Provably Private Distributed Averaging Consensus: An Information-Theoretic Approach

Mohammad Fereydounian, Graduate Student Member, IEEE, Aryan Mokhtari, Ramtin Pedarsani, Senior Member, IEEE, and Hamed Hassani, Member, IEEE

Abstract—In this work, we focus on solving a decentralized consensus problem in a private manner. Specifically, we consider a setting in which a group of nodes, connected through a network, aim at computing the mean of their local values without revealing those values to each other. The distributed consensus problem is a classic problem that has been extensively studied and its convergence characteristics are well-known. However, state-of-the-art consensus methods build on the idea of exchanging local information with neighboring nodes which leaks information about the users' local values. We propose an algorithmic framework that is capable of achieving the convergence limit and rate of classic consensus algorithms while keeping the users local values private. The key idea of our proposed method is to carefully design noisy messages that are passed from each node to its neighbors such that the consensus algorithm still converges precisely to the average of local values, while a minimum amount of information about local values is leaked. We formalize this by precisely characterizing the mutual information between the private message of a node and all the messages that another adversary collects over time. We prove that our method is capable of preserving users privacy for any network without a so-called generalized leaf, and formalize the trade-off between privacy and convergence time. Unlike many private algorithms, any desired accuracy is achievable by our method, and the required level of privacy only affects the convergence time.

Index Terms—Distributed learning, private learning, leakage measure, information-theoretic privacy, algebraic graph theory.

I. Introduction

In this paper, we focus on a classic distributed computing problem in which a group of connected agents aim to find the average of their local values, also known as the consensus problem. Due to applications of the consensus problem in many domains, such as sensor fusion [1]–[3], distributed energy management [4], Internet of Things [5], and large-scale machine learning and federated learning [6], [7], it has been widely studied in the literature [8]–[10].

A common feature in most classic consensus algorithms is the requirement for sharing the local values with neighboring nodes. However, this approach is, indeed, problematic in settings that nodes are not willing to share their exact local value (opinion or belief) due to privacy concerns [11]. An example would be in the case where members of a social network aim to compute their common opinion on a subject, but want to keep their personal opinions secret [12]. Another example rises in the case where multiple parties in a financial system seek to reach a summation (e.g., of bank capital) while individual data is extremely sensitive. This issue has motivated a new line of research which focuses on the possibility of achieving consensus in a fully decentralized manner without disclosing the initial nodes' value.

To provide privacy in the consensus problem, the first step is to define a measure that quantifies it. Differential privacy [13], [14] is the most studied measure of privacy for an algorithm running over a dataset. This notion measures the privacy based on the statistical dependency of an algorithm's output to the perturbation of a single element of the input dataset. Noting that the classic consensus method is a deterministic procedure, the most common approach for a private consensus algorithm in the literature is that each node perturbs its signals by some form of noise such that the resulting stochastic algorithm is differentially private [15]–[25]. However, these approaches suffer from some drawbacks. Adding perturbing noises affects the convergence properties compared to non-private consensus algorithms in two ways: (i) It leads to a non-exact limit, and (ii) it compromises the convergence rate, i.e., it leads to slower convergence rates, e.g., slower than the rates achieved by [8]. While some methods address the non-exact limit by adding zero-sum correlated noise [11], [22], no prior study has addressed both (i) and (ii) simultaneously. This issue persists for any iterative method that perturbs in some way the communication messages at each iteration either randomly or in a deterministic manner. This includes works that use methods like output masking [26], state decomposition [27], splitting messages into segments and adding noise to each segment [21], observability methods [28], [29], edge-based perturbation methods [30], [31], and Homomorphic encryption-based methods [32]–[34].

To explain how this paper deviates from part of the existing literature, we highlight some specific characteristics of our setting: (i) We aim to present a simple algorithm in terms of both analysis and extension. In particular, we aim to fully preserve the original classical averaging consensus structure and protocol so that our methods can be implemented on the existing averaging consensus platforms; (ii) We consider a given graph structure rather than assuming that all nodes can

Manuscript received May 16, 2022; revised March 1, 2023; accepted July 11, 2023.

Mohammad Fereydounian and Hamed Hassani are with the Electrical and Systems Engineering Department, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: mferey@seas.upenn.edu; hassani@seas.upenn.edu).

Aryan Mokhtari is with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712 USA (e-mail: mokhtari@austin.utexas.edu)

Ramtin Pedarsani is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA (e-mail: ramtin@ece.ucsb.edu)

Communicated by Anand Sarwate, Associate Editor for the IEEE Transactions on Signal and Information Processing over Networks.

communicate; (iii) We suppose that initial values are random variables; (iv) We seek to deviate from the classical differential privacy as a measure of privacy in our method, because given the item (iii), the data collected at each node will be also a vector-valued random variable Y. Assuming that the initial value of interest is a random variable X due to (iii), the most natural way of conducting a privacy measure is to ask: "Does Y give any hint (information) about X" This is exactly what the mutual information between X and Y projects.

With attention to the item (iv), investigating the connection of mutual information and differential privacy [35] may be relevant here. However, the results of [35] do not enforce any implications on the subject we study in this paper. Note that [35] defines a concept of mutual information differential privacy where the concept of differential privacy is mapped to the mutual information between two constructed distributions and the paper shows an equivalency between such a measure and the classical differential privacy. However, the notion of the privacy discussed in this paper is not mutual information differential privacy. In fact, it is based on the mutual information between two quantities that are fundamentally irrelevant to that of [35]. Hence, it is hard to project the results of [35] here.

To elaborate how the specifications discussed in the earlier paragraph separates the current paper from the existing work, some instances are mentioned in the following. The works on multi-party computation [36] in general try to design specific and mostly discrete noise structures using coding-like schemes so that they cancel out and return the desired consensus value. In particular, [36] considers a setting in which all nodes can communicate. These properties imply that [36] deviates from our work according to the items (i) to (iv) mentioned in thee earlier paragraph. Moreover, [11] can be differentiated from our work in setting due to the items (i), (iii), and (iv).

Indeed, differential-privacy is a powerful measure for indicating the level of privacy of a learning algorithm running over a dataset. This is done by quantifying the statistical dependency of the algorithm's output to a perturbation of a single input data point. This approach is best applicable when a full list of data points is accessible by the adversary. However, in a consensus setting, the data available at one user, i.e., the messages it collects from its neighbors over time, is different from the data available at other users. By injecting extra noise to all communication messages in a consensus algorithm, the differential-privacy approaches mask the private values against an adversary that can eavesdrop all messages. The cost to defeat this adversary appears on convergence side as discussed above. However, in a setting that the adversary can gain access to only a single node's information, such approaches provide more than necessary privacy with an unwanted cost. In fact, the approaches based on the local adaptation of differential privacy that build a dataset based on a single node's perspective are most compatible with extra noise injection procedures that, as discussed above, fail to guarantee the exact limit and the fastest rate simultaneously. Hence, to handle the consensus setting with such type of adversary and asymmetric data distribution, differential privacy may be replaced with a more compatible privacy measure. There have been few information-theoretic measures proposed to model the leakage of a random variable to another. In particular, [37] proposes a leakage measure $\mathsf{L}(X \,!\, Y)$ that is based on some axiomatic properties. This novel measure is neatly for telecommunication purposes. However, finding an analytical closed-form for L is challenging when the inputs consist of high-dimensional continuous random variables that evolve over time. Such a setting arises for the concatenation of all observed data by a node over time. Based on this, for a consensus setting, using simpler information-theoretic measures such as mutual information is preferable.

Motivated by this point, we provide a novel information-theoretic scheme to measure and analyze privacy leakage in consensus problems. We further provide simple noise-aggregation methods to reach the exact consensus in a private manner while achieving the fastest rate that a non-private consensus method can achieve. These advantages are based on the fact that our proposed provably private averaging consensus (PPAC) method only modifies the initial values of the consensus dynamic and leaves the consensus procedure intact. Moreover, we use a probabilistic model where the private values are random variables, which makes our method fundamentally different from most of the prior work. In this probabilistic framework, we introduce a new measure of privacy leakage based on the mutual information between the initial value of a node and the set of messages that is available at another node. In this way, we can capture how private the initial value of a node is with respect to any other node in the network. Most of our theoretical results hold regardless of the shape of the distribution of noises and private values. Next, we state our contributions:

1) Given a network structure and a consensus matrix, we propose an algorithm for reaching consensus among the nodes, while keeping the initial values private throughout the algorithm and preserving the fastest rate that a non-private consensus method can achieve.
2) We formalize a continuous notion based on the mutual information function that measures the privacy leakage of any (victim) node at another (adversary) node that can be (optimally) tuned by our adjustable noise parameters.
3) We provide a precise characterization of the information flow of a private value over the network over time. Having this, even in ill-shaped structures that leak privacy, we can determine the amount of information that an adversary receives as a function of time and compute the waiting time until recovering a private value.

