1$ to $\log 3$. Given $\epsilon$, the above definition specifies the exact relationship required of private queries of $D_1$ and $D_2$, and the strong privacy protections provided by this definition have been well-studied [10].

Indeed, several key properties of differential privacy can be derived from this definition. First, differential privacy is resilient to post-processing, so that $M$ being differentially private implies that $f \circ M$ is as well for any deterministic $f$ such that the composition is well-defined. This allows differentially private data to be freely processed without threatening the privacy guarantees afforded to the underlying sensitive data. Second, differential privacy is robust to side information, and this property implies that learning some database entries does not, in general, fully defeat differential privacy [15]. These properties make differential privacy an appealing choice for protecting data because they allow for the release of private data without concern for what an eavesdropper will do with this data or what else they may learn in the future.
B. Differential Privacy on Metric Spaces

In this paper, we do not use the above definition for differential privacy, instead opting for a generalized definition for arbitrary metric spaces. The ordinary definition of differential privacy requires one to bound the sensitivity of the query \( q \), which is an upper bound on how much \( q(D_1) \) differs from \( q(D_2) \) when \( D_1 \) and \( D_2 \) differ by a single element. For set-valued data, queries can themselves be set-valued or scalar-valued, and the wide range of possible query outputs makes it difficult to compute general-purpose sensitivity bounds. To avoid doing so, we use the formulation of differential privacy for metric spaces.

To define this type of differential privacy, we consider an arbitrary space \( S \), and we use a metric over this space, defined as \( d: S \times S \rightarrow \mathbb{R} \). A mechanism \( M: S \rightarrow S \) is then said to be \( \epsilon_d \)-differentially private if

\[
P[M(S_1) \in A] \leq e^{\epsilon_d(S_1, S_2)} P[M(S_2) \in A]
\]

for all \( S_1, S_2 \in S \) and all measurable sets \( A \) in the range of \( M \).

In addition to eliminating the need for sensitivity bounds, this definition has the feature that it requires some level of indistinguishability between all choices of \( S_1 \) and \( S_2 \), regardless of how different they are. In particular, while ordinary differential privacy only pertains to databases differing by a single entry, this definition applies to all pairs of input data and scales privacy based on the extent to which these data differ, providing some level of privacy in all cases. With this definition in hand, we now formally define differential privacy for set-valued data.

C. Differential Privacy for Set-Valued Data

We now state the definition for differential privacy that is used throughout the remainder of the paper. This definition builds upon the aforementioned definition of privacy in metric spaces, though it adds an additional term which accounts for this definition being in terms of capacity functionals rather than probability distributions. We first state the definition itself and, below that, we discuss the need for this difference.

**Definition 4:** (Differential privacy for sets) A mechanism \( M: \mathcal{F} \rightarrow \mathcal{F} \) is \( \epsilon_{d_H} \)-differentially private if, for any two sets \( X, Y \in \mathcal{F} \),

\[
T_M(X)(K) \leq e^{\epsilon_{d}(K)} e^{\epsilon_{d_H}(X,Y)} T_M(Y)(K)
\]

for all \( K \in \mathcal{K} \), where \( d_H \) is the Hausdorff metric on \( \mathcal{F} \), and \( T_M(X) \) and \( T_M(Y) \) are the capacity functionals of the sets \( M(X) \) and \( M(Y) \), respectively.

**Remark 1:** The presence of the multiplicative term \( e^{\epsilon_{d}(K)} \) causes this definition for differential privacy to differ from the framework established in [3]. The reason for this change stems from using capacity functionals instead of probability distributions, though its interpretation is intuitive: the capacity functionals of \( M(X) \) and \( M(Y) \) are parameterized by the choice of set \( K \in \mathcal{K} \), and as \( K \) changes, the privacy needs of the underlying data must change in response. In particular, if \( K \) is a small set, then determining that \( M(X) \) intersects \( K \) but \( M(Y) \) does not allows for fine differences between \( M(X) \) and \( M(Y) \) to be discerned. Thus, in this case, the definition of privacy should be informed by the small size of \( K \), and therefore the term \( e^{\epsilon_{d}(K)} \) is incorporated, requiring further closeness
between the capacities of \( M(X) \) and \( M(Y) \) when \( K \) is small. Similarly, if \( K \) is a large set, then knowing that both \( M(X) \) and \( M(Y) \) intersect \( K \) does not threaten privacy much, simply because many sets will intersect a large \( K \). Accordingly, the term \( e^{\epsilon \cdot \text{diam}(K)} \) encodes this fact by requiring weaker privacy guarantees for very large choices of \( K \).

