ANNUAL REVIEWS

Annual Review of Control, Robotics, and Autonomous Systems

Privacy in Control and Dynamical Systems

Shuo Han[1] and George J. Pappas[2]

[1]Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, Illinois 60607, USA; email: hanshuo@uic.edu

[2]Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, Pennsylvania 19104, USA; email: pappasg@seas.upenn.edu

Annu. Rev. Control Robot. Auton. Syst. 2018. 1:309–32

First published as a Review in Advance on January 12, 2018

The *Annual Review of Control, Robotics, and Autonomous Systems* is online at control.annualreviews.org

https://doi.org/10.1146/annurev-control-060117-105018

Keywords

differential privacy, Kalman filter, gradient method, distributed optimization

Abstract

Many modern dynamical systems, such as smart grids and traffic networks, rely on user data for efficient operation. These data often contain sensitive information that the participating users do not wish to reveal to the public. One major challenge is to protect the privacy of participating users when utilizing user data. Over the past decade, differential privacy has emerged as a mathematically rigorous approach that provides strong privacy guarantees. In particular, differential privacy has several useful properties, including resistance to both postprocessing and the use of side information by adversaries. Although differential privacy was first proposed for static-database applications, this review focuses on its use in the context of control systems, in which the data under processing often take the form of data streams. Through two major applications—filtering and optimization algorithms—we illustrate the use of mathematical tools from control and optimization to convert a nonprivate algorithm to its private counterpart. These tools also enable us to quantify the trade-offs between privacy and system performance.

1. INTRODUCTION

With advances in real-time computing and sensor technology, a growing number of dynamical systems have begun to utilize user data for more efficient operation. For example, utility companies are now able to collect nearly real-time power consumption data from individual households through advanced metering infrastructures in order to improve the accuracy of demand forecasts and facilitate the operation of power plants (1). At the same time, however, individual customers are exposed to the risk that the utility company or a potential eavesdropper can learn information that the customer did not intend to share, which may include marketable information (such as the types of appliances being used) or even sensitive information (such as the customer's daily activities). Concerns about such privacy issues have been raised (2, 3) and have started to become one major hindrance to effective user participation (4). Similar issues arise in other dynamical systems, such as traffic networks (5, 6), energy-efficient buildings (7), and remote personal health monitoring (8).

The effort to protect the privacy of participating users started with ad hoc solutions (9–11) that did not provide formal mathematical guarantees. It was later recognized that these solutions were inadequate to protect privacy owing to the presence of public side information. One of the most famous instances was the identification of certain subscribers to Netflix (an online video-on-demand service provider) in the anonymized Netflix Prize data set, which was made possible through cross-validation with the Internet Movie Database (IMDb) (12). The development of rigorous solutions for preserving privacy has become an active area of research. Popular frameworks include differential privacy (13), information-theoretic privacy (14), and privacy based on secure multiparty computation (15), to name a few. These frameworks differ mostly in the strength of their privacy guarantees, which is reflected in the assumption of potential adversaries. In control systems, recent work on privacy includes, among other examples, privacy-preserving filtering of streaming data (16), privacy in smart metering (17), privacy in traffic monitoring (18), privacy in stochastic control (19), and privacy-preserving consensus (20, 21).

Recently, the notion of differential privacy proposed by Dwork et al. (13) has received attention because of its strong privacy guarantees. The original setting assumes that the sensitive database is held by a trustworthy party, which we call a mediator. The mediator needs to answer external queries about the sensitive database that potentially come from an adversary who is interested in learning information about a certain user in the database. Informally, preserving differential privacy requires that the mediator ensure that the results revealed by the mediator remain approximately unchanged if data belonging to any single user in the database are modified. Under such a requirement, an adversary can know little about any single user's information from the revealed results.

A natural and important aspect to consider when protecting privacy is the price of privacy. Introducing privacy often leads to a degradation of system performance. Indeed, without any considerations of system performance, one could protect privacy by choosing to ignore user data completely. A rule of thumb is that more privacy will result in a greater loss in performance. As discussed in detail in Section 2.4, differential privacy works by adding noise before the answer is revealed. The magnitude of the noise needs to be large enough to hide useful information with a certain probability, which is related to the level of privacy. The relationship between privacy and performance applies not only to differential privacy but also to various notions of privacy. For instance, information-theoretic privacy usually yields a weaker guarantee of privacy than differential privacy but has the advantage of sacrificing less system performance.

The review is organized as follows. Section 2 introduces the necessary background about differential privacy; for more details, especially regarding how differential privacy is applied in a more traditional setting of static databases, interested readers can refer to a recent book on

differential privacy (22). Section 3 presents the application of differential privacy to dynamic filters, including both general filters (when the user data are generic time series) and Kalman filters (when the user data are generated by a dynamical system). For the case of Kalman filters, we discuss the use of system-theoretic tools for optimal filter design, which leads to better estimation accuracy under the same privacy requirement. Section 4 presents the application of differential privacy to distributed optimization algorithms that are common for decision making in multiagent systems. We discuss two major cases: when the sensitive information appears in the objective function, and when the sensitive information appears in the constraints. Some material in Sections 2–4 can also be found elsewhere (16, 23–25).

2. BACKGROUND ON DIFFERENTIAL PRIVACY

The purpose of this section is to introduce differential privacy, which is the notion of privacy used throughout this review. We start by giving a formal definition of differential privacy and discussing its implications. We then discuss several important properties of differential privacy. In the end, we introduce several commonly used mechanisms that guarantee differential privacy. These mechanisms often serve as the building blocks for designing more complicated mechanisms.

2.1. Motivation of Differential Privacy

We consider the setting of protecting the privacy of individual users whose information is stored collectively as a database. Examples of databases include patient records in a hospital, salaries of employees of a company, and census records. The database is used to extract useful aggregate information from the users, and the result is often made available to the public. One such example is computing the average salary of all employees from a database of salaries.

One may wonder how the publicly available result, which contains only aggregate information of all users in a database, can compromise the privacy of any individual user. This can, nevertheless, happen in certain extreme cases. For example, suppose that n voters participate in an anonymous vote that involves two candidates, Alice and Bob. The result of the vote shows that $n - 1$ people voted for Alice, and 1 person voted for Bob. The person who voted for Bob is then able to learn from the result that all other people voted for Alice. Such a result apparently compromises the privacy of the $n - 1$ people who voted for Alice, even though the process was anonymous.

Aside from this extreme case, the privacy of individual users can also be compromised in the presence of side information. Consider the example of computing the average salary of employees from a salary database. It is true that one cannot generally infer the salary of any particular user in the database from the average salary. However, a powerful adversary who is able to collaborate with all but one user in the database can obtain the exact salary of that remaining user by learning from the average, even if that user is not willing to collaborate with the adversary.

We now face a dilemma: On the one hand, we would like to release useful aggregate information from a given database; on the other hand, we need to make sure that no one can learn much about any individual user, regardless of the released result and the presence of possible side information. Differential privacy was proposed to resolve this dilemma. The notion of differential privacy may differ from our common understanding of privacy. In differential privacy, privacy is not treated as a binary concept (i.e., information is either private or nonprivate) but is instead measured on a level that changes continuously from total privacy to nonprivacy. The basic idea used by differential privacy is to perturb the exact result upon release. As one can imagine, the amount of perturbation affects both the usefulness of the result and the level of privacy: More perturbation leads to a less useful result and a higher level of privacy. As seen in Sections 2.4, 4.1 (Theorem 11), and 4.2

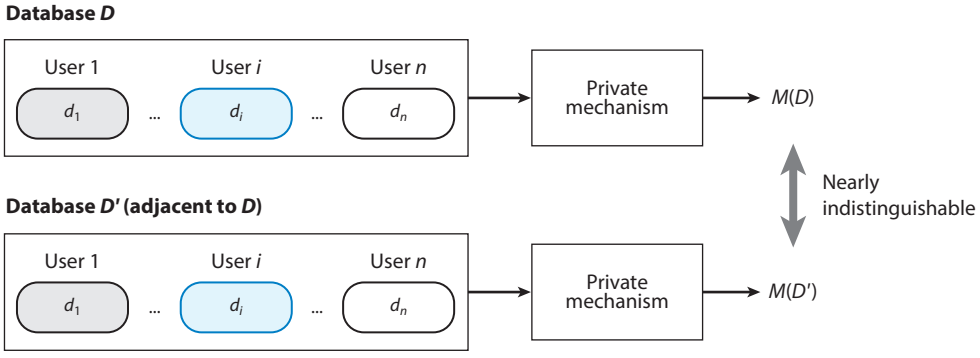

Figure 1

General setup of differential privacy with adjacent databases D and D'. A differentially private mechanism ensures that the outputs are nearly indistinguishable when the inputs are D and D', respectively.

(Theorem 13), such a trade-off between the usefulness of the result and the level of privacy can often be quantified and can provide guidelines for choosing an appropriate level of privacy.

2.2. Definition of Differential Privacy

We now present the setting of differential privacy. After giving a formal definition of differential privacy, we discuss its practical implications for how user information can be protected.

2.2.1. Terminology and definition. In differential privacy, user information that needs to be protected is contained in a set (called a database) D, in which each element corresponds to information from an individual user. For convenience, we denote by $\mathcal{D}$ the universe of all possible databases of interest. The quantity (to be released to the public) that we would like to compute from database D is modeled by $q(D)$ for some mapping q (called the query) that acts on D; the range of q is denoted by $\mathcal{Q}$.

> **Example 1.** For a database containing the salaries of a group of people, we can define $D = \{d_i\}_{i=1}^n$, where $d_i \in \mathbb{R}_+$ is the salary of user i (assuming no minimum denomination). Suppose that someone is interested in the average salary of people in the database. Then the query can be defined as $q(D) = \sum_{i=1}^n d_i/n$.

Figure 1 illustrates the general setup of differential privacy. Informally, differential privacy is able to guarantee that the result of computation on a database does not change much when any individual user's information is modified. In other words, preserving privacy is equivalent to hiding changes in the database. Formally, changes in a database are defined by a symmetric binary relation on $\mathcal{D} \times \mathcal{D}$ called an adjacency relation and are denoted by $\mathrm{Adj}(\cdot,\cdot)$; two databases D and D' that satisfy $\mathrm{Adj}(D, D')$ are called adjacent databases.

