Differential privacy in control and network systems

Conference Paper · December 2016

DOI: 10.1109/CDC.2016.7798915

CITATIONS
86

READS
487

6 authors, including:

Shuo Han
University of Illinois at Chicago
43 PUBLICATIONS **676** CITATIONS
SEE PROFILE

Jerome Le Ny
Polytechnique Montréal
106 PUBLICATIONS **1,713** CITATIONS
SEE PROFILE

Sayan Mitra
University of Illinois, Urbana-Champaign
174 PUBLICATIONS **2,137** CITATIONS
SEE PROFILE

George J. Pappas
University of Pennsylvania
682 PUBLICATIONS **27,147** CITATIONS
SEE PROFILE

Differential Privacy in Control and Network Systems

Jorge Cortés Geir E. Dullerud Shuo Han Jerome Le Ny Sayan Mitra George J. Pappas

***Abstract*— As intelligent automation and large-scale distributed monitoring and control systems become more widespread, concerns are growing about the way these systems collect and make use of privacy-sensitive data obtained from individuals. This tutorial paper gives a systems and control perspective on the topic of privacy preserving data analysis, with a particular emphasis on the processing of dynamic data as well as data exchanged in networks. Specifically, we consider mechanisms enforcing *differential privacy*, a state-of-the-art definition of privacy initially introduced to analyze large, static datasets, and whose guarantees hold against adversaries with arbitrary side information. We discuss in particular how to perform tasks such as signal estimation, consensus and distributed optimization between multiple agents under differential privacy constraints.**

I. Introduction

An important concern raised by many of the emerging distributed automated systems around us, from smart homes and buildings to intelligent transportation systems and even smart cities, is their heavy reliance on data collected from private individuals and more generally from entities who naturally would wish to preserve the confidentiality of their own data [1]–[3]. Data obtained from these privacy-sensitive sources is fused to estimate the current state of a target system, with this estimate subsequently used for effective decision making and control. Traditionally, much attention has been focused on the design, improvement, and sophistication of these data processing technologies by assuming that aggregators and decision makers are trustworthy. However, this assumption turns out to be risky or undesirable in a wide variety of situations, and even when it is valid, it does not cover the many cases in which some analysis based on the sensitive data is released publicly or to third-parties, implicitly leaking some of the information contained in the original datasets. An extreme example of this scenario was the Netflix Prize: although the release format of the data sets provided by Netflix was meant to preserve customer privacy, the work [4] was able to identify individual users by using side information, specifically film ratings on the Internet Movie Database.

Another example is given by smart grids, where optimized power forecast, generation, and distribution might push network operators and dispatch units to access fine-grained usage time series of consumers measured by advanced metering infrastructures. It is well known, however, that these signals reveal detailed information about the presence, absence and even specific activities of a home's occupants [5]. Similarly, smart transportation services require traffic state estimates and forecasts, which in turn rely on the on the measurement of individual location traces by an increasingly large number and variety of sensors, either static like induction loops and cameras, or moving with the traffic like GPS devices [6] or RFID tags [7] inside individual vehicles.

Similar issues arise for data analysis in a variety of other areas, from economics to social networks and healthcare, and are only amplified by the current trend within companies and government agencies to collect ever more information about private individuals. Hence, privacy preserving data analysis, once a topic mostly of interest to statisticians and econometrists [8], has received an increasing amount of attention in the last decade. In particular, since offering privacy guarantees for a system generally involves sacrificing some level of performance, researchers have proposed various quantitative definitions of privacy to evaluate the resulting trade-offs rigorously. These definitions include k-anonymity [9] and its various extensions [10], information-theoretic privacy [11], or conditions based on observability [12]–[14]. However, in the last few years the notion of differential privacy, the focus of this paper, has emerged essentially as a standard privacy specification [15], [16]. A system processing privacy sensitive inputs from individuals is made differentially private by randomizing its answers in such a way that the distribution over published outputs is not too sensitive to the data provided by any single participant. As a result, it is provably difficult for an adversary, no matter how powerful, to make inferences about individual records from the published outputs, or even to detect the presence of an individual in the dataset. Differential privacy is already being employed by technological giants such Google and Apple, making its way into commercial products. Google has successfully implemented it in its web browser Chrome [17] and also has several projects on tracking urban mobility using differential privacy to ensure that no individual user's journey can be identified [18]. Apple recently announced that iOS 10 uses technologies for learning usage patters from a large number of users without compromising individual privacy using differential privacy [19].

The notion of differential privacy has also made its way into systems and controls, where researchers have used it to

J. Cortés is with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, cortes@ucsd.edu; G. E. Dullerud and S. Mitra are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, {dullerud,mitras}@illinois.edu; J. Le Ny is with the Department of Electrical Engineering, Polytechnique Montreal, and GERAD, jerome.le-ny@polymtl.ca; S. Han and G. J. Pappas are with the Department of Electrical and Systems Engineering, University of Pennsylvania, {hanshuo,pappasg}@seas.upenn.edu

design privacy-aware algorithms for a diverse set of objectives, including control [20], [21], average consensus [22], [23], network topology [24], estimation and filtering [25], [26], and convex optimization [27]–[30]. Our goal in this tutorial paper is to introduce the main concepts in differential privacy and provide an overview of recent advances relevant to control and network systems in this emerging area.

Organization: Section II introduces the fundamental concepts in differential privacy, along with motivational scenarios and the basic set of results on which the rest of the exposition builds. Section III deals with the application of differential privacy in estimation and filtering problems. Section IV studies the trade-offs between differential privacy and accuracy in the context of linear distributed control systems for consensus. Section V examines the role of differential privacy in solving distributed optimization problems. Finally, Section VI summarizes our conclusions and outlines avenues for further research in differential privacy for network systems.

Notation: We denote by $\mathbb{N}$, $\mathbb{R}_{>0}$, and $\mathbb{R}$, the sets of natural, positive real, and real numbers, respectively. We denote the set of d-dimensional real vectors and the set of $m \times d$ real matrices by $\mathbb{R}^d$ and $\mathbb{R}^{m\times d}$. For $x \in \mathbb{R}$, we denote its absolute value by $|x|$. For $x \in \mathbb{R}^d$, with $d \in \mathbb{N} \cup \{+\infty\}$ we define its ℓ_p-norm by

$$\|x\|_p = \Big(\sum_{i=1}^{n} |x_i|^p\Big)^{1/p},$$

where x_i is the i^{th} component of x. For G a linear time-invariant (LTI) system, we let $\|G\|_2$ denote its $\mathcal{H}_2$ norm and $\|G\|_\infty$ its $\mathcal{H}_\infty$ norm.

A scalar random variable x obeys the Laplace distribution with parameter λ (and zero mean), if its probability distribution function satisfies

$$p(x) = \frac{1}{2\lambda}\exp(-\frac{|x|}{\lambda}).$$

We write this as $x \sim \text{Lap}(\lambda)$. The definition extends to n-dimensional random vectors by using the ℓ_1-norm, namely, $x \sim \text{Lap}(\lambda, n)$ if

$$p(x) = \left(\frac{1}{2\lambda}\right)^n \exp(-\frac{\|x\|_1}{\lambda}).$$

The components of the Laplace random vector are independent.

II. Foundations of Differential Privacy

The purpose of this section is to introduce the concept of differential privacy. We start by giving a formal definition of differential privacy and discuss its implications. Then, we discuss several important properties of differential privacy. In the end, we introduce a number of commonly used mechanisms that guarantee differential privacy. These mechanisms often serve as the building blocks for designing more complicated mechanisms, as we illustrate in the forthcoming sections.

A. Motivation for Differential Privacy

Throughout the paper, we consider the setting of protecting the privacy of individual users whose information is stored collectively as a database. Examples of a database include patient records in a hospital, salaries of employees of a company, or census records, to name a few. The database is used to extract useful aggregate information from the users, and the result is often available to the public. For example, one may be interested in computing the average salary of all employees from a database of salaries.

One may wonder how the publicly available result, which only contains aggregate information of all users in a database, can compromise the privacy of any individual user. This can, nevertheless, happen in certain extreme cases. Consider the following example. Suppose n voters participate in an anonymous vote that involves two candidates, Alice and Bob. Suppose the result of the vote shows that $n-1$ people voted for Alice, and one person voted for Bob. Then, the person who voted for Bob is 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 is anonymous.

Aside from the extreme case in the previous example, 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 is able to 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 are now faced with 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 infer the information of any individual user, regardless of the released result and the presence of possible side information. The notion of differential privacy squarely addresses this dilemma. The notion of differential privacy may differ from our common understanding of privacy: privacy is not treated as a binary concept (i.e., either being private or non-private) but is instead measured on a level that changes continuously from total privacy to non-privacy.

The basic idea used by differential privacy is to "perturb" the exact result before release. As one can imagine, the amount of perturbation affects both the usefulness of the result and the level of privacy. The more perturbation used, the less useful the result and the higher the level of privacy. As will be seen in later sections, such a trade-off between usefulness of the result and level of privacy can often be quantified and provide guidelines on choosing an appropriate level of privacy.

B. Definition of Differential Privacy

1) Terminology and Definition: In differential privacy, user information that needs to be protected is contained in a set (called *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 a database D is modeled by $q(D)$ for some mapping q (called *query*) that acts on D; the range of q is denoted by $\mathcal{Q}$.

Example 2.1: (Salary database): For a database containing the salaries of a group of people, we can define $D = \{d_i\}_{i=1}^n$, where $d_i \in \mathbb{R}_{>0}$ is the salary of user i (assuming no minimum denomination). Suppose someone is interested in the average salary of people in the database. Then the query can be written as $q(D) = \sum_{i=1}^n d_i/n$. •

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

Definition 2.2 (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 $i \in \{1, \ldots, n\}$ such that $d_j = d_j'$ for all $j \neq i$.

When differential privacy was first proposed [15], 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/removing one user from the other database. The motivation behind the original definition is 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 2.2 generalizes the original notion of adjacency relation in order to handle databases consisting of numeric values. In fact, as we illustrate later, this definition can be extended further to incorporate more complex objects, such as vector norms and databases of functions.

Besides the conditions stated in Definition 2.2, 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 on constraining the difference between d_i and d_i' determines the "granularity" that an individual's value can be protected, as will be illustrated in the following example.

Example 2.3: (Salary database – cont'd): Consider again the database of salaries in Example 2.1. We can define an adjacency relation as in Definition 2.2 with $|d_i - d_i'| \leq \$1{,}000$. Then the privacy guarantee given by a differentially private mechanism would become "an adversary cannot determine the salary of any user in the database within an accuracy of \$1,000 (with high probability)". Note, however, that the adversary may, however, still be able to tell that some user's salary is \$1 versus \$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. Namely, an adversary cannot know anything about the salary of any user except that it lies within $[0, d_{\max}]$ (which is public knowledge). •

As we mentioned in Section II-A, directly making $q(D)$ available to the public may cause users in the database to lose their privacy. In order to preserve privacy, for any given query q, one needs to develop a *mechanism* M that approximates q. Naturally, the range of M is the same as that of q, i.e., $\mathrm{range}(M) = \mathrm{range}(q) = \mathcal{Q}$. In the framework of differential privacy, all mechanisms under consideration are *randomized*. Namely, 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 just looking at the output of the mechanism.

Definition 2.4 (ϵ-Differential privacy [15]): Given $\epsilon \geq 0$, a mechanism M preserves *ϵ-differential privacy* if for all $\mathcal{R} \subseteq \mathrm{range}(M)$ and all adjacent databases D and D' in $\mathcal{D}$, it holds that

$$\mathbb{P}\left[M(D) \in \mathcal{R}\right] \leq e^{\epsilon}\mathbb{P}\left[M(D') \in \mathcal{R}\right]. \tag{1}$$

The probability measure in (1) is taken from the probability space used for defining the randomized mechanism M. The constant ϵ indicates the level of privacy: smaller ϵ implies higher level of privacy. Notice that the relationship (1) also implies

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

due to the symmetric nature of the 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 (1) that any non-constant 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 2.5: ((ϵ, δ)-Differential privacy [15]): Given $\epsilon, \delta \geq 0$, a mechanism M preserves *(ϵ, δ)-differential privacy* if for all $\mathcal{R} \subseteq \mathrm{range}(M)$ and all adjacent databases D and D' in $\mathcal{D}$, it holds that

$$\mathbb{P}\left[M(D) \in \mathcal{R}\right] \leq e^{\epsilon}\mathbb{P}\left[M(D') \in \mathcal{R}\right] + \delta. \tag{2}$$

It can be seen from Definition 2.5 that the notion of (ϵ, δ)-differential privacy reduces to ϵ-differential privacy when $\delta = 0$. The introduction of the additive term δ in (2)

yields a weaker privacy guarantee than ϵ-differential privacy. When $\delta > 0$, even if ϵ is small, it can still happen that $\mathbb{P}[M(D) \in \mathcal{R}]$ is large compared to $\mathbb{P}[M(D') \in \mathcal{R}]$; as a result, one can still potentially tell whether the input database is D or D'.