In summary, this paper has two main messages: (i) Privacy from a node's perspective can be measured and efficiently analyzed based on the concept of mutual information. (ii) Splitting private messages into fragments prior to running the classical consensus method maintains the convergence guarantees and ensures the privacy for most structures. No further additive noise is required.

II. Preliminaries

In this section, we first mention the notation used throughout the paper and then recap some basic concepts required for presenting our framework.

Notation. Column vectors are denoted by small letters in bold font, $\mathbf{a}, \mathbf{b}$, while matrices are denoted by capital non-bold letters, A, B, and scalars by small letters a, b. For a matrix A, wee denote the transpose of A by $A^\top$ while the notation A^t for a positive integer t indicates powers of the matrix A, i.e., self-multiplying of the matrix A for t times, e.g., $A^2 = A\ A$ and A^0 to be the identity matrix with same size as A. Moreover, $\{a_i\}_{i\in I}$ is considered as an ordered sequence of mathematical objects a_i, $i \in I$. Concatenation of ordered sequences is denoted by Cartesian product and when the number of elements are finite, these ordered sequences are considered as column vectors. For scalars a_i, the following example illustrates these notations:

$$\{a_i\}_{i\in\{1,2\}}\ (a_3, a_4)\ \{a_5\} = [a_1, a_2, a_3, a_4, a_5]^\top. \tag{1}$$

The right-hand side of (1) indicates a column vector with elements a_1, a_2, a_3, a_4, a_5. When a_i s in (1) are replaced with row vectors, (1) refers to the matrix with rows a_1, a_2, a_3, a_4, a_5. Moreover, note that $\{a_i\}_{i\in\{1,2\}}$ can be interpreted as either $[a_1, a_2]^\top$ or $[a_2, a_1]^\top$. In this paper, whenever the choice of order is not specified, the argument holds regardless of the choice. The identity matrix of size k is I_k and all ones (column) vector of length k is 1_k. We use both A_{ij} and $[A]_{ij}$ to denote the ij-th element of the matrix A. Also, we use $[A]_i$ and $[A]_j$ to represent the i-th row of A as a row vector and the j-th column of A as a column vector. Similarly, $[v]_i$ refers to the i-th coordinate of the vector v. Further, $A \succeq 0$ means $A_{ij} \geq 0$ for all i, j and $\mathrm{diag}(a_1, \ldots, a_n)$ represents an $n \times n$ diagonal matrix with a_i as its i-th diagonal element. If $A = \{a_1, \ldots, a_k\} \subseteq \mathbb{R}^n$, then $\mathrm{span}(A) = \{\sum_{i=1}^k t_i a_i \mid t_1, \ldots, t_k \in \mathbb{R}\}$. Further, $[n] = \{1, \ldots, n\}$ and $A \setminus B$ represents set difference for sets A and B. Finally, $|A|$ denotes the size of set A and for (real or complex) scalar a, $|a|$ denotes the absolute value of a.

Graph Theory. By $G = (V, E)$, we denote a simple graph where V is the set of nodes and E is the set of (undirected) edges. The distinct nodes i and j are called neighbors if $\{i, j\} \in E$. Further, the neighborhood of node i, denoted by N_i, is the set of all neighbors of node i. The degree of node i is $\deg(i) = |N_i|$. A walk of length k between i and j, is the sequence of nodes $(\ell_0, \ell_1, \ldots, \ell_k)$, where $\ell_0 = i$, $\ell_k = j$, and all consecutive nodes are neighbors. A graph is called connected if for every node pair, there is a walk between them. The minimum k for which a walk of length k exists between i and j is called the distance between i and j and denoted by $d_G(i, j)$. The eccentricity of node i is the largest distance we can get from node i, i.e., $\mathrm{ecc}_G(i) = \max_{j\in V} d_G(i, j)$ and the radius of G is defined as $r(G) = \min_{i\in V} \mathrm{ecc}_G(i)$. The adjacency matrix of a graph G with n nodes is an $n \times n$ matrix denoted by A_G, where $[A_G]_{ij}$ is 1 if $\{i, j\} \in E$ and 0 otherwise.

Matrix Theory. For matrices A and B, we define $A \succeq B$ if for every $i = j$, $A_{ij} = 0$ if and only if $B_{ij} = 0$. Spectral radius of $A \in \mathbb{R}^{n\times n}$ with eigenvalues $\lambda_1, \ldots, \lambda_n$ is $\rho(A) = \max_{i\in[n]} |\lambda_i|$. For $A \in \mathbb{R}^{n\times n}$, its minimal polynomial ψ_A is defined as the unique monic polynomial with minimum degree such that $\psi_A(A) = 0$. By Cayley-Hamilton theorem, $\deg(\psi_A) \leq n$ (see [38]).

Given an infinite sequence of vectors $\{a_i\}_{i=0}^{\infty}$, where $a_i \in \mathbb{R}^n$ and $a_0 = 0$, we consider a basis pursuit procedure which constructs a finite subset B of the elements of the sequence $\{a_i\}_{i=0}^{\infty}$ such that the elements of B are linearly independent and $a_i \in \mathrm{span}(B)$ for all i and in this sense, it is a called a basis for $\{a_i\}_{i=0}^{\infty}$ and is denoted by $B = B(\{a_i\}_{i=0}^{\infty})$. The procedure is as follows: Add a_0 to B. For $i > 0$, if $a_i \notin \mathrm{span}(B)$, then add a_i to B and go to the next iteration. We formally prove in the supplementary material that this procedure results in a basis for $\{a_i\}_{i=0}^{\infty}$.

Information Theory. Consider continuous random variables X and Y which are defined over spaces $\mathcal{X}$ and $\mathcal{Y}$ with probability density functions (PDFs) f_X and f_Y, respectively. Moreover, let $f_{XY}(x, y)$ denote joint PDF of X and Y, then their mutual information $I(X; Y)$ is defined as

$$I(X; Y) = \int_{\mathcal{X}\mathcal{Y}} f_{XY}(x, y) \log \frac{f_{XY}(x, y)}{f_X(x) f_Y(y)}\, dx\, dy. \tag{2}$$

III. Problem Setup

In this section, we discuss the problem setup and the restrictions under which the problem is solved. To do this, we start by introducing the classical averaging consensus whose concepts are used in the our setting.

A. Classical Averaging Consensus

Consider a network represented by a simple connected graph $G = (V, E)$ over a set of nodes $V = \{1, \ldots, n\}$ with $m = |E|$. We assume only adjacent nodes can exchange information with each other. Moreover, suppose $u_i \in \mathbb{R}$ is the initial value of user i. The main goal in the consensus problem is to reach a state that all nodes in the network learn the average of initial vectors, i.e., $u = \frac{1}{n}\sum_{i=1}^n u_i$. Letting $v_i(t)$ be the value of node i at iteration t which is initially set to $v_i(0) = u_i$, one can consider a linear update given by

$$v_i(t+1) = W_{ii}\, v_i(t) + \sum_{j\in N_i} W_{ij} v_j(t). \tag{3}$$

Here, W_{ii} and W_{ij} denote the corresponding coefficients of $v_i(t)$ and $v_j(t)$, respectively, in the update formula of $v_i(t+1)$. One can form a matrix $W \in \mathbb{R}^{n\times n}$ whose ij-th element is W_{ij}, called the consensus matrix. Having this, the averaging consensus aims to have $\lim_{t\to\infty} v_i(t) = u$ for all $i \in [n]$. Letting $v(t) = [v_1(t), \ldots, v_n(t)]^\top$, one can write $v(t) = W^t v(0)$ seeking to have $\lim_{t\to\infty} v(t) = u$, where $u = u 1_n$. This is equivalent to

$$\lim_{t\to\infty} W^t = \frac{1}{n} 1_n 1_n^\top. \tag{4}$$

It is known that Equation (4) holds if and only if (i) $W 1_n = W^\top 1_n = 1_n$ and (ii) $\rho(W - 1/n\, 1_n 1_n^\top) < 1$, where $\rho(\cdot)$ denotes the spectral radius of a matrix; see [8] for more details. Further, the convergence rate of (4) can be computed as follows:

$$\sup_{v(0)\neq u} \lim_{t\to\infty} \left(\frac{\|v(t) - u\|_2}{\|v(0) - u\|_2}\right)^{\frac{1}{t}} = \rho\left(W - \frac{1}{n} 1_n 1_n^\top\right). \tag{5}$$

B. Problem Definition

Definition 1: Basics. We consider a network over n nodes (users) with an underlying simple graph $G = ([n]; E)$. We assume that each node $i \in [n]$ has a (secret) fixed initial value $u_i \in \mathbb{R}$ and the quantities $u_i; i \in [n]$ are continuous independent random variables.

