Privacy-Preserving Distributed Processing
Metrics, Bounds and Algorithms

Li, Qiongxiu ; Gundersen, Jaron Skovsted ; Heusdens, Richard; Christensen, Mads Græsbøll

DOI
10.1109/TIFS.2021.3050064

Publication date
2021

Document Version
Accepted author manuscript

Published in
IEEE Transactions on Information Forensics and Security

Privacy-Preserving Distributed Processing: Metrics, Bounds and Algorithms

Qiongxiu Li, Jaron Skovsted Gundersen, Richard Heusdens and Mads Græsbøll Christensen

***Abstract*—Privacy-preserving distributed processing has recently attracted considerable attention. It aims to design solutions for conducting signal processing tasks over networks in a decentralized fashion without violating privacy. Many existing algorithms can be adopted to solve this problem such as differential privacy, secure multiparty computation, and the recently proposed distributed optimization based subspace perturbation algorithms. However, since each of them is derived from a different context and has different metrics and assumptions, it is hard to choose or design an appropriate algorithm in the context of distributed processing. In order to address this problem, we first propose general mutual information based information-theoretical metrics that are able to compare and relate these existing algorithms in terms of two key aspects: output utility and individual privacy. We consider two widely-used adversary models, the passive and eavesdropping adversary. Moreover, we derive a lower bound on individual privacy which helps to understand the nature of the problem and provides insights on which algorithm is preferred given different conditions. To validate the above claims, we investigate a concrete example and compare a number of state-of-the-art approaches in terms of the concerned aspects using not only theoretical analysis but also numerical validation. Finally, we discuss and provide principles for designing appropriate algorithms for different applications.**

***Index Terms*—Distributed processing, differential privacy, secure multiparty computation, subspace perturbation, information-theoretic, privacy-utility metric, consensus.**

I. Introduction

Big data is accompanied by big challenges. Currently, data are collected and simultaneously stored on various local devices, such as phones, tablets and wearable devices [1], [2]. In these cases, three critical challenges exist in processing such large amounts of data: (1) the emerging demand for distributed signal processing tools, as these devices are distributed in nature and often rely on wireless communication to form a network that allows devices to cooperate for solving a problem; (2) the requirement for both computational and communication efficient solutions, due to the fact that these devices are usually resource-constrained, for example in wireless sensor networks; and (3) privacy concerns, as sensors from these devices, such as GPS and cameras, usually contain sensitive personal information. Consequently, having efficient privacy-preserving distributed processing solutions, which are able to address the privacy concerns, is highly important and usually requires interdisciplinary research across fields such as distributed signal processing, information theory and cryptography.

There are two primary types of security models: (1) computational security, in which the adversary is assumed to be computationally bounded such that it cannot decrypt a secret efficiently (i.e., in polynomial time) and (2) information-theoretic security, in which the adversary is assumed to be computationally unbounded but does not have sufficient information for inferring the secret. In this paper we focus on information-theoretic security since it assumes a stronger adversary and is more efficient in terms of both communication and computational demands [3].

A. Related works

Many information-theoretic approaches have been proposed for addressing privacy issues in various distributed processing problems like distributed average consensus [4]–[16], distributed least squares [17], [18], distributed optimization [19]–[27] and distributed graph filtering [28]. These approaches can be broadly classified into three classes. The first two classes combine distributed signal processing with commonly used cryptographic tools, such as secure multiparty computation (SMPC) [29], [30], and privacy primitives, such as differential privacy (DP) [31], [32], respectively. The third class directly explores the potential of existing distributed signal processing tools for privacy preservation, such as distributed optimization based subspace perturbation (DOSP) [7], [18], [27]. Among these approaches, SMPC aims to securely compute a function over a number of parties' private data without revealing it. DP, on the other hand, is defined to add noise to ensure that the posterior guess relating to the private data is only slightly better (quantified by the parameter ϵ) than the prior guess. DOSP protects the private data by inserting noise in a specific subspace depending on the graph topology.

Even though all the above mentioned algorithms can in principle be applied in distributed processing, it is still very challenging to design an appropriate algorithm given a specific application at hand. For example, whether choosing one single algorithm is good enough or if we should combine them to have a hybrid approach. The main difficulty comes from the fact that the metrics of these approaches are different and are defined based on different motivations and contexts. There are cases where these approaches are mutually exclusive. For example, it has been shown that, in distributed

Q. Li and M. G. Christensen are with the Audio Analysis Lab, CREATE, Aalborg University, Rendsburggade 14, Aalborg, Denmark (emails: {qili,mgc}@create.aau.dk).

J. S. Gundersen is with the Department of Mathematical Sciences, Aalborg University, Skjernvej 4A, Aalborg, Denmark (e-mail: jaron@math.aau.dk).

R. Heusdens is with the Netherlands Defence Academy (NLDA), Het Nieuwe Diep 8, 1781 AC Den Helder, The Netherlands, and with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands (email: r.heusdens@{mindef.nl,tudelft.nl}).

average consensus applications, the exact average result and differential privacy cannot be achieved simultaneously [10]. This implies that a DOSP or a perfect SMPC protocol, which guarantees accurate results, can never be differentially private in distributed average consensus. Another issue is that the privacy defined by these approaches might not be the same as the individual privacy defined in the context of distributed processing. For example, a perfect SMPC protocol does not necessarily guarantee that no private information is revealed (see Section IV-A). In addition, a perfect DP based approach ($\epsilon = 0$) also does not guarantee that no private information is revealed if the private data are correlated [33] (see Section IV-B). Therefore, it is highly desired to have general metrics that are able to compare and relate these algorithms in a consistent fashion, so that appropriate privacy-preserving distributed algorithms can be designed based on their performance and underlying assumptions.

In addition to the above mentioned challenges in algorithm design, another challenge lies in how to analyze the algorithm performance in a distributed setting. Due to the fact that distributed processing algorithms are usually iterative, it is complex to analytically track the privacy analysis over the iterations.

B. Paper contributions

In this paper, we attempt to solve the above mentioned problems. The main contributions of this paper can be summarized as follows:

- To the best of our knowledge, this is the first paper proposing formal and general information-theoretic metrics for quantifying privacy-preserving distributed processing algorithms in terms of output utility and individual privacy. Additionally, we prove that existing well-known metrics in SMPC and DP can be considered special cases of the proposed metrics under certain assumptions/conditions. Moreover, by analyzing the lower bound on individual privacy which provides insights on the nature of a problem, we give suggestions and discuss principles on how to design appropriate algorithms.
- We demonstrate how to analyze, quantify, compare, and understand the nature of a number of existing privacy-preserving distributed processing algorithms including DP, SMPC and DOSP.

C. Outline and notation

This paper is organized as follows. Section II introduces fundamentals and states the problem to be solved. Section III introduces the proposed metrics. Section IV relates the well-known SMPC and DP to the proposed metrics. Sections V and VI describe a concrete example of distributed average consensus. The former section defines the problem and shows that traditional approaches leak privacy, while the latter section first presents a theoretical result for achieving privacy-preservation and then analyzes existing privacy-preserving distributed average consensus algorithms using the proposed metrics. Numerical validations are given in Section VII. Section VIII gives suggestions on algorithm design and Section IX concludes the paper.

The following notations are used in this paper. We will use lowercase letters (x) for scalars, lowercase boldface letters $(\boldsymbol{x})$ for vectors, uppercase boldface letters $(\boldsymbol{X})$ for matrices, overlined uppercase letters $(\bar{X})$ for subspaces, calligraphic letters $(\mathcal{X})$ for arbitrary sets and $|\cdot|$ for the cardinality of a set. Uppercase letters (X) denote random variables having realizations x. $\mathrm{span}\{\cdot\}$ and $\mathrm{null}\{\cdot\}$ denote the span and nullspace of their argument, respectively. $(\boldsymbol{X})^\top$ denotes the transpose of $\boldsymbol{X}$. x_i denotes the i-th entry of the vector $\boldsymbol{x}$ and $\boldsymbol{X}_{ij}$ denotes the (i,j)-th entry of the matrix $\boldsymbol{X}$. $\mathbf{0}$, $\mathbf{1}$ and $\boldsymbol{I}$ denote the vectors with all zeros and all ones, and the identity matrix of appropriate size, respectively.

II. Preliminaries

In this section, we first introduce the problem setup and the adversary models. After that we summarize the key aspects to be considered when evaluating an algorithm.

A. Privacy-preserving distributed processing over networks

A network can be modelled as a graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ where $\mathcal{N} = \{1, \ldots, n\}$ denotes the set of n nodes and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ denotes the set of m (undirected) edges. Note that node i and j can communicate with each other only if there is an edge between them, i.e., $(i,j) \in \mathcal{E}$. Let $\mathcal{N}_i = \{j \mid (i,j) \in \mathcal{E}\}$ denote the neighborhood of node i and $d_i = |\mathcal{N}_i|$, called the degree of node i. Assume each node i has private data s_i and let $\boldsymbol{s} = [s_1, \ldots, s_n]^\top$. Note that for simplicity, s_i is assumed to be scalar but the results can easily be generalized to arbitrary dimensions.

The goal of privacy-preserving distributed processing over a network is to compute a function

$$f : \mathbb{R}^n \mapsto \mathbb{R}^n, \boldsymbol{y} = f(\boldsymbol{s}), \tag{1}$$

in a distributed manner without revealing each node's private data s_i to other nodes, where y_i denotes the desired output of node i. By a distributed manner we mean that only data exchange between neighboring nodes is allowed.

B. Adversary models

Adversary models are used to evaluate the robustness of the system under different security attacks. In this paper, we consider two types of adversary models: the passive and eavesdropping model.

1) Passive adversary: The passive adversary model is a typical model to be addressed in distributed networks [34]. It works by colluding a number of nodes to infer the private data of the other nodes. These colluding nodes are referred to as corrupted nodes, and the others are called honest nodes. The corrupted nodes are assumed to follow the algorithm instructions (called the protocol) but will share information together to infer the private data of the honest nodes. We call an edge in the graph corrupted when there is one corrupted node at its ends, see Fig. 1 for a toy example. Hence, all the messages transmitted along such an edge will be known to the

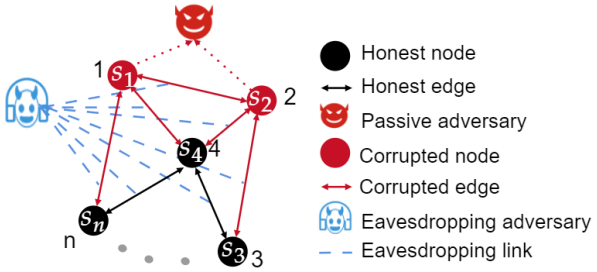


Fig. 1: System setup and adversary models.

passive adversary. In the following, we will denote $\mathcal{N}_c$ and $\mathcal{N}_h$ as the set of corrupted nodes and honest nodes, respectively. Additionally, we will denote $\mathcal{E}_c = \{(i,j) \in \mathcal{E} : (i,j) \notin \mathcal{N}_h \times \mathcal{N}_h\}$ as the set of corrupted edges. An algorithm is more robust if it can tolerate more corrupted nodes without revealing the private data of the honest nodes.

2) Eavesdropping adversary: The eavesdropping adversary, on the other hand, is assumed to listen to all communication channels, i.e., edges, between nodes with the purpose of inferring the private data. This model is relatively unexplored in the context of privacy-preserving distributed processing. The main reason is that many SMPC based approaches, such as those based on secret sharing [17], [19], [35], assume that all messages are transmitted through securely encrypted channels [36] so that the transmitted messages cannot be eavesdropped. However, channel encryption is computationally demanding for iterative approaches like the distributed processing algorithms considered here, since the channels are used many times before the algorithm converges. As a consequence, the cost for channel encryption is also an important factor to be considered when designing privacy-preserving algorithms.

Throughout this paper we will assume that these two adversaries cooperate. That is, they will share information together to increase the chance of inferring the private data of the honest nodes.