2) Choosing the Privacy Level: One useful interpretation of differential privacy can be made in the context of detection theory [31], [32]. 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 $D = \{d_i\}_{i=1}^n \in \{0,1\}^n$, and 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 $p_{\rm FP} = \mathbb{P}[M(D') \in \mathcal{R}^*]$ (i.e., $d_i = 0$, but the detection rule reports $d_i = 1$) and false negative probability $p_{\rm FN} = \mathbb{P}[M(D) \notin \mathcal{R}^*] = \mathbb{P}[M(D) \in \mathcal{Q}\backslash\mathcal{R}^*]$. For a good detection rule, both probabilities are desired to be small. Since D and D' are adjacent, we know from Definition 2.4 that

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

which lead to

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

The conditions (3) imply that $p_{\rm FN}$ and $p_{\rm FP}$ cannot be both too small. In particular, we have

$$p_{\rm FN} + p_{\rm FP} \ge \frac{2}{1+e^\epsilon}. \tag{4}$$

Namely, 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 $p_{\rm FN} = 0.05$, then the false positive probability $p_{\rm FP} \ge \max\{1 - e^\epsilon p_{\rm FN}, e^{-\epsilon}(1 - p_{\rm FN})\} \approx 0.94$, which is quite large. The relationship (4) provides a guideline for choosing ϵ when implementing a differentially private mechanism. Often, the error probabilities $p_{\rm FN}$ and $p_{\rm FP}$ are more straightforward to specify than the level of privacy ϵ. Once the lower bounds for $p_{\rm FN}$ and $p_{\rm FP}$ are specified, one can choose ϵ accordingly from the relationship (4).

C. Properties of Differential Privacy

Differential privacy enjoys several important properties. First, differential privacy is immune to post-processing. Namely, without any additional knowledge on 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 mechanism is often released to the public, after which the released result can be potentially utilized arbitrarily by other parties. The immunity to post-processing guarantees that no privacy can be further lost through the handling by other parties. The property of immunity to post-processing is formalized below.

Theorem 2.6 (Post-processing [33]): Suppose a mechanism $M\colon \mathcal{D} \to \mathcal{Q}$ preserves ϵ-differential privacy. Then for any function f, the (functional) composition $f \circ M$ also preserves ϵ-differential privacy.

Next, we introduce a number of composition rules that are often used for constructing 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 2.7 (Sequential composition [33]): Suppose a mechanism M_1 preserves ϵ_1-differential privacy, and another mechanism M_2 preserves ϵ_2-differential privacy. Define a new mechanism $M(D) := (M_1(D), M_2(D))$. Then the mechanism M preserves $(\epsilon_1 + \epsilon_2)$-differential privacy.

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 2.7, 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 2.7. Theorem 2.7 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 2.7 can be loose and may not be the best guarantee. This is because Theorem 2.7 does not take into account the correlation between M_1 and M_2. Consider the trivial case where in the joint mechanism, $M_2(D)$ simply repeats the output produced by $M_1(D)$. In this case, adding M_2 does not reveal any additional information about the database compared to using M_1 alone. Namely, the joint mechanism M is ϵ_1-differentially private, whereas Theorem 2.7 says that M is $2\epsilon_1$-differentially private.

For certain applications, the mechanism we would like to design is a result of 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 gives privacy guarantees for cases that involve iterative computation.

Theorem 2.8: (Adaptive composition [33]): Consider a mechanism $M_1\colon \mathcal{D} \to \mathcal{Q}_1$ that preserves ϵ_1-differential privacy, and another mechanism $M_2\colon \mathcal{D} \times \mathcal{Q}_1 \to \mathcal{Q}_2$ such that $M_2(\cdot, y_1)$ preserves ϵ_2-differential privacy for any $y_1 \in \mathcal{Q}_1$. Define a new mechanism $M(D) := M_2(D, M_1(D))$. Then the mechanism M preserves $(\epsilon_1 + \epsilon_2)$-differential privacy.

The adaptive composition rule generalizes the post-processing rule (cf. Theorem 2.6), 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 (cf. Theorem 2.7).

The post-processing rule and composition rules also hold 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. However, one can actually relax the guarantee for δ in favor for a smaller ϵ, as stated in the following result. For simplicity, the theorem considers the case of composing k mechanisms, each of which is (ϵ, δ)-private.

Theorem 2.9: (Advanced composition [33]): For all $\epsilon, \delta, \delta' \geq 0$, the mechanism formed by adaptive composition of k mechanisms with (ϵ, δ)-differential privacy preserves $(\epsilon', k\delta + \delta')$-differential privacy with

$$\epsilon' = \sqrt{2k \log(1/\delta')}\epsilon + k\epsilon(e^\epsilon - 1). \tag{5}$$

When e^ϵ is close to 1, the right-hand side of (5) is dominated by the first term. Compared to the $(k\epsilon, k\delta)$-privacy guarantee given by the original composition theorem, the new guarantee then ensures a slower degradation of ϵ at the rate of $O(\sqrt{k})$ (instead of $O(k)$), at the expense of introducing an extra additive factor δ'.

D. Differentially Private Mechanisms

We now present a number of commonly used mechanisms that preserve differential privacy. This is by no means an exhaustive list of differentially private mechanisms, and the mechanisms in the list are not necessarily optimal mechanisms (defined as the one that achieves the best accuracy/utility for a specified privacy level). Interested readers can refer to related work [32], [34] on the discussions about optimal mechanisms.

1) Laplace Mechanism: When the range of query $\mathcal{Q}$ is $\mathbb{R}$, one commonly used differentially private mechanism is the *Laplace mechanism* [15]. The Laplace mechanism works by introducing additive noise drawn from the Laplace distribution.

Theorem 2.10 (Laplace mechanism [33]): 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.

The Laplace mechanism reveals an intrinsic trade-off between privacy and accuracy of the result. Notice that the mean squared error of the result is given by

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

Namely, as ϵ becomes smaller (i.e., more privacy is preserved), the result becomes less accurate. To illustrate how the Laplace mechanism can be applied, we give in the following a simple example on computing the average salary while preserving differential privacy.

Example 2.11: (Salary database –cont'd): Consider the database of salaries given in Example 2.1, 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 2.3. Then the sensitivity of q can be obtained as follows:

$$\Delta = \max_{D,D'} |q(D) - q(D')| = \frac{1}{n} \max_{i \in \{1,\ldots,n\}} \max_{d_i, d_i'} |d_i - d_i'| = \frac{d_{\max}}{n}.$$

From Theorem 2.10, 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 matches 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 $\mathbb{R}^k$. Suppose the sensitivity of query q, defined as

$$\Delta := \max_{D,D'} \|q(D) - q(D')\|_\infty,$$

is bounded. One way to achieve ϵ-differential privacy is to add i.i.d. Laplace noise $\text{Lap}(k\Delta/\epsilon)$ to each component of q, which is guaranteed by the sequential composition theorem (cf. Theorem 2.7). However, a similar mechanism that requires less noise can be adopted in this case by using the fact that the ℓ_2-sensitivity Δ_2 of the query is also bounded:

$$\Delta_2 := \max_{D,D'} \|q(D) - q(D')\|_2 \leq \sqrt{k}\Delta.$$

Theorem 2.12: For a given query q, let $\Delta_2 = \max_{D,D'} \|q(D) - q(D')\|_2$ be the ℓ_2-sensitivity of q. Then the mechanism $M(D) = q(D) + w$, where w is a random vector whose probability distribution is proportional to $\exp(-\epsilon \|w\|_2 / \Delta_2)$, preserves ϵ-differential privacy.

2) Exponential Mechanism: Another useful and quite general mechanism is the exponential mechanism. This mechanism requires a scoring function $u\colon \mathcal{Q} \times \mathcal{D} \to \mathbb{R}$. Although the scoring function can be chosen arbitrarily, there is often a natural choice when the query corresponds to the optimal solution to an optimization problem. For minimization problems, one can usually choose the negative objective function as the scoring function. The exponential mechanism $M_E(D; u)$ guarantees ϵ-differential privacy by randomly reporting q according to the probability density function

$$\frac{\exp(\epsilon u(q, D)/2\Delta_u)}{\int_{q' \in \mathcal{Q}} \exp(\epsilon u(q', D)/2\Delta_u)\, dq'}, \tag{6}$$

where

$$\Delta_u := \max_x \max_{D,D' :\, \mathrm{Adj}(D,D')} |u(x,D) - u(x,D')|$$

is the sensitivity of the scoring function u.

Theorem 2.13 (Exponential mechanism [35]): The exponential mechanism M_E is ϵ-differentially private.

Compared with the Laplace mechanism, the exponential mechanism is more general in that it does not restrict the query to be numeric. However, it is generally difficult to obtain the probability density function (6) in closed form. In practice, the exponential mechanism is more widely used in cases where the range $\mathcal{Q}$ is finite, so that the probability density function (6) has finite support and can be computed. In the context where the query is the optimal solution to an optimization problem, the exponential mechanism provides guarantees on the quality of the approximate solution. When the range $\mathcal{Q}$ is finite, i.e., $|\mathcal{Q}| < \infty$, the exponential mechanism has the following probabilistic guarantee on the suboptimality with respect to the scoring function.

Theorem 2.14: (Probabilistic guarantee on suboptimality [35]): Consider the exponential mechanism $M_E(D;u)$ acting on a database D under a scoring function u. If $\mathcal{Q}$ is finite, i.e., $|\mathcal{Q}| < \infty$, then M_E satisfies

$$\mathbb{P}\Big[u_{\mathrm{opt}} - u(M_E(D;u),D) \geq \frac{2\Delta_u}{\epsilon}(\log|\mathcal{Q}| + t)\Big] \leq e^{-t},$$

where $u_{\mathrm{opt}} = \max_{q\in\mathcal{Q}} u(q,D)$.

It is also possible to obtain the expected suboptimality using the observation that if a random variable X satisfies: (1) $X \geq 0$ and (2) $\mathbb{P}[X \geq t] \leq e^{-\alpha t}$ for some $\alpha > 0$, then it holds that $\mathbb{E}[X] \leq 1/\alpha$.

Theorem 2.15 (Expected suboptimility): Under the assumptions of Theorem 2.14, the exponential mechanism $M_E(D;u)$ satisfies

$$\mathbb{E}\left[u_{\mathrm{opt}} - u(M_E(D;u),D)\right] \leq 2\Delta_u(1+\log|\mathcal{Q}|)/\epsilon.$$

Similar to the Laplace mechanism, we can see from both Theorems 2.14 and 2.15 the trade-off between optimality and privacy: the suboptimality gap of the response $M_E(D;u)$ increases as ϵ becomes smaller.

3) 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 2.16 (Gaussian mechanism [33]): For a given query q, let $\Delta_2 = \max_{D,D'} \|q(D) - q(D')\|_2$ 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 i.i.d. zero-mean Gaussian with variance $\sigma^2 = \kappa^2_{\delta,\epsilon}\Delta_2^2$, with $\kappa_{\delta,\epsilon} = \sqrt{2\log(1.25/\delta)}/\epsilon$.

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, due to the fact that any linear transformation of a Gaussian random vector remains Gaussian.

4) Design of Differentially Private Mechanisms: The mechanisms presented above (Laplace, exponential, and Gaussian) are quite general and straightforward to implement. For implementing these algorithms, the only 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, by using the composition rules from Section II-C, one can construct the desired private mechanism.

III. Differentially Private Filtering

An important trend motivating this section is the increasing emphasis on systems processing *streams* of dynamic data collected from many sources around us, from websites and smartphones to surveillance cameras, smart meters and house thermostats. The production of useful statistics in real-time (road traffic state estimate, power consumption in a small neighborhood, detection of disease outbreaks) subject to privacy constraints on the input signals brings new challenges to the fields of signal processing, systems and control.

A. Generic Architecture for Privacy Preserving Filtering

In general, the goal is to compute an approximation of a desired output signal while providing a differential privacy guarantee on the space of input signals from which this output is computed. In words, it should be hard to infer just from looking at the published output signal which of two adjacent input signals was used. As we mentioned above (cf. our discussion after Definition 2.2), we have some freedom in the choice of adjacency relation on the space of input signals, which allows us to model various situations. In essence, the adjacency relation captures which signals we want to make indistinguishable. The difficulty is in producing outputs that are useful, i.e., close according to some performance measure to the signals we would like to have in the absence of privacy constraint, see Figure 1. Ideally, we would also want to obtain trade-off curves characterizing the fundamental limit in terms of achievable utility for a given privacy level,

although this topic is currently mostly unexplored for the processing of real-time signals.

Figure 1 depicts an architecture for the approximation of a desired signal in a differentially private way, which has proved useful [25], [36] when trying to optimize utility for a given requested level of privacy, i.e., given parameters (ϵ, δ) in the differential privacy definition. In fact, this architecture appears to be useful in a broader context than differentially private filtering. It can be recognized in the work of Li and Miklau [37] on providing differentially private answers to linear queries about static databases, in [11], where quantization is used to enforce an information theoretic definition of privacy rather than random perturbation, or more recently in [38], where the problem considered can be interpreted as publishing a signal under an information theoretic definition of privacy, although one that does not taken auxiliary information into account.

One can interpret Figure 1 in the context of communication systems. The "signal preparation block" is a transmitter (or sensor) design problem, shaping the input signal before it is sent into a sanitization mechanism, which can be viewed as a transmission channel. In the context of differential privacy, this channel typically just adds additive (Laplace or Gaussian) noise to its input signal. Finally, the "Output signal reconstruction block" is a receiver design problem, attempting to remove the effect of the sanitization block to estimate the desired output signal. This architecture enforces differential privacy as long as the sanitization block does, by the resilience to post-processing, cf. Theorem 2.6. Generally, the sanitization block is fixed initially, as in the following result, and we are faced with a joint transmitter-receiver design problem, an analogy that is further discussed in [26].

Theorem 3.1: (Gaussian mechanism for signals [25]): Let G be a dynamic system with n inputs and p outputs. For Adj an adjacency relation on the space of input signals for G, define its ℓ_p-sensitivity as

$$\Delta_p G = \sup_{\text{Adj(u,u')}} \|Gu - Gu'\|_p$$

Then the mechanism $M(u) = Gu + w$, where w is a white Gaussian noise with covariance matrix $\kappa_{\delta,\epsilon}^2 (\Delta_2 G)^2 I_p$, with $\kappa_{\delta,\epsilon}$ defined in Theorem 2.16, is (ϵ, δ)-differentially private.