Communication. We assume that only adjacent nodes are allowed to directly communicate. Except for one initial round, all the communications are restricted to be synchronous broadcasting, i.e., there is a clock that determines the communication iteration for all users and in each iteration, each user has to broadcast some value to all its neighbors following by an update of the form (3) for some given fixed weight matrix $W \in \mathbb{R}^{n \times n}$. Only in the initial round, users have still synchronous but possibly non-broadcasting communications to the neighbors, i.e., they may send different values to different neighbors in the initial round.

Data. We assume that each node has only access to what it receives from the neighboring nodes over time following the communication protocol described above. This excludes the possibility of colluding among nodes.

Goal. The goal is that each node i obtains $u = \frac{1}{n} \sum_{i=1}^{n} u_i$ without being able to recover u_j for $j = i$ from the data described above.

To explain the setting we introduce in Theorem 1, the adversary and privacy models are described in the following.

Adversary model. We assume all nodes are honest but curious, meaning that they follow the protocols and communication principles of the network, but they might be willing to recover the initial value of other nodes in the network. This is equivalent to the situation in which an adversary obtains access to the messages received by a single node. Moreover, we assume that nodes (adversaries) do not collude in recovering private messages and the underlying graph and the full consensus matrix are accessible by all nodes (adversaries).

Privacy model. In this part, we further determine what is means to recover an the initial in Theorem 1. Consider arbitrary distinct nodes i and j and suppose node i is curious about node j's private value u_j. Node i only has access to the u_i and all messages it sends to or receives from its neighbors over time. Denote the concatenation of all these by D_i. The privacy fails if node i can deterministically recover u_j from D_i. Otherwise, it is important to know how close node i can statistically estimate u_j using D_i. This can be evaluated by finding how much information D_i reveals about u_j which can be formalized by the mutual information between the joint distribution of the elements of D_i and u_j, denoted by $I(D_i; u_j) \in [0; 1]$. Achieving zero mutual information is impossible since the consensus goal u itself reveals some information about u_j. Upcoming sections reveal how tuning the model's parameters push $I(D_i; u_j)$ toward 0 as much as possible.

Next, we briefly mention our approach in achieving the goal expressed in Theorem 1.

Approach. We consider the averaging dynamic described in (3) under the constraint that the original messages u_i must be kept private from other nodes. Satisfying this requirement rules out the naive initialization of $v_i(0) = u_i$. Therefore, we seek new initializations $v_i(0)$ in the first round of communication to fulfill the privacy and convergence requirements while following the consensus dynamic (3). We present this initialization and rigorously analyze the privacy and convergence under the model described earlier in this section.

C. Problem Extension

We assume that the initial values u_i are scalar-valued continuous random variables, but a similar argument is applicable to a discrete case. Also, for a vector-valued u_i with independent coordinates, our results can be applied to each coordinate. As we will discuss in Section V, the result of our privacy analysis is proved for the case where the randomness sources of the model are all Gaussian. However, the structure of our analysis allows one to rebuild the results for other distributions.

As an interesting direction, one may think of extending the problem to the case where the underlying graph is directed rather than simple. It is important to note that the current assembly of the analysis requires the graph to be simple. To incorporate such an assumption, one must first setup a meaningful model. To explain this necessity, suppose one node i_0 has only inward edges. Following our model described in Theorem 1, this means no other node can incorporate u_{i_0} as a partial term in any update, i.e., the value of u_{i_0} will not appear in the final consensus and thus u cannot be achieved. Despite these confusions, extension of this paper to directed graphs can be an interesting direction if the model definition is properly investigated.

IV. Private Consensus

In this section, we first explain our method and then introduce a privacy leakage measure in Section IV-A. As mentioned earlier, the regular averaging consensus method reveals the private messages at $t = 0$ due to the initialization $v_i(0) = u_i$. We propose a new initialization for $v_i(0)$ that preserves privacy while leaving most convergence properties intact. The main idea behind the proposed method is the following: If we modify the initial vectors $v_i(0)$ such that their sum preserves the sum of the right local values u_i, i.e., $\sum_{i=1}^{n} v_i(0) = \sum_{i=1}^{n} u_i$, then by following the averaging consensus dynamic all nodes converge to the optimal value $u = \frac{1}{n} \sum_{i=1}^{n} u_i$. Now our goal is to select these values so that they do not reveal the original values $\{u_i\}_{i \in [n]}$ or minimize the amount of information that is revealed after a specific number of communication rounds.

To do so, we proceed as follows. In the first round, which we call the preparation phase, unlike the traditional consensus approach, each node i does not send the same signal to all its neighbors. Instead, it splits its private message u_i into multiple pieces and distributes it among its neighbors. All but one of these pieces are pure noise terms, independent from u_i. More precisely, in the preparation phase, each node i splits its private value u_i into fragments $_{ij}$, where $j \in N_i$. Hence, the number of fragments is equal to the number of neighbors of node i. These elements are selected such that all of them except one are pure noise and their sum recovers the original signal, i.e.,

$$u_i = \sum_{j \in N_i} {}_{ij}: \tag{6}$$

Algorithm 1 Provably Private Averaging Consensus (PPAC)

Preparation-Phase:

1: for $i \in [n]$ do
2: Node i picks $m_i \in N_i$ arbitrarily.
3: For all $j \in N_i \setminus \{m_i\}$: node i generates noise γ_{ij} and sends it to node j
4: Node i computes $\gamma_{im_i} = u_i - \sum_{j \in N_i \setminus \{m_i\}} \gamma_{ij}$ and sends it to node m_i.
5: end for
6: for $i \in [n]$ do
7: Node i computes: $v_i(0) = \sum_{k \in N_i} \gamma_{ki}$.
8: end for

Consensus-Phase:

1: Nodes follow the consensus dynamic in (3) with initial values $\{v_i(0)\}_{i \in [n]}$.

Formally, to create such signals, node i arbitrarily chooses one of its neighbors, which we denote by m_i. For $j = m_i$, node i sets γ_{ij} to be a continuous random variable, independent from all other randomness sources. Then, node i sets γ_{im_i} such that (6) holds, i.e., it sets

$$\gamma_{im_i} = u_i - \sum_{j \in N_i \setminus \{m_i\}} \gamma_{ij}. \tag{7}$$

The following indexing set S distinguishes all pairs (i,j) such that γ_{ij} is a pure noise term.

$$S = \{(i,j) \mid i \in [n], j \in N_i \setminus \{m_i\}\}. \tag{8}$$

Note that $|S| = 2m - n$, where $m = |E|$. Randomness sources in this model consist of n private values and $2m - n$ pure noisy messages. Next, we state the independence requirement.

Assumption 2: The concatenation of all $2m$ randomness sources in Algorithm 1, i.e., $\{u_i\}_{i \in [n]} \cup \{\gamma_{ij}\}_{(i,j) \in S}$, where S is defined as (8) consists of independent elements.

The choice of distribution for randomness sources is arbitrary among continuous random variables with a valid PDF unless otherwise is specified. However, it is beneficial to fix a notation for their mean and variances as follows.

Definition 3: For $i \in [n]$ and $(k,\ell) \in S$, let μ_i and σ_i^2 be the mean and variance of u_i and $\gamma_{k\ell}$ and σ_k^2 be the mean and variance of $\gamma_{k\ell}$, respectively. Moreover, let $\mu_{\max}$ be the maximum of the absolute value of all μ_i and $\gamma_{k\ell}$, and define $\sigma_{\max}$ in a similar manner for variances.

Once the preparation phase is done and the messages γ_{ij} are communicated, every node computes its consensus initialization by summing up the messages that it has received in the preparation phase, that is, $v_i(0) = \sum_{k \in N_i} \gamma_{ki}$. After that, all nodes follow the consensus dynamic described in (3). The steps of our proposed method (PPAC) are summarized in Algorithm 1.

Note that PPAC preserves the sum property after the preparation phase, i.e., $\sum_{i=1}^n v_i(0) = \sum_{i=1}^n u_i$. To achieve the exact limit, it is required to have $W\mathbf{1}_n = W^\top \mathbf{1}_n = \mathbf{1}_n$. This implies that the sum property is preserved during the consensus phase too, i.e., $\sum_{i=1}^n v_i(t) = \sum_{i=1}^n v_i(0) = \sum_{i=1}^n u_i$.