C. Key aspects for algorithm evaluation

We will evaluate the performance of privacy-preserving distributed processing algorithms in terms of the following two aspects: output utility and individual privacy.

1) Output utility: Let $\hat{\boldsymbol{y}} \in \mathbb{R}^n$ denote the estimated output of a privacy-preserving distributed processing algorithm. For each node i, the output utility should measure how close the estimate $\hat{y}_i$ is to its desired output y_i.

2) Individual privacy: Based on the definition of the adversary models, the corrupted nodes are willing to share their private data to the passive adversary. Therefore, privacy is only relevant for the honest nodes. The individual privacy of honest node $i \in \mathcal{N}_h$ should measure how much information regarding its private data s_i is revealed to the adversaries, both passive and eavesdropping, given all the information available to them.

In next section we will introduce the proposed metrics for quantifying the output utility and individual privacy.

III. Proposed metrics

In this section we will introduce the proposed metrics. We first motivate why we adopt mutual information for defining these metrics and then give details on how to quantify both the output utility and individual privacy stated above.

A. Motivation of using mutual information

To quantify the privacy for information-theoretic approaches, a natural language is to use information theory. For an overview of information-theoretic metrics the reader is referred to [37]. In the context of privacy-preserving distributed processing, two types of metrics are widely adopted: mutual information and ϵ-DP (their definitions will be given later in Section III-B and IV-B, respectively). The reasons for choosing mutual information over ϵ-DP are:

(1) ϵ-DP is very difficult to realize in practice as it is a worst-case metric that provides strong privacy assurance in any situation, e.g., for all prior distributions of the private data [38]–[40]. Mutual information is easier to implement in practice as it can be seen as a relaxed version of ϵ-DP [41].

(2) The privacy measured by ϵ-DP only reflects the privacy in the worst-case scenario which can be very far from the typical privacy of the average users; mutual information, on the other hand, is more preferred in quantifying the privacy of the average users [42].

(3) ϵ-DP has problems in working with correlated data [33].

To quantify the output utility, we also adopt mutual information as the metric because it has been widely used in the literature [43], [44].

B. Definition of mutual information

Let X denote a continuous random variable with probability density function $f_X(x)$ and differential entropy $h(X) = -\int f_X(x) \log f_X(x) dx$, assuming it exists. Given a random variable Y, the conditional entropy $h(X|Y)$ quantifies how much uncertainty is remained in X after knowing Y. The mutual information $I(X;Y)$ [45] measures the dependence between X and Y. It quantifies how much information can be learned about X after knowing Y, or vice versa, which is given by[1]

$$I(X;Y) = h(X) - h(X|Y). \tag{2}$$

C. Output utility u_i

We quantify the output utility as:

$$\forall i \in \mathcal{N} : u_i = I(Y_i; \hat{Y}_i). \tag{3}$$

Hence $0 \leq u_i \leq I(Y_i;Y_i)$ where $u_i = I(Y_i;Y_i)$ implies prefect output utility.

[1]For the case of discrete random variables, the condition is given in terms of the Shannon entropy $H(\cdot)$

D. Individual privacy ρ_i

Let $\mathcal{V}$ denote the set of random variables containing all the information collected by the adversaries throughout the whole algorithm. The individual privacy of honest node i quantifies the amount of information about the private data s_i learned by the adversaries, which we define as

$$\forall i \in \mathcal{N}_h : \rho_i = I(S_i, \mathcal{V}), \tag{4}$$

and we conclude that $0 \leq \rho_i \leq I(S_i; S_i)$. The smaller ρ_i, the more private the data is. Given the definition of the adversary models, we conclude that the adversaries always have knowledge of the private data $\{s_j\}_{j\in\mathcal{N}_c}$ and estimated outputs $\{\hat{y}_j\}_{j\in\mathcal{N}_c}$, regardless of the algorithm used. Therefore, we conclude that $\{S_j, \hat{Y}_j\}_{j\in\mathcal{N}_c} \subseteq \mathcal{V}$ which give rise to the following lower bound.

1) lower bound on individual privacy: The individual privacy ρ_i is lower bounded by

$$\rho_{i,\min} = I(S_i; \{S_j, \hat{Y}_j\}_{j\in\mathcal{N}_c}). \tag{5}$$

Hence, we have $\rho_{i,\min} \leq \rho_i \leq I(S_i; S_i)$.

There are two more parameters to consider regarding the individual privacy, namely the maximum number of corrupted nodes, giving information about the robustness of the algorithm, and the cost for channel encryption.

2) Maximum number of corrupted nodes under a passive adversary: The maximum number of corrupted nodes allowed in the network under a passive adversary will be denoted by $k_i \in \{0, \ldots, n-1\}$. That is, the algorithm is guaranteed to achieve individual privacy ρ_i for honest node i if there are at most k_i corrupted nodes in the network.

3) Cost for channel encryption under an eavesdropping adversary: Let $\mathcal{T} = \{0, \ldots, T\}$, where T is the maximum number of iterations. The cost $c_i \in \mathcal{T}$ indicates how many iterations require channel encryption to guarantee individual privacy ρ_i.

We propose a new definition of perfect individual privacy in the context of distributed processing. Intuitively, perfect individual privacy means $\rho_i = 0$. However, due to the fact that in many cases the lower bound $\rho_{i,\min} > 0$, it is in general impossible to achieve zero individual privacy. In addition, we assume $\rho_{i,\min} \neq I(S_i; S_i)$, otherwise there is no privacy at all. We have the following definition of perfect individual privacy.

Definition 1. *(Perfect individual privacy in the context of privacy-preserving distributed processing.) Given $\rho_{i,\min}$, $0 \leq \rho_{i,\min} < I(S_i; S_i)$, a privacy-preserving algorithm achieves perfect individual privacy if and only if $\rho_i = \rho_{i,\min}$.*

IV. Linking the proposed metrics to SMPC and DP

In this section, we will show that the well-known SMPC and DP can be considered special cases of the proposed metrics based on different setups or assumptions.

A. Secure multiparty computation

An important concept in SMPC is the definition of an ideal world, in which a trusted third party (TTP) is assumed to be available. A TTP first collects all private data from the nodes and computes the output $\boldsymbol{y} = f(\boldsymbol{s})$ after which the outputs y_i are transmitted to each and every node. This scenario is considered secure since a TTP is assumed to be non-corrupted. However, there is a distinction between security and privacy. In the ideal scenario, each node obtains its desired output y_i directly from the TTP. As a consequence, the set of random variables containing the information collected by the adversaries is given by $\mathcal{V} = \{S_j, Y_j\}_{j\in\mathcal{N}_c}$. Therefore, the individual privacy in the ideal world is given by

$$\forall i \in \mathcal{N}_h : \rho_{i,\text{ideal}} = I(S_i; \{S_j, Y_j\}_{j\in\mathcal{N}_c}). \tag{6}$$

Apparently, $\rho_{i,\text{ideal}}$ is not necessarily zero and it depends on several factors such as the output function and whether the private data are correlated or not.

The motivation for using SMPC comes from the fact that in practice a third party might not be available or trustworthy. The goal of SMPC is thus to design a protocol that can replace a TTP, i.e., simulates an ideal world. To do so, SMPC has to exchange information between nodes in the network and could, therefore, reveal some information about the private data. Let $\rho_{i,\text{smpc}}$ denote the individual privacy when using SMPC. An SMPC protocol is considered to be perfect when (1) it achieves perfect output utility and (2) the adversaries do not learn more about each honest node's private data than what will be revealed in an ideal world. That is, SMPC is perfect if

$$\begin{aligned} &\forall i \in \mathcal{N} : u_i = I(Y_i; Y_i), \\ &\forall i \in \mathcal{N}_h : \rho_{i,\text{smpc}} = \rho_{i,\text{ideal}}. \end{aligned} \tag{7}$$

As mentioned before, there is a distinction between security and privacy. As an example in which an SMPC protocol is perfect according to (7) but reveals maximum individual privacy, i.e., $\rho_{i,\text{smpc}} = I(S_i; S_i)$, consider the situation in which $\boldsymbol{y}$ is a permuted version of the private data $\boldsymbol{s}$. That is, $y_i = s_{i-1 \bmod n}$. Assume that node $i+1$ is corrupted. Using (6) we conclude that $\rho_{i,\text{ideal}} = I(S_i; \{S_{i+1}, Y_{i+1} = S_i\} = I(S_i; S_i)$. As $\rho_{i,\text{ideal}}$ is already maximum, any SMPC protocol giving perfect output utility will be considered perfect as $\rho_{i,\text{smpc}} = I(S_i; S_i) = \rho_{i,\text{ideal}}$. Hence, (7) is satisfied but there is no privacy at all.

We remark that $\rho_{i,\text{smpc}}$ and $\rho_{i,\text{ideal}}$ in SMPC correspond to the individual privacy ρ_i and its lower bound $\rho_{i,\min}$ under the condition of achieving full output utility in the proposed metrics, respectively. In the above example, in order to achieve meaningful individual privacy $\rho_i < I(S_i; S_i)$, we have to compromise the output utility to decrease the lower bound $\rho_{i,\min}$. That is, perfect output utility and individual privacy are not achievable simultaneously in this example.

B. Differential privacy

DP assumes an extreme scenario in which all nodes in the network are corrupted ($k_i = n-1$) except for node i [31], [32]. Let $\boldsymbol{s}^{-i} \in \mathbb{R}^{n-1}$ be a so-called adjacent vector of $\boldsymbol{s}$, obtained by excluding the private data s_i from $\boldsymbol{s}$. Denote Ω_i as the range of s_i. Let $\hat{F}$ be a randomized algorithm that protects the privacy of its input and $\mathcal{Y}$ denotes its output range. Given

$\epsilon \geq 0$, algorithm $\hat{F}$ achieves ϵ-DP if for any pair of adjacent vectors $\boldsymbol{s}$ and $\boldsymbol{s}^{-i}$, and for all sets $\mathcal{Y}_s \subseteq \mathcal{Y}$, we have

$$\forall s_i \in \Omega_i : \quad \frac{P(\hat{F}(\boldsymbol{s}) \in \mathcal{Y}_s)}{P(\hat{F}(\boldsymbol{s}^{-i}) \in \mathcal{Y}_s)} \leq e^{\epsilon}. \tag{8}$$

It has been shown [41, Theorem 1] that by relaxing the right-hand side of (8) to an expected value rather than a statement about all $s_i \in \Omega_i$, (8) is related to the Kullback-Leibler divergence and can be further relaxed to the following conditional mutual information (also called mutual information differential privacy):

$$I(S_i; Y | \{S_j\}_{j \in \mathcal{N} \setminus \{i\}}) \leq \epsilon. \tag{9}$$

The upper bound ϵ in (9) can be interpreted as the difference of the posterior and prior individual privacy. The prior individual privacy, in which the adversaries have the knowledge of $\boldsymbol{s}^{-i}$ and the related output $y' = \hat{F}(\boldsymbol{s}^{-i})$, can be quantified as

$$\begin{aligned} \rho_{i,\text{prior}} &= I(S_i; \{S_j\}_{j \in \mathcal{N} \setminus \{i\}}, Y') \\ &= I(S_i; \{S_j\}_{j \in \mathcal{N} \setminus \{i\}}), \end{aligned} \tag{10}$$

where the last equality holds because Y' is redundant information as $\{S_j\}_{j \in \mathcal{N} \setminus \{i\}}$ can determine Y'. The posterior individual privacy on the other hand, where the adversaries have the knowledge of the algorithm output $y = \hat{F}(\boldsymbol{s})$, is given by

$$\rho_{i,\text{post}} = I(S_i; \{S_j\}_{j \in \mathcal{N} \setminus \{i\}}, Y). \tag{11}$$

Based on the definition of conditional mutual information, we can rewrite (9) as

$$\begin{aligned} \epsilon &\geq I(S_i; \{S_j\}_{j \in \mathcal{N} \setminus \{i\}}, Y) - I(S_i; \{S_j\}_{j \in \mathcal{N} \setminus \{i\}}) \\ &= \rho_{i,\text{post}} - \rho_{i,\text{prior}}, \end{aligned} \tag{12}$$

showing the interpretation mentioned above.