> **Definition 1 (adjacent databases).** Two databases $D = \{d_i\}_{i=1}^n$ and $D' = \{d_i'\}_{i=1}^n$ are said to be adjacent if there exists $i \in [n]$ such that $d_j = d_j'$ for all $j \neq i$.

Here, we use the notation $[n]$ to denote the set $\{1, 2, \ldots, n\}$. When differential privacy was first proposed (13), the adjacency relation was defined in a slightly different way: Two databases

are adjacent if and only if one database is a result of adding or removing one user from the other database. The motivation behind the original definition was to hide the participation of any individual in the database; a typical example is a database of patients with a certain type of disease (e.g., AIDS). Definition 1 generalizes the original notion of an adjacency relation in order to protect sensitive information beyond participation. Besides the conditions stated in Definition 1, we often also need some constraint on the difference between d_i and d_i' as a design choice. Recall that differential privacy guarantees that any two adjacent databases D and D' are nearly indistinguishable. The choice of constraining the difference between d_i and d_i' determines the granularity that an individual's value can be protected, as illustrated in the following example.

Example 2. Consider again the database of salaries in Example 1. We can define an adjacency relation as in Definition 1 with $|d_i - d_i'| \leq \$1{,}000$. Then the privacy guarantee given by a differentially private mechanism would become that an adversary cannot determine the salary of any user in the database within the accuracy of \$1,000 (with high probability). (The adversary may, however, still be able to tell that some user's salary is \$1 as opposed to \$10,000.) Alternatively, if it is publicly known that the maximum salary for any person is $d_{\max}$, we can define an adjacency relation with $|d_i - d_i'| \leq d_{\max}$. In this way, the salary of each user is fully protected—that is, an adversary cannot know anything about the salary of any user except that it lies within $[0, d_{\max}]$ (which is public knowledge).

As mentioned in Section 2.1, directly making $q(D)$ available to the public may cause users in the database to lose their privacy. To preserve privacy, for any given query q, we need to develop a mechanism M that approximates q. Naturally, the range of M is the same as that of q; that is, $\text{range}(M) = \text{range}(q) = \mathcal{Q}$. In the framework of differential privacy, all mechanisms under consideration are randomized; that is, for a given database, the output of such a mechanism obeys a certain probability distribution. A mechanism that acts on a database is said to be differentially private if it is able to ensure that two adjacent databases are nearly indistinguishable (in a probabilistic sense) from the output of the mechanism.

Definition 2 (ϵ-differential privacy). Given $\epsilon \geq 0$, a mechanism M preserves ϵ-differential privacy if for all $\mathcal{R} \subseteq \text{range}(M)$ and all adjacent databases D and D' in $\mathcal{D}$, it holds that

$$\mathbb{P}(M(D) \in \mathcal{R}) \leq e^{\epsilon}\mathbb{P}(M(D') \in \mathcal{R}). \quad 1.$$

The probability measure in Equation 1 is taken from the probability space used to define the randomized mechanism M. The constant ϵ indicates the level of privacy: Smaller ϵ implies a higher level of privacy. Notice that Equation 1 also implies

$$\mathbb{P}(M(D') \in \mathcal{R}) \leq e^{\epsilon}\mathbb{P}(M(D) \in \mathcal{R})$$

owing to the symmetric nature of an adjacency relation. The notion of differential privacy promises that an adversary cannot tell from the output of M with high probability whether data corresponding to a single user in the database have changed. It can be seen from Equation 1 that any nonconstant differentially private mechanism is necessarily randomized.

In certain cases, it is also useful to consider a relaxed and more general notion of differential privacy called (ϵ, δ)-differential privacy, which is defined as follows.

Definition 3 [(ϵ, δ)-differential privacy]. Given $\epsilon, \delta \geq 0$, a mechanism M preserves (ϵ, δ)-differential privacy if for all $\mathcal{R} \subseteq \text{range}(M)$ and all adjacent databases D and D' in $\mathcal{D}$, it holds that

$$\mathbb{P}(M(D) \in \mathcal{R}) \leq e^{\epsilon}\mathbb{P}(M(D') \in \mathcal{R}) + \delta. \quad 2.$$

It can be seen from Definition 3 that the notion of (ϵ, δ)-differential privacy reduces to ϵ-differential privacy when $\delta = 0$. The introduction of the additive term δ in Equation 2 yields a weaker privacy guarantee than ϵ-differential privacy. When $\delta > 0$, even if ϵ is small, the difference between $\mathbb{P}(M(D) \in \mathcal{R})$ and $\mathbb{P}(M(D') \in \mathcal{R})$ can still be large; as a result, an adversary can still potentially tell whether the input database is D or D'.

We now make a final note on the language used throughout this review. To avoid confusion, we reserve the word private for descriptions of mechanisms (e.g., a differentially private mechanism) and restrain ourselves from using it for user information (i.e., d_i) in the database. Instead, we use the word sensitive for user information (i.e., sensitive user information as opposed to private user information). The one exception is the term user privacy, which is used mostly in a nonmathematical context.

2.2.2. Choosing the privacy level. One useful interpretation of differential privacy can be made in the context of detection theory (26, 27). The interpretation also provides a guideline for choosing the level of privacy ϵ when implementing differentially private mechanisms. Consider a simple case involving a binary database $D = \{d_i\}_{i=1}^n \in \{0, 1\}^n$, where the goal of the adversary is to infer the value d_i of a particular user i from the output of an ϵ-differentially private mechanism M. The inference procedure used by the adversary can be modeled as the following detection rule: Report $d_i = 1$ if the output of M lies in some set $\mathcal{R}^*$ and $d_i = 0$ otherwise. Let D be the database with $d_i = 1$ and D' be the one with $d_i = 0$. We are interested in the probabilities of two types of detection errors: false positive probability $p_{\text{FP}} = \mathbb{P}(M(D') \in \mathcal{R}^*)$ (i.e., $d_i = 0$, but the detection rule reports $d_i = 1$) and false negative probability $p_{\text{FN}} = \mathbb{P}(M(D) \notin \mathcal{R}^*) = \mathbb{P}(M(D) \in \mathcal{Q}\backslash\mathcal{R}^*)$. For a good detection rule, both probabilities should be small. Since D and D' are adjacent, we know from Definition 2 that

$$\mathbb{P}(M(D) \in \mathcal{R}^*) \leq e^{\epsilon}\mathbb{P}(M(D') \in \mathcal{R}^*),$$
$$\mathbb{P}(M(D') \in \mathcal{Q}\backslash\mathcal{R}^*) \leq e^{\epsilon}\mathbb{P}(M(D) \in \mathcal{Q}\backslash\mathcal{R}^*),$$

which lead to

$$p_{\text{FN}} + e^{\epsilon}p_{\text{FP}} \geq 1 \quad \text{and} \quad e^{\epsilon}p_{\text{FN}} + p_{\text{FP}} \geq 1. \quad 3.$$

The conditions represented in Equation 3 imply that p_{FN} and p_{FP} cannot both be too small. In particular, we have

$$p_{\text{FN}} + p_{\text{FP}} \geq \frac{2}{1 + e^{\epsilon}}. \quad 4.$$

That is, these conditions limit the detection capability of the adversary so that the privacy of user i is protected. For example, if $\epsilon = 0.1$ and the false negative probability $p_{\text{FN}} = 0.05$, then the false positive probability $p_{\text{FP}} \geq \max\{1 - e^{\epsilon}p_{\text{FN}}, e^{-\epsilon}(1 - p_{\text{FN}})\} \approx 0.94$, which is quite large. Equation 4 provides a guideline for choosing ϵ when implementing a differentially private mechanism. The error probabilities p_{FN} and p_{FP} are often more straightforward to specify than the level of privacy ϵ. Once the lower bounds for p_{FN} and p_{FP} are specified, one can choose ϵ accordingly from Equation 4.

2.3. Properties of Differential Privacy

Differential privacy enjoys several important properties. First, it is immune to postprocessing—that is, without any additional knowledge of the original database, no one can perform computation on the output of a differentially private mechanism and make the result less private. This property has an important direct consequence. Recall that the output of a differentially private mechanism is often released to the public, after which the released result can potentially be utilized arbitrarily by other parties. The immunity to postprocessing guarantees that no privacy can be further lost through handling by other parties. The property of immunity to postprocessing is formalized below.

Theorem 1 (postprocessing). Suppose a mechanism $M : \mathcal{D} \to \mathcal{Q}$ preserves ϵ-differential privacy (22). Then for any function f, the (functional) composition $f \circ M$ also preserves ϵ-differential privacy.

Next, we introduce composition rules that are often used to construct new differentially private mechanisms from existing ones. The sequential composition rule given in the following is useful when one needs to release multiple quantities computed from the same database.

Theorem 2 (sequential composition). Suppose a mechanism M_1 preserves ϵ_1-differential privacy, and another mechanism M_2 preserves ϵ_2-differential privacy. Define a new mechanism $M(D) \triangleq (M_1(D), M_2(D))$. Then the mechanism M preserves $(\epsilon_1 + \epsilon_2)$-differential privacy (22).

For example, in the case of the salary database, one may wish to release both the average and the standard deviation of the salaries. As implied by Theorem 2, one can design two differentially private mechanisms for the average and the standard deviation separately and later combine them with the level of privacy given by Theorem 2. Theorem 2 also reveals a fundamental fact about differential privacy: More privacy is lost as more queries are made to the same database and released to the public.

It should be noted that the privacy guarantee given by Theorem 2 can be loose and may not be the best guarantee. This is because Theorem 2 does not take into account the correlation between M_1 and M_2. Consider the trivial case where $M_1 = M_2$. In this case, introducing M_2 in the joint mechanism $M = (M_1, M_2)$ does not reveal any more information about the database than using M_1 alone. That is, the joint mechanism M should also be ϵ_1-differentially private, whereas Theorem 2 dictates that M is $2\epsilon_1$-differentially private.

For certain applications, the mechanism we would like to design is a result of the adaptive composition of several other mechanisms. This is common in iterative computation (such as in optimization algorithms), where the result of each step depends on the result from previous steps. The adaptive composition rule given in the following provides privacy guarantees for cases that involve iterative computation.