A. Privacy Leakage Measure

We first provide a mathematical definition to formalize the notion of privacy leakage that was earlier described in Section III. Then, we rigorously analyze this notion.

We start with mathematically formalizing the concatenation of all data that a node i collects, which was earlier denoted by D_i in Section III. The first data available at node i is its own private value u_i. Next, in the preparation phase, it generates $\{\gamma_{i\ell}\}_{\ell \in N_i \setminus \{m_i\}}$. Note that γ_{im_i} is a redundant data, since it is simply the subtraction of other pure noises from u_i. In the preparation phase, node i receives $\{\gamma_{\ell i}\}_{\ell \in N_i}$ from its neighbors. Moreover, when the consensus procedure starts, at time $t \geq 0$, node $\ell \in N_i$ transmits $v_\ell(t)$ to node i so that it can compute its new update for the next iteration. Hence, up to time t, node i has received values $v_\ell(\cdot)$ for all $\ell \in N_i$ and $\cdot \in \{0,\dots,t\}$. Hence, the concatenation of node i's observed data up to time t can be written as

$$D_i(t) = \{u_i\} \cup \{\gamma_{i\ell}\}_{\ell \in N_i \setminus \{m_i\}} \cup \{\gamma_{\ell i}\}_{\ell \in N_i} \cup \{v_\ell(\cdot)\}_{\ell \in N_i; \cdot \in t}. \tag{9}$$

The notation $D_i(\infty)$ is also applicable and represents all the data that node i can gather if the consensus runs forever. To measure the privacy leakage of u_j at node i up to time t, we consider the mutual information between $D_i(t)$ and u_j. This is formalized in the following definition.

Definition 4 (Privacy Leakage Measure): Considering the definition of $D_i(t)$ in (9), the privacy leakage of node j from the perspective of node i up to time t is defined as

$$\mathcal{P}_i^{(j)}(t) := I(D_i(t); u_j) \in [0, \infty). \tag{10}$$

Note that smaller $\mathcal{P}_i^{(j)}(t)$ means u_j is more private at node i as less information is revealed about it.

V. Main Results

Based on the private consensus method in Section IV, our main results can be stated informally in Theorem 6. Formal statements regarding privacy and convergence are provided in Section VI and Section VII, respectively. To present the main results, we first must mention the definition of a generalised leaf in the following.

Definition 5 (informal): We say there is a generalized leaf with head j and tail i if either of the following cases occurs: (i) When node i is the only neighbor of node j; (ii) When node j is not adjacent to node i, and node j only has neighbors of degree two and node i is adjacent to all of them; (iii) The situation of case (ii) while the edge $\{i,j\}$ is also added. An illustration of these cases is provided in Figure 1.

Theorem 6 (informal): Running Algorithm 1 (PPAC) over a network and assuming all randomness sources are independent, we can conclude the following results:

1) For each node i, there are only finitely many iterations $t_1, \dots, t_k$ that the information of node i about some private values strictly increases (Theorem 8). For the latest of such iterations, i.e., t_k, we find an upper-bound $t_k \leq n - 1$, where n is the number of all nodes (Proposition 9), and a lower-bound $\mathrm{ecc}_G(i) \geq 2t_k$, where $\mathrm{ecc}_G(i)$ is the eccentricity of node i (Proposition 10).

2) If the network contains a specific sub-structure called a generalized leaf, defined in Definition 13, some nodes can deterministically recover the private values of some other nodes in the first consensus iteration, i.e., privacy fails.
3) If the network does not contain a generalised leaf, no node can fully recover a private value (Lemma 11 and Theorem 14). In this case, $\Gamma_i^{(j)}$, defined in Definition 4, measures the amount of leakage of u_j to node i, statistically. We obtain a closed form $\Gamma_i^{(j)}(t) = 1/2 \log(1+\alpha^\top_j a^\top \Sigma^{-1} a)$ for some vector a and matrix Σ (Theorem 12). In Section VI, we discuss the construction of a and Σ through several steps. In this construction, the quantities σ_k^2, the variances of noisy fragments $\xi_{k,\ell}, (k,\ell) \in \mathcal{S}$, appear as linear terms in the elements of Σ. This shows that $\Gamma_i^{(j)}$ is a decreasing function in terms of σ_k^2. Thus, increasing the tunable noise variances decreases the information leakage.
4) PPAC leaves the convergence limit and rate of the classical averaging consensus intact. Moreover, the convergence time to achieve an ε-accurate solution by PPAC scales as $O(\log(1/\varepsilon))$ and it also increases proportional to the logarithm of noise parameters (Theorem 16). This shows that tuning σ_k^2 adjusts the trade-off between privacy and convergence time.

Next, we discuss the novel techniques used to achieve the above results. The baseline in obtaining our results is transforming the information-/graph-theoretical formulation of the problem into a linear-algebraic form by writing the data collection at each node as a matrix-vector decomposition $D_i(t) = R_i(t)g$, where g is the vector consisting of the $2m$ randomness sources of the model. As we show later, the possibility of leakage and properties of information flow over the graph can be translated into constraints on $R_i(t)$.

While the size of g is fixed, each new iteration adds $\deg(i)$ rows to $D_i(t)$ and $R_i(t)$. The decomposition $D_i(t) = R_i(t)g$ reveals that, after sufficiently many iterations, all new rows are linear combinations of previous ones. This leads to the fact that finitely many iterations $t_1, \dots, t_k$ have new information. Moreover, we constructively find $t_1, \dots, t_k$ by running a basis-pursuit procedure on the rows of $R_i(t)$. As a next step, we write each element $R_i(t)$ as a function of the elements of W^t. Having this, we then use Cayley-Hamilton theorem (see [38]) on matrix powers to show that $t_k \leq n - 1$. The aforementioned decomposition also paves the way to translate the waiting time for node i to receive the first message containing non-redundant information about u_j in terms of the distance $d_G(i, m_j)$, which leads to $\mathrm{ecc}_G(i) \leq t_k$.

Considering the data collection $D_i(t)$ on only non-redundant data points corresponding to $t_1, \dots, t_r$ for $r \leq k$ gives D_i^r and accordingly R_i^r. The next tool to obtain the results of Theorem 6 is the fact that we translated the ability of deterministic recovery of u_j by node i into the rank-deficiency of R_i^r when its j-th column is removed. Having this, the core technique in obtaining the main results of this paper is transforming the aforementioned rank-deficiency condition into the existence of a specific substructure in the graph called a generalized leaf. A generalized leaf with head j and tail i consists of i being connected to all degree 2 neighbors of j (and possibly j itself) which is illustrated in Figure 1. While it is straightforward to see why in a generalized leaf with head j and tail i, node i can fully recover u_j, it is far more challenging to prove that this is indeed the only possible scenario for a full recovery of some private value. We prove this fact based on a refined analysis (see Section B-G) that shows, assuming there are no generalized leaves, the j-th column of R_i^r can be written in terms of other columns and thus removing the j-th column does not decrease the rank of R_i^r.

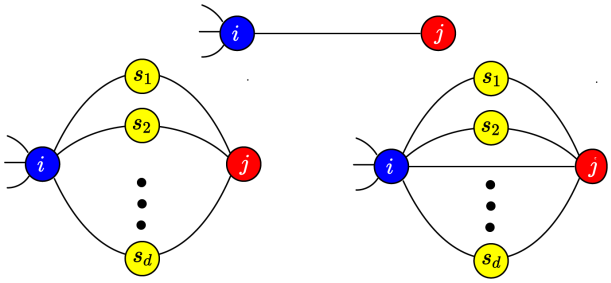

Fig. 1. All possible forms of a generalized leaf with head j and tail i: (i) When node i is the only neighbor of node j; (ii) When node j is not adjacent to node i and only has neighbors of degree two and node i is adjacent to all of them; (iii) The situation of case (ii) while the edge $\{i, j\}$ is also added.

By distinguishing the only case for a full (deterministic) recovery, the next step is to analyse and reduce the partial (stochastic) recovery of private values. With no generalized leaves in the graph, under Gaussian distribution for all randomness sources in the model, we then show that joint distribution of the elements of D_i^r is a non-degenerate Gaussian that has an invertible covariance matrix and compute the mutual information between this joint distribution and $u_{\cdot j}$. This leads to a closed-form expression for $\Gamma_i^{(j)}(t)$ and shows how it can be decreased by the model's tunable parameters.

VI. Privacy Analysis

In this section, we first introduce a representation of $D_i(t)$ in the form of a matrix-vector decomposition (Section VI-A), and then use it to formally state our results on privacy (Section VI-B).