We can see that the above $\rho_{i,\text{post}}$ and $\rho_{i,\text{prior}}$ are related to the individual privacy ρ_i and its lower bound $\rho_{i,\min}$, respectively, in the context of distributed processing when we assume that there are $k_i = n - 1$ corrupted nodes. Again, similar to SMPC, $\epsilon = 0$ does not imply zero individual privacy but only means that no additional information is leaked.

C. Proposed metrics for SMPC and DP

We end this section by concluding that both the SMPC and DP metrics can be considered as special cases of the proposed metrics under certain assumptions/requirements. For example, a privacy-preserving distributed processing algorithm can be considered as a perfect SMPC protocol if $u_i = I(Y_i; Y_i)$ and $\rho_i = \rho_{i,\min}$, and as an ϵ-DP protocol if $u_i = I(Y_i; Y_i), \rho_i \leq \epsilon + \rho_{i,\min}$, and $k_i = n - 1$.

V. Example I: Distributed average consensus

To demonstrate the benefits using the proposed metrics, we use the distributed average consensus as a canonical example. The two main reasons for choosing this problem are that it has general applicability in many signal processing tasks, such as denoising [46] and interpolation [47], and that its privacy-preserving solutions have been widely investigated in the literature [4]–[16].

In this section, we first define the problem. After that, we introduce traditional distributed average consensus approaches and show that they are not privacy-preserving; maximum individual privacy is revealed as $\forall i \in \mathcal{N}_h : \rho_i = I(S_i; S_i)$.

A. Problem definition

The goal of the distributed average consensus algorithm is to compute the global average of all the private data over the network, i.e.,

$$\boldsymbol{y} = s_{\text{ave}} \mathbf{1}, \tag{13}$$

where $s_{\text{ave}} = n^{-1} \sum_{i \in \mathcal{N}} s_i$. Hence, we have that $\boldsymbol{y} = n^{-1} \mathbf{1}\mathbf{1}^\top \boldsymbol{s}$. As the nodes in the network can only communicate with the neighboring nodes, the solution is obtained iteratively. Many distributed average consensus algorithms have been proposed to achieve this goal. Below, we introduce two types of approaches that serve as baselines for the coming sections.

Before describing the details, we will make the following assumptions.

Assumption 1. *The private data are statistically independent, i.e.,* $\forall i, j \in \mathcal{N}, i \neq j : I(S_i; S_j) = 0$.

Assumption 2. *The passive adversary has knowledge of the number of nodes* n *in the network and the degree* d_i *of all nodes.*

Let $\mathcal{N}_{i,c} = \mathcal{N}_i \cap \mathcal{N}_c$ and $\mathcal{N}_{i,h} = \mathcal{N}_i \cap \mathcal{N}_h$ denote the set of corrupted and honest neighbors of node i, respectively. In order to consider the worst-case scenario in which all information transmitted by honest nodes is known to the passive adversary, we have the following additional assumption.

Assumption 3. *Every honest node has a non-empty corrupted neighborhood, i.e.,* $\forall i \in \mathcal{N}_h : \mathcal{N}_{i,c} \neq \emptyset$.

B. Distributed linear iteration approaches

Distributed average consensus can be obtained by applying, at every iteration $t \in \mathcal{T}$ a linear transformation $\boldsymbol{W} \in \mathcal{W}$ where

$$\mathcal{W} = \left\{ \boldsymbol{W} \in \mathbb{R}^{n \times n} \,|\, \boldsymbol{W}_{ij} = 0 \text{ if } (i,j) \notin \mathcal{E} \text{ and } i \neq j \right\}, \tag{14}$$

such that the state vector $\boldsymbol{x}$ is updated as

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{W} \boldsymbol{x}^{(t)}, \tag{15}$$

and it is initialized with the private data, i.e.,

$$\boldsymbol{x}^{(0)} = \boldsymbol{s}. \tag{16}$$

The structure of $\boldsymbol{W}$ reflects the connectivity of the network[2]. In order to correctly compute the average, that is, $\boldsymbol{x}^{(t)} \to \boldsymbol{y} = n^{-1} \mathbf{1}\mathbf{1}^\top \boldsymbol{s}$ as $t \to \infty$, necessary and sufficient conditions for $\boldsymbol{W}$ are given by (i) $\mathbf{1}^\top \boldsymbol{W} = \mathbf{1}^\top$, (ii) $\boldsymbol{W}\mathbf{1} = \mathbf{1}$,

[2]For simplicity, we assume that $\boldsymbol{W}$ is constant for every iteration, which corresponds to a synchronous implementation of the algorithm. In the case of an asynchronous implementation, the transformation depends on which node will update. The results shown here are easily generalized to asynchronous systems by working with expected values.

(iii) $\alpha\left(\boldsymbol{W}-\frac{\mathbf{1}\mathbf{1}^\top}{n}\right)<1$, where $\alpha(\cdot)$ denotes the spectral radius [48].

Individual privacy: By inspecting (15), we can see that each node i needs to send its state values $x_i^{(t)}$ to all of its neighbours for updating $\{x_j^{(t+1)}\}_{j\in\mathcal{N}_i}$. Hence, we have $X_i^{(0)}=S_i\in\mathcal{V}$ and we conclude that

$$\rho_i=I(S_i,\mathcal{V})\geq I(S_i,X_i^{(0)})=I(S_i,S_i). \tag{17}$$

The algorithm is not private in the sense that it reveals all private information.

C. Distributed optimization approaches

The average consensus problem can also be stated as a linear-constrained convex optimization problem given by

$$\begin{aligned}\min_{x_i}\quad & \sum_{i\in\mathcal{N}}\frac{1}{2}\|x_i-s_i\|_2^2\\ \text{s.t.}\quad & \forall(i,j)\in\mathcal{E}:\quad x_i=x_j.\end{aligned} \tag{18}$$

Many distributed optimizers have been proposed to solve the above problem, such as ADMM [49] and PDMM [50], [51]. Here, we provide an example using PDMM. The corresponding (extended) augmented Lagrangian function is given by:

$$\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{s}\|_2^2+(\boldsymbol{P}\boldsymbol{\lambda}^{(t)})^T\boldsymbol{C}\boldsymbol{x}+\frac{c}{2}\|\boldsymbol{C}\boldsymbol{x}+\boldsymbol{P}\boldsymbol{C}\boldsymbol{x}^{(t)}\|_2^2, \tag{19}$$

and the updating equations are

$$\boldsymbol{x}^{(t+1)}=\left(\boldsymbol{I}+c\boldsymbol{C}^\top\boldsymbol{C}\right)^{-1}\left(\boldsymbol{s}-c\boldsymbol{C}^\top\boldsymbol{P}\boldsymbol{C}\boldsymbol{x}^{(t)}-\boldsymbol{C}^\top\boldsymbol{P}\boldsymbol{\lambda}^{(t)}\right), \tag{20}$$

$$\boldsymbol{\lambda}^{(t+1)}=\boldsymbol{P}\boldsymbol{\lambda}^{(t)}+c(\boldsymbol{C}\boldsymbol{x}^{(t+1)}+\boldsymbol{P}\boldsymbol{C}\boldsymbol{x}^{(t)}), \tag{21}$$

where $c>0$ is a constant for controlling the convergence rate and $\boldsymbol{\lambda}\in\mathbb{R}^{2m}$ is a dual variable. Let the subscript $i|j$ be a directed identifier that denotes the directed edge from node i to j. We first denote $\boldsymbol{B}\in\mathbb{R}^{m\times n}$ as the graph incidence matrix defined as $\boldsymbol{B}_{li}=1$, $\boldsymbol{B}_{lj}=-1$ if and only if $(i,j)\in\mathcal{E}$ and $i<j$. Denote $e_l=(i,j)\in\mathcal{E}$, where $l\in\{1,\dots,m\}$, as the l-th edge. The dual variable $\boldsymbol{\lambda}$ is defined as $\lambda_l=\lambda_{i|j}$ and $\lambda_{l+m}=\lambda_{j|i}$. Hence, with PDMM, each edge is associated with two dual variables, $\lambda_{i|j}$ and $\lambda_{j|i}$. The matrix $\boldsymbol{C}\in\mathbb{R}^{2m\times n}$ is related to the graph incidence matrix and defined as $\boldsymbol{C}_{li}=\boldsymbol{B}_{i|j}=1$ and $\boldsymbol{C}_{(l+m)j}=\boldsymbol{B}_{j|i}=-1$ if and only if $i<j$. Of note, $\boldsymbol{P}\in\mathbb{R}^{2m\times 2m}$ denotes a symmetric permutation matrix exchanging the first m with the last m rows. Thus, $\forall(i,j)\in\mathcal{E}:\lambda_{j|i}=(\boldsymbol{P}\boldsymbol{\lambda})_{i|j}$. and $\boldsymbol{C}+\boldsymbol{P}\boldsymbol{C}=[\boldsymbol{B}^\top\,\boldsymbol{B}^\top]^\top$.

The local updating functions for each node become

$$x_i^{(t+1)}=\frac{s_i+\sum_{j\in\mathcal{N}_i}\left(cx_j^{(t)}-\boldsymbol{B}_{i|j}\lambda_{j|i}^{(t)}\right)}{1+cd_i}, \tag{22}$$

$$\lambda_{i|j}^{(t+1)}=\lambda_{j|i}^{(t)}+c\boldsymbol{B}_{i|j}\left(x_i^{(t+1)}-x_j^{(t)}\right). \tag{23}$$

It has been shown that $\boldsymbol{x}^{(t)}$ converges geometrically (linearly on a logarithmic scale) to the global optimum $\boldsymbol{x}^*=s_{\text{ave}}\mathbf{1}$, given arbitrary initialization of both $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ [50].

Individual privacy: Note that traditional distributed optimization algorithms generally initialize both $\boldsymbol{x}^{(0)}$ and $\boldsymbol{\lambda}^{(0)}$ with all zeros as it gives the smallest initial error resulting in the smallest number of iterations to converge. As a consequence, by inspecting (22) we have

$$x_i^{(1)}=\frac{s_i}{1+cd_i}. \tag{24}$$

As the constant c is globally known to all nodes and the degree d_i is known to the adversaries based on Assumption 2, the private data s_i can be reconstructed by the adversaries from $x_i^{(1)}$. Since $X_i^{(1)}\in\mathcal{V}$ we conclude that

$$\rho_i=I(S_i,\mathcal{V})\geq I(S_i,X_i^{(1)})=I(S_i,S_i). \tag{25}$$

Based on (17) and (25), we conclude that traditional distributed average consensus algorithms, including distributed linear iteration and distributed optimization algorithms, are not privacy-preserving at all; they reveal all private data.

VI. Example II: Privacy-preserving distributed average consensus

From the previous section, we can see that the reason why the traditional distributed average consensus algorithms are not privacy-preserving is because the private data, either itself or a scaled version, is directly sent to the neighboring nodes during the data exchange step. As a consequence, one way to protect privacy is to not exchange the private data directly, but to first insert noise to obtain an obfuscated version of it and then exchange the obfuscated data with the neighboring nodes. In what follows, we will first present an information-theoretic result regarding noise insertion to achieve privacy-preservation. After that, we will introduce existing privacy-preserving distributed average consensus approaches and quantify their performances using the proposed metrics.

A. Noise insertion for privacy preservation

Proposition 1. *(Arbitrary small information loss can be achieved through noise insertion.) Let private data s and inserted noise r denote realizations of independent random variables S and R with variance $\sigma_S^2,\sigma_R^2<\infty$, respectively. Let $Z=S+R$. Given arbitrary small $\delta>0$, there exists $\beta>0$ such that for $\sigma_R^2\geq\beta$*

$$I(S;Z)\leq\delta. \tag{26}$$

In the case of Gaussian distributed noise, we have

$$\beta=\frac{\sigma_S^2}{2^{2\delta}-1}. \tag{27}$$

Proof. See Appendix A. □

Proposition 1 shows that the mutual information $I(S;Z)$, where Z is a noisy version of S obtained by adding independent noise, can be made arbitrarily small by making the noise variance sufficiently large.