Although the architecture of Figure 1 achieves the optimal trade-off between utility and obfuscation (or communication rate) under certain information-theoretic formulations [38], to the best of our knowledge no such result is available for differentially private signal analysis. Nonetheless, optimizing this architecture generally leads to interesting computational problems, and typically provides a significant improvement over more basic schemes such as input perturbation (where only the output signal reconstruction block is considered) and output perturbation (where only the signal preparation block is considered). A general approach to attempt to optimize this architecture can proceed with the following steps:

(i) Fix the adjacency relation on the space of measured signals and the sanitization mechanism to provide differential privacy. For example, it could be the Gaussian mechanism of Theorem 3.1, with Gaussian noise proportional to the sensitivity of the signal preparation block. The computation of the sensitivity itself depends on the choice of adjacency relation.

(ii) Fix the family of output signal reconstruction blocks, and express the "best" such block as a function of the signal preparation block. In general, this step depends on the performance measure considered, on the properties of the signal preparation blocks considered (ex: linear or nonlinear), and on any public information one might have about the input signals, which could help in the estimation of the desired output signal from the output of the sanitization mechanism.

(iii) Express the performance measured now solely as a function of the signal preparation block, and optimize over the latter.

In the rest of this section, we illustrate the application of this procedure to a few different scenarios. In some cases, it can lead to apparently intractable optimization problems. However, even suboptimal designs can be worth considering under this point of view.

B. Zero-Forcing Equalization Mechanism

Consider a scenario where n sensors each report at regular intervals a scalar $u_{i,t} \in \mathbb{R}$, for $t \geq 0$, $i \in \{1, \dots, n\}$. Let $u_t = [u_{1,t}, \dots, u_{n,t}]^T \in \mathbb{R}^n$. In traffic or building monitoring systems for example, $u_{i,t}$ could be the number of cars or people detected in front of sensor i during period t, a measurement that could then be integrated by a density estimation algorithm. In a smart metering scenario for a household, $u_{i,t}$ could be the average power consumption level of appliance i during period t. We would like to publish in real-time a signal $y = Fu$ computed from the detected events, with $y_t \in \mathbb{R}^p$, where F in this section is assumed to be a causal, linear time-invariant (LTI) system. However, the signals u are assumed to be privacy sensitive, which prevents from releasing exactly y, since F could be partially or completely invertible for example. Instead, we release a sanitized, differentially private approximation $\hat{y}$ and measure the utility of this approximation by the average mean square error for example, i.e., we wish to minimize the quantity $\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[|y_t - \hat{y}_t|_2^2]$.

1) Adjacency Relation and Sanitization Mechanism: To instantiate the differential privacy definition, consider the following adjacency relation on the input signals u

$$\begin{aligned} &\text{Adj}(u, u') \text{ iff } \forall i \in \{1, \dots, n\}, \\ &\exists t_i \in \mathbb{N}, \alpha_i \in \mathbb{R}, \text{ s.t. } u_i' - u_i = \alpha_i \delta_{t_i}, |\alpha_i| \leq k_i, \end{aligned} \tag{7}$$

parametrized by a vector $k \in \mathbb{R}^n$ with components $k_i > 0$. Here δ_{t_i} denotes the discrete impulse signal with impulse at t_i. This adjacency relation, called *event-level adjacency*, generalizes to vector-valued signals the adjacency relation

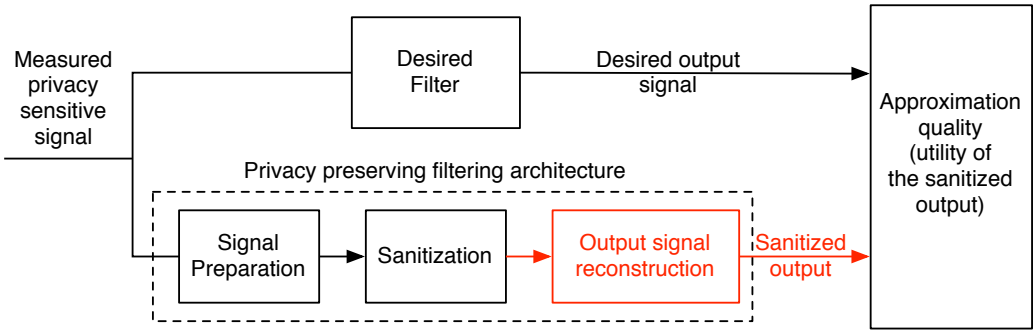

Fig. 1. Architecture for privacy preserving filtering. The signals exiting the sanitization block are differentially private.

considered in [39]–[41]. In words, differential privacy with respect to (7) aims to make it hard to detect deviations on each input signal component at any single time period, and by at most k_i. In terms of applications, consider a location based service, a traffic monitoring sensor network or a system recording hospital visits. Differential privacy with respect to (7) makes it hard to answer questions of the type: "Did person X visit location i?", at least for people who visited location i only once. It does not protect a person visiting the same location twice however, although this situation could be mitigated to some extent by duplicating sensor i for different time windows, and putting a lower bound on the frequency at which the person visits a given location. Naturally, from a privacy point of view, it would be preferable to remove this bound on the visit frequency at the n locations. However, this would allow a person to have potentially a very strong influence on any sensed signal, and would likely require adding an amount of perturbation that would render interesting output signals useless or require very large values for the privacy parameters ϵ and δ.

For simplicity, we discuss in the following only the case where u is a scalar signal, i.e., $n = 1$, although we might want to produce several output signals, so $p > 1$ is allowed. In other words, F is a single-input multiple-output (SIMO) LTI system. In this case, the following simple observation is key for sensitivity computations.

Theorem 3.2: (Event-Level Sensitivity for SIMO Systems): Let G be an LTI system with one input, p outputs and such that $\|G\|_2 < \infty$. For the adjacency relation (7), we have $\Delta_2 G = k_1 \|G\|_2$.

Proof: For u and u' adjacent

$$\|G(u - u')\|_2^2 = |\alpha_1|^2 \|G\delta_{t_1}\|_2^2 \leq k_1^2 \|G\|_2^2,$$

and the bound is attained if $|\alpha_1| = k_1$. ■

As a result of Theorem 3.2, we obtain the following input perturbation and output perturbation differentially private mechanisms. Either add white Gaussian noise w_1 to u with standard deviation $k_1 \kappa_{\delta,\epsilon}$ (since the identity system has sensitivity 1) and output $F(u + w_1)$ (which is private by resilience to post-processing, cf. Theorem 2.6), or add white Gaussian noise w_2 to $y = Fu$ with standard deviation $k_1 \kappa_{\delta,\epsilon} \|F\|_2$. In this case, both schemes achieve the same performance as measured by the Mean Squared Error (MSE), namely an MSE of $k_1^2 \kappa_{\delta,\epsilon}^2 \|F\|_2^2$.

2) Signal Reconstruction and Optimization over Linear Signal Shaping Blocks: To improve over the performance of the input or output perturbation mechanisms, we can consider the more general mechanism of Figure 2, called the Zero-Forcing Equalization (ZFE) mechanism [25]. This

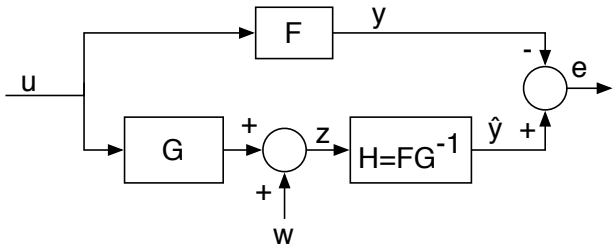

Fig. 2. Zero-Forcing Mechanism.

mechanism follows the architecture of Figure 1, with the Gaussian mechanism as sanitization mechanism. Indeed, the signal w is a white Gaussian noise with standard deviation $k_1 \kappa_{\delta,\epsilon} \|G\|_2$, so that the signal z is differentially private. The output reconstruction filter H simply inverts the filter G and then passes the result through the desired filter F. The goal is then to choose the signal shaping block G to optimize the MSE

$$\begin{aligned} & e_{mse}^{ZFE}(G) = \\ & \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[|(Fu)_t - (HGu)_t - (Hw)_t|_2^2\right] \\ & = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[|(Hw)_t|_2^2\right] \\ & = k_1^2 \kappa_{\delta,\epsilon}^2 \|FG^{-1}\|_2^2 \|G\|_2^2. \end{aligned} \tag{8}$$

From the expression (8), a straightforward application of the Cauchy-Schwarz inequality gives the following lower bound together with the filter G achieving this lower bound.

Theorem 3.3 (SIMO ZFE mechanism [26]): Let F be a SIMO LTI system with $\|F\|_2 < \infty$. For any LTI system G such that $\|G\|_2 < \infty$, we have

$$e_{mse}^{ZFE}(G) \geq k_1^2 \kappa_{\delta,\epsilon}^2 \left(\frac{1}{2\pi} \int_{-\pi}^{\pi} |F(e^{j\omega})|_2 \, d\omega \right)^2. \quad (9)$$

If moreover F satisfies the Paley-Wiener condition $\frac{1}{2\pi}\int_{-\pi}^{\pi} \ln |F(e^{j\omega})|_2 \;\; d\omega > -\infty$, this lower bound on the mean square error of the ZFE mechanism is attained by some SISO system G with causal inverse L, such that

$$|G(e^{j\omega})|^2 = \|F(e^{j\omega})\|_2 \quad (10)$$

for almost every $\omega \in [-\pi, \pi)$.

Finding G with causal inverse satisfying (10) is a spectral factorization problem [42]. Note also from Jensen's inequality that the achievable MSE (9) is always lower or equal to $k_1^2 \kappa_{\delta,\epsilon}^2 \|F\|_2^2$, the MSE of the input or output mechanism. In practice the ZFE mechanism can lead to substantial performance improvements [25], [26].

3) Extension to MIMO systems: For MIMO systems, the sensitivity with respect to the adjacency relation (7) has a much more complicated expression than in Theorem 3.2, because of the superposition at the output of the effect of one individual on the different input channels [26]. A consequence is that at this time, an optimal MIMO ZFE mechanism is not known, but one can derive lower bounds on its steady-state estimation error, together with sub-optimal mechanism using diagonal signal shaping blocks, in effect reducing the problem to a set of SIMO problems, see [26].

C. Model-Based Signal Reconstruction

Leveraging a priori known models of datasets, e.g., statistical models, is a relatively unexplored topic in the differential privacy literature, in contrast to standard approaches in control and signal processing. A few papers related to differentially private analysis of time series do make assumptions about the coefficients of the discrete Fourier transform of the input [43], or about the existence of a sparse representation in some basis [44]. However, in contrast to our focus, the resulting mechanisms are not causal, i.e., the output signal can only be published when the input signal is entirely known. In many cases, we have a significant amount a knowledge about the input data even before receiving it, which we would like to use to provide either better accuracy in our estimates, or better privacy, e.g., by sampling less frequently. We explore two topics illustrating this idea: leveraging statistical models of stationary input signals, and differentially private Kalman filtering for input signals that are produced by a state-space model, such as location traces that are constrained by kinematic equations.

1) Leveraging Statistical Models of Stationary Signals: With zero-forcing equalization, one tries to cancel the effect of a channel G simply by inverting this channel at the receiver. One issue with this approach is the noise amplification at frequencies where $|G(e^{j\omega})|$ is small [45], although this is not as problematic for us because $|F(e^{j\omega})|$ and $|G(e^{j\omega})|$ in Theorem 3.3 are both small at the same frequencies, see (10). The main advantage of the ZFE mechanism is that it can be implemented in absence of any model of the input signal. However, one can improve on it by using more advanced equalization schemes in the design of the post-filter H of Figure 2, if some additional information about the input signal is known.

Suppose for example that the input signal u on Figure 2 is known to be wide-sense stationary (WSS) with know mean vector μ and matrix-valued autocorrelation sequence $R_u[k] = \mathbb{E}[u_t u_{t-k}^T] = R_u[-k]^T, \forall k$. The z-spectrum matrix of u is denoted $P_u(z) = \sum_{k=-\infty}^{\infty} R[k] z^{-k}$. We can assume μ to be zero, since the output y is also WSS with known mean equal to $F(1)\mu$. Note that for privacy reasons we assume here that the parameters R_u and μ have *not* been estimated directly from the signal u, but instead they correspond to general knowledge one might have about an application, and perhaps could have been estimated from an additional dataset that is not privacy-sensitive. Since w a white Gaussian noise (with variance proportional to $\|G\|_2^2$), the task of the filter H is to reconstruct the WSS signal y from the WSS signal $z = Gu + w$. In this case, the linear reconstruction filter H that is optimal with respect to the minimization of the MSE is the Wiener filter [42]. Ignoring for now any requirement that H be causal, the optimal *smoother* H for a given signal shaping filter G is then

$$\begin{aligned} H(z) =& F(z) P_u(z) G(z^{-1})^T \\ & \times \left(G(z) P_u(z) G(z^{-1})^T + \kappa_{\delta,\epsilon}^2 \|G\|_2^2 \, I_n \right)^{-1}. \end{aligned} \quad (11)$$

At this point, we have followed steps (i) and (ii) in the procedure outline in Section III-A. For step (iii), we express the MSE between y at the output of F and $\hat{y}$ at the output of H, as a function of G, which can be written here $\frac{1}{2\pi}\int_{-\pi}^{\pi} \mathrm{Tr}(P_y(e^{j\omega}) - P_{\hat{y}}(e^{j\omega}))d\omega$. From (11), as detailed in [26, Section VI], we can obtain an explicit expression for $P_{\hat{y}}(e^{j\omega})$, which depends on $G(e^{j\omega})$. Then, it turns out that optimizing the MSE over G can be done in certain cases by solving a certain "waterfilling" type optimal allocation problem of the same nature as those encountered in the communications literature on joint transmitter-receiver design [46], [47], or via semidefinite programming [26].