**Remark 2:** Asymptotically, as \( K \) becomes very large, the term \( e^{\epsilon \cdot \text{diam}(K)} \) goes to infinity, encoding the fact that no privacy protections would be needed for an infinite set. Indeed, so many sets would intersect such a \( K \) that there would be no privacy loss in detecting its intersection with another set, and this intuition is captured in the \( e^{\epsilon \cdot \text{diam}(K)} \) term. Similarly, if \( K = \{ \xi \} \) is a singleton, then the capacity functional of a set determines the probability that a set contains \( \xi \). Here, there are significant privacy concerns because the knowledge that \( \xi \in M(X) \) and \( \xi \notin M(Y) \) allows for very fine distinctions to be made between \( M(X) \) and \( M(Y) \), leading to substantial losses of privacy. In such a case, \( \text{diam}(K) = 0 \) and \( e^{\epsilon \cdot \text{diam}(K)} = 1 \), reducing the above privacy definition to the one seen for metric spaces in Equation (1), and enforcing stronger privacy requirements. In light of this fact, the privacy definition given in Definition 4 can be seen to be a direct generalization of the one in [3].

Having defined privacy for set-valued data, the next section defines the Laplacian Perturbation Mechanism and proves that it provides \( \epsilon d_H \) privacy as defined by Definition 4.

**IV. Enforcing Differential Privacy for Set-Valued Data**

In this section we define the Laplacian Perturbation Mechanism and show that it provides differential privacy as defined in Section III. Then we discuss why we do not use the exponential mechanism, despite its common use for non-numerical data. Below, we use the notation\[ X + a := \{ x + a \mid x \in X \} \]for all \( X \subseteq \mathbb{R}^n \) and all \( a \in \mathbb{R}^n \).

**A. The Laplacian Perturbation Mechanism**

In this subsection we define the Laplacian Perturbation Mechanism for privacy of set-valued data, which, as its name suggests, applies perturbations to sets which are randomly drawn from a Laplace distribution. The scalar Laplace distribution has probability density
\[ \text{Lap}(\mu, b) := \frac{1}{2b} \exp \left( -\frac{|x - \mu|}{b} \right), \]
where \( \mu \in \mathbb{R} \) is the mean of the distribution and \( b \in \mathbb{R} \) is its scale parameter. Using this distribution we have the following result.

**Theorem 1:** The Laplacian Perturbation Mechanism in \( \mathbb{R}^n \) defined by
\[ M(X) := X + w, \quad \text{where} \quad w \sim \text{Lap} \left( 0, \frac{1}{\epsilon} \right)^n \]
is \( \epsilon d_H \)-differentially private for all sets \( X \in \mathcal{F} \).
Proof: Consider an arbitrary compact set \( K \in \mathcal{K} \), along with two sets \( X, Y \in \mathcal{F} \). We find that, for a vector \( w \in \mathbb{R}^n \) with \( w_i \sim \text{Lap}(0, 1/\epsilon) \), expanding definitions gives

\[
\frac{T_{M(X)}(K)}{T_{M(Y)}(K)} = \frac{T_{X+w}(K)}{T_{Y+w}(K)}
\]

\[
= \frac{\Pr[X + w \cap K \neq \emptyset]}{\Pr[Y + w \cap K \neq \emptyset]}
\]

\[
= \frac{\Pr[w = x - k \text{ for some } x \in X, k \in K]}{\Pr[w = y - \xi \text{ for some } y \in Y, \xi \in K]}
\]

We emphasize that \( k \) and \( \xi \) need not have any relationship because the capacity functionals here capture the probability of intersecting with the set \( K \), though this analysis does not require that \( M(X) \) and \( M(Y) \) intersect \( K \) in the same place, allowing \( k \) and \( \xi \) to differ. Continuing, we examine these probabilities component-wise, and use the definition of the Laplace distribution given above, to find

\[
\frac{\Pr[w = x - k \text{ for some } x \in X, k \in K]}{\Pr[w = y - \xi \text{ for some } y \in Y, \xi \in K]}
\]