A. Matrix-Vector Decomposition for $D_i(t)$

To obtain Theorem 6, we start by finding a representation of the elements of $D_i(t)$ in terms of the randomness sources of our model. To this aim, let $\Phi \in \mathbb{R}^{n \times n}$ be the matrix whose ij-th element is ϕ_{ij} defined in Algorithm 1. Based on the preparation phase, we have $v(0) = \Phi^\top 1_n$ which results in

$$\begin{aligned} v(t) &= W^t v(0) = W^t \Phi^\top 1_n \\ \Rightarrow v_i(t) &= [v(t)]_i = [W^t \Phi^\top 1_n]_i = [W^t]_i \Phi^\top 1_n. \end{aligned} \tag{11}$$

Recall that $[\,]_i$ denotes the i-th row. Consider all the (independent) sources of randomness in our model which are $\{u_i\}_{i \in [n]}$ and $\{\phi_{ij}\}_{(i,j) \in \mathcal{S}}$. Equation (11) implies that $v_i(t)$ can be written as a linear combination of these quantities.

The following expression suggests a notation for this linear combination:

$$v_i(t) = \sum_{j \in [n]} {}^{i}_{j}(t)\, u_j + \sum_{(k,\ell) \in S} {}^{i}_{k,\ell}(t)\, p_{k,\ell}: \quad (12)$$

(11) implies that the coefficients in (12) can be obtained in terms of the elements of $[W^t]_i$ as

$${}^{i}_{j}(t) = W^t_{i,m_j}; \qquad {}^{i}_{k,\ell}(t) = W^t_{i,\ell}, \quad W^t_{i,m_k}: \quad (13)$$

To express (12) in a vector-multiplication form, we define g to be a $2m$-dimensional (column) vector that is the concatenation of all sources of randomness and $p_i(t)$ to be the concatenation of the coefficients as follows: (Note that $g, p_i(t) \in \mathbb{R}^{2m}$.)

$$\begin{aligned} g &= \{u_j\}_{j\in[n]} \cup \{p_{k,\ell}\}_{(k,\ell)\in S}; \\ p_i(t) &= \{{}^{i}_{j}(t)\}_{j\in[n]} \cup \{{}^{i}_{k,\ell}(t)\}_{(k,\ell)\in S}: \end{aligned} \quad (14)$$

Hence, $v_i(t)$ can be written as

$$v_i(t) = p_i^{\top}(t)\, g: \quad (15)$$

Similar to $v_i(t)$ in (15), other elements of $D_i(t)$ are also linear combinations of the elements of g and can be written in vector-multiplication forms in terms of g. To this aim, we can consider matrices $_i$ and $_i$ (where both are of size $\deg_G(i) \times 2m$) such that

$$\{u_i\} \cup \{p_{i,\ell}\}_{\ell \in N_i \setminus \{m_i\}} = {}_i g; \qquad \{p_{i,\ell}\}_{\ell \in N_i} = {}_i g: \quad (16)$$

Putting together the representation of the elements of $D_i(t)$ in terms of g in (15), and (16), we obtain the matrix $R_i(t)$ by vertically concatenating the matrices $_i$, $_i$, and appending vectors $p_{i,\ell}(t)$ to the end. More formally, to be compatible with our notation for concatenation of vectors, we write

$$R_i(t) = \begin{bmatrix} [_i]_s \\ [_i]_s \\ p_{i,\ell}(t)_{\ell \in N_i;\, 0 \le t < t_k} \end{bmatrix}; \quad D_i(t) = R_i(t)\, g: \quad (17)$$

Since we set the notation only for the concatenation of vectors, to concatenate matrices $_i$ and $_i$, we first split them into their rows and then concatenate the rows.

B. Formal Statement of Privacy Results

In this section, we formally state the privacy results using the linear-algebraic representations obtained in Section VI-A. As the first step, we seek to spot the iterations at which node i receives a message that contains new information. To do so, we define a_s as the s-th row of $R_i(t)$ for some $t \ge s$. We then run a basis-pursuit procedure over $\{a_s\}_{s=0}^{1}$, as described in Section II. Suppose $B(\{a_s\}_{s=0}^{1})$ is the basis obtained by the basis-pursuit scheme. It is straightforward to confirm that the elements $[_i]_s$ and $[_i]_s$, for all s, are all linearly independent and lie in this basis. Denote the rest of the elements of the basis by $p_{j_1}(t_1), \ldots, p_{j_k}(t_k)$, where $j_1, \ldots, j_k \in N_i$ and $0 \le t_1 \le \cdots \le t_k$. More formally,

$$\begin{aligned} &B(\{a_s\}_{s=0}^{1}) = \\ &\{[_i]_s : s \in [\ ]\} \cup \{[_i]_s : s \in [\ ]\} \cup \{p_{j_1}(t_1), \ldots, p_{j_k}(t_k)\}: \end{aligned} \quad (18)$$

We want to show that the elements of $B(\{a_s\}_{s=0}^{1})$ are all the data points at node i that matter and the rest of data points are redundant. To this aim, consider the following definition.

Definition 7: Define $((j_1, t_1), \ldots, (j_k, t_k))$ in (18) to be the informative sequence of node i. Moreover, for a given integer $0 \le t \le 1$, define $q(t)$ to be the largest $r \in \{1, \ldots, k\}$ such that $t_r \le t$.

Next, let D_i^r be the accumulated data that considers the data points corresponding to the informative sequence at node i, i.e., the messages coming from neighbors $j_1, \ldots, j_r$ at times $t_1, \ldots, t_r$ for some $r \in \{1, \ldots, k\}$. More formally,

$$\begin{aligned} D_i^r = \{u_i\} \cup \{p_{i,\ell}\}_{\ell \in N_i \setminus \{m_i\}} \cup \{p_{i,\ell}\}_{\ell \in N_i} (&v_{j_1}(t_1), \ldots, v_{j_r}(t_r)): \end{aligned} \quad (19)$$

As a next step, we seek to formally show that the information content of $D_i(t)$ equals that of $D_i^{q(t)}$, with $q(t)$ defined in Definition 7. This claim is proved in the following statement.

Theorem 8: Consider Algorithm 1 with $n \ge 2$ under Assumption 2. Moreover, consider $q(t)$ from Definition 7. Suppose $0 \le t \le 1$ is given and let $r = q(t)$. Having D_i^r defined in (19), for any $j \ne i$, we have $I(D_i(t); u_j) = I(D_i^r; u_j)$.

Proof Sketch: We consider the rows of $D_i(t)$ as a sequence of vectors. We then run a basis-pursuit procedure described in Section II that leads to the construction of D_i^r. As a result, each row of $D_i(t)$ can be written as a linear combination of the rows of D_i^r and thus it can be easily shown that the information content of $D_i(t)$ lies in D_i^r. A detailed proof is provided in Section B-B. ∎

Theorem 8 implies that $I(D_i(1); u_j) = I(D_i^k; u_j)$, suggesting that all but finitely many messages are redundant. The following proposition upperbounds the largest informative time instance t_k.

Proposition 9: Consider Algorithm 1 under Assumption 2 and let $((j_1, t_1), \ldots, (j_k, t_k))$ be the informative sequence at node i. If $_W$ is the minimal polynomial of matrix W, then

$$t_k \le \deg(_W) - 1 \le n - 1: \quad (20)$$

Proof Sketch: On one hand, Equation (13) shows that the coefficients in (12) can be obtained in terms of the elements of $[W^t]_i$. On the other hand, having $= \deg(_W)$, one can write W^t for t in terms of smaller powers of W. This means the rows of $D_i(t)$ for t can be written in terms of previous rows of $D_i(t)$ and no information is added for t. Hence, for the last informative instance t_k, we get $t_k \le 1$. Further, by Cayley-Hamilton theorem from linear algebra, we have $\le n$. A detailed proof is provided in Section B-C. ∎

Proposition 9 states that no more information about private values will be exchanged after n rounds of consensus and from then all the messages throughout the network are redundant. Hence, up to $t = n$, it can be determined whether or not an adversary is able to recover a private value. It is also interesting to know how many consensus rounds a (curious) node i must wait to hear about u_j for the first time. The following proposition is dedicated to this result.

Proposition 10: Consider Algorithm 1 under Assumption 2 and let $((j_1, t_1), \ldots, (j_k, t_k))$ be the informative sequence at node i from Definition 7. Further, suppose the underlying

graph is connected and for the consensus matrix, W, we have $W \succeq 0$, and $W \in \mathcal{A}_G$. Let $t = \tau_j^{(i)}$ be the minimum t such that $\Psi_j^{(i)}(t)$ defined in (13) is not zero. Then $\tau_j^{(i)} = d_G(i,j)$ and $r(G) \geq \mathrm{ecc}_G(i) \geq t_k$.