Theorem 3 (adaptive composition). Consider a mechanism $M_1 : \mathcal{D} \to \mathcal{Q}_1$ that preserves ϵ_1-differential privacy, and another mechanism $M_2 : \mathcal{D} \times \mathcal{Q}_1 \to \mathcal{Q}_2$ such that $M_2(\cdot, y_1)$ preserves ϵ_2-differential privacy for any $y_1 \in \mathcal{Q}_1$. Define a new mechanism $M(D) \triangleq M_2(D, M_1(D))$. Then the mechanism M preserves $(\epsilon_1 + \epsilon_2)$-differential privacy (22).

The adaptive composition rule generalizes the postprocessing rule (Theorem 1), because any function f that does not depend on the database can be treated as a 0-differentially private mechanism. It is also straightforward to see that the adaptive composition rule generalizes the sequential composition rule (Theorem 2). The postprocessing rule and composition rules also hold similarly for (ϵ, δ)-differential privacy. For the composition rules, the final privacy guarantee becomes $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$ when the original mechanisms are (ϵ_1, δ_1)- and (ϵ_2, δ_2)-differentially private.

2.4. Differentially Private Mechanisms

We now present some commonly used mechanisms that preserve differential privacy. This is by no means an exhaustive list of such mechanisms, nor are the mechanisms in the list optimal mechanisms (defined as ones that achieve the best accuracy—sometimes called utility—for a specified privacy level). Interested readers can refer to related work (27, 28) for discussions of optimal mechanisms.

2.4.1. Laplace mechanism. When the range of query $\mathcal{Q}$ is $\mathbb{R}$, one commonly used differentially private mechanism is the Laplace mechanism (13). This mechanism works by introducing additive noise drawn from the Laplace distribution. We denote by $\text{Lap}(b)$ the Laplace distribution with parameter b. For a random variable w, we write $w \sim \text{Lap}(b)$ if the probability density function of w is given by $p(w) = \frac{1}{2b} \exp(-|w|/b)$.

> **Theorem 4 (Laplace mechanism).** For a given query q with $\text{range}(q) = \mathbb{R}$, let $\Delta = \max_{D,D'} |q(D) - q(D')|$ be the sensitivity of q. Then the mechanism $M(D) = q(D) + w$ with $w \sim \text{Lap}(\Delta/\epsilon)$ preserves ϵ-differential privacy (22).

The Laplace mechanism reveals an intrinsic trade-off between the privacy and accuracy of the result. Indeed, without any system performance requirement, one can achieve perfect privacy by replacing the original query with a random output that is independent of the input. Notice that the mean squared error (MSE) of the Laplace mechanism is given by

$$\mathbb{E}[M(D) - q(D)]^2 = \text{var}(w) = \frac{2\Delta^2}{\epsilon^2}.$$

That is, as ϵ becomes smaller (i.e., more privacy is preserved), the result becomes less accurate. It is often important to analyze the trade-off between accuracy and privacy for a given application; we revisit this type of analysis in Sections 3 and 4 when we discuss the application of differential privacy to dynamical systems. To illustrate how the Laplace mechanism can be applied, we give the following simple example of computing the average salary while preserving differential privacy.

> **Example 3.** Consider the database of salaries given in Example 1, with the query $q(D) = \frac{1}{n}\sum_{i=1}^{n} d_i$ (average salary). Suppose $d_i \in [0, d_{\max}]$, where $d_{\max}$ is the maximum salary, and we use the adjacency relation with $|d_i - d_i'| \leq d_{\max}$ as in Example 2. Then the sensitivity of q can be obtained as follows:
>
> $$\Delta = \max_{D,D'} |q(D) - q(D')| = \frac{1}{n} \max_{i \in [n]} \max_{d_i, d_i'} |d_i - d_i'| = \frac{d_{\max}}{n}.$$
>
> From Theorem 4, we know that the (randomized) mechanism $M(D) = \frac{1}{n}\sum_{i=1}^{n} d_i + \text{Lap}\left(\frac{d_{\max}}{n\epsilon}\right)$ preserves ϵ-differential privacy. Notice that the magnitude of the Laplace noise is inversely proportional to the number of users in the database. In other words,

with more users in the database, we can introduce less noise in order to achieve the same privacy guarantee. This coincides with our intuition that it is easier to preserve individual privacy with more participating users.

The Laplace mechanism can be generalized to the multidimensional case for queries that lie in $\mathbb{R}^m$. Define the ℓ_p sensitivity Δ_p of query q as

$$\Delta_p \triangleq \max_{D,D'} \left\| q(D) - q(D') \right\|_p .$$

The following theorem shows how the Laplace mechanism is generalized for various common choices of p.

Theorem 5. For a given query q whose range is $\mathbb{R}^m$, let $\Delta_p = \max_{D,D'} \left\| q(D) - q(D') \right\|_p$ be the ℓ_p sensitivity of q. Then the mechanism $M(D) = q(D) + w$, where w is a random vector, preserves ϵ-differential privacy when

1. $p = 1$ [the distribution of each component of w is independent and identically distributed $\mathrm{Lap}(\Delta_1/\epsilon)$],
2. $p = 2$ [the probability distribution of w is proportional to $\exp(-\epsilon \|w\|_2 / \Delta_2)$], and
3. $p = \infty$ [the distribution of each component of w is independent and identically distributed $\mathrm{Lap}(k\Delta_\infty/\epsilon)$].

In the case of $p = 1$ and $p = \infty$ (when each component of w is independent and identically distributed), we also write the noise w as $w \sim \mathrm{Lap}(\Delta_1/\epsilon)^m$ and $w \sim \mathrm{Lap}(k\Delta_\infty/\epsilon)^m$. As a quick note, the result for $p = 1$ and $p = \infty$ can be derived immediately by making use of the sequential composition theorem (Theorem 2).

2.4.2. Gaussian mechanism. For the relaxed notion of (ϵ, δ)-differential privacy and when the query is numeric, a common choice is the Gaussian mechanism, which introduces additive Gaussian noise to the query.

Theorem 6 (Gaussian mechanism). For a given query q, let $\Delta_2 = \max_{D,D'} \left\| q(D) - q(D') \right\|_2$ be the ℓ_2 sensitivity of q. Then, for $\epsilon \in (0, 1)$ and $\delta > 0$, the mechanism $M(D) = q(D) + w$ preserves (ϵ, δ)-differential privacy when w is a random vector whose entries are independent and identically distributed zero-mean Gaussian with variance $\sigma^2 = \kappa^2(\delta, \epsilon)\Delta_2^2$, where $\kappa^2(\delta, \epsilon) = 2\log(1.25/\delta)/\epsilon^2$ (22).

The Gaussian mechanism indicates that, for certain applications (such as in linear systems), the notion of (ϵ, δ)-differential privacy may be favored over ϵ-differential privacy even with its weaker privacy guarantees. This is because the Gaussian mechanism often simplifies the analysis of the performance of the mechanism, owing to the fact that any linear transformation of a Gaussian random vector remains Gaussian.

2.4.3. Design of differentially private mechanisms. The Laplace and Gaussian mechanisms presented above are quite general and straightforward to implement. When implementing these algorithms, the most important quantity that one needs to compute is the sensitivity. Although the sensitivity can be easy to compute for simple queries (e.g., the average), it may be difficult to compute for complicated queries (e.g., the optimal solution of a nonlinear optimization problem). When the queries are complicated, a common strategy is to decompose the query under

investigation so that the sensitivity of each part of the query can be easily computed. For example, although the sensitivity of the optimal solution of a nonlinear optimization problem may not be easy to compute, the sensitivity of the intermediate results used to iteratively compute the optimal solution is often much easier to compute. Then the desired private mechanism can often be constructed by using the composition rules from Section 2.3.

3. DIFFERENTIALLY PRIVATE FILTERING

Filtering of streaming data is one of the first applications of differential privacy studied in the context of dynamical systems. Unlike the static databases that are traditionally studied in the database literature, the main challenge in filtering of streaming data is to define proper adjacency relations and compute the corresponding sensitivity. As shown below, once the sensitivity can be computed, it is relatively straightforward to make use of established results (e.g., the Laplace mechanism) from the database literature to develop private mechanisms for filtering streaming data. For linear time-invariant systems, sensitivity computation often reduces to computing the system norm of the dynamic filter. As a result, mathematical tools developed in control theory can not only quantify the level of privacy guarantee for a given filter but also provide better filter designs under a given privacy requirement.

We now introduce the notion of differential privacy for dynamic filtering used in this section. All signals are discrete-time signals indexed by t starting from $t = 0$; a signal x is an infinite sequence $x = (x(0), x(1), \ldots)$. For each time T, let P_T be the truncation operator, so that for any signal x we have

$$(P_T x)(t) = \begin{cases} x(t), & t \le T \\ 0, & t > T. \end{cases}$$

We denote by ℓ_p^m the space of sequences with values in $\mathbb{R}^m$ and such that $x \in \ell_p^m$ if and only if $P_T x$ has a finite p-norm for all integers T. All systems considered are assumed to be causal and linear time invariant. For a given dynamical system, we use uppercase calligraphic letters (e.g., $\mathcal{G}$) to represent both the system itself and the mathematical operator that maps the input to the output of the system; we use corresponding uppercase roman letters (e.g., G) to denote its transfer function. For a dynamical system $\mathcal{G}$, we say that $\mathcal{G}$ is causal if and only if $P_T\mathcal{G} = P_T\mathcal{G}P_T$. The $\mathcal{H}_2$- and $\mathcal{H}_\infty$-norms of a stable (matrix) transfer function G are defined as $\|G\|_2 = \left(\frac{1}{2\pi}\int_{-\pi}^{\pi} \mathrm{tr}(G^*(e^{i\omega})G(e^{i\omega}))\,d\omega\right)^{1/2}$ and $\|G\|_\infty = \sup_{\omega\in[-\pi,\pi)} \sigma_{\max}(G(e^{i\omega}))$, respectively, where $\sigma_{\max}(A)$ denotes the maximum singular value of a matrix A (29).