Based on the design of the noise insertion process, we will classify existing approaches into two classes: zero-sum noise insertion and subspace noise insertion. We first introduce the former case.

The main idea of zero-sum noise insertion comes from the nature of the distributed average consensus. Let r_i denote the noise added by node i to its private data s_i. The estimated output is then given by

$$\hat{y}_i = \frac{1}{n}\sum_{j\in\mathcal{N}}(s_j + r_j) = s_{\rm ave} + \frac{1}{n}\sum_{j\in\mathcal{N}} r_j. \tag{28}$$

Clearly, if the sum of all inserted noise is zero, perfect output utility will be achieved as $\hat{y}_i = s_{\rm ave} = y_i$ in that case. Next we will proceed to introduce two different approaches, including DP and SMPC, which aim to insert zero-sum noise in a distributed manner.

B. Statistical zero-sum noise insertion using DP

DP-based approaches [8]–[10] mostly apply zero-mean noise insertion to achieve zero-sum in a statistical sense. That is, according to the law of the large numbers, the average of a large number of noise realizations should be close to the expected value, which is zero in this case, and will tend to become closer to the expected value as more realizations are involved. As a consequence, these algorithms only obtain asymptotically perfect output utility as $n \to \infty$. Variants exist in designing the noise insertion process, but here we will focus on one simple example to illustrate the main idea, which was proposed in [8] and [10]. Each node i initializes its state value by adding zero-mean noise r_i to its private data. That is, the state value initialization (16) becomes

$$\forall i \in \mathcal{N} : x_i^{(0)} = s_i + r_i, \tag{29}$$

and then arbitrary distributed average consensus algorithms (e.g., linear iterations [48] or distributed optimization [49]–[51]) can be adopted to compute the average.

1) Output utility analysis: Assume that all inserted noise are realizations of independent and identically distributed random variables with zero-mean and variance σ^2. Denote $r_{\rm tot} = \sum_{i\in\mathcal{N}} r_i$ and $r_{\rm ave} = r_{\rm tot}/n$ as the sum of all inserted noise realizations and its average, respectively. As a consequence, $R_{\rm tot}$ and $R_{\rm ave}$ are also zero-mean, and their variances are $n\sigma^2$ and σ^2/n, respectively. Based on (28) the output utility of node i is

$$\forall i \in \mathcal{N} :\ u_i = I(Y_i; Y_i + R_{\rm ave}). \tag{30}$$

Indeed, as mention before, we obtain perfect output utility only when $n \to \infty$ since $\lim_{n\to\infty} R_{\rm ave} = 0$.

2) Individual privacy analysis: DP based approaches do not require any channel encryption and assume $n-1$ corrupted nodes, i.e., $\mathcal{N}_c = \mathcal{N}\backslash\{i\}$. Collecting all state random variables $X_i^{(t)}$ in the vector $X^{(t)} = [X_1^{(t)}, \ldots, X_n^{(t)}]^\top$, we conclude that all information seen by the adversaries throughout the algorithm is

$$\begin{aligned}\mathcal{V} &= \{\hat{Y}_j, S_j, R_j, X^{(t)}\}_{j\in\mathcal{N}_c, t\in\mathcal{T}} \\ &= \{S_j, R_j, X^{(t)}\}_{j\in\mathcal{N}_c, t\in\mathcal{T}},\end{aligned} \tag{31}$$

since $\hat{Y}_j = X_j^{(T)}$. Note that we assume that all messages $\{X^{(t)}\}_{t\in\mathcal{T}}$ transmitted through the communication channels can be eavesdropped and are thus known to the adversaries. We see that computing $I(S_i;\mathcal{V})$ requires to analyze the information flow over the whole iterative process. This imposes challenges as keeping track of information loss throughout all iterations is difficult. We can, however, simplify the privacy analysis through the following result.

Lemma 1. *(Information release of successive iterations.)*

$$I(S_i; X^{(0)}, \ldots, X^{(T)}) = I(S_i; X^{(0)}).$$

Proof. The sequence $S_i \to X^{(0)} \to X^{(t)}$ forms a Markov chain in that order. As a consequence, by the chain rule of mutual information, we have

$$\begin{aligned}I(S_i; X^{(0)}, \ldots, X^{(T)}) &= \sum_{t=0}^{T} I(S_i; X^{(t)} | X^{(t-1)}, \ldots, X^{(0)}) \\ &= I(S_i; X^{(0)}).\end{aligned}$$

□

Lemma 1 states that it is sufficient to analyze the privacy leakage of the initial state vector only as successive iterations will not reveal additional information about the private data. Given this result, we conclude that

$$\begin{aligned}I(S_i;\mathcal{V}) &= I(S_i; \{S_j, R_j, X^{(0)}\}_{j\in\mathcal{N}_c}) \\ &\overset{\text{(a)}}{=} I(S_i; X_i^{(0)}) \\ &\quad + I(S_i; \{S_j, R_j, X_j^{(0)}\}_{j\in\mathcal{N}_c} | X_i^{(0)}) \\ &\overset{\text{(b)}}{=} I(S_i; X_i^{(0)}),\end{aligned} \tag{32}$$

where (a) follows from the chain rule of mutual information, and (b) holds as $\{S_j, R_j, X_j^{(0)}\}_{j\in\mathcal{N}_c}$ is independent of both S_i and $X_i^{(0)}$. The individual privacy thus becomes

$$\rho_i = I(S_i; X_i^{(0)}) = I(S_i; S_i + R_i). \tag{33}$$

Lower bound analysis. The lower bound on individual privacy is given by

$$\begin{aligned}\rho_{i,\min} &= I(S_i; \{\hat{Y}_j, S_j\}_{j\in\mathcal{N}_c}) \\ &\overset{\text{(a)}}{=} I(S_i; \sum_{j\in\mathcal{N}} S_j + R_{\rm tot}, \{S_j\}_{j\in\mathcal{N}_c}) \\ &= I(S_i; S_i + R_{\rm tot}, \{S_j\}_{j\in\mathcal{N}_c}) \\ &\overset{\text{(b)}}{=} I(S_i; S_i + R_{\rm tot}),\end{aligned} \tag{34}$$

where (a) follows from (28) and the fact that n is known to the adversaries (Assumption 2) and (b) from the fact that $\{S_j\}_{j\in\mathcal{N}_c}$ is independent of $S_i + R_{\rm tot}$. By inspection of (33) and (34) we conclude that for $n > 1$ we have $\rho_{i,\min} < \rho_i$, except for $r_i = 0$, so that DP does not achieve perfect individual privacy for the average consensus problem.

Maximum number of corrupted nodes and cost for channel encryption. Since $\mathcal{N}_c = \mathcal{N} \setminus \{i\}$, we have $k_i = |\mathcal{N}_c| = n-1$ being the maximum number of corrupted nodes. As no channel encryption is needed, we have $c_i = 0$.

Summarizing, with the proposed metrics, DP-based approaches achieve

$$\begin{aligned} u_i &= I(Y_i; Y_i + R_{\text{ave}}), \\ \rho_i &= I(S_i; S_i + R_i), \\ \rho_{i,\min} &= I(S_i; S_i + R_{\text{tot}}), \\ k_i &= n-1, \\ c_i &= 0. \end{aligned} \tag{35}$$

We have the following remark.

Remark 1. *(In the distributed average consensus, DP always has a trade-off between the output utility and individual privacy.) As both output utility* (30) *and individual privacy* (33) *are dependent on the inserted noise, we conclude, using Proposition 1, that*

$$\sigma^2 \to \infty \quad \Rightarrow \quad u_i = 0, \rho_i = 0, \tag{36}$$

$$\sigma^2 = 0 \quad \Rightarrow \quad u_i = I(Y_i; Y_i),\ \rho_i = I(S_i; S_i). \tag{37}$$

Hence DP has a trade-off between privacy and utility. Of note, the conclusion that DP based approaches cannot achieve perfect full utility has been shown before in [10]. Here, we provide a simpler proof in terms of mutual information.

C. Exact zero-sum noise insertion using SMPC

Unlike DP based approaches, which have a privacy-utility trade-off, SMPC based approaches can obtain full utility without compromising privacy. However, there is no "free lunch"; the price to be paid is that the robustness over $n-1$ corrupted nodes is no longer achievable. Existing SMPC based approaches [4]–[6] have applied additive secret sharing [30] to construct exact zero-sum noise through coordinated noise insertion. To do so, at the initialization phase, each node i first sends each neighbor $j \in \mathcal{N}_i$ a random number r_i^j and receives a random number r_j^i from each of its neighbors. After that node i constructs its noise realization as

$$r_i = \sum_{j \in \mathcal{N}_i} r_{i|j}, \tag{38}$$

where

$$r_{i|j} = r_j^i - r_i^j. \tag{39}$$

Of note, all the random numbers $\{r_i^j\}_{(i,j)\in\mathcal{E}}$ are independent of each other. After constructing the noise realizations, similar as DP based approaches, each node initializes its state value using (29) after which an arbitrary distributed average consensus algorithm can be used.

1) Output utility analysis: In SMPC the noise is constructed such that it sums to zero:

$$\sum_{i\in\mathcal{N}} r_i = \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}_i} r_{i|j} = \sum_{(i,j)\in\mathcal{E}} \big(r_{i|j} + r_{j|i}\big) = 0, \tag{40}$$

as $r_{i|j} = -r_{j|i}$ by (39). Full utility is thus obtained as $\hat{y}_i = y_i$:

$$\forall i \in \mathcal{N}:\ u_i = I(Y_i; Y_i). \tag{41}$$

2) Individual privacy analysis: SMPC based approaches assume that the communication channels are not securely encrypted except for transmitting the random numbers $\{r_i^j\}_{(i,j)\in\mathcal{E}}$ (initialization phase). As a consequence, all information that the adversaries see throughout the algorithm is given by

$$\begin{aligned} \mathcal{V} &= \{\{Y_j, S_j\}_{j\in\mathcal{N}_c}, \{R_i^j\}_{(i,j)\in\mathcal{E}_c}, \{X^{(t)}\}_{t\in\mathcal{T}}\} \\ &= \{\{S_j\}_{j\in\mathcal{N}_c}, \{R_i^j\}_{(i,j)\in\mathcal{E}_c}, \{X^{(t)}\}_{t\in\mathcal{T}}\}, \end{aligned} \tag{42}$$

since $Y_j = X_j^{(T)}$ and $X^{(t)}$ is known by Assumption 3.

Let $\mathcal{G}_h \subseteq \mathcal{G}$ denote the graph obtained by removing all corrupted nodes from $\mathcal{G}$. Moreover, let $\mathcal{G}_h = \cup_q \mathcal{C}_q$, where $\mathcal{C}_q$ is a component or connected subgraph of $\mathcal{G}_h$. The set of nodes in $\mathcal{C}_q$ is denoted by $\mathcal{N}_{h_q}$ so that $\mathcal{N}_h = \cup_q \mathcal{N}_{h_q}$. We have the following result which simplifies the individual privacy analysis.