Proof Sketch: In the proof, we use the construction of the D_i^r and the fact that each element of W^t can be written as a sum of products of the elements of W over all walks of length t in the graph. The result verifies the intuition that in order to make sure that all the information content of the graph has reached out to node i, we must wait at least until the time when the information of the farthest nodes to i (i.e., nodes whose distance to i is $\mathrm{ecc}_G(i)$) has arrived to node i. A detailed proof is provided in Section B-D. ∎

Analogous to (16), we can define the data collection on non-redundant data based on the concept of the informative sequence at one node and its representation in terms of g in a matrix-vector multiplication format. To this aim, first consider the informative sequence $((j_1,t_1),\dots,(j_k,t_k))$ at node i and for $r \geq 1$, define

$$R_i^r = [i]_s g_s \quad [i]_s g_s \quad p_{j_\ell}(t_\ell)\ _{1\leq \ell\leq r,s}; \tag{21}$$

$$D_i^r = R_i^r g;$$

and note that R_i^r is a full row-rank matrix. Equation (21) represents all the informative data points at node i in terms of the model's randomness sources, i.e., in terms of the elements of g. In this representation, each column of R_i^r corresponds to the coefficients of one randomness source for all data points. As we show later, the difference between the rank of R_i^r before and after removing the column corresponding to a private value determines the privacy-preserving properties of the network. Towards this argument, let $R_{i,\setminus j}^r$ be the matrix obtained by removing the column corresponding to u_j. We use $\setminus j$ to emphasize that this column is removed. Two cases are possible in this situation:

$$\text{Case 1: } \mathrm{rank}\ R_{i,\setminus j}^r = \mathrm{rank}\,(R_i^r); \tag{22}$$

$$\text{Case 2: } \mathrm{rank}\ R_{i,\setminus j}^r = \mathrm{rank}\,(R_i^r) - 1. \tag{23}$$

We will show that under Case 2, node i can deterministically recover u_j while under Case 1, this is not possible. We start by presenting the following lemma stating that under Case 2, the privacy fails.

Lemma 11: Consider Algorithm 1 with $n \geq 2$ under Assumption 2. Moreover, consider $r = q(t)$ from Definition 7 and suppose (23) holds for some $j \neq i$. Then node i can fully recover u_j using D_i^r.

Proof Sketch: Using the facts from linear algebra, we explicitly obtain u_j in terms of the elements of D_i^r. A detailed proof is provided in Section B-E. ∎

Lemma 11 shows that under Case 2, node i can deterministically recover u_j, meaning that u_j is a deterministic function of $D_i(t)$. Since we assumed all the random variables in our model are continuous random variables, this leads to $\Psi_i^{(j)}(t) = 1$. Next, in Theorem 12, we obtain a closed-form expression for $\Psi_i^{(j)}(t)$ under Case 1, which implies that $\Psi_i^{(j)}(t_i)$ is finite under Case 1. This implies that u_j is not a deterministic function of $D_i(t)$ and thus node i cannot fully recover u_j using $D_i(t)$. Thus, Case 2 holds if and only if node i can deterministically recover u_j. To analyse Case 1, we need to define $S_{\setminus j}$ in the following, which is the covariance matrix of g when the element u_j is removed:

$$S_{\setminus j} = \mathrm{diag}\ \ {}^2{}_{i\in[n]\setminus\{j\}}\ {}_{k'}{}^{2(\beta_k)}{}_{k\in\mathcal{S}}\ : \tag{24}$$

Note that the parameters in (24) are previously defined in Definition 3. The following theorem computes $\Psi_i^{(j)}(t)$ under Case 1, for a Gaussian model.

Theorem 12: Consider Algorithm 1 with $n \geq 2$ under Assumption 2. Suppose $0 \leq t \leq 1$ is given and let $r = q(t)$ as defined in Definition 7. Moreover, suppose Equation (22) holds and assume that all random variables u_s, $s \in [n]$, and ${}_{k'}$, $(k,\ell) \in \mathcal{S}$, have Gaussian distributions. Then, for $j \neq i$

$$\Psi_i^{(j)}(t) = \frac{1}{2}\log\left(1 + {}^2 a_j^{\top}\ {}^{-1} a\right); \tag{25}$$

where $a = [R_i^r]_j$, and $= R_{i,\setminus j}^r S_{\setminus j} R_{i,\setminus j}^{r\top}$.

Proof Sketch: Considering $X = u_j$, we show that the information content of D_i^r about X lies in the variables of the form $X + a_k Y_k$ where Y_k's are independent. We then compute the mutual information of X and the concatenation of such variables. A detailed proof is provided in Section B-F. ∎

Note that for $(k,\ell) \in \mathcal{S}$, the quantity ${}^2{}_k$, i.e., the variance of the generated noise term ${}_{k'}$ lies in the diagonal of $S_{\setminus j}$. As a result, $\Psi_i^{(j)}(t)$ is a decreasing function of ${}^2{}_k$. Since we are free to choose ${}^2{}_k$, we set them large enough to decrease $\Psi_i^{(j)}(t)$ close to its minimum. However, the cost of increasing ${}^2{}_k$ appears in the convergence time. This result is formalised in Theorem 16 and the effect of increasing ${}^2{}_k$ on $\Psi_i^{(j)}(t_i)$ is also evaluated in the simulations (see Section A).

As the next step, to avoid Case 2, we investigate under what conditions Case 1 and Case 2 hold. To this aim, the notion of a generalized leaf is introduced.

Definition 13 (Generalized Leaf): Consider graph $G = (V,E)$ with distinct vertices $i,j \in V$ where $\{i,j\}$ may or may not be in E. Then, G has a generalized leaf with head j and tail i if either $\mathcal{N}_j = \{i\}$ or for every $s \in \mathcal{N}_j \setminus \{i\}$, we have $\deg_G(s) = 2$ and $s \in \mathcal{N}_i$.

A generalized leaf is a generalization of a leaf (i.e., a node with degree 1). Figure 1 illustrates all possible forms of a generalized leaf with head j and tail i. For a generalized leaf of head j and tail i, node i can recover u_j in the first iteration of the consensus. To see this, suppose $s \in \mathcal{N}_j$. Node s receives ${}_{is}$ and ${}_{js}$ in the preparation phase and initializes its consensus by ${}_{is} + {}_{js}$. Node i knows ${}_{is}$ and has also received ${}_{is} + {}_{js}$ from node s in the first iteration of the consensus, thus it obtains ${}_{js}$. Since node i is adjacent to all neighbors of j, node i recovers ${}_{js}$ for all $s \in \mathcal{N}_j$. Summing these reveals u_j to node i. Next theorem guarantees that this is the only troubling case.

Theorem 14: Consider Algorithm 1 over a connected graph G. Then (22) holds for every distinct pair $i,j \in [n]$ and all $r \geq 1$ if G does not contain any generalized leaf.

Proof Sketch: While the non-existence of a generalised leaf is enforced, one can discuss about the elements of R_i^r and uses a subtle case by case proof to show that in all cases, the column of R_i^r corresponding to u_j can be written as a linear

combination of other columns and thus it can be removed from R_i^r without rank reduction. Hence, (22) holds. A detailed proof is provided in Section B-G. ■

VII. Convergence Analysis

Next, we study the convergence properties of Algorithm 1. We first formalize the notion of convergence time based on the waiting time required to achieve an ε-accurate estimation of the convergence limit. We denote this quantity by t_ε and define it as follows.

Definition 15: For $\varepsilon > 0$ and $v(t)$ in (11), let t_ε be the minimum time t such that $\|v(t) - u\|_2 \le \varepsilon$.

Note that the convergence rate (5) is conceptually different from the convergence time defined in Definition 15. While the rate represents the "slope" of a convergence, the convergence time refers to the time when the iterations arrive to the vicinity of the solution. This for example, implies that a larger distance between the initial values and the final solution leads to a larger convergence time under the same convergence rate. Having this definition, the convergence results of this paper are summarized in the following theorem.

Theorem 16: Consider Algorithm 1 (PPAC) under Assumption 2. Further, suppose $W\mathbf{1}_n = W^\top \mathbf{1}_n = \mathbf{1}_n$ and $\rho(W - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) < 1$. Then, we have:

1) In the limit, PPAC converges to the exact solution, i.e., $\lim_{t\to\infty} v(t) = u$.
2) The convergence rate in part (i) is the same as the convergence rate of an ordinary (non-private) consensus, meaning that (5) holds for PPAC too.
3) The expected convergence time for achieving an ε-accurate solution is

$$\mathbb{E}[t_\varepsilon] = O\left(\log \frac{1 + \alpha_{\max} + \beta_{\max}}{\varepsilon}\right); \quad (26)$$

where $\alpha_{\max}$ and $\beta_{\max}^2$ are from Definition 3.