3.1. Private Mechanisms for General Filtering

We consider the case in which a mediator needs to aggregate information from n participants, as shown in **Figure 2**. Each participant $i \in [n]$ owns a dynamical system $\mathcal{G}_i : \ell_r^m \to \ell_s^{m'}$ whose input signal is $u_i \in \ell_p^m$ for some $m, m' \in \mathbb{N}$ and $p \in [1, \infty]$. The action of the mediator can be modeled by a system $\mathcal{G} : \ell_r^m \to \ell_s^{m'}$ defined by

$$\mathcal{G}(u_1, \ldots, u_n) = \sum_{i=1}^{n} \mathcal{G}_i u_i; \qquad 5.$$

that is, the mediator releases the sum of outputs from n dynamical systems. A simple example is that of a dynamic system releasing at each period the average over the past l periods of the sum of the input values of the participants, that is, with output $\frac{1}{l}\sum_{k=t-l+1}^{t}\sum_{i=1}^{n} u_i(k)$ at time t. We denote by $u = (u_1, \ldots, u_n)$ the collection of all input signals. An adjacency relation Adj can be

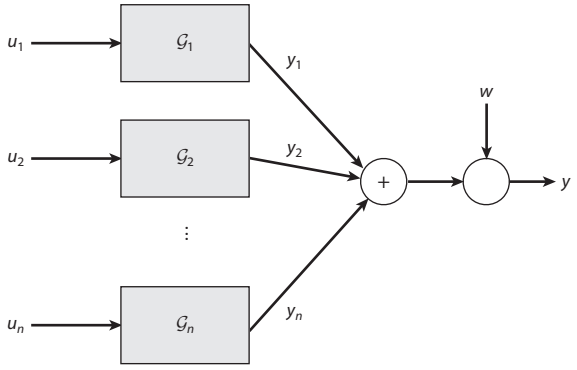

Figure 2

General differentially private filtering. Each participant $i \in [n]$ owns a dynamical system $\mathcal{G}_i$ whose input u_i contains sensitive information. In the absence of a privacy requirement, a mediator needs to collect outputs $\{y_i\}_{i=1}^n$ from all the systems and release the sum $\sum_{i=1}^n y_i$. To protect privacy, however, the mediator needs to introduce additive random perturbation w to the sum, where the magnitude of w depends on the level of privacy.

defined on the space of input signals as

$$\text{Adj}(u, u') \text{ iff for some } i, \|u_i - u_i'\|_p \le b, \text{ and } u_j = u_j' \text{ for all } j \ne i, \qquad 6.$$

for some nonnegative number $b \ge 0$. In plain words, $\text{Adj}(u, u')$ if and only if u and u' differ by exactly one component signal, and the difference for this component is bounded.

Recall that for a system $\mathcal{G}$ with input in ℓ_r^m and output in $\ell_s^{m'}$, its ℓ_r-to-ℓ_s gain $\gamma_{r,s}(\mathcal{G})$ (30) is defined as the smallest number γ such that

$$\|P_T \mathcal{G} v\|_s \le \gamma \|P_T v\|_r, \quad \forall v \in \ell_r^m, \ \forall T.$$

The next theorem generalizes the Laplace and Gaussian mechanisms in Theorems 5 and 6 to causal dynamic systems.

Theorem 7. Let $\mathcal{G}$ be defined as in Equation 5 and consider the adjacency relation shown in Equation 6. Then the mechanism $Mu = \mathcal{G}u + w$, where w is a white noise with $w(t) \sim \text{Lap}(B/\epsilon)^{m'}$ and $B \ge b \cdot \max_{1 \le i \le n}\{\gamma_{r,1}(\mathcal{G}_i)\}$, is ϵ-differentially private. The mechanism is (ϵ, δ)-differentially private if $w(t) \sim \mathcal{N}(0, \sigma^2 I_{m'})$, with $\sigma \ge b \cdot \kappa(\delta, \epsilon) \max_{1 \le i \le n}\{\gamma_{r,2}(\mathcal{G}_i)\}$ (16).

Proof. Consider two adjacent signals u, u' differing in their ith component. Then, for $\alpha \in \{1, 2\}$, we have

$$\begin{aligned} \|P_T \mathcal{G} u - P_T \mathcal{G} u'\|_\alpha &= \|P_T \mathcal{G}_i u_i - P_T \mathcal{G}_i u_i'\|_\alpha \le \gamma_{r,\alpha} \|P_T u_i - P_T u_i'\|_r \\ &\le \gamma_{r,\alpha} \|u_i - u_i'\|_r \le \gamma_{r,\alpha} b. \end{aligned}$$

Here we have used the fact that $\mathcal{G}_i$ is linear, so that $\mathcal{G}_i u_i - \mathcal{G}_i u_i' = \mathcal{G}_i(u_i - u_i')$. The above inequality leads to a bound on the ℓ_1 and ℓ_2 sensitivity of $P_T \mathcal{G}$, valid for all T. The theorem is then an application of Theorems 5 and 6. □

Later in this section, we further discuss the case when $r = 2$ and (ϵ, δ)-differential privacy is needed. In this case, the system gain can be computed from the $\mathcal{H}_\infty$-norm of the system as given in the following corollary.

Corollary 1. Let $\mathcal{G}$ be defined as in Equation 5 and $r = 2$. Then the mechanism $Mu = \mathcal{G}u + w$, where w is a white Gaussian noise with $w_t \sim \mathcal{N}(0, \sigma^2 I_{m'})$ and $\sigma \geq b \cdot \kappa(\delta, \epsilon) \max_{1 \leq i \leq n} \{\|\mathcal{G}_i\|_\infty\}$, is (ϵ, δ)-differentially private for Equation 6 (16).

3.2. Differentially Private Kalman Filtering

We now discuss the Kalman filtering problem subject to a differential privacy constraint. In this problem, the purpose of each dynamic filter $\mathcal{G}_i$ is to estimate the state of an autonomous stochastic system given by

$$x_i(t+1) = A_i x_i(t) + B_i w_i(t),$$
$$y_i(t) = C_i x_i(t) + D_i w_i(t).$$

Here, the state of the autonomous system is x_i and the output is y_i. The term w_i is a standard zero-mean Gaussian white noise with covariance

$$\mathbb{E}[w_i(t) w_i(t')] = \begin{cases} I & t = t' \\ 0 & t \neq t'. \end{cases}$$

The initial condition $x_i(0)$ is a Gaussian random variable with mean $\bar{x}_i(0)$, independent of the noise process w_i. The mediator aims at releasing a signal $\hat{z}$ that provides an optimal estimate of $z = \sum_{i=1}^n L_i x_i$ from y_i for some given matrices L_i. The optimal estimator $\hat{z}$ needs to asymptotically minimize the MSE; that is, we are looking for the solution of

$$\min_{\hat{z}} \quad \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \left\| z(t) - \hat{z}(t) \right\|_2^2.$$

For all $i \in [n]$, we assume that the data $\bar{x}_i(0), A_i, B_i, C_i, D_i, L_i$ are public information; that the pairs (A_i, C_i) are detectable; and that the pairs (A_i, B_i) are stabilizable. In the absence of a privacy constraint, the optimal estimator is given by $\hat{z} = \sum_{i=1}^n L_i \hat{x}_i$, with $\hat{x}_i = \mathcal{K}_i y_i$ provided by the steady-state Kalman filter $\mathcal{K}_i$ estimating the state of system i from y_i (31).

Suppose that the state trajectory x_i contains sensitive information, and we need to develop a mechanism that releases an estimate of z while protecting the privacy of the participants. We first specify an adjacency relation on the space of databases consisting of state trajectories. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ denote the global state and measurement trajectories. In this application, our goal is to guarantee differential privacy with respect to a subset $\mathcal{S}_i \triangleq \{i_1, \ldots, i_k\}$ of the coordinates of the state trajectory x_i. Let the selection matrix S_i be the diagonal matrix with $[S_i]_{jj} = 1$ if $j \in \mathcal{S}_i$, and $[S_i]_{jj} = 0$ otherwise. $S_i v$ therefore sets the coordinates of a vector v that do not belong to the set $\mathcal{S}_i$ to zero. The adjacency relation considered here is

$$\begin{aligned} \mathrm{Adj}_{\mathcal{S}}^{\rho}(x, x') \text{ iff for some } i,\ \|S_i x_i - S_i x_i'\|_2 \leq \rho,\ (I - S_i) x_i = (I - S_i) x_i', \\ \text{and } x_j = x_j' \text{ for all } j \neq i \end{aligned} \qquad 7.$$

for some fixed $\rho > 0$. In other words, two adjacent databases of state trajectories differ by x_i from a single participant i. In addition, the difference can happen only in the coordinates $\mathcal{S}_i$ with the amount of deviation bounded by ρ.

The first differentially private mechanism works by directly adding output noise to the original estimator. We first compute the ℓ_2 sensitivity of the estimator $\hat{z}$. Consider now two state trajectories

x, x', adjacent according to Equation 7, and let $\hat{z}, \hat{z}'$ be the corresponding estimates produced by the Kalman filters. We have

$$\hat{z} - \hat{z}' = L_i \mathcal{K}_i (y_i - y_i') = L_i \mathcal{K}_i C_i S_i (x_i - x_i') = L_i \mathcal{K}_i C_i S_i S_i (x_i - x_i'),$$

where we recall that $\mathcal{K}_i$ is the Kalman filter from user i. Therefore, $\|\hat{z} - \hat{z}'\|_2 \leq \gamma_i \rho$ for all $i \in [n]$, where γ_i is the $\mathcal{H}_\infty$-norm of the system $L_i \mathcal{K}_i C_i S_i$. We thus have the following theorem.

Theorem 8. A mechanism releasing $\left(\sum_{i=1}^n L_i \mathcal{K}_i y_i\right) + \gamma\rho\ \kappa(\delta, \epsilon)\ \nu$, where ν is a standard white Gaussian noise independent of $\{w_i\}_{i=1}^n$, $\{x_i(0)\}_{i=1}^n$, and $\gamma = \max_{i \in [n]}\{\gamma_i\}$, with γ_i the $\mathcal{H}_\infty$-norm of $L_i \mathcal{K}_i C_i S_i$, is (ϵ, δ)-differentially private for the adjacency relation represented by Equation 7 (16).

Although the Kalman filter is the optimal estimator (in the sense of minimum MSE) in the absence of privacy constraints, it is no longer optimal with the additive Gaussian noise used for ensuring privacy. In the second private mechanism, we show that the filter can be redesigned to achieve better MSE performance. Formally, we consider the design of n filters of the form

$$\begin{aligned} \hat{x}_i(t+1) &= F_i \hat{x}_i(t) + G_i y_i(t), \\ \hat{z}_i(t) &= H_i \hat{x}_i(t) + K_i y_i(t), \end{aligned}$$

for $i \in [n]$, where F_i, G_i, H_i, K_i are matrices to determine. The estimator considered is $\hat{z} = \sum_{i=1}^n \hat{z}_i$, so that each filter output $\hat{z}_i$ should minimize the steady-state MSE with $z_i = L_i x_i$, and the released signal should guarantee differential privacy with respect to Equation 7. For simplicity, we assume that the system matrices A_i are stable, in which case we also restrict the filter matrices F_i to be stable (for the case of unstable systems, see Reference 16). Moreover, we consider only the design of full order filters—that is, filters in which the dimensions of F_i are greater than or equal to those of A_i, for all $i \in [n]$.