One important remaining issue however is that the procedure outlined above relies on the expression (11), which is that of the Wiener smoother, i.e., the filter H is in general non causal. In certain cases, we can still implement the resulting mechanism, at least approximately, by adding a small delay. Otherwise, we need to constrain H explicitly to be a causal Wiener filter. How to compute H for a given G in this case is well known [42], but the expression of the performance measure as a function of G becomes more complicated and not as explicit as in the case where H is unconstrained. As a result, it is not currently known how to optimize over causal mechanisms for the problem of this section.

So in general, the optimization procedure based on (11) only provides a bound on achievable performance for the design of causal mechanisms. One can then resort to suboptimal heuristics. For example, a scheme that seems to work well in practice is to use the square root signal shaping block of Theorem 3.3, which was optimal for the ZFE mechanism, followed by a causal Wiener filter H replacing the zero-forcing equalizer. The performance obtained can then be compared to the performance bound to provide an indication of how far from optimal the scheme is.

2) Privacy Preserving Kalman Filtering: Suppose now that instead of having the type of statistical model for stationary signals of the previous subsection, we know that the input signals provided by individuals are generated according to a known linear state space model. For example, consider n individuals periodically reporting their positions $x_i^{(1)} := p_i$, and velocities $x_i^{(2)} := v_i$, $i \in \{1, \dots, n\}$, obtained for example via GPS devices that they carry. From this data we would like to publish a differentially private estimate of the average velocity of the group, i.e., $v = \frac{1}{n}\sum_{i=1}^n v_i$. For $x_{i,t} = [p_{i,t}^T, v_{i,t}^T]^T \in \mathbb{R}^{2d}$, we can consider the noisy kinematic model

$$\begin{aligned} x_{i,t+1} &= \begin{bmatrix} I_d & T_s I_d \\ 0 & I_d \end{bmatrix} x_{i,t} + \begin{bmatrix} \frac{T_s^2}{2} I_d \\ T_s I_d \end{bmatrix} w_{i,t} \qquad (12) \\ &=: Ax_{i,t} + Bw_{i,t}, \end{aligned}$$

where T_s is the sampling period, and w_i, $i \in \{1, \dots, n\}$, is a white Gaussian noise with covariance matrix $W \in \mathbb{R}^{d\times d}$, assumed known. The dynamics (12) model the velocity as a Gaussian random walk, and take into account the physical relationship between speed and velocity. The measurement model is then

$$y_{i,t} = x_{i,t} + \xi_{i,t}, \qquad (13)$$

where ξ_i, $i \in \{1, \dots, n\}$, is a white Gaussian noise with known covariance matrix V.

In the absence of privacy constraint, the estimate $\hat{v}$ of v minimizing the mean squared error is obtained by Kalman filtering [42]. In this example, where all individual dynamics are independent, we can run n independent Kalman filters to produce estimates $\hat{x}_i$, and then let $\hat{v}_t = \frac{1}{n}\sum_{i=1}^n \hat{x}_{i,t}^{(2)}$, as follows

$$\begin{aligned} &\hat{x}_{i,t+1}^- = A\hat{x}_{i,t}^+, \;\; \Sigma_{i,t+1}^- = A\Sigma_{i,t}^+ A^T + BWB^T, \\ &K_{i,t+1} = \Sigma_{i,t+1}^-(\Sigma_{i,t+1}^- + V)^{-1}, \qquad (14) \\ &\hat{x}_{i,t+1}^+ = \hat{x}_{i,t+1}^- + K(y_{i,t+1} - \hat{x}_{i,t+1}^-). \end{aligned}$$

We write $\hat{v}_i = \mathcal{K} y_i$ for the estimate of the velocity obtained through this filter.

To produce a differentially private estimate, once again we start by defining an adjacency relation of interest on the datasets, which are the measured signals $y^T = [y_1^T, \dots, y_n^T]$. Consider for example the following relation, for $S = \mathrm{diag}(s_1 I_d, s_2 I_d)$ a diagonal matrix

$$\mathrm{Adj}(y, \tilde{y}) \text{ iff for some } i, \; \|S(y_i - \tilde{y}_i)\|_2 \le \rho, \qquad (15)$$

and $y_j = \tilde{y}_j$ for all $j \neq i$.

Differential privacy with respect to (15) aims at hiding weighted ℓ_2-variations of size ρ in the signal y_i of a single individual. Higher values of s_1 (resp. s_2) provide less protection to component p_i (resp. v_i). Then, a simple differentially private mechanism is the input perturbation scheme, where each individual directly perturbs its data. Since we have

$$\max_{\mathrm{Adj}(y,\tilde{y})} \|y - \tilde{y}\|_2 = \max_{\|z-\tilde{z}\|_2 \le \rho} \|S^{-1}(z - \tilde{z})\|_2 \le \rho \|S^{-1}\|_2,$$

we see from Theorem 3.1 that $\check{y}_{i,t} = y_{i,t} + \zeta_{i,t}$ is differentially private if ζ_i a white Gaussian noise with covariance matrix $\Xi = \kappa_{\delta,\epsilon}\, \rho \|S^{-1}\|_2 I_{2d}$. The advantage of this scheme is that individuals can directly release signals that are differentially private, i.e., there is no need to trust the dataset manager. Once these signals are received, they can be integrated through the same previous Kalman filter, with only a modification in the computation (14) of the Kalman gain, where the matrix V should be replaced by $V + \Xi$, since the privacy preserving noise can simply be viewed as an additional observation noise. We note this filter $\check{\mathcal{K}}$ and the estimate $\check{v}_i$, so that $\check{v}_i = \check{\mathcal{K}}\check{y}_i$.

In general however, this input perturbation scheme for an adjacency relation such as (15) does not achieve the optimal scaling for the MSE as a function of n. Indeed, we have that

$$\mathrm{Var}(\check{v} - v) = \frac{1}{n^2}\sum_{i=1}^n \mathrm{Var}(\check{\mathcal{K}}_i \check{y}_i - v_i),$$

i.e., the estimator error variance scales as $1/n$. In contrast, consider an output perturbation scheme where $\hat{v}$ is first computed using a steady-state version $\mathcal{K}_{ss}$ of the Kalman filter

$$\hat{v} = \frac{1}{n}\sum_{i=1}^n \mathcal{K}_{ss} y_i, \qquad (16)$$

and then white Gaussian noise is added to $\hat{v}$ with covariance proportional to the squared sensitivity, producing a final differentially private estimate. The ℓ_2-sensitivity of (16) with respect to (15) is $\frac{1}{n}\|\mathcal{K}_{ss} S^{-1}\|_\infty$ [25] (an ℓ_2-gain computation), so that this output perturbation scheme will have an MSE scaling as $1/n^2$. For a sufficiently small number of individuals, the input perturbation scheme might still be advantageous, although other considerations also have to be taken into account, such as the fact that the matrix Ξ in $\check{\mathcal{K}}$ can slow down the filter's convergence significantly, i.e., the MSE and especially long-term MSE is not necessarily the only performance measure of interest, see Figure 3. Moreover, instead of simply using a Kalman filter in the output perturbation mechanism, $\mathcal{K}_{ss}$ can be redesigned to balance estimation error with $\mathcal{H}_\infty$ norm, i.e., sensitivity, in order to minimize the overall MSE in the final differentially private estimate, see [25, Section IV].

The output perturbation scheme does not change the speed of the transient response of the filter, however, it produces a final estimate still containing raw, white noise in it, which

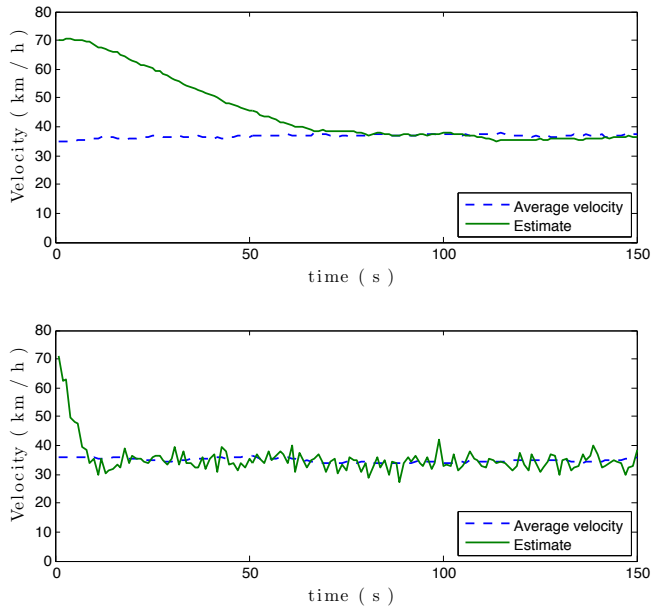

Fig. 3. Examples of differentially private velocity estimates obtained by Kalman filtering with input (top) and output (bottom) perturbation (see [25] for details). The input perturbation scheme gives a better steady-state MSE here, but its convergence is much slower.

intuitively should be furthered filtered. Hence, one should consider once again a more general architecture following Figure 2, containing a signal shaping block G and where now H is a Kalman filter, a choice that is optimal as long as G is taken to be linear. This Kalman filter H will depend on G, and one can then optimize the overall performance with respect to G, following again the procedure outlined in Section III-A.

3) Further Extensions: Nonlinear Systems: The previous paragraphs have only been concerned with linear systems, e.g., in the choice of filters G and H in the ZFE mechanism or its extension using Wiener filtering, or in the use of linear state space systems to model the measured signals in differentially private Kalman filtering. There are however many ways in which nonlinearities should be considered as well. For example, if the input signals u in (7) describe a number of events captured by detectors during a certain period, then they will take discrete values, and intuitively we should take advantage of this information to improve the performance of our mechanism. In this case, a suboptimal but computationally tractable scheme consists in replacing the Wiener filter by a decision-feedback equalizer, which includes a quantization block in the reconstruction filter, see [48].

When we wish to publish a differentially private estimate of an aggregate quantity such as an estimate of the proportion of sick people in a population, or of the proportion of connections between two categories of people in a social network, the underlying dynamics of these quantities are often strongly nonlinear. As an example, consider the following so-called SIR model [49], [50], which models the evolution of an epidemic in a population. Individuals are divided into 3 categories: susceptible (S) individuals who might become infected if exposed, infectious (I) individuals who can transmit the infection and recovered (R) individuals, who are then immune to the infection. The continuous-time version of this model is bilinear

$$\begin{aligned} \frac{ds}{dt} &= -\mu\mathcal{R}_o is \\ \frac{di}{dt} &= \mu\mathcal{R}_o is - \mu i. \end{aligned} \tag{17}$$

Here i and s represent the proportion of the total population in the classes I and S. The parameter $\mathcal{R}_o$ is called the "basic reproduction number" and represents the average number of individuals infected by a sick person. The epidemic can propagate when $\mathcal{R}_o > 1$. The parameter μ represents the rate at which infectious people recover.

Syndromic surveillance systems monitor health related data in real-time in a population to facilitate early detection of epidemic outbreaks [51]. One can view the data they collect as noisy measurements $y_k = i_k + \nu_k$ of the proportion $i(t)$ of infected individuals at times $\{t_k\}_{k\geq 0}$, which should be fused with the model (17) to produce a refined estimate $\{\hat{i}_k\}_{k\geq 0}$. However, the signal y_k is an aggregation of typically privacy-sensitive data, e.g., records of visits to emergency rooms. To ensure that $\hat{i}_k$ is differentially private, the distribution of this estimate should not be too sensitive to variations in y_k that can be attributed to a single individual. This leads us to consider in [52] to consider the following adjacency relation

$$\begin{aligned} &\text{Adj}(y,\tilde{y}) \text{ iff} \\ &\exists k_0 \geq 0 \text{ s.t. } \begin{cases} y_k = \tilde{y}_k, & k < k_0 \\ |y_k - \tilde{y}_k| \leq K\alpha^{k-k_0}, & k \geq k_0, \end{cases} \end{aligned} \tag{18}$$

where $K > 0$, $0 \leq \alpha < 1$ are given constants. Differential privacy with respect to (18) aims at hiding transient deviations starting at any time k_0 that subsequently decrease geometrically, and which could be due to a single individual's data influencing the syndromic surveillance dataset for some time as he becomes infected. With a larger population sampled, we can choose K smaller as the effect of one individual becomes smaller, which then allows us to decrease the level of noise in our estimate $\hat{i}$.

Let us denote $\mathcal{F}$ our estimator of $\hat{i}$, i.e., $\hat{i} = \mathcal{F}y$. A model based estimator should generally rely on (17) and capture the nonlinear dynamics of this model internally, making the operator $\mathcal{F}$ nonlinear. Designing a simple differentially private output perturbation mechanism $\check{i} = \mathcal{F}y + w_1$, with w_1 Gaussian, requires computing the sensitivity of $\mathcal{F}$ with respect to (18), a problem that can be much more difficult than in the linear case, but for which tools from nonlinear system analysis can help [52]. Output perturbation might be preferable to an even simpler input perturbation scheme $\check{i} = \mathcal{F}(y+w_2)$, since the latter can produce a systematic bias at the output when w_2 passes through the nonlinear system

$\mathcal{F}$. Naturally, optimizing the more general architecture of Figure 1 is desirable in this context as well.