Proof Sketch: We follow simple algebraic computations after writing the initial values $v_i(0)$ in terms of the randomness sources of the model. A detailed proof is provided in Section B-H. ■

Note that the conditions $W\mathbf{1}_n = W^\top \mathbf{1}_n = \mathbf{1}_n$ and $\rho(W - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) < 1$ are essential for the fundamental convergence properties of the classical averaging consensus, as mentioned in Section II. Further, we refer to Section A for simulation results on convergence.

Remark 17: The trade-off between privacy and the convergence time in our model can be explained as follows. In Section VI, we mentioned that increasing α_k^2 for $(k,\ell) \in \mathcal{S}$ (or similarly $\beta_{\max}^2$) decreases the privacy leakage $\mathcal{P}_i^{(j)}(t)$ due to Equation (25), while it increases the convergence time due to Equation (26). Hence, the quantities α_k^2 for $(k,\ell) \in \mathcal{S}$ can be seen as the tuning parameters that adjust the position in the trade-off between privacy and the convergence time.

VIII. Conclusion

In this work, we introduced a novel private consensus averaging algorithm and analyzed its privacy from an information-theoretic perspective. We specifically used the mutual information function between all the information available at the adversary node and the initial value of the victim to measure how private the value of the victim is with respect to the attacker. Our results show that the convergence rate of our proposed method is similar to the convergence rate of a classic non-private consensus method, for any level of privacy. More importantly, any level of accuracy can be achieved via our proposed method. Our results also show that there exists a trade-off between the level of information-theoretic privacy and convergence time.

Appendix A
Simulations

In this section, we provide simulations as a visualization to theoretical results of this paper. Consider a private consensus problem with the underlying graph $\mathcal{G}$ consisting of $n = 6$ nodes as shown in Figure 2. The vector of private values is denoted by $u = [u_1; \dots; u_6]^\top$, where each component is independently generated from the normal distribution with mean 0 and standard deviation $\sigma_0 = 10$, i.e., $u_i \sim \mathcal{N}(0; \sigma_0^2) = \mathcal{N}(0; 100)$ for all $i \in [6]$. The specific realization of this example was $u = [2.30; 4.40; \dots; 6.17; 2.75; 6.01; 0.92]^\top$ with average $\bar{u} = 1.7$. We then ran Algorithm 1 (PPAC) over this model. For each node i, we chose m_i uniformly at random in $\mathcal{N}_i$ and the realization for this example is $(m_1; \dots; m_6) = (2; 4; 4; 3; 4; 1)$. All the pure noises are independently generated from normal distribution with mean 0 and standard deviation σ_N, i.e., $\eta_{ij} \sim \mathcal{N}(0; \sigma_N^2)$ for all $(i,j) \in \mathcal{S}$. In all cases, we use the following consensus matrix: $W = I_n - 1/d_{\max}(D_{\mathcal{G}} - A_{\mathcal{G}})$, where I_n is the identity matrix, $D_{\mathcal{G}} = \mathrm{diag}(\deg(1); \dots; \deg(n))$ and $d_{\max} = \max_{i\in[n]} \deg(i)$.

First, one might be interested in observing a realization of the pure noises and the corresponding dynamics (convergence) of nodes' messages for such a realization. For this purpose, we set $\sigma_N = 15$. Note that PPAC generates new initial values for the consensus different from u but with the same average. Under the aforementioned realization, the new initial values were $v(0) = [v_1(0); \dots; v_6(0)]^\top = [12.57; 16.49; 20.11; 20.78; 23.08; 28.70]^\top$. Figure 3 shows the convergence of $v_i(t)$ for all nodes in terms of the iteration number (time) t. As it can be seen, all nodes converge to the same value, which is the true average $\bar{u} = 1.7$.

The second plot of interest answers the following question: While the initial values are kept fixed, how does the total convergence error over time behave when we increase the variance of pure noises σ_k^2 generated by Algorithm 1? To this aim, we

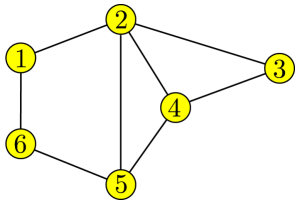

Fig. 2. The graph $\mathcal{G}$.

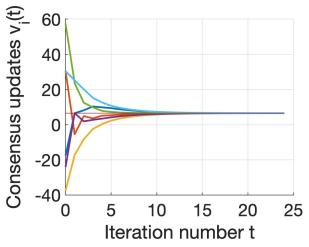

Fig. 3. One realization of the convergence of $v_i(t)$ in terms of t for all nodes using $\sigma_N = 15$.

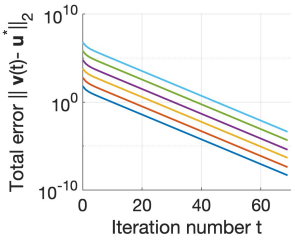

Fig. 4. The total convergence error $\|v(t) - u\|_2$ in terms of iteration number t for a wide range of generated noises' standard deviation σ_N, for values $\sigma_N = 15, 150, 1500, 15000, 150000, 1500000$.

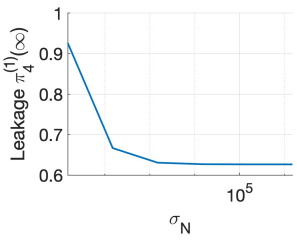

Fig. 5. The leakage of node 1 at node 4, i.e., $\pi_4^{(1)}(1)$ in terms of generated noises' standard deviation σ_N for values $\sigma_N = 15, 150, 1500$, i.e., $15000, 150000, 1500000$ on a logarithmic scale.

plot the total convergence error, i.e., $\|v(t) - u\|_2$ in terms of time t for different values of σ_N. We consider a wide range of values, that is $\sigma_N = 15; 150; 1500; 15000; 150000; 1500000$. Recall that the generated noises $_{ij}; i \in [n]; j = m_i$ are realizations from the normal distribution with mean zero and variance ^2_N and thus the quantity $\|v(t) - u\|_2$ is affected by particular realization of the values $_{ij}$. To release the plots from this randomness, we generated 100 realizations of $_{ij}$ and averaged $\|v(t) - u\|_2$ for each t over these realizations. Figure 4 shows the resulted plots.[2] As it can be seen, Figure 4 agrees with Theorem 16 and shows that increasing the variance of the generated noises $_{ij}; (i; j) \in S$ affects the convergence time t of achieving a given error only by a logarithmic factor. Note that the y-axis is in logarithmic scale.

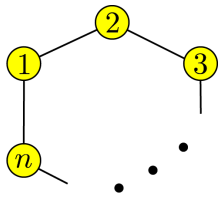

Fig. 6. The graph $G = C_n$.

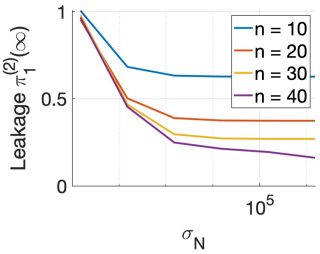

Fig. 7. The leakage of node 2 at node 1 in $G = C_n$, i.e., $\pi_1^{(2)}(1)$ in terms of generated noises' standard deviation σ_N for values $\sigma_N = 15, 150, 1500, 15000, 150000, 1500000$ on a logarithmic scale for values $n = 10; 20; 30; 40$.

As the next step, we run experiments to visualize our privacy results. Note that graph G in Figure 2 does not contain any generalized leaf that are illustrated in Figure 1. More generally, it is interesting to know that any graph over $n \geq 5$ nodes that has a cycle of length n cannot contain a generalized leaf and is therefore a private structure. Hence, due to Theorem 6, the privacy of every node is preserved at every other node in this example. Having this, one might be interested in computing the leakage measure $_i^{(j)}(t)$ defined by Definition 4. This quantity can be computed based on Theorem 12. We are interested in considering all the data that one node can collect over all times through Algorithm 1. Therefore, we consider $_i^{(j)}(1)$, i.e., the maximum leakage of node j at node i. Here, we consider node $i = 4$ is curious about $j = 1$. Recall that in this experiment, we let all the generated noises have a common variance ^2_N. Due to (25), we know that increasing ^2_N decreases $_i^{(j)}(1) = I(D(_i^1); u_j)$ through a logarithmic function. This is plotted in Figure 5 for $_1^{(4)}(1)$. Note that the x-axis is in logarithmic scale. Few important items are worth mentioning about this figure which are discussed in the following:

First, as it can be observed, there is a ceiling on the achievable amount of privacy. This is due to the fact that some amount of information will unavoidably be transmitted by the construction of consensus model. For instance, the average of private values to which all nodes will converge, contains some amount of information about each private value. Second, since the scale is logarithmic, choosing smaller values of σ_N can provide almost the same privacy level that extremely large values can but smaller values of σ_N provide better convergence time as discussed in Figure 4. Hence, one might choose σ_N to be smallest value for which the plot in Figure 5 is almost flat. $\sigma_N = 1500$ seems a good option.