Define $\tilde{x}_i \triangleq (x_i, \hat{x}_i)$ so that the combined dynamics from w_i to the estimation error $e_i \triangleq z_i - \hat{z}_i$ can be written as

$$\begin{aligned} \tilde{x}_i(t+1) &= \tilde{A}_i \tilde{x}_i(t) + \tilde{B}_i w_i(t), \\ e_i(t) &= \tilde{C}_i \tilde{x}_i(t) + \tilde{D}_i w_i(t), \end{aligned}$$

where

$$\tilde{A}_i = \begin{bmatrix} A_i & 0 \\ G_i C_i & F_i \end{bmatrix}, \quad \tilde{B}_i = \begin{bmatrix} B_i \\ G_i D_i \end{bmatrix}, \quad \tilde{C}_i = \begin{bmatrix} L_i - K_i C_i & -H_i \end{bmatrix}, \quad \tilde{D}_i = -K_i D_i.$$

The total MSE consists of two terms. One is from the steady-state MSE $\lim_{t \to \infty} \mathbb{E}[e_i(t)^T e_i(t)]$ of the ith estimator and is given by

$$\|\tilde{C}_i (zI - \tilde{A}_i)^{-1} \tilde{B}_i + \tilde{D}_i\|_2^2.$$

The other term is from noise to ensure privacy and depends on the $\mathcal{H}_\infty$-norm of the filters. Considering two adjacent state trajectories x and x' according to Equation 7 and defining $\delta x_i = x_i - x_i' = S_i(x_i - x_i') = S_i \delta x_i$, we see that the change in the output of filter i follows the dynamics

$$\begin{aligned} \delta\hat{x}_i(t+1) &= F_i \delta\hat{x}_i(t) + G_i C_i S_i \delta x_i, \\ \delta\hat{z}_i &= H_i \delta\hat{x}_i(t) + K_i C_i S_i \delta x_i. \end{aligned}$$

Therefore, the ℓ_2 sensitivity γ_i of the ith filter can be measured by the $\mathcal{H}_\infty$-norm of the transfer function

$$\left[\begin{array}{c|c} F_i & G_i C_i S_i \\ \hline H_i & K_i C_i S_i \end{array}\right].$$

That is, we have

$$\gamma_i = \|H_i(zI - F_i)^{-1} G_i C_i S_i + K_i C_i S_i\|_\infty$$

and the corresponding MSE to be $\rho\kappa(\delta,\epsilon)^2 \max_{i\in[n]}\{\gamma_i^2\}$ for the noise that guarantees (ϵ,δ)-differential privacy. To summarize, the problem of optimal filter design that minimizes the total MSE can be written as follows:

$$\text{min.} \quad \sum_{i=1}^{n} \mu_i + \kappa(\delta,\epsilon)^2 \lambda \qquad (\text{over } \mu_i, \lambda, F_i, G_i, H_i, K_i) \tag{8.}$$

$$\text{s.t.} \quad \|\tilde{C}_i(zI - \tilde{A}_i)^{-1}\tilde{B}_i + \tilde{D}_i\|_2^2 \le \mu_i, \tag{9.}$$

$$\rho^2 \|H_i(zI - F_i)^{-1} G_i C_i S_i + K_i C_i S_i\|_\infty^2 \le \lambda, \quad i \in [n]. \tag{10.}$$

It can be shown (16) that the constraints represented in Equations 9 and 10 can be relaxed into linear matrix inequalities (32). In other words, Equation 8 can be relaxed into a semidefinite program so that the optimal filter matrices F_i, G_i, H_i, K_i can be obtained efficiently using convex optimization solvers.

4. DIFFERENTIALLY PRIVATE DISTRIBUTED OPTIMIZATION

Another application of differential privacy in the field of control and dynamical systems is optimization algorithms. Many optimization algorithms, including various versions of gradient methods and Newton's method, can be viewed as autonomous (nonlinear) dynamical systems. (For recent views on the connection between control theory and optimization algorithms, see Reference 33.) In certain applications, the input to an optimization algorithm often contains sensitive data from participants. For example, in energy demand response, the optimal pricing is computed by solving an optimization problem defined from individual utility functions representing the energy usage pattern of the participants (34). In most cases, the participants do not wish to reveal their utility function to the public. In general, the sensitive information can appear in the optimization problem in one of two ways: in the objective function or in the constraints of the problem.

Differential privacy has been applied to optimization problems in both centralized and distributed settings. In the centralized setting, data from participants are first collected by a trusted mediator, who is responsible for solving the optimization problem in a way that preserves privacy. In the framework of differential privacy, the query corresponds to the result of the optimization problem, which can be the optimal value or the optimal solution. The centralized setting is mostly used in machine learning. For example, the training of a support vector machine classifier requires solving an optimization problem defined from the training data set. On the other hand, in the distributed setting, each participant keeps its own data without sending them to a central mediator. The distributed setting is more common in the field of control, particularly control of multiagent systems (35). In order to solve the optimization problem, each participant needs to iteratively perform local computation and exchange the results by passing messages with either other participants or a central mediator. Although data are kept by the participants, the results of local computation still depend on each participant's data and may reveal sensitive information. In the framework of differential privacy, the query corresponds to the messages being passed

throughout the entire execution of the optimization algorithm. In this section, we focus on the distributed setting and discuss two cases: one in which the sensitive information appears in the objective function and one in which it appears in the constraints.

4.1. Private Optimization with Sensitive Objective Functions

In this section, we consider a group of n users who attempt to solve the following optimization problem collectively:

$$\min_{x\in\mathcal{X}} \quad \sum_{i=1}^{n} f_i(x). \tag{11}$$

In this optimization problem, the objective function is the sum of n cost functions $\{f_1, f_2, \ldots, f_n\}$. Each cost function f_i is the cost function of user i and must be kept private; the set $\mathcal{X}$ is the domain of the optimization problem and is considered public knowledge. We make the following assumptions about f_i and $\mathcal{X}$:

1. The set $\mathcal{X}$ is compact and convex. We denote by $C_1 \triangleq \sup_{x,y\in\mathcal{X}} \|x - y\|$ the diameter of $\mathcal{X}$.
2. Each function f_i is differentiable.
3. The gradient of f_i is uniformly bounded: There exists $C_2 > 0$ such that $\|\nabla f_i(x)\| \le C_2$ for all $x \in \mathcal{X}$.
4. Each function f_i is strongly convex: There exists $C_3 > 0$ such that $\nabla f_i(x)^T(y - x) \le f_i(y) - f_i(x) - \frac{C_3}{2}\|y - x\|^2$ for all $x, y \in \mathcal{X}$.

These are standard assumptions that guarantee the existence of an optimal solution and guarantee that gradient methods can be applied effectively (36). In this section, we use the shorthand notation $\|\cdot\|$ to denote the ℓ_2-norm $\|\cdot\|_2$.

> **Example 4 (consensus).** Suppose that each of the n users keeps a (sensitive) quantity, which we denote by $v_i \in \mathbb{R}^m$ for user i. The goal of the consensus problem is for all users to reach an agreement that corresponds to the average $\bar{v} = \sum_{i=1}^{n} v_i/n$. If we set $f_i(x) = \|x - v_i\|^2$, then it can be verified that the optimal solution x^* of Equation 11 satisfies $x^* = \bar{v}$. Note that f_i contains the sensitive local information v_i.

The general goal of distributed optimization is to solve Equation 11 without requiring a central mediator to collect all the sensitive cost functions f_i. In distributed optimization, each user needs to communicate with other users interactively and eventually converge to an optimal solution of the problem. Among various distributed optimization methods, we focus on distributed projected gradient methods for their simplicity. Without any privacy considerations, one version of distributed projected gradient methods is presented in Algorithm 1. The matrix $A = [a_{ij}] \in \mathbb{R}^{n\times n}$, called the connectivity matrix, is a nonnegative matrix that captures how users communicate as the optimization algorithm is being executed. If user i is able to receive information from user j, then $a_{ij} > 0$; otherwise $a_{ij} = 0$. The connectivity matrix needs to satisfy several requirements:

1. The matrix A is doubly stochastic: $\sum_{j=1}^{n} a_{ij} = 1$ for all i and $\sum_{i=1}^{n} a_{ij} = 1$ for all j.
2. The matrix A is irreducible. This implies that the communication graph represented by A is strongly connected.
3. There exists $\eta \in (0, 1]$ such that $a_{ij} \ge \eta$ if user i is able to receive information from user j. As a convention, each user is able to communicate with itself, so that we also have $a_{ii} \ge \eta$ for all i.

The sequence $\{\gamma_t\}_{t=0}^{T-1}$ is called step sizes, which need to satisfy $\gamma_t \to 0$ as $t \to \infty$ (for discussions on how to choose γ_t, see Reference 36). In the following, we choose

$$\gamma_t = cq^t \tag{12.}$$

for some $c > 0$ and $q \in (0, 1)$. The operator $\text{proj}_{\mathcal{X}}$ is the (Euclidean) projection onto the (convex) set $\mathcal{X}$. Under the assumptions on A, Algorithm 1 is guaranteed to converge to the optimal solution x^* of Equation 11 as $T \to \infty$.

Algorithm 1 (distributed projected gradient method). *Input*: (arbitrary) initial conditions $x_i(0) \in \mathbb{R}^m$ $(i = 1, 2, \ldots, n)$, connectivity matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$, number of iterations T, and step sizes $\{\gamma_t\}_{t=0}^{T-1}$.
Output: $x_i(T)$ $(i = 1, 2, \ldots, n)$.

for $t = 0, 1, \ldots, T - 1$ **do**

Sending: For each user i, send $x_i(t)$ to user j if $a_{ji} > 0$.