Proposition 2.

$$\forall i \in \mathcal{N}_{h_q} : I(S_i; \mathcal{V}) = I(S_i; \{S_j + \sum_{k\in\mathcal{N}_{j,h}} R_{j|k}\}_{j\in\mathcal{N}_{h_q}}).$$

Proof. See Appendix B. □

We conclude from Proposition 2 that node i should have at least one honest neighbor. If not, S_i will be revealed as in that case $\mathcal{N}_{h_q} = \{i\}$ and $\mathcal{N}_{j,h} = \emptyset$. Moreover, the adversaries can compute the partial sum of the private data in each component $\mathcal{C}_q$ since

$$\sum_{j\in\mathcal{N}_{h_q}} (S_j + \sum_{k\in\mathcal{N}_{j,h}} R_{j|k}) = \sum_{j\in\mathcal{N}_{h_q}} S_j, \tag{43}$$

as $R_{j|k} = -R_{k|j}$. Since this partial sum can always be determined regardless of the amount of noise insertion, we have

$$\rho_i = I(S_i; \mathcal{V}) \geq I(S_i; \sum_{j\in\mathcal{N}_{h_q}} S_j). \tag{44}$$

We have equality in (44) when the partial sum (43) is all the adversaries know and no additional information can be inferred from the individual noisy observations. That is, we have equality if $\forall j \in \mathcal{N}_{h_q} : I(S_i; S_j + \sum_{k\in\mathcal{N}_{j,h}} R_{j|k}) = 0$, which can, by Proposition 1, be achieved asymptotically by adding independent noise to the private data. Therefore, the privacy level SMPC based approaches can achieve is given by

$$\rho_i = I(S_i; \sum_{j\in\mathcal{N}_{h_q}} S_j). \tag{45}$$

Lower bound analysis. With perfect output utility, the lower bound (5) becomes

$$\begin{aligned} \rho_{i,\min} &= I(S_i; \{Y_j, S_j\}_{j\in\mathcal{N}_c}) \\ &\overset{\text{(a)}}{=} I(S_i; \sum_{j\in\mathcal{N}} S_j, \{S_j\}_{j\in\mathcal{N}_c}) \\ &\overset{\text{(b)}}{=} I(S_i; \sum_{j\in\mathcal{N}_h} S_j, \{S_j\}_{j\in\mathcal{N}_c}) \\ &\overset{\text{(c)}}{=} I(S_i; \sum_{j\in\mathcal{N}_h} S_j), \end{aligned} \tag{46}$$

where (a) holds as $\forall j \in \mathcal{N} : y_j = n^{-1}\sum_{j\in\mathcal{N}} S_j$ and n is known by Assumption 2, (b) holds as $\sum_{j\in\mathcal{N}} S_j, \{S_j\}_{j\in\mathcal{N}_c}$ can be determined by $\sum_{j\in\mathcal{N}_h} S_j, \{S_j\}_{j\in\mathcal{N}_c}$ as $S_j, j \in \mathcal{N}_c$, are known to the adversaries, and (c) holds as $\{S_j\}_{j\in\mathcal{N}_c}$ is independent of both S_i and $\sum_{j\in\mathcal{N}_h} S_j$ by Assumption 1.

Maximum number of corrupted nodes and cost for channel encryption. As mentioned before, to guarantee the individual privacy $\rho_i < I(S_i; S_i)$, node i should have at least one honest neighbor, i.e., $\mathcal{N}_{i,h} \neq \emptyset$. The maximum number of corrupted nodes is therefore $k_i = d_i - 1$ and only depends on the degree d_i. For a fully connected graph we have $k_i = n - 2$. The amount of channel encryption is $c_i = 1$ as only the communication channels in the initialization phase need to be securely encrypted.

In conclusion, with the proposed metrics, SMPC based approaches achieve

$$\begin{aligned} &u_i = I(Y_i; Y_i),\\ &\rho_i = I(S_i; \textstyle\sum_{j\in\mathcal{N}_{h_q}} S_j),\\ &\rho_{i,\min} = I(S_i; \textstyle\sum_{j\in\mathcal{N}_h} S_j),\\ &k_i = d_i - 1,\\ &c_i = 1. \end{aligned} \tag{47}$$

We can see that u_i is independent of ρ_i. Hence, SMPC has no trade-off between privacy and utility in distributed average consensus. Hence, we have the following remark.

Remark 2. *(Conditions for achieving perfect individual privacy and perfect output utility using the SMPC based approaches in the distributed average consensus.) By inspection of* (45) *and* (46)*, if $\mathcal{G}_h$ is connected and $|\mathcal{N}_h| \geq 2$, we have only one component so that $\mathcal{N}_{h_q} = \mathcal{N}_h$ and thus $\rho_i = \rho_{i,\min}$; the algorithm achieves both perfect individual privacy (Definition 1) and perfect output utility.*

The main limitation of the above zero-sum noise insertion approaches is that it is hard to be generalized to problems other than distributed average consensus. To mitigate this problem, recently subspace noise-insertion based algorithms have been proposed which are able to solve more general (convex) optimization problems. In the next subsection we will introduce such an approach referred to as distributed optimization based subspace perturbation (DOSP).

D. Subspace noise insertion using DOSP

The DOSP algorithm [7], [27] differentiates from the DP and SMPC based approaches in the sense that it can ensure full output utility without compromising privacy and does not require coordinated noise insertion. In particular, DOSP does not introduce zero-sum noise but exploits the fact that the dual variables, if properly initialized, can obfuscate the private data throughout the algorithm. As a consequence, in order to analyze privacy, we have to consider the convergence behavior of the dual variable $\boldsymbol{\lambda}$.

To do so, consider two successive $\boldsymbol{\lambda}$-update in (21). We have

$$\boldsymbol{\lambda}^{(t+2)} = \boldsymbol{\lambda}^{(t)} + c(\boldsymbol{C}\boldsymbol{x}^{(t+2)} + 2\boldsymbol{P}\boldsymbol{C}\boldsymbol{x}^{(t+1)} + \boldsymbol{C}\boldsymbol{x}^{(t)}), \tag{48}$$

as $\boldsymbol{P}^2 = \boldsymbol{I}$. Let $\bar{H} = \mathrm{span}(\boldsymbol{C}) + \mathrm{span}(\boldsymbol{P}\boldsymbol{C})$ and $\bar{H}^{\perp} = \mathrm{null}(\boldsymbol{C}^{\top}) \cap \mathrm{null}((\boldsymbol{P}\boldsymbol{C})^{\top})$. We can see that every two $\boldsymbol{\lambda}$-updates affect only $\Pi_{\bar{H}}\boldsymbol{\lambda} \in \bar{H}$ where $\Pi_{\bar{H}}$ denotes the orthogonal projection onto $\bar{H}$. As shown in [27], the dual variable $\boldsymbol{\lambda}^{(t)}$ composites of two parts: a so-called convergent component $\Pi_{\bar{H}}\boldsymbol{\lambda}^{(t)}$ which will converge to a fixed point $\boldsymbol{\lambda}^*$, and a so-called non-convergent component $(\boldsymbol{I} - \Pi_{\bar{H}})\boldsymbol{\lambda}^{(t)} = \boldsymbol{P}^t(\boldsymbol{I} - \Pi_{\bar{H}})\boldsymbol{\lambda}^{(0)}$ which will not converge ($\boldsymbol{P}^t = \boldsymbol{P}$ for t odd and $\boldsymbol{P}^t = \boldsymbol{I}$ for t even) and only depends on the initialization $\boldsymbol{\lambda}^{(0)}$.

By inspecting (22), the noise for protecting s_i of honest node i is constructed as

$$\begin{aligned} \forall t \in \mathcal{T} : r_i^{(t)} &= \sum_{j\in\mathcal{N}_i} (\boldsymbol{B}_{i|j}\lambda_{j|i}^{(t)})\\ &= \sum_{j\in\mathcal{N}_{i,c}} (\boldsymbol{B}_{i|j}\lambda_{j|i}^{(t)}) + \sum_{j\in\mathcal{N}_{i,h}} (\boldsymbol{B}_{i|j}\lambda_{j|i}^{(t)}), \end{aligned} \tag{49}$$

where the dual variables $\{\lambda_{j|i}^{(t)}\}_{j\in\mathcal{N}_{i,c}}$ of the corrupted neighbors are known to the adversaries. As a consequence, only $\sum_{j\in\mathcal{N}_{i,h}}(\boldsymbol{B}_{i|j}\lambda_{j|i}^{(t)})$ is unknown to the adversaries. Separating the convergent and non-convergent component of $\boldsymbol{\lambda}^{(t)}$, we have

$$\begin{aligned} \sum_{j\in\mathcal{N}_{i,h}} (\boldsymbol{B}_{i|j}\lambda_{j|i}^{(t)}) &= \sum_{j\in\mathcal{N}_{i,h}} (\boldsymbol{B}_{i|j}(\Pi_{\bar{H}}\boldsymbol{\lambda}^{(t)})_{j|i})\\ &+ \sum_{j\in\mathcal{N}_{i,h}} \left(\boldsymbol{B}_{i|j}(\boldsymbol{P}^t(\boldsymbol{I} - \Pi_{\bar{H}})\boldsymbol{\lambda}^{(0)})_{j|i}\right). \end{aligned} \tag{50}$$

The main idea of subspace noise insertion is to exploit the non-convergent component of the dual variables as subspace noise for guaranteeing the privacy. That is, $\sum_{j\in\mathcal{N}_{i,h}} \left(\boldsymbol{B}_{i|j}(\boldsymbol{P}^t(\boldsymbol{I} - \Pi_{\bar{H}})\boldsymbol{\lambda}^{(0)})_{j|i}\right)$ protects the private data s_i from being revealed to others. By controlling $\boldsymbol{\lambda}^{(0)}$, the variance of the above subspace noise can be made arbitrarily large so that, by Proposition 1, we can achieve an arbitrarily small information loss.

Before discussing how to implement the subspace noise, we first state the following remark.

Remark 3. *(There is always a non-empty subspace for noise insertion as long as $m \geq n$.) Since $[\boldsymbol{C}\ \boldsymbol{P}\boldsymbol{C}] \in \mathbb{R}^{2m\times 2n}$ can be viewed as a new graph incidence matrix with $2n$ nodes and $2m$ edges [27], we thus have $\dim(\bar{H}) \leq 2n - 1$, and $\bar{H}^{\perp}$ is non-empty if $m \geq n$.*

In DOSP, each node only needs to randomly initialize its own dual variables $\{\lambda_{i|j}^{(0)}\}_{j\in\mathcal{N}_i}$ as in that case we have $(\boldsymbol{I} - \Pi_{\bar{H}})\boldsymbol{\lambda}^{(0)} \neq \boldsymbol{0}$ with probability 1 as long as $m \geq n$. Hence, DOSP does not require any coordination between nodes for noise construction. In the remainder of this section we will investigate the output utility and individual privacy of DOSP.

1) Output utility analysis: As mentioned before, $\boldsymbol{x}^{(t)}$ converges geometrically to the global optimum $\boldsymbol{x}^* = s_{\mathrm{ave}}\boldsymbol{1}$, given arbitrary initialization of both $\boldsymbol{x}$ and $\boldsymbol{\lambda}$, even though $\boldsymbol{\lambda}^{(t)}$ does not necessarily converge. Indeed, by inspection of (20), we see that the non-converging component of $\boldsymbol{\lambda}^{(t)}$ does not affect the $\boldsymbol{x}$-update since

$$\boldsymbol{C}^{\top}\boldsymbol{P}\,(\boldsymbol{I} - \Pi_{\bar{H}})\,\boldsymbol{\lambda}^{(t)} = (\boldsymbol{P}\boldsymbol{C})^{\top}\,(\boldsymbol{I} - \Pi_{\bar{H}})\,\boldsymbol{\lambda}^{(t)} = 0. \tag{51}$$

Hence, DOSP achieves perfect output utility.

2) Individual privacy analysis: Similar as the above SMPC based approaches, DOSP assumes that the communication channels are not securely encrypted except for the initialization phase where the initialized $\lambda_{i|j}^{(0)}$ are transmitted to all neighboring nodes. Therefore, the information collected by the adversaries throughout the course of the algorithm is given by

$$\begin{aligned}\mathcal{V} &= \{\{Y_j, S_j\}_{j\in\mathcal{N}_c}, \{\Lambda_{i|j}^{(0)}, X^{(t)}\}_{(i,j)\in\mathcal{E}_c, t\in\mathcal{T}}\} \\ &= \{\{S_j\}_{j\in\mathcal{N}_c}, \{\Lambda_{i|j}^{(0)}, X^{(t)}\}_{(i,j)\in\mathcal{E}_c, t\in\mathcal{T}}\}, \end{aligned} \tag{52}$$

since $Y_j = X_j^{(T)}$. Note that all the $\{\Lambda_{i|j}^{(t)}\}_{(i,j)\in\mathcal{E}_c, t>0}$ are not included here because they are not transmitted through the network, and they can be determined by $\{X^{(t)}\}_{t\in\mathcal{T}}$ and $\{\Lambda_{i|j}^{(0)}\}_{(i,j)\in\mathcal{E}_c}$ from (21). We have the following result which simplifies the privacy analysis of DOSP.