\[
= \prod_{i=1}^{n} \frac{\Pr[w_i = x_i - k_i, x \in X, k \in K]}{\Pr[w_i = y_i - \xi_i, y \in Y, \xi \in K]}
\]

\[
= \prod_{i=1}^{n} \frac{\exp(-\epsilon|x_i - k_i|)}{\exp(-\epsilon|y_i - \xi_i|)}, \text{ for } x \in X, y \in Y, k, \xi \in K
\]

\[
= \exp \left( -\epsilon \sum_{i=1}^{n} |x_i - k_i| + \epsilon \sum_{i=1}^{n} |y_i - \xi_i| \right), \text{ for } x \in X, y \in Y, k, \xi \in K
\]

\[
= \exp \left( \epsilon\|y - \xi\|_1 - \|x - k\|_1 \right), \text{ for } x \in X, y \in Y, k, \xi \in K
\]

\[
\leq \exp \left( \epsilon\|y - x\| + \epsilon\|k - \xi\|_1 \right), \text{ for } x \in X, y \in Y, k, \xi \in K,
\]

which follows from the reverse triangle inequality. Continuing, we use the ordinary triangle inequality to find

\[
\exp(\epsilon\|y - x\| + \epsilon\|k - \xi\|_1) \leq \exp(\epsilon\|k - \xi\|_1) \exp(\epsilon\|x - y\|_1) \text{ for } x \in X, y \in Y, k, \xi \in K,
\]

whereupon we use \( \|k - \xi\|_1 \leq \text{diam}(K) \) and \( \|x - y\|_1 \leq d_\mathcal{H}(X, Y) \) to conclude. \( \blacksquare \)

Theorem 1 shows that a set can be kept differentially private by applying a random perturbation in each coordinate. The introduction specified that the privacy mechanism developed in this paper should be computationally simple, and we see that this is indeed the case as keeping a set in \( \mathbb{R}^n \) private can be achieved by generating \( n \) random numbers. Next we comment on why we deliberately do not use the exponential mechanism for privacy in this setting.

B. On Avoiding the Exponential Mechanism

Perhaps the best known privacy mechanism for non-numerical data is the exponential mechanism. In using it, output data are assigned a quality score, which encodes how well each possible output approximates the true output of a query. Then, private outputs of non-numerical queries are generated in a way that outputs high-utility outputs with high probability and low-utility outputs with low probability. To define the exponential mechanism for closed sets in \( \mathbb{R}^n \), one would need to define a utility function

\[
u : \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}
\]
which provides a quantitative measure of the utility of outputting the set $\bar{S} \in \mathcal{F}$ when the input set is $S \in \mathcal{F}$.

A known theoretical hurdle [10, Page 38] in implementing the exponential mechanism is the complexity associated with generating quality scores for all possible input/output pairs, and this can be prohibitive for some data types. Here, this hurdle precludes the use of the exponential mechanism for the same reason we are precluded from working directly with probability distributions: the space of all subsets is simply too large to assign each subset a quality score, and thus we avoid doing so and likewise avoid using the exponential mechanism. However, as was shown above, we have successfully used the Laplace mechanism for privacy, meaning that differential privacy can be provided for sets despite the inability to use the exponential mechanism.

Next, we give numerical implementations of the Laplacian Perturbation Mechanism to show its use in practice.

V. NUMERICAL RESULTS

In this section we present two sets of numerical results implementing the Laplacian Perturbation Mechanism defined in Theorem 1. In order to generate results that are easily plotted, we do two numerical runs, one in $\mathbb{R}^2$ and one in $\mathbb{R}^3$.

A. Numerical Results in $\mathbb{R}^2$

To demonstrate the effects of the Laplacian Perturbation Mechanism in $\mathbb{R}^2$, we implement it on a set of four points. In particular, we begin with the set

\[ X = \{(0,0), (0,1), (1,0), (1,1)\} \subset \mathbb{R}^2 \]

and generate four outputs of the Laplacian Perturbation Mechanism with this set as the input. Here the privacy parameter was selected to be $\epsilon = 1$. In this case, each of the four numerical runs implemented privacy by adding noise $w$, where

\[ w \sim \text{Lap}(0,1)^2. \]

The results from these runs are shown in Figure 2. The original dataset is shown in the black square, with the points connected by solid lines and an $X$-shape in the center. We emphasize that each set only consists of the vertices of each square, and we connect points in this way simply to make clear which points belong to the same set. We see that the private outputs of the Laplacian Perturbation Mechanism are able to preserve privacy of the set $X$ while still allowing its outputs to remain close to $X$ itself, indicating that privacy can be implemented while still retaining reasonably accurate set-valued data.