Receiving: For each user i, compute

$$\begin{aligned} z_i(t) &= \sum_{j=1}^{n} a_{ij} x_j(t), \\ x_i(t+1) &= \text{proj}_{\mathcal{X}} \left[z_i(t) - \gamma_t \nabla f_i(z_i(t)) \right]. \end{aligned} \tag{13.}$$

end for

As can be seen in Algorithm 1, each user needs to communicate its iterates $x_i(t)$ with other users. Notice that the computation of $x_i(t)$ depends on the gradient ∇f_i. As a result, directly sending $x_i(t)$ to other users may reveal sensitive information contained in f_i to the recipients or potential adversaries who can eavesdrop on the communication channel. Like previous settings in differential privacy, our goal is to develop a private mechanism $M = (M_i(\cdot, t))$ $(i = 1, 2, \ldots, n;$ $t = 1, 2, \ldots, T)$ for each user to publish its iterates $x_i(t)$. Algorithm 2 shows how the private mechanism is integrated into the original algorithm. **Figure 3*a*** illustrates the architecture of the private distributed optimization algorithm.

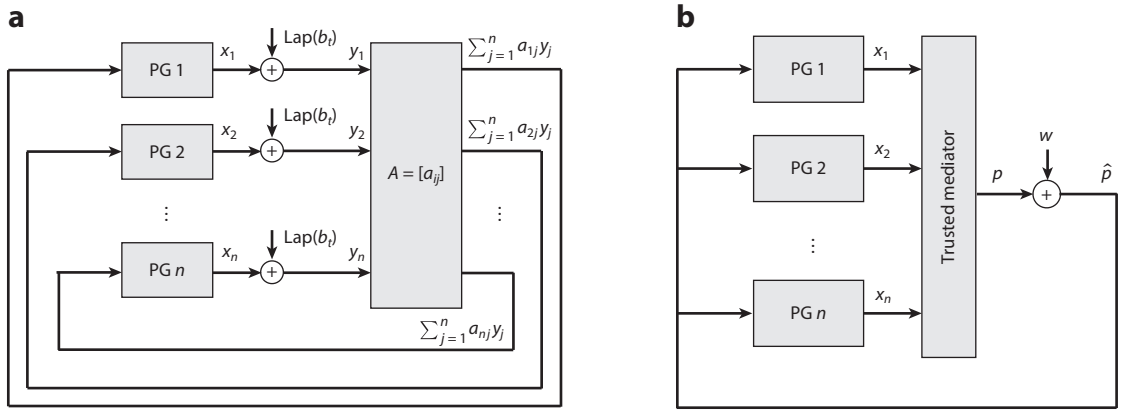

Figure 3

Differentially private distributed optimization. (*a*) Optimization with sensitive objective functions. Each user applies a local projected gradient (PG) update rule using information from other users as specified by the communication matrix $A = [a_{ij}]$. (*b*) Optimization with sensitive constraints. The trusted mediator aggregates user information and broadcasts the gradient for each user to apply the update rule.

Algorithm 2 (private distributed projected gradient method, meta-algorithm only). *Input*: (arbitrary) initial conditions $x_i(0) \in \mathbb{R}^m$ $(i = 1, 2, \ldots, n)$, connectivity matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$, number of iterations T, and step sizes $\{\gamma_t\}_{t=0}^{T-1}$. *Output*: $x_i(T)$ $(i = 1, 2, \ldots, n)$.

for $t = 0, 1, \ldots, T - 1$ **do**

Sending: For each user i, compute $y_i(t) = M_i(x_i(t), t)$ through a (possibly time-dependent) private mechanism $M_i(\cdot, t)$, then send $y_i(t)$ to user j if $a_{ji} > 0$.

Receiving: For each user i, compute

$$z_i(t) = \sum_{j=1}^{n} a_{ij} y_j(t),$$
$$x_i(t+1) = \mathrm{proj}_{\mathcal{X}} \left[z_i(t) - \gamma_t \nabla f_i(z_i(t)) \right].$$

end for

Before introducing the private mechanisms, we formally define the privacy guarantees that we wish to obtain under the framework of differential privacy. Since the sensitive information is in the individual cost functions, the database D can be defined as $D = \{f_i\}_{i=1}^n$. We use the following adjacency relation defined between two databases D and D'.

Definition 4 (adjacency relation for objective functions). We say that two databases $D = \{f_i\}_{i=1}^n$ and $D' = \{f_i'\}_{i=1}^n$ are adjacent if there exists $i \in \{1, 2, \ldots, n\}$ such that $f_j = f_j'$ for all $j \neq i$.

In Definition 4, we did not define the deviation between f_i and f_i' for the following reason. Recall that the sensitive information can be revealed only through the gradient ∇f_i. Therefore, only the deviation between ∇f_i and $\nabla f_i'$ will affect the design of private mechanisms. It is not difficult to see that a bound on this deviation can be derived from the assumption on ∇f_i (related to the constant C_2). With the defined adjacency relation, we formally state the problem of ϵ-differentially private distributed optimization (with sensitive objective functions) as follows.

Problem 1 (private optimization with sensitive objectives). For a given $\epsilon > 0$, find a mechanism $M = (M_i(\cdot, t))$ $(i = 1, 2, \ldots, n;\ t = 1, 2, \ldots, T)$ that maps $x_i(t)$ to $y_i(t) = M_i(x_i(t), t)$ (in Algorithm 2) such that, for any adjacent databases of functions $D = \{f_i\}_{i=1}^n$ and $D' = \{f_i'\}_{i=1}^n$ as defined in Definition 4, any initial condition $x(0)$, and any set $\mathcal{R} \subseteq \mathrm{range}(M)$, we have

$$\mathbb{P}(M(D) \in \mathcal{R}) \leq e^{\epsilon} \mathbb{P}(M(D') \in \mathcal{R}).$$

As mentioned above, the key to developing a private mechanism is computing the sensitivity of the query $(x(0), x(1), \ldots, x(T))$ to changes in the database D. In order to make use of the adaptive composition theorem, we need to define the sensitivity at time t while assuming that the previous output of the mechanism is identical. Specifically, we define the sensitivity at t as

$$\Delta(t) \triangleq \sup_{D, D', y} \left\| x(t; D, y(0), \ldots, y(t-1)) - x(t; D', y(0), \ldots, y(t-1)) \right\|_1,$$

where we have used the notation $x(t; D, y(0), \ldots, y(t-1))$ to indicate the states $x(t)$ when the database of cost functions is D and the previous output of the mechanism from 0 to $t - 1$ is $(y(0), \ldots, y(t-1))$. Recall that the definition of the adjacency relation requires $f_j = f_j'$ for all

$j \neq i$. Therefore, we have

$$\Delta(t) = \sup_{f_i, f_i'} \left\| \text{proj}_{\mathcal{X}} \left[z_i(t) - \gamma_t \nabla f_i(z_i(t)) \right] - \text{proj}_{\mathcal{X}} \left[z_i(t) - \gamma_t \nabla f_i'(z_i(t)) \right] \right\|_1,$$

where we have used the fact that $y(t-1)$ [and hence $z(t-1)$] is fixed when $\Delta(t)$ is evaluated. Instead of bounding the ℓ_1-norm directly, we derive by bounding the ℓ_2-norm

$$\begin{aligned} &\left\| \text{proj}_{\mathcal{X}} \left[z_i(t) - \gamma_t \nabla f_i(z_i(t)) \right] - \text{proj}_{\mathcal{X}} \left[z_i(t) - \gamma_t \nabla f_i'(z_i(t)) \right] \right\| \\ &\quad \leq \left\| \left[z_i(t) - \gamma_t \nabla f_i(z_i(t)) \right] - \left[z_i(t) - \gamma_t \nabla f_i'(z_i(t)) \right] \right\| \\ &\quad = \gamma_t \left\| \nabla f_i(z_i(t)) - \nabla f_i'(z_i(t)) \right\| \\ &\quad \leq 2C_2 \gamma_t. \end{aligned}$$

The first inequality used a property of the projection operator: $\|\text{proj}_{\mathcal{X}}(u) - \text{proj}_{\mathcal{X}}(v)\| \leq \|u - v\|$ for all u and v. The bound on the ℓ_2-norm implies

$$\Delta(t) \leq 2C_2 \sqrt{n} \gamma_t.$$

With a bound on the sensitivity, we have the following private mechanism that guarantees ϵ-differential privacy.

Theorem 9. The distributed optimization algorithm (Algorithm 2) is ϵ-differentially private when each $M_i(\cdot, t)$ is the Laplace mechanism $\text{Lap}(b_t)^m$ with

$$b_t = 2C_2 \sqrt{n} \frac{cp}{\epsilon(p-q)} p^t$$

for any $p \in (q, 1)$ when the step size γ_t is chosen according to Equation 12 (23). The constant C_2 is in the assumption on the bound of ∇f_i.

Proof. From the adaptive composition theorem (Theorem 3), we know that Algorithm 2 is ϵ-differentially private when each $M_i(\cdot, t)$ is a Laplace mechanism with parameter b_t and when b_t satisfies $\sum_{t=0}^{T-1} \Delta(t)/b_t \leq \epsilon$, which can be readily verified. □

Although introducing noise into the original distributed optimization algorithm (Algorithm 1) provides privacy guarantees, it also affects the convergence of the algorithm. The following result ensures that the private version of the algorithm still converges (in a probabilistic sense) as $T \to \infty$.

Theorem 10. The output of Algorithm 2 satisfies (23)

$$\lim_{t \to \infty} \mathbb{E} \left\| x_i(t) - x_j(t) \right\| = 0.$$

Theorem 10 implies that all $x_i(t)$ converge to the same value as $t \to \infty$. For convenience, we define

$$\bar{x}(t) \triangleq \frac{1}{n} \sum_{i=1}^{n} x_i(t);$$

Theorem 10 then ensures $\mathbb{E} \left\| x_i(t) - \bar{x}(t) \right\| \to 0$ for all i. Although Algorithm 2 still converges with the introduction of additive noise, it is no longer guaranteed to converge to the optimal solution x^* of Equation 11. The accuracy, measured by the deviation between $\bar{x}(t)$ and x^*, of Algorithm 2 is given by the following result.