There are many other variations and types of problems worth considering in privacy preserving signal processing. For a recent survey of other problems currently under active study, we refer the reader to [53].

IV. Differential Privacy, Entropy, and Consensus

In distributed control systems with shared resources, participating agents can improve the overall performance of the system by sharing data about their personal preferences. Examples include crowd-sourced traffic estimation and navigation applications such as those provided by Google Maps and Waze and also smart meter-based "peak-shaving" in power generation based on user demands.

In this section, we formulate and study a natural trade-off arising in these problems between the privacy of the agent's data and the performance of the control system. Each agent seeks to preserve the privacy of its preference vector, for example, the sequence of way points in a navigation system. The overall control system consists of a group of agents with linear discrete-time coupled dynamics, each controlled to track its preference vector. Performance of the system is measured by the mean squared tracking error. Here, following [20], we discuss a mechanism that achieves differential privacy by adding Laplace noise to the shared information in a way that depends on the sensitivity of the control system to the private data. A special case of this problem is studied in [22], where the agents need to achieve average consensus through iterative interactions while preserving the privacy of their initial states. We also investigate the best estimates that can be derived from a sequence of the noisy shared data from an adversary's point of view. We show that there exists a lower-bound for the entropy of any unbiased estimator of the private data from any noise-adding mechanism that gives ε-differential privacy, and that the mechanism achieving this lower-bound is a randomized mechanism that also uses Laplace noise.

A. System Formulation

Consider a linear distributed control system with n agents, cf. Figure 4, whose dynamics are coupled

$$x_i(t+1) = Ax_i(t) + v_i(t) + \frac{c}{n}\sum_{j=1}^{n} x_j(t),$$

where (a) $x_i(t) \in \mathbb{R}^d$ is the state of agent i at time $t < T$; (b) $v_i(t) \in \mathbb{R}^d$ is the local control input; (c) $c \in \mathbb{R}$ is a coupling coefficient capturing the aggregate influence of the other agents. This model of coupling via average states is adopted for the sake of simplicity. The techniques described here can be extended to general linear couplings between the agents with moderate revisions.

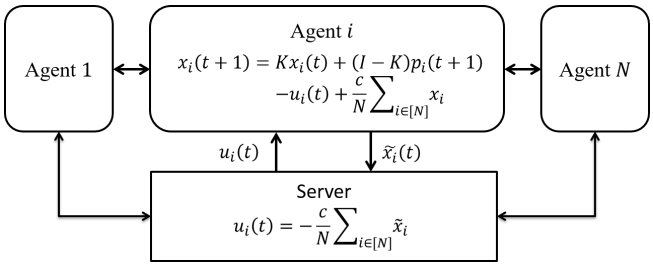

Fig. 4. Diagram of a distributed control system.

To achieve better performance, the agents need to exchange information about their states. Since the communication between the agents may be compromised, the agents choose to share only noisy versions of their states via a *randomized mechanism* $\mathcal{M}$ for the sake of privacy. Specifically, at each time $t \geq 0$, the i^{th} agent adds mean-zero noise $n_i(t)$ to its state and reports this noisy state $\tilde{x}_i(t)$ to the other agents

$$\tilde{x}_i(t) = x_i(t) + n_i(t). \tag{19}$$

The aggregation and dissemination of the noisy states can be performed either via a central server, as shown in Figure 4, or in a fully distributed or peer-to-peer fashion.

In addition, each agent i is also associated with a sequence $p_i(t)$ of (possibly constant) preferences or waypoints that it aims to track. To achieve this, the agent uses a feedback control $v_i(t)$ based on the information of *average state* $u'(t) = -\frac{c}{n}\sum_{j=1}^{n}\tilde{x}_j$ received from the server and adopts the linear feedback control law

$$v_i(t) = K'(x_i(t) - p_i(t+1)) + (I-A)p_i(t+1) - u'(t),$$

where the $K'(x_i(t) - p_i(t+1))$ is a linear feedback term of the tracking error, $(I-A)p_i(t+1)$ is an additive term to move the equilibrium of $x_i(t)$ to $p_i(t+1)$, and $-u'(t)$ tries to cancel the effect of the aggregate state. Thus, we have

$$\begin{aligned}
\tilde{x}_i(t) &= x_i(t) + n_i(t),\\
u'(t) &= \frac{c}{n}\sum_{j=1}^{n}\tilde{x}_j,\\
x_i(t+1) &= Kx_i(t) + (I-K)p_i(t+1)\\
&\qquad - u'(t) + \frac{c}{n}\sum_{j=1}^{n} x_j(t),
\end{aligned}$$

where $K = K' + A \in \mathbb{R}^{d\times d}$ is the closed loop dynamics matrix. If a different linear feedback control scheme is used, then the system equations can be modified accordingly. Combining these equations, the closed-loop dynamics of agent $i \in \{1,\dots,n\}$ is

$$x_i(t+1) = Kx_i(t) + (I-K)p_i(t+1) - \frac{c}{n}\sum_{j=1}^{n} n_j(t). \tag{20}$$

The state of the i^{th} agent at time $t+1$ can be written as a function of its preference sequence $\{p_i(s)\}_{s\leq t}$ and the

sequence $\{n_i(s)|i \in \{1,\dots,n\}, s \le t\}$ of noise vectors added in all previous rounds.

Let $x(t)$, $\tilde{x}(t)$, $n(t)$ and $p(t)$ be the aggregated state, noisy reported state, noise and preference respectively. From (20), the aggregated closed loop dynamics can be written as

$$x(t+1) = \mathbf{K}x(t) + (I-\mathbf{K})p(t+1) - \mathbf{C}n(t),$$

or equivalently

$$x(t+1) = (\mathbf{K}+\mathbf{C})x(t) - \mathbf{C}\tilde{x}(t) + (I-\mathbf{K})p(t+1),$$

where $\mathbf{K} = I_n \otimes K$, $\mathbf{C} = \mathbf{1}_n \otimes \frac{cI_d}{d}$, I_n is the $n \times n$ identity matrix, $\mathbf{1}_n$ is the $n \times n$ matrix with all elements being 1, and $\otimes$ denotes the Kronecker product.

Solving the above two equations gives

$$\begin{aligned} x(t) = (\mathbf{K}+\mathbf{C})^t x(0) - \sum_{s=0}^{t-1}(\mathbf{K}+\mathbf{C})^{t-s}u(t) \\ + \sum_{s=1}^{t}(\mathbf{K}+\mathbf{C})^{t-s}(I-\mathbf{K})p(s), \end{aligned} \tag{21}$$

where $u(t) = (u'(t),\dots,u'(t))^{\mathrm{T}}$ is the aggregated average state, and

$$\begin{aligned} x(t) =& (\mathbf{K}+\mathbf{C})^t x(0) \\ &+ \sum_{s=0}^{t-1}(\mathbf{K}+\mathbf{C})^{t-s-1}((I-\mathbf{K})p(s+1) - \mathbf{C}\tilde{x}(t)). \end{aligned} \tag{22}$$

1) Differential Privacy and Bayesian Inference: For a time horizon T, the agents need to keep the *private data* $D = \{d_i\}_{i\in\{1,\dots,n\}} = \{(x_i(0), p_i(1),\dots,p_i(T-1))\}_{i\in\{1,\dots,n\}}$, namely their initial states and preferences ϵ-differentially private, under the observation $O_D = \{\tilde{x}(t)\}_{t<T} \in \mathbb{R}^{dnT}$ of the randomized aggregated reported states, parametrized by D. This leads us to the following definition of ϵ-differential privacy for the private data.

Definition 4.1: (Differential privacy over time horizon): Given a time horizon $T > 0$ and a parameter $\epsilon > 0$, a randomized mechanism $\mathcal{M} : D \to O_D$ is ϵ-differentially private up to time $T-1$, if for any subset $\mathcal{O} \subseteq \mathbb{R}^{dnT}$ and any two data sets D, D', the inequality

$$\mathbb{P}[O_D \in \mathcal{O}] \le e^{\epsilon\|D-D'\|_1}\mathbb{P}[O_{D'} \in \mathcal{O}] \tag{23}$$

holds, where the random variables O_D and $O_{D'}$ are the observations generated by the two data sets D and D'.

Note that, if the system is ϵ-differentially private up to time $T-1$, then it is ϵ-differentially private up to any time $S < T$. Consider the unbiased estimator

$$\hat{D} = \{(\hat{x}_i(0), \hat{p}_i(1),\dots,\hat{p}_i(T-1)) \mid i \in \{1,\dots,n\}\}$$

of the private data set from a sequence of reported states O_D. We will show in Section IV-C that if the private data D is ϵ-differentially private, then there is a lower bound on the entropy of $H(\hat{D})$ and the minimum is achieved by mechanisms that add Laplace noise.

2) Measuring the Tracking Performance: We use the second moment of the tracking error to define a cost function for agent i up to time $T-1$

$$\mathcal{C}(\epsilon, D, i) = \mathbb{E}\left[\sum_{t=1}^{T-1}\|x_i(t) - p_i(t)\|_2^2\right].$$

It increases with time T. This cost is non-zero even when there is no noise in the communication, namely $n_i(t) = 0$ for all t. In this case, $\epsilon \to \infty$, and we denote the zero-noise cost by

$$\mathcal{C}_0(D, i) = \lim_{\epsilon\to\infty}\mathcal{C}(\epsilon, D, i).$$

The *cost of privacy* is defined as the supremum in the change of single agent's cost over all data sets relative to the non-private mechanism

$$\Delta(\epsilon, T) = \sup_{i\in\{1,\dots,n\}, D\in\mathcal{D}}\left(\mathcal{C}(\epsilon, D, i) - \mathcal{C}_0(D, i)\right).$$

We will show in Section IV-B that if the private data D is ϵ-differentially private, then there is a lower bound on the entropy of $H(\hat{D})$ and the minimum is achieved by mechanisms that add Laplace noise.

B. Differentially Private Linear Distributed Control

The feedback nature of the system leads to the following observation: given the private data set D, the system trajectory $\{x(t)\}_{t<T}$ is uniquely determined by the value of the sequence of reported states $O_D = \{\tilde{x}(t)\}_{t<T}$, which we denote by $\rho(D, O_D)$. The influence on $\rho(D, O_D)$ of changing D is captured by the notion of *sensitivity*.

Definition 4.2 (Sensitivity): The *sensitivity* of a randomized mechanism $\mathcal{M}$ at time $t \ge 0$ is

$$S(t) = \sup_{D,D'\in\mathcal{D},\ O\in\mathbb{R}^{dnT}} \frac{\|\rho(D, O_D)(t) - \rho(D', O_{D'})(t)\|_1}{\|D - D'\|_1}.$$

A randomized mechanism $\mathcal{M}$ that keeps the data D ϵ-differentially private can be designed by using Laplace noise to cover the change in private data [20].

Theorem 4.3: (Differential privacy over time horizon via Laplace noise): For $\epsilon > 0$ and a time horizon $T > 0$, let $M_t = TS(t)/\epsilon$. A randomized mechanism defined by

$$n(t) \sim \mathrm{Lap}(M_t, dn)$$

for $t < T$ in (19) is ϵ-differentially private.

The following result gives a bound on the sensitivity for the system. To prove it, we fix two private data sets D and D', and calculate the bound on the distance between the two corresponding trajectories under the same observation by decomposing it into (1) the change in agent i's state, and (2) the sum of changes in other agents' state.

Theorem 4.4: (Upper bound on sensitivity): For the linear distributed control system, for all $t \in \mathbb{N}$ the sensitivity

$S(t) \leq \kappa(t)$, where κ is defined as

$$\kappa(t) := \|G^t - K^t\|_1 + \|K^t\|_1 + \|H\|_1 \sum_{s=0}^{t-1} (\|G^s - K^s\|_1 + \|K^s\|_1)$$

with $G := cI + K$ and $H := I - K$.

The upper bound $\kappa(t)$ on the sensitivity at time t is independent of the number of agents. It only depends on the matrix K (specified by the individual's control function) and the coupling coefficient c and time t. The upper bound κ has two components:

(i) $\|K^t\|_1 + \|H\|_1 \sum_{s=1}^{t} \|K^s\|_1$ over-approximates the change in the i^{th} agent's state x_i if its own preference changes at each time up to t, and

(ii) $\|G^t - K^t\|_1 + \|H\|_1 \sum_{s=0}^{t-1} \|G^s - K^s\|_1$ over-approximates the sum of the changes in other agents' state given agent i's preference changes up to t.

When K is stable, $\|K^t\|_1$ decays to 0. The coupling coefficient c quantifies the influence of the aggregate on each individual agent. The matrix $G = cI + K$ captures the combined dynamics under the influence of the environment and the dynamics of the individual agents. The weaker the physical coupling, the smaller $\|G^t\|_1$. As the individual agent dynamics becomes more stable or the physical coupling between agents becomes weaker, the sensitivity of the system decreases. This gives the following estimation on the growth rate of the cost of privacy with time.

Theorem 4.5 (Cost of privacy): The cost of privacy of the ϵ-differentially private mechanism $\mathcal{M}$ of Theorem 4.3 is of the order of $O(\frac{T^3}{n\epsilon^2})$ if the matrix K is Hurwitz. Otherwise it grows exponentially with T.

C. Estimation of Differentially Private Linear Distributed Systems

Here, we study the problem of estimating the private data using privacy preserving mechanisms like the ones discussed above from the point of view of Bayesian inference. Let $\hat{D}$ be an unbiased estimator of the private data set D given observation O_D up to time $T-1$. We show here that there is a lower bound on the entropy of such estimators for any ϵ-differentially private mechanism $\mathcal{M}$.