B. Numerical Results in $\mathbb{R}^3$

To further illustrate the use of the Laplacian Perturbation Mechanism, we give numerical results for a set contained in $\mathbb{R}^3$. The set we begin with is the polytope in the center of Figure 3 shown in white. As above, four total numerical runs were conducted, each with $\epsilon = 1$, giving

\[ w \sim \text{Lap}(0,1)^3. \]
Fig. 2. The numerical results of four runs of the Laplacian Perturbation Mechanism in $\mathbb{R}^2$. The original set to keep private consists of four isolated points, and these points are connected by a square with an “X” in the middle to indicate that it is the sensitive data we want to protect. Each query of this set is shown with its four points connected by dotted lines to indicate which points belong to the same set. Here we can visually verify that privacy is provided while requiring only mild perturbations to the original set, indicating that private set-valued data will still be useful in the computations required by the applications of interest.

Fig. 3. The numerical results of four applications of the Laplacian Perturbation Mechanism in $\mathbb{R}^3$. The original sensitive data is the polytope in the center, and the outputs of the four numerical runs are the other four polytopes. As in Figure 2, we see that privacy is attained with reasonably sized perturbations, allowing for accurate responses to private queries of sensitive set-valued data.
in each case.

The results of these runs are shown in Figure 3, where the center polytope is the original set which we intend to keep private. Each of the other four polytopes results from a numerical run we conduct, and it can be readily verified in Figure 3 that the random displacements needed for differential privacy are on the order of the diameter of the set itself, indicating that set-valued data can be protected with reasonable random perturbations.

The results of this subsection and those above in Section V-A indicate that the privacy framework we develop meets the three criteria given in the introduction, namely, that collections of singletons should be accommodated, that the shape of the set should be preserved, and that the framework should be computationally simple. Moreover, we have done all of this while largely preserving the integrity of the underlying set-valued data, requiring only small perturbations while providing strong privacy guarantees to the data of interest, letting us simultaneously keep data private and preserve its usefulness in applications.

VI. CONCLUSION

Future work includes broadening the types of privacy that can be provided to sets. In this work, we deliberately preserved the shape of sets when keeping them private, though in other applications it may be desirable to distort the boundaries of sets, and future work includes defining privacy of this kind. Similarly, the orientation of a set may be revealing, and another direction of future work includes protecting a set’s orientation using differential privacy. Beyond that, many applications have been characterized when Gaussian disturbances are added, and developing a Gaussian Perturbation Mechanism would let users keep set-valued data private while relying on existing convergence and stability guarantees that already accommodate Gaussian noise.

REFERENCES

[1] Vladimir I Bogachev. Measure theory, volume 1. Springer Science & Business Media, 2007.
[2] S. Caron and G. Kesidis. Incentive-based energy consumption scheduling algorithms for the smart grid. In Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on, pages 391–396, Oct 2010.
[3] Konstantinos Chatzikokolakis, Miguel E. Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. Broadening the Scope of Differential Privacy Using Metrics, pages 82–102. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
[4] Guang-ya Chen, Xuexiang Huang, and Xiaogi Yang. Vector optimization: set-valued and variational analysis, volume 541. Springer Science & Business Media, 2006.
[5] Rui Chen, Noman Mohammed, Benjamin CM Fung, Bipin C Desai, and Li Xiong. Publishing set-valued data via differential privacy. Proceedings of the VLDB Endowment, 4(11):1087–1098, 2011.
[6] Mung Chiang, S.H. Low, A.R. Calderbank, and J.C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. Proceedings of the IEEE, 95(1):255–312, Jan 2007.
[7] Cynthia Dwork. Differential privacy. In 33rd International Colloquium on Automata, Languages and Programming, Part II (ICALP 2006), volume 4052 of Lecture Notes in Computer Science, pages 1–12, Venice, Italy, July 2006. Springer-Verlag.
[8] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Advances in Cryptology (EUROCRYPT 2006), volume 4004 of Lecture Notes in Computer Science, page 486503, Saint Petersburg, Russia, May 2006. Springer-Verlag.
[9] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Proceedings of the Third Conference on Theory of Cryptography, TCC’06, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag.