Proposition 3.

$$\begin{aligned} I(S_i;\mathcal{V}) =& I(S_i; \{S_j - \sum_{k\in\mathcal{N}_{j,h}} \boldsymbol{B}_{j|k}\Lambda_{k|j}^{(t)}\}_{j\in\mathcal{N}_h, t=0,1} \\ & |\{S_j\}_{j\in\mathcal{N}_c}, \{\Lambda_{i|j}^{(0)}\}_{(i,j)\in\mathcal{E}_c}). \end{aligned} \tag{53}$$

Proof. See Appendix C. □

We note that, similar to the SMPC based approach, the partial sum $\sum_{j\in\mathcal{N}_{h_q}} S_j$ can be computed by the adversaries. Indeed, the partial sum can be constructed as

$$\begin{aligned} \sum_{j\in\mathcal{N}_{h_q}} S_j =& \frac{1}{2}\Bigg(\sum_{t=0,1}\sum_{j\in\mathcal{N}_{h_q}} \Big(S_j - \sum_{k\in\mathcal{N}_{j,h}} \boldsymbol{B}_{j|k}\Lambda_{k|j}^{(t)}\Big) \\ & + \sum_{t=0,1}\sum_{j\in\mathcal{N}_{h_q}}\sum_{k\in\mathcal{N}_{j,h}} \boldsymbol{B}_{j|k}\Lambda_{k|j}^{(t)}\Bigg). \end{aligned} \tag{54}$$

The first term of the right-hand side of (54) is the addition of terms that are known by the adversaries, as shown by (53). Let $\mathcal{E}_{h_q} = \{(i,j)\in\mathcal{E} : (i,j)\in\mathcal{N}_{h_q}\times\mathcal{N}_{h_q}\}$ denote the set of all edges between the honest nodes in component $\mathcal{C}_q$. With this, the second term of (54) can be expressed as

$$\begin{aligned} &\sum_{t=0,1}\sum_{j\in\mathcal{N}_{h_q}}\sum_{k\in\mathcal{N}_{j,h}} \boldsymbol{B}_{j|k}\Lambda_{k|j}^{(t)} \\ &= \sum_{t=0,1}\sum_{(i,j)\in\mathcal{E}_{h_q}} \left(\boldsymbol{B}_{j|k}\Lambda_{k|j}^{(t)} + \boldsymbol{B}_{k|j}\Lambda_{j|k}^{(t)}\right) \\ &= \sum_{t=0,1}\sum_{(i,j)\in\mathcal{E}_{h_q}} \boldsymbol{B}_{j|k}\left(\Lambda_{k|j}^{(t)} - \Lambda_{j|k}^{(t)}\right) \\ &= \sum_{(i,j)\in\mathcal{E}_{h_q}} \boldsymbol{B}_{j|k}\left(\left(\Lambda_{k|j}^{(1)} - \Lambda_{j|k}^{(0)}\right) - \left(\Lambda_{j|k}^{(1)} - \Lambda_{k|j}^{(0)}\right)\right), \end{aligned}$$

which can be determined by the adversaries since, by inspection of (23), the difference $\Lambda_{i|j}^{(1)} - \Lambda_{j|i}^{(0)}$ only depends on $x_i^{(1)}$ and $x_j^{(0)}$, all of which are known by the adversaries (based on (52)).

As the partial sum can be computed, the analysis of DOSP follows along the same line as the one presented for SMPC and we conclude that the performance indicators for DOSP, as measured by the proposed metrics, are also given by (47). In addition, Remark 2 also holds for DOSP.

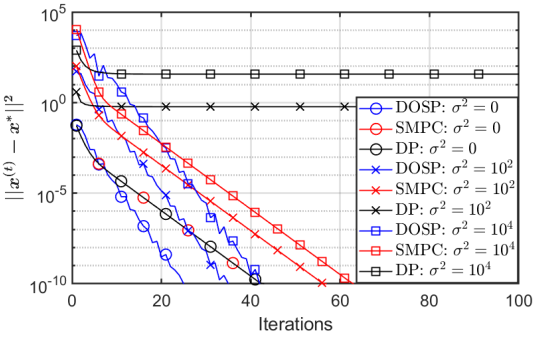


Fig. 2: Convergence behaviors of DOSP, SMPC and DP based approaches under three different amounts of noise insertion.

E. Comparisons of existing approaches

In Table I we summarize the performances of the discussed DP, SMPC and DOSP approaches for distributed average consensus. We can see that SMPC and DOSP achieve exactly the same performances, except the fact that SMPC requires coordination between nodes to construct zero-sum noise. Moreover, DP is robust against $n-1$ corrupted nodes and does not require channel encryption at all but suffers from a privacy-utility trade-off. On the other hand, SMPC and DOSP do not have privacy-utility trade-off but are only robust to $d_i - 1$ corrupted nodes and require channel encryption for the first iteration.

VII. Numerical Results

In this section we compare DP, SMPC and DOSP using computer simulations. The comparisons are conducted in terms of (1) convergence behavior and (2) utility/privacy behavior. Their metrics are given below.

- Convergence behavior: mean square error to measure the distance between the state value $\boldsymbol{x}^{(t)}$ and the desired average result $\boldsymbol{x}^* = s_{\text{ave}}\mathbf{1}$ for each iteration t, i.e., $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\|^2$.
- Privacy/utility behavior: normalized mutual information (NMI)[3] to measure the information-theoretical performances, i.e., $u_i/I(Y_i;Y_i)$ for the output utility, $\rho_i/I(S_i;S_i)$ for the individual privacy and $\rho_{i,\min}/I(S_i;S_i)$ for the lower bound on individual privacy.

We simulated a geometrical graph with $n = 10$ nodes, and set the radius as $r^2 = 2\frac{\log n}{n}$ to ensure a connected graph with high probability [52]. For simplicity, all private data have a zero-mean unit variance Gaussian distribution, and all the noise used in the DP, SMPC and DOSP approaches follow a zero-mean Gaussian distribution with variance σ^2.

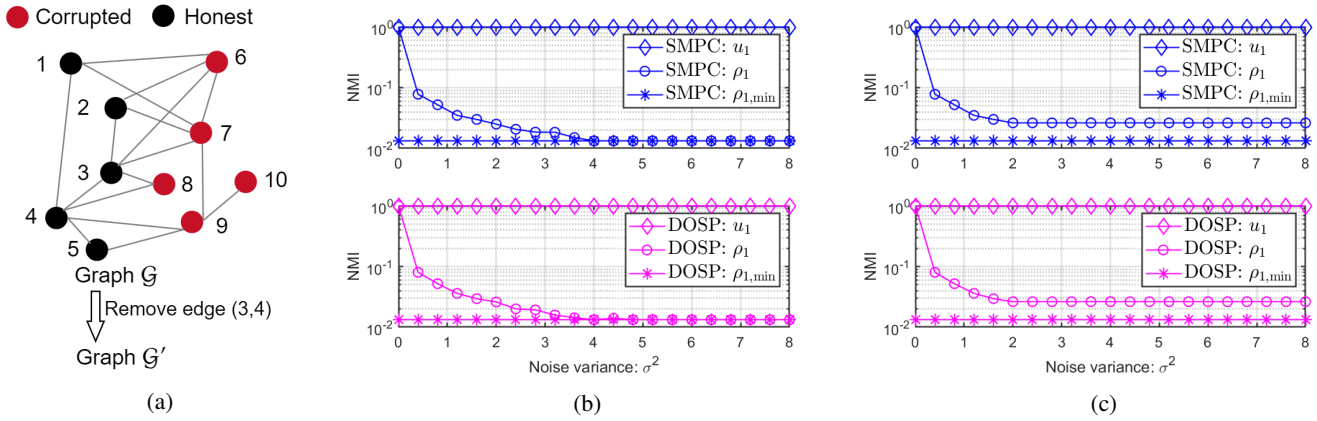
A. Convergence behavior

In Fig. 2 we present the convergence behavior of the algorithms under different amounts of noise insertion, i.e., different noise variances. We can see that all algorithms achieve the

[3]Since the experiments are done using discrete data, the mutual information $I(X;X)$ is bounded by $H(X) < \infty$.

TABLE I: Comparisons of existing information-theoretic solutions for the distributed average consensus

	DP [8]–[10]	SMPC [4]–[6]	DOSP [7], [27]
Adversary models	Passive, Eavesdropping		
Coordinated noise insertion	No	Yes	No
Output utility	$u_i = I(Y_i; Y_i + R_{\text{ave}})$	$u_i = I(Y_i; Y_i)$	
Individual privacy	$\rho_i = I(S_i; S_i + R_i)$	$\rho_i = I(S_i; \sum_{j \in \mathcal{N}_{h_q}} S_j)$	
Lower bound on individual privacy	$\rho_{i,\min} = I(S_i; S_i + R_{\text{tot}})$	$\rho_{i,\min} = I(S_i; \sum_{j \in \mathcal{N}_h} S_j)$	
Maximum number of corrupted nodes	$k_i = n - 1$ out of n	$k_i = d_i - 1$ out of d_i	
Cost of channel encryption	$c_i = 0$	$c_i = 1$	



(a) (b) (c)

Fig. 3: (a) Two sample graphs in which $\mathcal{G}'$ and $\mathcal{G}$ differ in only one edge. Normalized mutual information of output utility, individual privacy, and its lower bound for honest node 1 in terms of the amount of noise insertion by using SMPC and DOSP approaches under (b) graph $\mathcal{G}$ and (c) graph $\mathcal{G}'$.

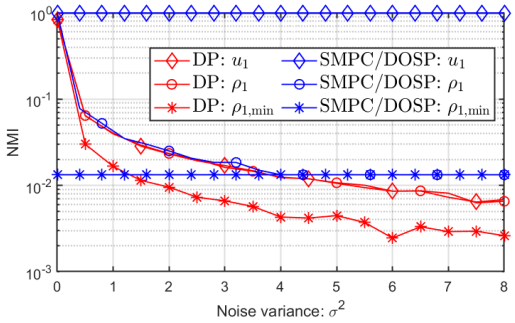


Fig. 4: NMI of output utility, individual privacy, and its lower bound for honest node i in terms of the amount of noise insertion using DP, SMPC and DOSP approaches.

correct average value in the absence of noise, i.e., $\sigma^2 = 0$. For nonzero noise variance, however, only the DOSP and SMPC based approaches achieve the correct average value, regardless of the amount of noise inserted, whereas the accuracy of the DP based approach is compromised by increasing the amount of noise insertion.

B. Utility and privacy

To validate the output utility, individual privacy, and its lower bound, we ran 10^4 Monte Carlo simulations and used the non-parametric entropy estimation toolbox (npeet) [53] to estimate the normalized mutual information.

1) Privacy-utility results of the DOSP and SMPC based approaches under different graph topologies: As shown in Table I, the performances of SMPC and DOSP are dependent on the number of corrupted nodes in the neighborhood and the graph topology. Note that we do not consider DP here because its performance is not dependent on graph topology as it assumes $n - 1$ corrupted nodes. To demonstrate the effects of graph topology, Fig. 3(a) shows a graph $\mathcal{G}$ satisfying Assumption 3; i.e., every honest node is connected to at least one corrupted node. In addition, we consider the graph $\mathcal{G}'$ which is obtained from $\mathcal{G}$ by removing edge $(3, 4)$. The main difference between graph $\mathcal{G}$ and $\mathcal{G}'$ is that, after removing all corrupted nodes, in the former all the honest nodes are connected and in the latter they are separated in two connected subgraphs. The privacy-utility results of the DOSP and SMPC based approaches over graph $\mathcal{G}$ and $\mathcal{G}'$ are shown in Fig. 3(b) and 3(c), respectively. We validate the following theoretical results regarding utility and privacy:

- SMPC and DOSP both ensure full utility regardless of the amount of noise, and thus the privacy level;
- The optimum individual privacy of node $i \in \mathcal{C}_q$ is only related to the partial sum of the private data in subgraph $\mathcal{C}_q$, i.e, $\rho_i = I(S_i; \sum_{j \in \mathcal{N}_{h_q}} S_j)$;
- For graph $\mathcal{G}$ both approaches are able to obtain perfect individual privacy, i.e., the result in Remark 2 is validated.