Theorem 11. Suppose the optimal solution of Equation 11 is x^*, the step size is chosen according to Equation 12, and the private mechanism from Theorem 9 is used. Then we have (23)

$$\lim_{t\to\infty} \mathbb{E}\left\|\bar{x}(t) - x^*\right\|^2 \le C_1 \exp\left(-\frac{C_3 c}{1-q}\right) + \frac{C_2^2 c^2}{1-q^2} + \frac{8C_2^2 n c^2 p^2}{\epsilon^2 (p-q)^2 (1-p^2)}.$$

The constants C_1, C_2, and C_3 appeared in the assumptions on $\mathcal{X}$ and f_i. If p, q, and c are fixed, the bound on accuracy depends on the level of privacy ϵ as $O(1/\epsilon^2)$. The result shows the trade-off between privacy and accuracy. As the privacy protection requirement tightens (i.e., as ϵ decreases), Algorithm 11 becomes less accurate. Nozari et al. (37) showed that this trade-off between privacy and accuracy exists as long as additive noise is introduced to the states x_i during communication.

4.2. Private Optimization with Sensitive Constraints

We consider a constrained optimization problem over n variables $x_1, x_2, \ldots, x_n \in \mathbb{R}^m$ in the following form:

$$\begin{aligned} \min_{\{x_i\}_{i=1}^n} \quad & f\left(\sum_{i=1}^n x_i\right) \\ \text{s.t.} \quad & x_i \in \mathcal{C}_i, \quad i \in [n]. \end{aligned} \tag{14.}$$

Throughout this review, we assume that the objective function $f : \mathbb{R}^m \to \mathbb{R}$ in Equation 14 is differentiable and convex, and its gradient ∇f is L-Lipschitz in the ℓ_2-norm—that is, there exists $L > 0$ such that

$$\left\|\nabla f(x) - \nabla f(y)\right\| \le L \left\|x - y\right\| \quad \text{for all } x, y.$$

The set $\mathcal{C}_i$ is assumed to be convex for all $i \in [n]$. Equation 14 is common in resource allocation problems, in which the variable x_i and the constraint set $\mathcal{C}_i$ are used to capture the allocation and constraints on the allocation for user i. In this setup, the objective function f is public, whereas the sensitive information is contained in the constraints $\mathcal{C}_i$. The following example illustrates a case in which $\mathcal{C}_i$ contains sensitive information.

Example 5. We consider an application in which n electric vehicles (EVs) need to be charged. The goal is to charge all vehicles over a horizon of m time steps with minimal influence on the power grid. For simplicity, we assume that each vehicle belongs to a single user. For any $i \in [n]$, the vector $x_i \in \mathbb{R}^m$ represents the charging rates of vehicle i over time. In the following, we denote by $x_i^{(j)}$ the jth component of x_i. Each vehicle needs to be charged a given amount of electricity $E_i > 0$ by the end of the scheduling horizon; in addition, for any $j \in [m]$, the charging rate $x_i^{(j)}$ cannot exceed the maximum rate $\bar{x}_i^{(j)}$ for some given constant vector $\bar{x}_i \in \mathbb{R}^m$. Under these constraints on x_i, the set $\mathcal{C}_i$ is described as follows:

$$0 \preceq x_i \preceq \bar{x}_i, \qquad \mathbf{1}^T x_i = E_i,$$

where the notation $\preceq$ denotes entrywise inequality.

The objective function f in Equation 14 can be used to quantify the influence of a charging schedule $\{x_i\}_{i=1}^n$ on the power grid. For instance, we can choose f as follows

for the purpose of minimizing load variance:

$$f\left(\sum_{i=1}^{n} x_i\right) = \frac{1}{2}\left\| d + \sum_{i=1}^{n} x_i/K \right\|^2. \qquad 15.$$

In Equation 15, K is the number of households, which is assumed to be proportional to the number of EVs—that is, there exists γ such that $n/K = \gamma$. The quantity $\sum_{i=1}^{n} x_i/K$ then becomes the aggregate EV load per household. The vector $d \in \mathbb{R}^m$ is the base load profile per household incurred by loads in the power grid other than EVs, so that $f\left(\sum_{i=1}^{n} x_i\right)$ quantifies the variation of the total load, including the base load and EVs. It can be verified that U is convex and differentiable and that ∇U is Lipschitz continuous.

The set $\mathcal{C}_i$ (defined by $\bar{x}_i$ and E_i) contains sensitive information because it can be associated with personal activities of the owner of vehicle i. For example, $\bar{x}_i^{(j)} = 0$ may indicate that the owner is temporarily away from the charging station (which may be colocated with the owner's residence), and so the vehicle is not ready to be charged. Similarly, $E_i = 0$ may indicate that the owner is not actively using the vehicle, and so the vehicle does not need to be charged.

Without privacy requirements, Equation 14 can be solved in a distributed way using a projected gradient method (Algorithm 3). This algorithm assumes that there exists a central mediator who collects the states $x_i(t)$ $(i = 1, 2, \ldots, n)$ and computes the gradient $p(t)$, which is then broadcast to users for them to update their states according to Equation 16. However, this algorithm is not private, because the broadcast $p(t)$ depends on $x_i(t)$, which contains information about the sensitive constraint $\mathcal{C}_i$ owing to Equation 16.

Algorithm 3 (distributed projected gradient method for solving Equation 14). *Input*: f, $\{\mathcal{C}_i\}_{i=1}^n$, T, and step sizes $\{\alpha_t\}_{t=0}^{T-1}$. *Output*: $x(T) = (x_1(T), x_2(T), \ldots, x_n(T))$.

Initialize $x(0)$ arbitrarily.

for $t = 0, 1, \ldots, T-1$ **do**

Compute $p(t) = \nabla f\left(\sum_{i=1}^{n} x_i(t)\right)$.

For $i \in [n]$, update $x_i(t+1)$ according to

$$x_i(t+1) = \mathrm{proj}_{\mathcal{C}_i}\left[x_i(t) - \alpha_t p(t)\right]. \qquad 16.$$

end for

To ensure privacy, our goal is to develop a mechanism for releasing $p(t)$, so that potential adversaries do not gain much knowledge about user information even with access to $p(t)$. To mathematically formulate our goal under the framework of differential privacy, we define the database D as the set $\{\mathcal{C}_i\}_{i=1}^n$ and the query as the T-tuple consisting of all the gradients $p = (p(0), p(1), \ldots, p(T-1))$. Without loss of generality, we consider the case where $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n$ belong to a family of sets parameterized by $\beta \in \mathbb{R}^s$. That is, there exists a parameterized set $\mathcal{C}$ such that for all $i \in [n]$, we can write $\mathcal{C}_i = \mathcal{C}(\beta_i)$ for some $\beta_i \in \mathbb{R}^s$. We also assume that there exists a metric $\rho : \mathbb{R}^s \times \mathbb{R}^s \to \mathbb{R}_+$, so that we can define the distance $\rho_{\mathcal{C}}(\mathcal{C}_i, \mathcal{C}_i')$ between any $\mathcal{C}_i = \mathcal{C}(\beta_i)$ and $\mathcal{C}_i' = \mathcal{C}(\beta_i')$ using the metric ρ as

$$\rho_{\mathcal{C}}(\mathcal{C}_i, \mathcal{C}_i') \triangleq \rho(\beta_i, \beta_i').$$

For any given $\delta\mathcal{C} \in \mathbb{R}_+$, we define the following adjacency relation between any two databases D and D' in the context of distributed constrained optimization.

Definition 5 (adjacency relation for constraints). For any databases $D = \{\mathcal{C}_i\}_{i=1}^n$ and $D' = \{\mathcal{C}_i'\}_{i=1}^n$, it holds that $\mathrm{Adj}(D, D')$ if and only if there exists $i \in [n]$ such that $\rho_{\mathcal{C}}(\mathcal{C}_i, \mathcal{C}_i') \leq \delta\mathcal{C}$, and $\mathcal{C}_j = \mathcal{C}_j'$ for all $j \neq i$.

With this adjacency relation defined, we state in the following the problem of designing a differentially private distributed algorithm for constrained optimization.

Problem 2 (private optimization with sensitive constraints). Find a randomized mechanism M_p that approximates the gradients $p = (p(0), p(1), \ldots, p(T-1))$ (defined in Algorithm 3) and preserves ϵ-differential privacy under the adjacency relation described in Definition 5. That is, for any adjacent databases D and D' and any $\mathcal{R} \subseteq \mathrm{range}(M_p)$, the mechanism M_p should satisfy

$$\mathbb{P}(M_p(D) \in \mathcal{R}) \leq e^{\epsilon}\mathbb{P}(M_p(D') \in \mathcal{R}).$$

Similarly to the case when the sensitive information is in the objective functions, a private version of Algorithm 3 can be derived by introducing Laplace noise to the gradients p; this is presented as Algorithm 4, the architecture of which is illustrated in **Figure 3*b***. The constant Δ that appears in the algorithm is defined as

$$\Delta \triangleq \max_{i\in[n]} \max\left\{ \left\| \Pi_{\mathcal{C}_i}(v) - \Pi_{\mathcal{C}_i'}(v) \right\| : v \in \mathbb{R}^m,\ \mathcal{C}_i \text{ and } \mathcal{C}_i' \text{ satisfy } \rho_{\mathcal{C}}(\mathcal{C}_i, \mathcal{C}_i') \leq \delta\mathcal{C} \right\}.$$

In other words, Δ can be viewed as a bound on the global ℓ_2 sensitivity of the projection operator $\Pi_{\mathcal{C}_i}$ to changes in $\mathcal{C}_i$ for all $i \in [n]$.

Algorithm 4 (differentially private distributed projected gradient method). *Input*: f, L, $\{\mathcal{C}_i\}_{i=1}^n$, T, $\{\alpha_t\}_{t=0}^{T-1}$, $\eta \geq 1$, Δ, and ϵ. *Output*: $\hat{x}(T) = (\hat{x}_1(T), \hat{x}_2(T), \ldots, \hat{x}_n(T))$.

Initialize $x(0)$ arbitrarily. Let $\hat{x}(0) = x(0)$ and $\theta_t = (\eta + 1)/(\eta + t)$.

for $t = 0, 1, \ldots, T-1$ **do**

If $t = 0$, then set $w_t = 0$; else draw a random vector $w_t \in \mathbb{R}^m$ from the distribution (proportional to) $\exp\left(-\frac{2\epsilon\|w_t\|}{T(T+1)L\Delta}\right)$.

Compute $\hat{p}(t) := \nabla f\left(\sum_{i=1}^n x_i(t)\right) + w_t$.