For a single timestep $T = 1$, we prove that Laplace-noise-adding mechanism minimizes that entropy of the unbiased estimator.

Theorem 4.6: (Lower bound on entropy of estimator: single timestep): Given a invertible $M \in \mathbb{R}^{d\times d}$ and a randomized mechanism $\tilde{x} = Mx + w$ that protects the ϵ-differential privacy of the private data set $x \in \mathbb{R}^d$ by adding mean-zero noise $w \in \mathbb{R}^d$ from one-shot observation $\tilde{x}$, the entropy of any unbiased estimator $\hat{x}$ from observation $\tilde{x}$ satisfies

$$H(\hat{x}) \geq H(M\lambda),$$

and the minimum is achieved by using $\hat{x} = M^{-1}\tilde{x}$ and adding noise $n = M\lambda$ where $\lambda \sim \text{Lap}(1/\epsilon, n)$. In particular, when $M = I$, we have

$$H(\hat{x}) \geq d(1 - \ln(\epsilon/2)).$$

The iterative application of Theorem 4.6 extends the result to an arbitrary time horizon $T \in \mathbb{N}$.

Theorem 4.7: (Lower bound on entropy of estimator: arbitrary horizon): If the private data set D is ϵ-differentially private up to time $T-1$ and $I - K$ is invertible, then the entropy of any unbiased estimator $\hat{D}$ of the private data set is at least

$$nd(1 - \ln(\epsilon/2)) + n(T-1)H((I-K)w),$$

where $w \sim \text{Lap}(1/\epsilon, d)$. The minimum is achieved by

$$n(0) = \lambda(0),$$

$$n(t) = (\mathbf{K} + \mathbf{C})^t \lambda(0) + \sum_{s=1}^{t} (\mathbf{K} + \mathbf{C})^{t-s} (I - \mathbf{K}) \lambda(s),$$

for $t \geq 1$, where $\lambda(t) \sim \text{Lap}(1/\epsilon, dn)$ are independent dn-dimensional Laplace noise for $t \in \{0, \ldots, T-1\}$.

As shown in Theorem 4.7, the minimal entropy of the estimator depends linearly on the number of agents n, the dimension n of the state of each agent and the time horizon T. In addition, the minimal entropy increases as the privacy level increases, namely ϵ decreases. Finally, it only depends on the dynamics of each agent and independent of the coupling coefficient c. This is because by communicating with others, the coupling in the dynamic of the agents has been canceled (with some noise left), thus the initial state and the preferences of each agent propagate only by the local dynamics K.

V. Differential Privacy and Distributed Optimization

In this section we consider convex optimization problems over networks, where the objective function can be written as the aggregate of local objective functions, each one known to one individual agent. In such scenarios, agents must cooperate with each other to determine the global optimizer given that their information about the optimization problem is incomplete. However, when doing so, they want to avoid revealing sensitive information about themselves such as, for instance, their private objective functions. This is where the concept of differential privacy comes in. The outline of the section is as follows: we start by formalizing the problem mathematically, then describe potential approaches to solve it in a distributed and differentially private way, and finally examine the pros and cons of each approach.

A. The Problem

Consider a group of n agents whose communication topology is described by an undirected graph $\mathcal{G}$. Each agent

corresponds to a vertex in the graph, whereas communication links are represented by the edges. Each agent $i \in \{1, \dots, n\}$ has a local objective function $f_i : D \to \mathbb{R}$, where $D \subset \mathbb{R}^d$ is convex and compact and has nonempty interior. We assume that each $f_i, i \in \{1, \dots, n\}$ is convex and twice continuously differentiable, and use the shorthand notation $F = \{f_i\}_{i=1}^n$. Consider the following convex optimization problem

$$\underset{x \in X}{\text{minimize}} \;\; f(x) := \sum_{i=1}^{n} f_i(x). \tag{24}$$

Here, $X \subseteq D$ is the feasibility set, which we assume is a global piece of information known to all agents.

The group objective is to solve the convex optimization problem (24) in a distributed and private way. By distributed, we mean that each agent can only interact with its neighbors in the graph $\mathcal{G}$. Regarding privacy, we consider the case where the function f_i (or some of its attributes) constitute the local and sensitive information known to agent $i \in \{1, \dots, n\}$ that has to be kept confidential. Other scenarios are also possible, such as preserving the privacy of the agent state or its local constraints. Each agent assumes that the adversary has access to all the "external" information (including all the network communications and all other objective functions). This setting is sometimes called local (differential) privacy in the literature, see e.g., [54].

Given that the objects to preserve the privacy of are functions, and therefore belong to an infinite-dimensional space, the standard definition of differential privacy needs some adjustments. Let us first introduce the notion of adjacency. We denote by $L_2(D)$ the set of square-integrable measurable functions defined on D. Given any normed vector space $(\mathcal{V}, \|\cdot\|_{\mathcal{V}})$ with $\mathcal{V} \subseteq L_2(D)$, two sets of functions $F, F' \subset L_2(D)$ are $\mathcal{V}$-adjacent if there exists $i_0 \in \{1, \dots, n\}$ such that

$$f_i = f_i', \; i \neq i_0 \quad \text{and} \quad f_{i_0} - f_{i_0}' \in \mathcal{V}.$$

The set $\mathcal{V}$ is a design choice that we specify later. This definition can be readily extended to the case where $\mathcal{V}$ is any subset of another normed vector space $\mathcal{W} \subseteq L_2(D)$. With this generalization, the conventional bounded-difference notion of adjacency becomes a special case of the definition above, where $\mathcal{V}$ is a closed ball around the origin. We provide next a general definition of differential privacy for a map.

Definition 5.1: (Functional differential privacy): Let $(\Omega, \Sigma, \mathbb{P})$ be a probability space and consider a random map

$$\mathcal{M} : L_2(D)^n \times \Omega \to \mathcal{X}$$

from the function space $L_2(D)^n$ to an arbitrary set $\mathcal{X}$. Given $\epsilon \in \mathbb{R}^n_{>0}$, the map $\mathcal{M}$ is ϵ-differentially private if, for any two $\mathcal{V}$-adjacent sets of functions F and F' that (at most) differ in their i_0'th element and any set $\mathcal{O} \subseteq \mathcal{X}$, one has

$$\begin{aligned} &\mathbb{P}\big[\{\omega \in \Omega \mid \mathcal{M}(F', \omega) \in \mathcal{O}\}\big] \\ &\qquad \leq e^{\epsilon_{i_0} \|f_{i_0} - f_{i_0}'\|_{\mathcal{V}}} \mathbb{P}\big[\{\omega \in \Omega \mid \mathcal{M}(F, \omega) \in \mathcal{O}\}\big]. \end{aligned} \tag{25}$$

Note that, in Definition 5.1, the map $\mathcal{M}$ has sets of n functions as argument, corresponding to the individual objective functions available to the agents. The meaning of this definition is the standard one in the context of differential privacy: the statistics of the output of $\mathcal{M}$ should change only (relatively) slightly if the objective function of one agent changes (and the change is in $\mathcal{V}$), making it hard to an "adversary" that observes the output of $\mathcal{M}$ to determine this change.

In the case of an iterative algorithm, where agents repeatedly interchange information with their neighbors and perform computations combining it with their own local private information, one should think of the map $\mathcal{M}$ as representing the action of the entire algorithm on the set of local functions F. The result of such action is observed by the adversary. In other words, $\mathcal{M}$ is a map, parameterized by the initial network state, that assigns to F the whole sequence of messages transmitted over the network. The underlying understanding is therefore that (25) has to hold for all allowable values of the initial network states. Having clarified this point, the objective of this section can be formalized as follows.

Problem 1: (Differentially private distributed optimization): Design a distributed and differentially private optimization algorithm whose guarantee on accuracy improves as the level of privacy decreases, leading to the exact optimizer of the aggregate objective function in the absence of privacy. •

The requirement of recovering the exact optimizer in the absence of privacy in our problem statement is motivated by the privacy-accuracy trade-off in differential privacy. The existence of this trade-off is well known, albeit its characterization for any specific task is in general challenging. This trade-off essentially states that there is always a cost for an algorithm to be differentially private, i.e., the algorithm inevitably suffers a performance loss that increases as the level of privacy increases. This phenomenon is a result of the noise added in the map $\mathcal{M}$, whose variance increases as ϵ decreases. With the requirement on the noise-free behavior of the algorithm made in our problem statement, we seek to ensure that the cause of this performance loss is *only* due to the noise added to guarantee privacy, and not to any other additional factor. We come back to this point later in our discussion.

B. Approaches to Algorithm Design

The two main requirements (distributed and differentially private) on the coordination algorithm raise major challenges to tackle its design. Various iterative algorithms have been proposed in the literature, e.g.. [55]–[59] and references therein, to solve optimization problems of the form (24) in a distributed fashion. Although we do not aim to survey these efforts here, it is instructive to consider a sample of this literature to illustrate the potential routes to address the algorithmic solution of Problem 1. To this effect, consider the algorithm proposed in [56] which has each agent

$i \in \{1, \dots, n\}$ start with an initial estimate $x_i(0)$ of the optimizer and, at each iteration k, update its estimate as

$$x_i(k+1) = \text{proj}_X(z_i(k) - \alpha_k \nabla f_i(z_i(k))), \tag{26a}$$

$$z_i(k) = \sum_{j=1}^{n} a_{ij} x_j(k), \tag{26b}$$

where $\{a_{ij}\}_{j=1}^n$ are the edge weights of the communication graph at node i and α_k is the stepsize. From (26b), one can see that agents only need to share their estimates with their neighbors to run the algorithm. Under standard assumptions on the connectivity of the communication graph, one can show [56] that $x_i(k)$ converges to the optimizer x^* asymptotically if the sequence of stepsizes is square-summable ($\sum_k \alpha_k^2 < \infty$) but not summable ($\sum_k \alpha_k = \infty$).

Algorithm (26) therefore solves Problem 1 in a distributed way. How can we endow distributed coordination algorithms such as this with privacy guarantees so that their execution does not reveal information about the local objective functions to the adversary? We consider two approaches to tackle this.

1) Message-Perturbing Strategies: Inspired by the developments presented in the previous sections of this paper, we could prescribe that the agents add noise to the messages that they send (either to their neighbors or a central aggregator, depending on the specific algorithm). Using some of the typical families of noise in differential privacy, such as Gaussian or Laplace noise, should render the resulting algorithm differentially private. For algorithm (26), this approach would result in each agent $i \in \{1, \dots, n\}$ executing

$$x_i(k+1) = \text{proj}_X(z_i(k) - \alpha_k \nabla f_i(z_i(k))), \tag{27a}$$

$$z_i(k) = \sum_{j=1}^{n} a_{ij} \xi_j(k), \tag{27b}$$

where $\xi_j(k) = x_j(k) + \eta_j(k)$ is the perturbed message received from agent j at time k. In fact, this is the algorithm considered in [28]. The challenge with this approach then is to determine how the noise affects the stability and convergence properties of the algorithm. As an example, if the original algorithm is not robust to disturbances, then the addition of noise might completely de-stabilize it, driving it arbitrarily far from the optimizer.

2) Objective-Perturbing Strategies: An alternative design approach consists of directly perturbing the agents' objective functions with noise in a differentially private way and then have them participate in a distributed optimization algorithm, but with the perturbed objective functions instead of their original ones. The latter in turn automatically adds noise to the estimates shared with neighbors (as in the message-perturbing approach, but possessing an intrinsically different structure). For algorithm (26), this approach would result in each agent $i \in \{1, \dots, n\}$ executing

$$x_i(k+1) = \text{proj}_X(z_i(k) - \alpha_k \nabla \tilde{f}_i(z_i(k))), \tag{28a}$$

$$z_i(k) = \sum_{j=1}^{n} a_{ij} x_j(k), \tag{28b}$$

where $\tilde{f}_i$ is a perturbed version of the local objective function f_i of agent i. In this design approach, the resilience to post-processing of differential privacy, cf. Theorem 2.6, ensures that the combination of objective perturbation with the distributed optimization algorithm does not affect the differential privacy at the functional level. There are of course challenges associated with this approach too, starting with the idea of properly formalizing a procedure to ensure functional differential privacy, passing through ensuring that the resulting perturbed functions enjoy the smoothness and regularity properties required by distributed optimization algorithms to converge, and finally characterizing the accuracy of the resulting strategy.

C. Message-Perturbing Strategies

As outlined above, we use the term *message-perturbing strategy* to refer to the result of modifying any of the distributed optimization algorithms available in the literature by adding (Gaussian or Laplace) noise to the messages agents send to either neighbors or a central aggregator in order to preserve privacy. A generic message-perturbing distributed algorithm takes the form

$$\begin{aligned} x(k+1) &= a_{\mathcal{I}}(x(k), \xi(k)), \\ \xi(k) &= x(k) + \eta(k), \end{aligned} \tag{29}$$

where $\boldsymbol{\xi}, \boldsymbol{\eta} : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ represent the sequences of messages and perturbations, respectively, and $a_{\mathcal{I}} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ depends on the agents' sensitive information set $\mathcal{I}$ with associated optimizer $x^*_{\mathcal{I}}$. This formulation of message-perturbing strategies is quite general and can also encode algorithmic solutions for optimization problems other than Problem 1, such as the ones studied in [27], [29]). In our setup here, the sensitive information set $\mathcal{I}$ are the agents' objective functions, $\mathcal{I} = F = \{f_i\}_{i=1}^n$.