2) Privacy-utility comparisons of the DP, SMPC and DOSP approaches: In Fig. 4 we compare DP, SMPC and DOSP in terms of the amount of noise insertion using graph $\mathcal{G}$. We show the performance of SMPC and DOSP together because they have identical performances as shown in Fig. 3(b). Fig. 4 shows that, in contrast to SMPC and DOSP which guarantee perfect output utility and a fixed individual privacy, DP can achieve a lower individual privacy by increasing the noise variance. However, the price to pay is a deterioration of output

utility, validating the fact that DP trades-off privacy versus utility.

VIII. Suggestions for algorithm design

We now provide some suggestions on how to design appropriate privacy-preserving algorithms for different applications. Typical ways to design a privacy-preserving solution are (1) choose one of the off-the-shelf tools such as DP, SMPC or DOSP; (2) combine them to obtain a hybrid approach. We concluded that the performances indicator of privacy-preserving distributed processing algorithms were bounded by $u_i \leq I(Y_i;Y_i)$ (perfect output utility), $I(S_i;S_i) > \rho_i \geq \rho_{i,\min}$ (perfect individual privacy), $k_i \leq n-1$ (maximum number of corrupted nodes being), and $c_i \geq 0$ (minimum (zero) cost for channel encryption). To provide insight on *when it is possible to achieve these optimum performances simultaneously*, we have the following result.

Remark 4. *(For any application satisfies $I(S_i;\{S_j,Y_j\}_{j\in\mathcal{N}\setminus\{i\}}) = I(S_i;S_i)$, it is impossible to protect privacy under the conditions of both perfect output utility and $k_i = n-1$ being the maximum number of corrupted nodes.) The reason is simply because the lower bound under such conditions $\rho_{i,\min} = I(S_i;\{S_j,Y_j\}_{j\in\mathcal{N}\setminus\{i\}}) = I(S_i;S_i)$ is already the maximum; there is no privacy at all. An immediate implication of this result is that a SMPC/DOSP, which achieves perfect output utility, can never be differentially private for such applications. In other words, DP and SMPC/DOSP are mutually exclusive for such applications.*

One conclusion for algorithm design can be drawn from the above result: given an application at hand, the first thing to do is to compute the lower bound under the condition of perfect output utility and $k_i = n-1$, i.e., $\rho_{i,\min} = I(S_i;\{S_j,Y_j\}_{j\in\mathcal{N}\setminus\{i\}})$. Based on this lower bound, we then classify applications into two classes and give related suggestions on how to design algorithms.

A. Applications for which $\rho_{i,\min} = I(S_i;S_i)$

One example of such applications is the distributed average consensus. For applications where $\rho_{i,\min} = I(S_i;S_i)$ (Remark 4), we should be aware that it is impossible to design privacy-preserving algorithms with all optimum performances. Therefore, we have to prioritize different performances, compromise one to achieve another. Here are some suggestions for algorithm designs:

1) If the application is in an extreme distrust scenario, i.e., $k_i = n-1$ is required, then adopt DP based approaches. But be aware that there is a trade-off between privacy and utility.
2) If the application is very sensitive in terms of the accuracy of function output, e.g., perfect output utility is a must, then both SMPC and DOSP are options. But be aware that $k_i < n-1$ and that the individual privacy depends on the graph topology.

B. Applications for which $\rho_{i,\min} < I(S_i;S_i)$

One such example is the application where the objective function is a function of the ℓ_1-norm, like $f(\boldsymbol{s}) = \sum_{i\in\mathcal{N}} |s_i|\mathbf{1}$. For applications where $\rho_{i,\min} < I(S_i;S_i)$, we have the following suggestions:

1) If $\rho_{i,\min}$ is tolerable, it is possible to achieve perfect individual privacy $\rho_i = \rho_{i,\min}$ under the condition of both perfect output utility and $k_i = n-1$. Try to use either SMPC or DOSP to achieve such optimum performances.
2) If the above cannot be achieved, one option is to compromise the requirement of $k_i = n-1$, i.e., decrease k_i, and try to use SMPC or DOSP to obtain both perfect individual privacy and perfect output utility only.
3) If $\rho_{i,\min}$ is not tolerable, one option is to combine SMPC or DOSP with DP to decrease this lower bound by compromising the output utility.

IX. Conclusions

In this paper, we first proposed information-theoretic metrics for quantifying the algorithm performance in terms of output utility and individual privacy. The proposed metrics are general and can reduce to well-known frameworks including SMPC and DP under certain conditions. We derived several theoretical results in terms of mutual information. We explicitly analyzed, compared and related the state-of-the-art algorithms including DP, SMPC and DOSP for the distributed average consensus problem, and validated the theoretical results by computer simulations. Given the lower bound on individual privacy, we gave suggestions on how to design privacy-preserving algorithms given different conditions/assumptions.

Appendix A
Proof of Proposition 1

Proof. As the private data S is independent of the noise R, we have $\sigma_Z^2 = \sigma_S^2 + \sigma_R^2$. Let $\gamma = 1/\sigma_Z$ and define $Z' = \gamma Z$ as the normalized random variable with unit variance. Since mutual information is invariant under scaling, we have

$$\begin{aligned}\lim_{\sigma_R^2\to\infty} I(S;Z) &= \lim_{\sigma_R^2\to\infty} I(\gamma S;\gamma Z)\\ &= \lim_{\gamma\to 0} I(\gamma S;Z')\\ &= I(0;Z')\\ &= 0.\end{aligned}$$

Hence we conclude that given arbitrary small $\delta > 0$, there exists $\beta > 0$ such that for $\sigma_R^2 \geq \beta$ we have $I(S;Z) \leq \delta$. In the case of Gaussian distributed noise, we find

$$\begin{aligned}I(S;Z) &= h(Z) - h(Z|S)\\ &= h(Z) - h(R)\\ &\overset{(a)}{=} h(Z) - \frac{1}{2}\log(2\pi e\sigma_R^2)\\ &\overset{(b)}{\leq} \frac{1}{2}\log(2\pi e\sigma_Z^2) - \frac{1}{2}\log(2\pi e\sigma_R^2)\\ &= \frac{1}{2}\log(1+\sigma_S^2/\sigma_R^2),\end{aligned}$$

where (a) holds as the differential entropy of a Gaussian random variable with variance σ^2 is given by $\frac{1}{2}\log(2\pi e\sigma^2)$, and (b) holds because the maximum entropy of a random variable with fixed variance is achieved by a Gaussian distribution. Hence

$$\delta = \frac{1}{2}\log(1+\sigma_S^2/\sigma_R^2) \quad \Leftrightarrow \quad \sigma_R^2 = \frac{\sigma_S^2}{2^{2\delta}-1} = \beta.$$

□

Appendix B
Proof of Proposition 2

Proof.

$$\begin{aligned}
I(S_i;\mathcal{V}) &= I(S_i;\{S_j\}_{j\in\mathcal{N}_c},\{R_i^j\}_{(i,j)\in\mathcal{E}_c},\{X^{(t)}\}_{t\in\mathcal{T}}) \\
&\overset{(a)}{=} I(S_i;\{S_j\}_{j\in\mathcal{N}_c},\{R_i^j\}_{(i,j)\in\mathcal{E}_c},X^{(0)}) \\
&\overset{(b)}{=} I(S_i;\{S_j\}_{j\in\mathcal{N}_c},\{R_i^j\}_{(i,j)\in\mathcal{E}_c},\{X_j^{(0)}\}_{j\in\mathcal{N}_h}) \\
&\overset{(c)}{=} I(S_i;\{R_i^j\}_{(i,j)\in\mathcal{E}_c},\{X_j^{(0)}\}_{j\in\mathcal{N}_h}) \\
&\overset{(d)}{=} I(S_i;\{R_i^j\}_{(i,j)\in\mathcal{E}_c},\{S_j+\sum_{k\in\mathcal{N}_j}R_{j|k}\}_{j\in\mathcal{N}_h}) \\
&\overset{(e)}{=} I(S_i;\{R_i^j\}_{(i,j)\in\mathcal{E}_c},\{S_j+\sum_{k\in\mathcal{N}_{j,h}}R_{j|k}\}_{j\in\mathcal{N}_h}) \\
&\overset{(f)}{=} I(S_i;\{S_j+\sum_{k\in\mathcal{N}_{j,h}}R_{j|k}\}_{j\in\mathcal{N}_h}) \\
&\overset{(g)}{=} I(S_i;\{S_j+\sum_{k\in\mathcal{N}_{j,h}}R_{j|k}\}_{j\in\mathcal{N}_{h_q}}),
\end{aligned}$$

where (a) holds by Lemma 1, as $\forall t\geq 1: S_i \to X^{(0)} \to X^{(t)}$ forms a Markov chain; (b) holds, as $\{X_j^{(0)}\}_{j\in\mathcal{N}_c}$ can be determined from $\{S_j\}_{j\in\mathcal{N}_c},\{R_i^j\}_{(i,j)\in\mathcal{E}_c}$ using (29), (39) and (38); (c) holds because $\{S_j\}_{j\in\mathcal{N}_c}$ is independent of $\{R_i^j\}_{(i,j)\in\mathcal{E}_c},\{X_j^{(0)}\}_{j\in\mathcal{N}_h}$ and S_i; (d) holds by representing $\{X_j^{(0)}\}_{j\in\mathcal{N}_h}$ by using (29) and (38); (e) follows as $\{\sum_{k\in\mathcal{N}_{j,c}}R_{j|k}\}_{j\in\mathcal{N}_h}$ can be determined from $\{R_i^j\}_{(i,j)\in\mathcal{E}_c}$ by using (39); (f) holds as $\{R_i^j\}_{(i,j)\in\mathcal{E}_c}$ is independent of both S_i and $\{S_j+\sum_{k\in\mathcal{N}_{j,h}}R_{j|k}\}_{j\in\mathcal{N}_h}$; and (g) holds as $\{S_j+\sum_{k\in\mathcal{N}_{j,h}}R_{j|k}\}_{j\in\mathcal{N}_h\setminus\mathcal{N}_{h_q}}$ is independent of both S_i and $\{S_j+\sum_{k\in\mathcal{N}_{j,h}}R_{j|k}\}_{j\in\mathcal{N}_{h_q}}$. □

Appendix C
Proof of equation (53)

Proof. By combining (48) and two successive $\boldsymbol{x}$-updates (20), it can be shown that

$$\begin{aligned}
\boldsymbol{x}^{(t+1)}-\boldsymbol{x}^{(t-1)} = &\left(\boldsymbol{I}+c\boldsymbol{C}^\top\boldsymbol{C}\right)^{-1} \\
&\left(-2c\boldsymbol{C}^\top\boldsymbol{P}\boldsymbol{C}\boldsymbol{x}^{(t)}-2c\boldsymbol{C}^\top\boldsymbol{C}\boldsymbol{x}^{(t-1)}\right). \quad (55)
\end{aligned}$$