For $i \in [n]$, compute

$$\begin{aligned} x_i(t+1) &= \mathrm{proj}_{\mathcal{C}_i}\left[x_i(t) - \alpha_t \hat{p}(t)\right], \\ \hat{x}_i(t+1) &= (1-\theta_t)\hat{x}_i(t) + \theta_t x_i(t+1). \end{aligned} \tag{17.}$$

end for

Theorem 12. Algorithm 4 ensures that $M_p \triangleq (\hat{p}(0), \hat{p}(1), \ldots, \hat{p}(T-1))$ preserves ϵ-differential privacy under the adjacency relation given by Definition 5 (25).

Algorithm 4 can be viewed as an instance of a stochastic gradient method that terminates after T iterations. The step size α_t is chosen as $\alpha_t = c/\sqrt{t}$ for some $c > 0$. The purpose of the additional $\hat{x}$ is to implement the polynomial-decay averaging method in order to improve the convergence rate, which is a common practice in the stochastic gradient method (38); introducing $\hat{x}$ does not affect privacy. The parameter $\eta \geq 1$ is used to control the averaging weight θ_t. Reference 38 provides details on choosing η.

Like most iterative optimization algorithms, the stochastic gradient method only converges in a probabilistic sense as the number of iterations $T \to \infty$. In practice, the number of iterations is always finite, so that it is desirable to analyze the suboptimality for finite T.

Theorem 13. The expected suboptimality of Algorithm 4 after T iterations is bounded as follows (25):

$$\mathbb{E}\left[f\left(\sum_{i=1}^{n} \hat{x}_i(T) \right) - f^* \right] \leq \mathcal{O}\left(\eta \sqrt{n} \rho \left(\frac{G}{\sqrt{T}} + \frac{\sqrt{2} m T^{3/2} L \Delta}{2\epsilon} \right) \right), \qquad 18.$$

where f^* is the optimal value of Equation 14, and

$$\rho = \max \left\{ \sqrt{\sum_{i=1}^{n} \|x_i\|^2} : x_i \in \mathcal{C}_i,\ i \in [n] \right\},$$

$$G = \max\{ \| \nabla f \left(\sum_{i=1}^{n} x_i \right) \| : x_i \in \mathcal{C}_i,\ i \in [n] \}.$$

Theorem 13 reveals an important trade-off in choosing the number of iterations T when running the differentially private optimization algorithm (Algorithm 4). If T is too small, then it will affect the convergence of the gradient method. On the other hand, if T is too large, then the amount of noise required by differential privacy will be too large and affect convergence as well. Note that this phenomenon is different from the case with sensitive objective functions. This is because in the gradient method with sensitive objective functions (Algorithm 2), the sensitive information ∇f_i enters the algorithm with a multiplicative factor γ_t according to Equation 13. Because of the choice of γ_t, the sensitivity of the query $(x(0), x(1), \ldots, x(T-1))$ converges as $T \to \infty$. However, when the sensitive information is in the constraint $\mathcal{C}_i$, the step sizes α_t do not change the sensitivity of the projection operations according to Equation 17.

5. CONCLUDING REMARKS

This review has discussed the integration of privacy guarantees in several applications in control and dynamical systems under the framework of differential privacy. When developing a differentially private mechanism, the most critical step is to compute the query sensitivity, which becomes more intricate for dynamical systems because the user data often take the form of data streams. We showcased the use of mathematical tools from systems theory in determining query sensitivity. For all private mechanisms, we highlighted the trade-offs between privacy and system performance, which provides a useful guideline in choosing the level of privacy and/or optimal system design.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

The authors acknowledge support from the National Science Foundation (CNS-1505799) and TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by the Microelectronics Advanced Research Corporation (MARCO) and the Defense Advanced Research Projects Agency (DARPA).

LITERATURE CITED

1. Quilumba FL, Lee WJ, Huang H, Wang DY, Szabados RL. 2015. Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities. *IEEE Trans. Smart Grid* 6:911–18
2. McDaniel P, McLaughlin S. 2009. Security and privacy challenges in the smart grid. *IEEE Secur. Priv*, 7:75–77
3. Molina-Markham A, Shenoy P, Fu K, Cecchet E, Irwin D. 2010. Private memoirs of a smart meter. In *BuildSys '10: Proceedings of the 2nd ACM Workshop on Embedded Systems Sensing for Energy-Efficiency in Building*, pp. 61–66. New York: ACM
4. Hoenkamp R, Huitema GB, de Moor-van Vugt AJC. 2011. The neglected consumer: the case of the smart meter rollout in the Netherlands. *Renew. Energy Law. Policy* 2011:269–82
5. Hubaux JP, Capkun S, Luo J. 2004. The security and privacy of smart vehicles. *IEEE Secur. Priv.* 2:49–55
6. Hoh B, Gruteser M, Xiong H, Alrabady A. 2006. Enhancing security and privacy in traffic-monitoring systems. *IEEE Pervasive Comput.* 5:38–46
7. Agarwal Y, Balaji B, Gupta R, Lyles J, Wei M, Weng T. 2010. Occupancy-driven energy management for smart building automation. In *BuildSys '10: Proceedings of the 2nd ACM Workshop on Embedded Systems Sensing for Energy-Efficiency in Building*, pp. 1–6. New York: ACM
8. Pantelopoulos A, Bourbakis NG. 2010. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Trans. Syst. Man Cybernet. C* 40:1–12
9. Agrawal R, Srikant R. 2000. Privacy-preserving data mining. In *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 439–50. New York: ACM
10. Sweeney L. 2002.*k* -anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 10:557–70
11. Machanavajjhala A, Gehrke J, Kifer D, Venkitasubramaniam M. 2006. *l*-diversity: privacy beyond *k*-anonymity. In *ICDE '06: Proceedings of the 22nd International Conference on Data Engineering*, p. 24. New York: IEEE
12. Narayanan A, Shmatikov V. 2008. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy*, pp. 111–25. New York: IEEE
13. Dwork C, McSherry F, Nissim K, Smith A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, ed. S Halevi, T Rabin, pp. 265–84. Berlin: Springer
14. Moulin P, O'Sullivan JA. 2003. Information-theoretic analysis of information hiding. *IEEE Trans. Inform. Theory* 49:563–93
15. Lindell Y, Pinkas B. 2009. Secure multiparty computation for privacy-preserving data mining. *J. Priv. Confident.* 1:5
16. Le Ny J, Pappas GJ. 2014. Differentially private filtering. *IEEE Trans. Autom. Control* 59:341–54
17. Sankar L, Rajagopalan SR, Mohajer S, Poor HV. 2013. Smart meter privacy: a theoretical framework. *IEEE Trans. Smart Grid* 4:837–46
18. Canepa ES, Claudel CG. 2013. A framework for privacy and security analysis of probe-based traffic information systems. In *HiCoNS '13: Proceedings of the 2nd ACM International Conference on High Confidence Networked Systems*, pp. 25–32. New York: ACM
19. Venkitasubramaniam P. 2013. Privacy in stochastic control: a Markov decision process perspective. In *2013 51st Annual Allerton Conference on Communication, Control, and Computing*, pp. 381–88. New York: IEEE
20. Huang Z, Mitra S, Dullerud G. 2012. Differentially private iterative synchronous consensus. In *WPES '12: Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*, pp. 81–90. New York: ACM
21. Mo Y, Murray RM. 2017. Privacy preserving average consensus. *IEEE Trans. Autom. Control* 62:753–65
22. Dwork C, Roth A. 2013. *The Algorithmic Foundations of Differential Privacy*. Found. Trends Theor. Comput. Sci. Vol. 9, No. 3–4. Hanover, MA: Now
23. Huang Z, Mitra S, Vaidya N. 2015. Differentially private distributed optimization. In *ICDCN '15: Proceedings of the 2015 International Conference on Distributed Computing and Networking*, chap. 4. New York: ACM

24. Cortés J, Dullerud GE, Han S, Le Ny J Mitra S, Pappas GJ. 2016. Differential privacy in control and network systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 4252–72. New York: IEEE
25. Han S, Topcu U, Pappas GJ. 2017. Differentially private distributed constrained optimization. *IEEE Trans. Autom. Control* 62:50–64
26. Wasserman L, Zhou S. 2010. A statistical framework for differential privacy. *J. Am. Stat. Assoc.* 105:375–89
27. Geng Q, Viswanath P. 2016. Optimal noise adding mechanisms for approximate differential privacy. *IEEE Trans. Inf. Theory* 62:952–69
28. Geng Q, Kairouz P, Oh S, Viswanath P. 2015. The staircase mechanism in differential privacy. *IEEE J. Sel. Top. Signal Process.* 9:1176–84
29. Doyle JC, Francis BA, Tannenbaum AR. 2013. *Feedback Control Theory*. New York: Courier
30. Khalil H. 2002. *Nonlinear Systems*. Upper Saddle River, NJ: Prentice Hall
31. Kailath T, Sayed A, Hassibi B. 2000. *Linear Estimation*. Upper Saddle River, NJ: Prentice Hall
32. Boyd S, Ghaoui L, Feron E, Balakrishnan V. 1994. *Linear Matrix Inequalities in System and Control Theory*. Philadelphia: SIAM
33. Lessard L, Recht B, Packard A. 2016. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM J. Optim.* 26:57–95
34. Li N, Chen L, Low SH. 2011. Optimal demand response based on utility maximization in power networks. In *2011 IEEE Power and Energy Society General Meeting*, pp. 1–8. New York: IEEE
35. Nedić A, Ozdaglar A. 2009. Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control* 54:48–61
36. Nocedal J, Wright S. 2006. *Numerical Optimization*. New York: Springer
37. Nozari E, Tallapragada P, Cortes J. 2016. Differentially private distributed convex optimization via functional perturbation. *IEEE Trans. Control Netw. Syst.* 5:395–408
38. Shamir O, Zhang T. 2013. Stochastic gradient descent for non-smooth optimization: convergence results and optimal averaging schemes. In *ICML '13: Proceedings of the 30th International Conference on International Conference on Machine Learning*, pp. I-71–79. New York: ACM

Annual Review of Control, Robotics, and Autonomous Systems

Volume 1, 2018

Contents

Errata

An online log of corrections to *Annual Review of Control, Robotics, and Autonomous Systems* articles may be found at http://www.annualreviews.org/errata/control