For convenience, we employ the short-hand notation $\tilde{a}_{\mathcal{I}}(x(k), \eta(k)) = a_{\mathcal{I}}(x(k), x(k) + \eta(k))$ to refer to (29). With this notation, the original distributed algorithm (without any message perturbation) would simply correspond to the dynamics

$$x(k+1) = \tilde{a}_{\mathcal{I}}(x(k), 0). \tag{30}$$

We assume this dynamics is globally asymptotically stable relative to the optimizer x^* of (24) so that, no matter where the network is initialized, the algorithm is guaranteed to solve the optimization problem. Remarkably, such convergence properties are inherently robust, meaning that the system still retains stability after the injection of noise. Formally, this can be expressed as follows [60]: the dynamics (30) is locally asymptotically stable relative to x^* if and only if is locally input-to-state stable relative to x^*. The latter notion means that there exist $\rho > 0$, a class $\mathcal{KL}$ function γ, and a class $\mathcal{K}$ function κ such that, for every initial condition

$x(0) \in B(x^*, \rho)$ and every disturbance satisfying $\|\boldsymbol{\eta}\|_\infty \leq \rho$, the trajectories of (29) satisfy

$$|x(k) - x^*| \leq \max\{\gamma(|x(0) - x^*|, k), \kappa(|\boldsymbol{\eta}_{k-1}|_\infty)\},$$

for all $k \in \mathbb{N}$. This equation precisely describes how the noise injection in the original distributed coordination algorithm disrupts exact convergence to the optimizer by an amount proportional to the size of the noise.

The following result makes an important observation regarding the message-perturbation algorithm design approach. Essentially, the result says that, if the underlying, noise-free dynamics (e.g., system (30)) is asymptotically stable (something which is reasonable to assume, since at least the algorithm should solve the optimization problem when there are no privacy concerns to take care of), then the algorithm that results from injecting asymptotically vanishing noise in the agents' messages cannot be differentially private.

Theorem 5.2: (Limitations to message-perturbing algorithms [30]): Consider the dynamics (29) with either $\eta_i(k) \sim \text{Lap}(b_i(k))$ or $\eta(k) \sim \mathcal{N}(0, b_i(k))$. If $\tilde{a}_{\mathcal{I}}$ is locally input-to-state stable relative to $x^*_{\mathcal{I}}$ for two information sets $\mathcal{I}$ and $\mathcal{I}'$ with different optimizers $x^*_{\mathcal{I}} \neq x^*_{\mathcal{I}'}$ and associated radii ρ and ρ', respectively, $b_i(k)$ is $O(\frac{1}{k^p})$ for all $i \in \{1, \dots, n\}$ and some $p > 0$, and at least one of the following holds,

(i) $x^*_{\mathcal{I}}$ is not an equilibrium point of $x(k+1) = \tilde{a}_{\mathcal{I}'}(x(k), 0)$ and $\tilde{a}_{\mathcal{I}'}$ is continuous,
(ii) $x^*_{\mathcal{I}}$ belongs to the interior of $B(x^*_{\mathcal{I}'}, \rho')$,

then no algorithm of the form (29) can preserve the ϵ-differentially privacy of the information set $\mathcal{I}$ for any $\epsilon > 0$.

Note that the hypotheses of Theorem 5.2 are mild and easily satisfied in most cases. In particular, the result holds if the dynamics are continuous and globally asymptotically stable relative to $x^*_{\mathcal{I}}$ for two information sets. The proof of the result essentially proceeds by establishing that, if the initial state is close to the equilibrium of the system for one information set, the state trajectory converges to that equilibrium with positive probability but to the equilibrium of the system with the other information set with probability zero. Using this fact, one can rule out differential privacy for the resulting coordination algorithm. The interested reader may refer to [30] for a complete exposition of the proof details.

As a consequence of Theorem 5.2, to make the message-perturbing design approach work, one can either make the algorithm terminate in a finite number of time steps (and adjust accordingly the noise level to make the strategy differentially private) or use stepsizes with finite sum to make the zero-input dynamics not asymptotically stable. The first route is pursued in [27], where the agents' local constraints are the sensitive information (instead of the objective function). This algorithmic solution uses a constant-variance noise, which would make the dynamics unstable if executed over an infinite time horizon. This problem is circumvented by having the algorithm terminate after a finite number of steps, and optimizing this number offline as a function of the desired level of privacy ϵ. The second route is pursued in [28], which proposes (27) and chooses a finite-sum sequence of stepsizes $\{\alpha_k\}$ (i.e., $\sum_k \alpha_k < \infty$) in the computation (26a), leading to a dynamical system which is not locally asymptotically stable in the absence of noise.

D. Objective-Perturbing Strategies

As outlined above, we use the term *objective-perturbing strategy* to refer to an algorithm where agents participate in a distributed optimization algorithm with the functions obtained by perturbing the original agents' objective functions with noise in a differentially private way. The first step in this approach is to formalize a methodology to ensure functional differential privacy, which we tackle next.

1) Functional Differential Privacy via Laplace Noise: Consider a function $f \in L_2(D)$ whose differential privacy has to be preserved. The basic procedure to achieve this is still to introduce noise to masquerade the function itself. To do this, we rely on the fact that $L_2(D)$ is a separable Hilbert space, and hence admits a countable orthonormal basis. This means that f can be written as an infinite sequence of coefficients, corresponding to the elements of that basis, which can be conveniently corrupted by noise. Formally, let $\{e_k\}_{k=1}^\infty$ be an orthonormal basis for $L_2(D)$. For any $f \in L_2(D)$, one has

$$f = \sum_{k=1}^{\infty} \langle f, e_k \rangle e_k.$$

We define the coefficient sequence $\boldsymbol{\theta} \in \mathbb{R}^{\mathbb{N}}$ by $\theta_k = \langle f, e_k \rangle$ for $k \in \mathbb{N}$. Then, $\boldsymbol{\theta} \in \ell_2$ (the space of square-summable infinite sequences) and, by Parseval's identity, $\|f\| = \|\boldsymbol{\theta}\|$. For ease of notation, we define $\Phi : \ell_2 \to L_2(D)$ to be the linear bijection that maps the coefficient sequence $\boldsymbol{\theta}$ to f. We are ready to formally describe the procedure to inject noise.

Definition 5.3: (Functional perturbation): Let the noise sequence $\boldsymbol{\eta} = \{\eta_k\}_{k=0}^\infty \in \mathbb{R}^{\mathbb{N}}$ be defined according to

$$\eta_k \sim \text{Lap}(b_k), \quad k \in \mathbb{N}. \tag{31a}$$

Then, for $f \in L_2(D)$, we define

$$\mathcal{M}(f, \boldsymbol{\eta}) = \Phi\left(\Phi^{-1}(f) + \boldsymbol{\eta}\right) = f + \Phi(\boldsymbol{\eta}). \tag{31b}$$

In Definition 5.3, for $\boldsymbol{\eta}$ to belong to ℓ_2 and for the series $\Phi(\boldsymbol{\eta})$ to converge, the sequence of scales $\{b_k\}_{k=1}^\infty$ cannot be arbitrary. One can show, for instance, if for some $K \in \mathbb{N}$, $p > \frac{1}{2}$ and $s > 1$,

$$b_k \leq \frac{1}{k^p \log k^s}, \quad \forall k \geq K,$$

holds, then $\boldsymbol{\eta}$ defined by (31a) belongs to ℓ_2 with probability one, and hence the map $\mathcal{M}$ in (31b) is well defined. In order to show that this map is differentially private, according to

Definition 5.1, we first specify our choice of adjacency space. Given $q > 1$, consider the weight sequence $\{k^q\}_{k=1}^{\infty}$ and define the adjacency vector space to be the image of the resulting weighted ℓ_2 space under Φ, i.e.,

$$\mathcal{V}_q = \Phi\big(\{\boldsymbol{\delta} \in \mathbb{R}^{\mathbb{N}} \mid \sum_{k=1}^{\infty}(k^q\delta_k)^2 < \infty\}\big).$$

It is not difficult to see that $\mathcal{V}_q$ is a vector space. Moreover,

$$\|f\|_{\mathcal{V}_q} := \Big(\sum_{k=1}^{\infty}(k^q\delta_k)^2\Big)^{\frac{1}{2}}, \quad \text{with } \boldsymbol{\delta} = \Phi^{-1}(f),$$

is a norm on $\mathcal{V}_q$. The next result establishes the differential privacy of the map $\mathcal{M}$.

Theorem 5.4: (Differential privacy of functional perturbation [30]): Given $q > 1$, $\gamma > 0$ and $p \in \big(\frac{1}{2}, q - \frac{1}{2}\big)$, let

$$b_k = \frac{\gamma}{k^p}, \quad k \in \mathbb{N}. \tag{32a}$$

Then, the map $\mathcal{M}$ in Definition 5.3 is ϵ-differentially private with

$$\epsilon = \frac{1}{\gamma}\sqrt{\zeta(2(q-p))}, \tag{32b}$$

where ζ denotes the Riemann zeta function.

Note that the map $\mathcal{M}$ is well defined because (32a) ensures that $\boldsymbol{\eta}$ belongs to ℓ_2 almost surely. The proof of this result proceeds by directly showing that $\mathcal{M}$ satisfies Definition 5.1.

With this procedure available, our idea is then to have each agent perturb their objective function in a differentially private way, using the map $\mathcal{M}$, and then participate in a distributed optimization algorithm with this function. This latter point raises some challenges: in general, the distributed optimization algorithms available in the literature have some basic requirements on the smoothness and convexity of the objective functions to ensure convergence. However, even if the original objective functions enjoy these properties, the addition of Laplace noise performed by $\mathcal{M}$ will in general completely destroy them in the perturbed ones. This is the problem we tackle next.

2) Smoothness and Regularity of the Perturbed Functions: To ensure that the perturbed functions have the smoothness and regularity properties required by the distributed coordination algorithm, we define here appropriate maps that, when composed with $\mathcal{M}$, yield functions with the desired properties. The resilience to post-processing of differential privacy, cf. Theorem 2.6, ensures that differential privacy is retained throughout this procedure.

To ensure *smoothness*, we rely on the fact that $C^2(D)$, the set of twice continuously differentiable functions over D, is dense in $L_2(D)$ and, therefore, given any function g in $L_2(D)$, there exists a smooth function arbitrarily close to it, i.e.,

$$\forall \varepsilon > 0,\ \exists \hat{g}^s \in C^2(D) \quad \text{such that} \quad \|g - \hat{g}^s\| < \varepsilon.$$

Here, ε is a design parameter and can be chosen sufficiently small (later, we show how to do this so that the accuracy of the coordination algorithm is not affected). A natural choice for the smoothening step, if the basis functions are smooth (i.e., $\{e_k\}_{k=1}^{\infty} \subset C^2(D)$), is truncating the infinite expansion of g. Such truncation is also necessary in practical implementations due to the impossibility of handling infinite series. The appropriate truncation order depends on the specific function, the basis set, and the noise decay rate.

To ensure *strong convexity* and *bounded Hessians*, we rely on the observation that the set of twice continuously differentiable functions with bounded gradients and Hessians is a closed subset of the space of twice continuously differentiable functions, and hence the projection onto the subspace is well defined. Formally, given $\overline{u} > 0$, $0 < \alpha < \beta$, let

$$\mathcal{S} = \{h \in C^2(D) \mid |\nabla h(x)| \le \overline{u},\ \forall x \in D \\ \text{and } \alpha I_d \le \nabla^2 h(x) \le \beta I_d,\ \forall x \in D^o\}.$$

This set is convex and closed as a subset of $\mathcal{S}_0$ under the 2-norm. Consequently, the best approximation in $\mathcal{S}$ of a function $h \in C^2(D)$ is its unique orthogonal projection onto $\mathcal{S}$, i.e., $\tilde{h} = \text{proj}_{\mathcal{S}}(h)$. By definition, the projected function has bounded gradient and Hessian.

3) Combination with the Distributed Optimization Algorithm: Putting the above pieces together, we can now have agents locally perturb their objective functions and use them in their computations for any desired distributed coordination algorithm, without adding any additional noise to the inter-agent messages. Specifically, each agent $i \in \{1, \dots, n\}$ first computes

$$\hat{f}_i = \mathcal{M}(f_i, \boldsymbol{\eta}_i) = f_i + \Phi(\boldsymbol{\eta}_i), \tag{33a}$$

where $\boldsymbol{\eta}_i$ is a sequence of Laplace noise generated by i according to (31a) with the choice (32a), then select $\hat{f}_i^s \in \mathcal{S}_0$ such that

$$\|\hat{f}_i - \hat{f}_i^s\| < \varepsilon_i, \tag{33b}$$

and finally compute

$$\tilde{f}_i = \text{proj}_{\mathcal{S}}(\hat{f}_i^s). \tag{33c}$$

After this process, agents participate in *any* distributed optimization algorithm with the perturbed objective functions $\{\tilde{f}_i\}_{i=1}^n$. The following result establishes the differentially private nature of the resulting coordination algorithm and characterizes its accuracy.

Theorem 5.5: (Accuracy of a class of distributed, differentially private coordination algorithms [30]): Consider a group of n agents which perturb their local objective functions $f_1, \dots, f_n \in \mathcal{S}$ according to (33) with Laplace noise (31a) of variance (32a), where $q_i > 1$, $\gamma_i > 0$, and $p_i \in \big(\frac{1}{2}, q_i - \frac{1}{2}\big)$ for all $i \in \{1, \dots, n\}$. Let the agents participate in any distributed coordination algorithm that asymptotically converges to the optimizer $\tilde{x}^*$ of the perturbed aggregate objective function. Then, ϵ_i-differential privacy of

each agent i's original objective function is preserved with $\epsilon_i = \sqrt{\zeta(2(q_i - p_i))}/\gamma_i$ and

$$|\mathbb{E}[\tilde{x}^*] - x^*| \leq \sum_{i=1}^{n} \kappa_n \left(\gamma_i \sqrt{\zeta(2p_i)}\right) + \kappa_n(\varepsilon_i),$$

where κ_n is class $\mathcal{K}_\infty$ function.