We have

$$\begin{aligned}
I(S_i;\mathcal{V}) &= I(S_i;\{S_j\}_{j\in\mathcal{N}_c},\{\Lambda_{i|j}^{(0)}\}_{(i,j)\in\mathcal{E}_c},\{X^{(t)}\}_{t\in\mathcal{T}}) \\
&\overset{(a)}{=} I(S_i;\{S_j\}_{j\in\mathcal{N}_c},\{\Lambda_{i|j}^{(0)}\}_{(i,j)\in\mathcal{E}_c},\{X^{(1)},X^{(2)}\}) \\
&\overset{(b)}{=} I(S_i;\{S_j\}_{j\in\mathcal{N}_c},\{\Lambda_{i|j}^{(0)}\}_{(i,j)\in\mathcal{E}_c},\{X_j^{(1)},X_j^{(2)}\}_{j\in\mathcal{N}_h}) \\
&\overset{(c)}{=} I(S_i;\{S_j\}_{j\in\mathcal{N}_c},\{\Lambda_{i|j}^{(0)}\}_{(i,j)\in\mathcal{E}_c} \\
&\quad ,\{S_j-\sum_{k\in\mathcal{N}_{j,h}}\boldsymbol{B}_{j|k}\Lambda_{k|j}^{(t)}\}_{j\in\mathcal{N}_h,t=0,1}) \\
&\overset{(d)}{=} I(S_i;\{S_j-\sum_{k\in\mathcal{N}_{j,h}}\boldsymbol{B}_{j|k}\Lambda_{k|j}^{(t)}\}_{j\in\mathcal{N}_h,t=0,1} \\
&\quad |\{S_j\}_{j\in\mathcal{N}_c},\{\Lambda_{i|j}^{(0)}\}_{(i,j)\in\mathcal{E}_c})
\end{aligned}$$

where (a) holds, as all $\{X^{(t)}\}_{t>2}$ can be determined by $X^{(1)}$ and $X^{(2)}$ using (55) (note that we omit $X^{(0)}$ by assuming $\boldsymbol{x}$ is initialized with all zeros); (b) holds, as $\{X_j^{(1)}\}_{j\in\mathcal{N}_c}$ can be constructed by $\{S_j\}_{j\in\mathcal{N}_c},\{\Lambda_{i|j}^{(0)}\}_{(i,j)\in\mathcal{E}_c}$; and similarly $\{X_j^{(2)}\}_{j\in\mathcal{N}_c}$ can be constructed by using $\{S_j\}_{j\in\mathcal{N}_c},X^{(1)},\{\Lambda_{i|j}^{(1)}\}_{(i,j)\in\mathcal{E}_c}$ based on (22), in which the last set can be determined using $\{S_j\}_{j\in\mathcal{N}_c},\{\Lambda_{i|j}^{(0)}\}_{(i,j)\in\mathcal{E}_c}$; (c) also follows from (22); and (d) follows from the definition of conditional mutual information and S_i being independent of both $\{S_j\}_{j\in\mathcal{N}_c}$ and $\{\Lambda_{i|j}^{(0)}\}_{(i,j)\in\mathcal{E}_c}$. □

References

[1] M. Anderson, *Technology device ownership, 2015*, Pew Research Center, 2015.

[2] J. Poushter and others, "Smartphone ownership and internet usage continues to climb in emerging economies," *Pew Research Center*, vol. 22, pp. 1–44, 2016.

[3] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Process. Magazine, vol. 30, no. 1, pp. 82–105*, 2013.

[4] Q. Li, I. Cascudo, and M. G. Christensen, "Privacy-preserving distributed average consensus based on additive secret sharing," in *Proc. Eur. Signal Process. Conf., pp. 1-5*, 2019.

[5] N. Gupta, J. Katz, N. Chopra, "Privacy in distributed average consensus," *IFAC-PapersOnLine, vol. 50, no. 1, pp. 9515-9520*, 2017.

[6] N. Gupta, J. Kat and N. Chopra, "Statistical privacy in distributed average consensus on bounded real inputs," in *ACC, pp 1836-1841*, 2019.

[7] Q. Li, R. Heusdens and M. G. Christensen, "Convex optimisation-based privacy-preserving distributed average consensus in wireless sensor networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process., pp. 5895-5899*, 2020.

[8] M. Kefayati, M. S. Talebi, B. H. Khalajand H. R. Rabiee , "Secure consensus averaging in sensor networks using random offsets," in *Proc. of the IEEE Int. Conf. on Telec., and Malaysia Int. Conf. on Commun., pp. 556–560*, 2007.

[9] Z. Huang, S. Mitra, and G. Dullerud, "Differentially private iterative synchronous consensus," in *ACM workshop Privacy electron. Soc., pp. 81–90*, 2012.

[10] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private average consensus: Obstructions, trade-offs, and optimal algorithm design," *Automatica, vol. 81, pp. 221–231*, 2017.

[11] N. E. Manitara and C. N. Hadjicostis, "Privacy-preserving asymptotic average consensus," in *ECC, pp. 760–765*, 2013.

[12] Y. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Trans. Automat Contr., vol. 62, no. 2, pp. 753–765*, 2017.

[13] J. He, L. Cai, C. Zhao, P. Cheng, X. Guan, "Privacy-preserving average consensus: privacy analysis and algorithm design," *IEEE Trans. Signal Inf. Process. Netw., vol. 5, no. 1, pp. 127–138*, 2019.

[14] P. Braca, R. Lazzeretti, S. Marano, and V. Matta, "Learning with privacy in consensus + obfuscation," *IEEE signal process. Lett., vol. 23, no. 9, pp. 1174–1178*, 2016.

[15] M. T. Hale, M. Egerstedt, "Differentially private cloud-based multi-agent optimization with constraints," in *Proc. American Control Conf., pp. 1235-1240*, 2015.

[16] M. T. Hale, M. Egerstedt, "Cloud-enabled differentially private multi-agent optimization with constraints," *IEEE Trans. Control Netw. Syst., vol. 5, no. 4, pp. 1693–1706*, 2018.

[17] K. Tjell and R. Wisniewski, "Privacy preservation in distributed optimization via dual decomposition and ADMM," in *Proc. IEEE 58th Conf. Decis. Control., pp. 7203–7208*, 2020.

[18] Q. Li, R. Heusdens and M. G. Christensen, "Convex optimization-based privacy-preserving distributed least squares via subspace perturbation," in *Proc. Eur. Signal Process. Conf., to appear*, 2020.

[19] K. Tjell, I. Cascudo and R. Wisniewski, "Privacy preserving recursive least squares solutions," in *Proc. Eur. Control Conf., pp.3490–3495*, 2019.

[20] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization., pp. 1–10," in *Proc. Int. Conf. Distrib. Comput. Netw*, 2015.

[21] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed constrained optimization," *IEEE Trans. Autom. Control., vol. 62, no. 1, pp 50-64*, 2016.

[22] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private distributed convex optimization via functional perturbation," *IEEE Trans. Control Netw. Syst., vol. 5, no. 1, pp 395-408*, 2018.

[23] T. Zhang and Q. Zhu, "Dynamic differential privacy for ADMM-based distributed classification learning," *IEEE Trans. Inf. Forensics Security, vol. 12, no. 1, pp. 172–187*, 2016.

[24] X. Zhang, M. M. Khalili, and M. Liu, "Recycled ADMM: Improve privacy and accuracy with less computation in distributed algorithms," in *in Proc. 56th Annu. Allerton Conf. Commun., Control, Comput. pp.959–965*, 2018.

[25] X. Zhang, M. M. Khalili, and M. Liu, "Improving the privacy and accuracy of ADMM-based distributed algorithms," *Proc. Int. Conf. Mach. Lear. pp. 5796–5805*, 2018.

[26] Y. Xiong, J. Xu, K. You, J. Liu and L. Wu, "Privacy preserving distributed online optimization over unbalanced digraphs via subgradient rescaling," *IEEE Trans. Control Netw. Syst.*, 2020.

[27] Q. Li, R. Heusdens and M. G. Christensen, "Privacy-preserving distributed optimization via subspace perturbation: A general framework," in *IEEE Trans. Signal Process., vol. 68, pp. 5983 - 5996*, 2020.

[28] Q. Li, M. Coutino, G. Leus and M. G. Christensen, "Privacy-preserving distributed graph filtering," in *Proc. Eur. Signal Process. Conf., to appear*, 2020.

[29] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Advances in Cryptology–CRYPTO, pp. 643–662*. Springer, 2012.

[30] R. Cramer, I. B. Damgård, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*, Cambridge University Press, 2015.

[31] C. Dwork, "Differential privacy," in *ICALP, pp. 1–12*, 2006.

[32] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *Proc. 41st Annu. ACM Symp. Theory Comput., pp. 371-380*, 2009.

[33] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *SIGMOD, pp. 193–204*, 2011.

[34] D. Bogdanov, S. Laur, J. Willemson, "Sharemind: A framework for fast privacy-preserving computations," in *Proc. 13th Eur. Symp. Res. Comput. Security: Comput. Security, pp. 192-206,*, 2008.

[35] Q. Li and M. G. Christensen, "A privacy-preserving asynchronous averaging algorithm based on shamir's secret sharing," in *Proc. Eur. Signal Process. Conf., pp. 1-5*, 2019.

[36] D. Dolev, C. Dwork, O. Waarts, M. Yung, "Perfectly secure message transmission," *J. Assoc. Comput. Mach., vol. 40, no. 1, pp. 17-47,*, 1993.

[37] I. Wagner and D. Eckhoff, "Technical privacy metrics: a systematic survey," *ACM Comput. Surv., vol. 51, no. 3, pp. 1–38*, 2018.

[38] M. Gtz, A. Machanavajjhala, G. Wang, X. Xiao, J. Gehrke, "Publishing search logs—a comparative study of privacy guarantees," *IEEE Trans. Knowledge and Data Eng., vol. 24, no. 3, pp. 520–532*, 2011.

[39] A. Haeberlen, B. C. Pierce, A. Narayan, "Differential privacy under fire.," in *Proc. 20th USENIX Conf. Security., vol. 33*, 2011.

[40] A. Korolova, K. Kenthapadi, N. Mishra, A. Ntoulas, "Releasing search queries and clicks privately," in *Proc. Int'l Conf. World Wide Web, pp. 171–180*, 2009.

[41] P. Cuff and L. Yu, "Differential privacy as a mutual information constraint," in *Proc. 23rd ACM SIGSAC Conf. Comput. Commun. Secur., pp 43–54*, 2016.

[42] M. Lopuhaä-Zwakenberg, B. Škorić and N. Li, "Information-theoretic metrics for local differential privacy protocols," *arXiv preprint arXiv:1910.07826*, 2019.

[43] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. IEEE Annu. Symp. Found. Comput. Sci., pp. 429–438*, 2013.

[44] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," in *NIPS., pp. 2879–2887*, 2014.

[45] T. M. Cover and J. A. Tomas, *Elements of information theory*, John Wiley & Sons, 2012.

[46] J. Pang, G. Cheung, A. Ortega, O. C. Au, "Optimal graph Laplacian regularization for natural image denoising," in *Proc. Int. Conf. Acoust., Speech, Signal Process., pp 2294-2298*, 2015.

[47] SK Narang, A Gadde, A Ortega, "Signal processing techniques for interpolation in graph structured data," in *Proc. Int. Conf. Acoust., Speech, Signal Process., pp 5445-5449*, 2013.

[48] A. Olshevsky and J. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM J. Control Optim., vol. 48, no. 1, pp. 33–55*, 2009.

[49] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine learning, vol. 3, no. 1, pp. 1–122*, 2011.

[50] T. Sherson, R. Heusdens, W. B. Kleijn, "Derivation and analysis of the primal-dual method of multipliers based on monotone operator theory," *IEEE Trans. Signal Inf. Process. Netw., vol. 5, no. 2, pp 334-347*, 2018.

[51] G. Zhang and R. Heusdens, "Distributed optimization using the primal-dual method of multipliers," *IEEE Trans. Signal Process., vol. 4, no. 1, pp. 173–187*, 2018.

[52] J. Dall and M. Christensen, "Random geometric graphs," *Physical review E, vol. 66, no. 1, pp. 016121*, 2002.

[53] G. Ver Steeg, "Non-parametric entropy estimation toolbox (npeet)," *https://github.com/gregversteeg/NPEET*, 2000.