Choosing $p_i = \frac{q_i}{2}$ in (32a) for all $i \in \{1, \ldots, n\}$, one can characterize the accuracy-privacy trade-off as

$$|\mathbb{E}[\tilde{x}^*] - x^*| \leq \sum_{i=1}^{n} \kappa_n \left(\frac{\zeta(q_i)}{\epsilon_i}\right) + \kappa_n(\varepsilon_i).$$

From this expression, it is clear that in order for the accuracy of the coordination algorithm not to be affected by the smoothening step, each agent $i \in \{1, \ldots, n\}$ has to take the value of ε_i sufficiently small so that it is negligible relative to $\zeta(q_i)/\epsilon_i$. In particular, this procedure can be executed for any arbitrarily large value of ϵ_i, so that in case of no privacy requirements at all, perfect accuracy is recovered, as specified in Problem 1.

VI. Conclusions and Beyond

Privacy preservation is a critical issue playing an increasingly key role in preventing catastrophic failures in physical infrastructure as well as easing the social adoption of new technology. Power networks, social networks, smartphones, manufacturing systems, and smart transportation are just but a few examples of cyberphysical applications in need of privacy-aware control and coordination strategies. In these scenarios, the ability of the networked system to optimize its operation, fuse information and filter noise, compute common estimates of unknown quantities, and agree on a common view of the world while protecting relevant sensitive information is critical. This paper has introduced the reader to the concept of differential privacy, starting from its definition in the context of preserving the privacy of individuals in large databases, discussing basic properties and mechanisms to ensure it, and illustrating its usefulness in achieving various control and networked tasks with privacy guarantees.

Many interesting questions and avenues for future research remain open. As an example, the accurate characterization of the optimal privacy-accuracy trade-off curve of differentially private mechanisms remains a challenging question with a direct impact on the design of optimized algorithms in a variety of applications. Another important question is how to ensure local privacy in the absence of a trusted central mediator. In the traditional setting of differential privacy, the database is centralized and can only be accessed by a trusted mediator. For network systems, however, the centralized setting does not apply in general, since the data may be kept by the users themselves and the mediator may not be trustworthy. In these situations, it is necessary to ensure privacy at the user level (i.e., local privacy) so that the data provided by users are no longer sensitive (but still useful) even before aggregation by the mediator. Moreover, for certain systems, it may happen that there is no central mediator at all, and computation on user data are fully decentralized through communications among neighbors in the network, as we illustrated in our discussion. The properties of the network itself, e.g., its structure and various parameters that define it such as edge weights and vertex degrees, might be the private information that agents seek to protect. Therefore, there is a need to investigate further how differential privacy should be applied in conjunction with decentralized computation in network systems. Promising solutions include using a relaxed notion privacy such as information-theoretic privacy, techniques from secure multiparty computation, and cryptography.

Another interesting direction of research is applying differential privacy to dynamic databases that are potentially prevalent in dynamical systems. As we learned from the sequential composition theorem, privacy guarantee weakens as more queries are made to the same database. However, if the database is not static and keeps changing over time, the sequential composition theorem will not apply, and one can expect a better privacy guarantee for dynamic databases. In the extreme case, if the database completely refreshes whenever a new query is made, the problem of deriving the privacy guarantee reduces to quantifying the privacy guarantee for a single query. One potential research topic is to model dynamic databases and quantify the relation between the privacy guarantee and the "rate of change" of the database.

Finally, certain barriers need to be overcome for broader adoption of these ideas in control systems. The formulations we have discussed in this tutorial, for example, rely on detailed knowledge of the objective functions and underlying dynamical models of the system for estimating sensitivity, which is essential for the privacy preservation, for example, through mechanisms like the Laplace mechanism. In reality, such models may not be available to the privacy architect and developing model-free approaches would become necessary. Also, quantifying privacy in terms of a single parameter has led to an elegant framework and mathematical guarantees, but translating these to applications where users may have different privacy attitudes for different streams of data remains a challenge. There is a need to further understand the appropriate scale of the privacy parameters for specific application domains.

Acknowledgments

The authors would like to thank the scientific input from joint collaboration with Z. Huang, M. Mohammady, E. Nozari, P. Tallapragada, U. Topcu, and Y. Wang in the material covered in this tutorial paper. This research was supported by NSF Award CNS-1329619 (JC), NSERC Grant RGPIN-435905-13 (JLN), and an NSA Science of Security Lablet grant (GD and SM).

REFERENCES

[1] R. H. Weber, "Internet of things - new security and privacy challenges," *Computer Law and Security Review*, vol. 26, pp. 23–30, 2010.

[2] President's Council of Advisors on Science and Technology, "Big data and privacy: A technological perspective," Report to the President, Executive Office of the President of the United States, Tech. Rep., May 2014.

[3] Electronic Privacy Information Center (epic). Online: http://epic.org/.

[4] A. Narayanan and V. Shmatikov, "How to break anonymity of the Netflix Prize dataset," 2006, arXiv:cs/0610105.

[5] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, December 1992.

[6] B. Hoh, T. Iwuchukwu, Q. Jacobson, M. Gruteser, A. Bayen, J.-C. Herrera, R. Herring, D. Work, M. Annavaram, and J. Ban, "Enhancing privacy and accuracy in probe vehicle based traffic monitoring via virtual trip lines," *IEEE Transactions on Mobile Computing*, vol. 11, no. 5, May 2012.

[7] W. Xin, J. Chang, S. Muthuswamy, and M. Talas, "Midtown in Motion: A new active traffic management methodology and its implementation in New York City," in *Transportation Research Board Annual Meeting*, 2013.

[8] G. Duncan and D. Lambert, "Disclosure-limited data dissemination," *Journal of the American Statistical Association*, vol. 81, no. 393, pp. 10–28, March 1986.

[9] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[10] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: privacy beyond k-anonymity and l-diversity," in *Proceedings of the 23rd IEEE International Conference on Data Engineering*, 2007.

[11] L. Sankar, S. R. Rajagopalan, and H. V. Poor, "A theory of privacy and utility in databases," in *Proceedings of the IEEE International Symposium on Information Theory*, June 2010.

[12] M. Xue, W. Wang, and S. Roy, "Security concepts for the dynamics of autonomous vehicle networks," *Automatica*, vol. 50, pp. 852–857, 2014.

[13] N. E. Manitara and C. N. Hadjicostis, "Privacy-preserving asymptotic average consensus," in *European Control Conference*, Zurich, Switzerland, 2013, pp. 760–765.

[14] Y. Mo and R. M. Murray, "Privacy preserving average consensus," in *IEEE Conf. on Decision and Control*, Los Angeles, CA, Dec. 2014, pp. 2154–2159.

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

[16] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP)*, Venice, Italy, July 2006, pp. 1–12.

[17] U. Erlingsson, "Learning statistics with privacy, aided by the flip of a coin," Google Security Blog, October 2014, https://security.googleblog.com/2014/10/learning-statistics-with-privacy-aided.html.

[18] A. Eland, "Tackling urban mobility with technology," Google Europe Blog, November 2015, https://europe.googleblog.com/2015/11/tackling-urban-mobility-with-technology.html.

[19] "Apple's 'differential privacy' is about collecting your data – but not *your* data," WIRED magazine, June 2016, https://www.wired.com/2016/06/apples-differential-privacy-collecting-data.

[20] Z. Huang, Y. Wang, S. Mitra, and G. E. Dullerud, "On the cost of differential privacy in distributed control systems," in *Proceedings of the 3rd International Conference on High Confidence Networked Systems*. ACM, 2014, pp. 105–114.

[21] Y. Wang, Z. Huang, S. Mitra, and G. E. Dullerud, "Entropy-minimizing mechanism for differential privacy of discrete-time linear feedback systems," in *IEEE Conf. on Decision and Control*, Los Angeles, CA, Dec. 2014, pp. 2130–2135.

[22] Z. Huang, S. Mitra, and G. Dullerud, "Differentially private iterative synchronous consensus," in *Proceedings of the 2012 ACM workshop on Privacy in the electronic society*, New York, NY, 2012, pp. 81–90.

[23] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private average consensus: obstructions, trade-offs, and optimal algorithm design," *Automatica*, 2016, submitted.

[24] V. Katewa, A. Chakrabortty, and V. Gupta, "Protecting privacy of topology in consensus networks," in *American Control Conference*. Chicago, IL: IEEE, July 2015, pp. 2476–2481.

[25] J. Le Ny and G. J. Pappas, "Differentially private filtering," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 341–354, February 2014.

[26] J. Le Ny and M. Mohammady, "Privacy-preserving filtering for event streams," submitted. Available at http://arxiv.org/abs/1407.5553.

[27] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed constrained optimization," *IEEE Transactions on Automatic Control*, 2016, to appear.

[28] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, Pilani, India, Jan. 2015.

[29] M. T. Hale and M. Egerstedt, "Differentially private cloud-based multi-agent optimization with constraints," in *American Control Conference*. Chicago, IL: IEEE, July 2015, pp. 1235–1240.

[30] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private distributed convex optimization via functional perturbation," *IEEE Transactions on Control of Network Systems*, 2017, to appear.

[31] L. Wasserman and S. Zhou, "A statistical framework for differential privacy," *Journal of the American Statistical Association*, vol. 105, no. 489, pp. 375–389, 2010.

[32] Q. Geng and P. Viswanath, "Optimal noise adding mechanisms for approximate differential privacy," *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 952–969, 2016.

[33] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2013.

[34] Q. Geng, P. Kairouz, S. Oh, and P. Viswanath, "The staircase mechanism in differential privacy," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 7, pp. 1176–1184, 2015.

[35] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *IEEE Symposium on Foundations of Computer Science*, 2007, pp. 94–103.

[36] J. Le Ny and G. J. Pappas, "Differentially private Kalman filtering," in *Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing*, October 2012.

[37] C. Li and G. Miklau, "An adaptive mechanism for accurate query answering under differential privacy," in *Proceedings of the Conference on Very Large Databases (VLDB)*, Istanbul, Turkey, 2012.

[38] T. Tanaka, K.-K. K. Kim, P. A. Parrilo, and S. K. Mitter, "Semidefinite programming approach to Gaussian sequential rate-distortion trade-offs," *IEEE Transactions on Automatic Control*, 2016, to appear. Available at http://arxiv.org/abs/1411.7632.

[39] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observations," in *Proceedings of the ACM Symposium on the Theory of Computing (STOC)*, Cambridge, MA, June 2010.

[40] T.-H. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *ACM Transactions on Information and System Security*, vol. 14, no. 3, pp. 26:1–26:24, November 2011.

[41] J. Bolot, N. Fawaz, S. Muthukrishnan, A. Nikolov, and N. Taft, "Private decayed sum estimation under continual observation," in *Proceedings of the 16th International Conference on Database Theory*, 2013.

[42] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, Prentice-Hall, Ed. Prentice Hall, 2000.

[43] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," Microsoft Research, Tech. Rep. MSR-TR-2009-165, 2009.

[44] Y. D. Li, Z. Zhang, M. Winslett, and Y. Yang, "Compressive mechanism: Utilizing sparse representation in differential privacy," in *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, October 2011.

[45] J. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2000.

[46] J. Salz, "Digital transmission over cross-coupled linear channels," *AT&T Technical Journal*, vol. 64, no. 6, pp. 1147–1159, July-August 1985.

[47] J. Yang and S. Roy, "On joint transmitter and receiver optimization for multiple-input-multiple-output (MIMO) transmission systems," *IEEE Journal on Communications*, vol. 42, no. 12, pp. 3221–3231, December 1994.

[48] J. Le Ny, "On differentially private filtering for event streams,"

in *Proceedings of the IEEE Conference on Decision and Control*, Florence, Italy, December 2013, pp. 3481–3486.

[49] W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," *Proceedings of the Royal Society of London Series A*, vol. 115, pp. 700–721, 1927.

[50] F. Brauer, P. van den Driessche, and J. Wu, Eds., *Mathematical Epidemiology*, ser. Lecture Notes in Mathematics. Berlin: Springer-Verlag, 2008, vol. 1945.

[51] A. B. Lawson and K. Kleinman, *Spatial and Syndromic Surveillance for Public Health*. Wiley, 2005.

[52] J. Le Ny, "Privacy-preserving nonlinear observer design using contraction analysis," in *Proceedings of the IEEE Conference on Decision and Control*, Osaka, Japan, July 2015, pp. 4499–4504. [Online]. Available: http://arxiv.org/abs/1507.02250

[53] A. Sarwate and K. Chaudhuri, "Signal processing and machine learning with differential privacy: theory, algorithms, and challenges," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 86–94, September 2013.

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

[55] A. Nedić, "Distributed optimization," in *Encyclopedia of Systems and Control*, J. Baillieul and T. Samad, Eds. New York: Springer, 2015.

[56] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[57] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM Journal on Control and Optimization*, vol. 20, no. 3, pp. 1157–1170, 2009.

[58] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2012.

[59] B. Gharesifard and J. Cortés, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781–786, 2014.

[60] C. Cai and A. R. Teel, "Results on input-to-state stability for hybrid systems," in *44th IEEE Conference on Decision and Control and European Control Conference*. Seville, Spain: IEEE, Dec. 2005, pp. 5403–5408.