Following the above discussion, we know that the consensus problem inherently reveals some amount of information. For instance, the limiting value u which is achievable by all nodes partially reveal information about each u_i. We know that part of the information is leaked due to achieving the exact average u. This can be computed as $I(u; u_j) = 0.5 \log(1+1/(n-1))$ in the case where u_i s are Gaussian. This expression also suggests that when n is large, less information will be naturally be revealed by u. One might ask if PPAC also provides a smaller leakage for a larger n and thus higher levels of privacy are achievable by PPAC as n gets larger. It turns out that the answer is a yes. To numerically evaluate this notion, we consider the underlying graph to be a cycle of n nodes $G = C_n$ (see Figure 6) and we suppose one node is curious about its neighbor. For the sake of clarity choose node 1 as the curious node (attacker) and node 2 as the victim. Any two neighbors can be chosen due to the symmetry. Figure 7 shows a similar plot as in Figure 5 for $G = C_n$, evaluating $_1^{(2)}(1)$ in terms of σ_N when the total number of nodes n is increasing, i.e., $n = 10; 20; 30; 40$. As it can be seen in Figure 7, larger networks provide lower levels of privacy leakage.

APPENDIX B
PROOFS

A. Further Notations

For a matrix A, $|A| = |\det(A)|$ and for a set A, $|A|$ is the number of elements of A.

For a matrix A, $A \succeq 0$ means that A is a positive definitive matrix.

For (column) vectors $a_1; \ldots; a_s$, $(a_1; \ldots; a_s)$ denotes the matrix with these columns.

For $a \in \mathbb{R}$, $\lceil a \rceil$ denotes the smallest integer greater than or equal to a.

$X \perp\!\!\!\perp Y : X$ and Y are statistically independent.

For a multidimensional random variable X: $_X = \mathrm{Cov}(X)$ is the covariance matrix of X.

For a one-dimensional random variable X: $\sigma_X^2 = Var(X)$ denotes the variance of X.
The indicator function with condition C is denoted by 1_C. Therefore, $1_C = 1$ if C holds and $1_C = 0$ otherwise.

B. Theorem 8

We first formally prove the fundamental property of a basis-pursuit procedure. Then we express a lemma stating that redundant data can be ignored when computing mutual information. Equivalently, if there is a spanning set that generates all other arguments of a mutual information function, considering this spanning set is enough and other elements can be ignored. The proof is based on the fact that the basis resulted from a basis-pursuit procedure is an example of such spanning sets. We start by formally proving fundamental properties of a basis-pursuit procedure.

Lemma 18: Consider $\{a_i\}_{i=0}^{\infty}$ where $a_i \in \mathbb{R}^n$ with $a_0 = 0$ and let $B = B(\{a_i\}_{i=0}^{\infty})$ be obtained from a basis pursuit procedure. Then, $|B| \le n$ and the elements of B are linearly independent. Let $B = \{a_{t_1}, \dots, a_{t_k}\}$ where $0 = t_1 < t_2 < \dots < t_k$. Then for every $i \ge 0$, letting r be the largest number such that $t_r \le i$, we have

$$a_i \in \text{span}(\{a_{t_1}, \dots, a_{t_r}\}). \tag{27}$$

Proof: By construction of $B()$, any finite subset of B is linearly independent. Thus, $|B| \le \dim(\mathbb{R}^n) = n$. Moreover, if $t_r = i$, (27) trivially holds. Therefore, suppose $t_r < i$ and $a_i \notin \text{span}(\{a_{t_1}, \dots, a_{t_r}\})$, then by construction of $B()$, we have $a_i \in B$ and thus $i = t_{r'}$ for some $r' > r$ which contradicts the definition of r. ∎

As the next step, we prove that a spanning subset of a dataset contains all the information of that dataset.

Lemma 19: Suppose Y is an n-dimensional and X is an arbitrary random variable. Further, suppose $a_1, \dots, a_r \in \mathbb{R}^n$ are constant and for $s \ge r$, let $a_1, \dots, a_r \in \text{span}(a_1, \dots, a_s)$. Then

$$I(a_1^\top Y, \dots, a_r^\top Y; X) = I(a_1^\top Y, \dots, a_s^\top Y; X). \tag{28}$$

Proof: If $s = r$ the statement is obvious. Suppose $s < r$ and note that since $a_1, \dots, a_r \in \text{span}(a_1, \dots, a_s)$, for every $i \in \{s+1, \dots, r\}$ there exits $\alpha_{i1}, \dots, \alpha_{is} \in \mathbb{R}$ such that $a_i = \alpha_{i1}a_1 + \dots + \alpha_{is}a_s$. Let α denote the matrix with elements α_{ij} and write

$$(a_{s+1}, \dots, a_r) = (a_1, \dots, a_s)\alpha^\top. \tag{29}$$

Hence,

$$\begin{aligned} (a_{s+1}^\top Y, \dots, a_r^\top Y) &= Y^\top (a_{s+1}, \dots, a_r) \\ &= Y^\top (a_1, \dots, a_s)\alpha^\top \\ &= (Y^\top a_1, \dots, Y^\top a_s)\alpha^\top \\ &= f(a_1^\top Y, \dots, a_s^\top Y). \end{aligned} \tag{30}$$

Let $Z = (a_1, \dots, a_s)$. From (30), $a_{s+1}^\top Y, \dots, a_r^\top Y = f(Z)$, where $f()$ is a deterministic function. This results in

$$\begin{aligned} I(a_1^\top Y, \dots, a_r^\top Y; X) &= I(a_1^\top Y, \dots, a_s^\top Y, a_{s+1}^\top Y, \dots, a_r^\top Y; X) \\ &= I(Z, f(Z); X) \\ &= I(Z; X) \\ &= I(a_1^\top Y, \dots, a_s^\top Y; X). \end{aligned} \tag{31}$$

Proof of Theorem 8: Consider $R_i(t)$ from (16). Then a_s is defined to be the s-th row of $R_i(1)$ or more rigorously, the s-th row of $R_i(t)$ for some $t \ge s$. Moreover, consider the related basis from (18). When finding $B(\{a_s\}_{s=0}^{\infty})$, the basis-pursuit procedure adds to the basis all the rows of matrices α_i and β_i. This holds because one can confirm that in the concatenation of these two matrices, i.e.,

$$\begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}, \tag{32}$$

each column has at most 1 non-zero value (the non-zero values are either 1 or -1) and each row has at least one non-zero value. The basis pursuit then continues by evaluating $p_{i'}(t)$ for $i' \in N_i$ and $t \ge 0$ and chooses $p_{j_1}(t_1), \dots, p_{j_k}(t_k)$, where $((j_1, t_1), \dots, (j_k, t_k))$ is the informative sequence of node i (see Definition 7). Define

$$\begin{aligned} A_r = \{&[i]_s : 1 \le s \le \deg_G(i)\} \\ &\cup \{[i]_s : 1 \le s \le \deg_G(i)\} \\ &\cup [p_{j_1}(t_1), \dots, p_{j_r}(t_r)]. \end{aligned} \tag{33}$$

Using the notation a_s, $D_i(t)$ and D_i^r (for every $r \ge k$), can be written as

$$\begin{aligned} D_i(t) &= a_s^\top g_{s \in \{0, \dots, t+2\deg_G(i)\}}, \\ D_i^r &= a_s^\top g_{a_s \in A_r}. \end{aligned} \tag{34}$$

Due to Lemma 19, for $0 \le s \le t + 2\deg_G(i)$ and $r = q(t)$,

$$a_s \in \text{span}(A_r). \tag{35}$$

Having (34) and (35), the result follows by applying Lemma 19 in which $Y = g$ and $X = u_j$. ∎

C. Proposition 9

Proof: Let $\nu = \deg(\mu_W)$ and denote the coefficients of the minimal polynomial $\mu_W()$ by

$$\mu_W(X) = X^\nu + \alpha_{\nu-1}X^{\nu-1} + \dots + \alpha_0 I. \tag{36}$$

Having $\mu_W(W) = 0$, W^ν can be written as a linear combination of $W^0, \dots, W^{\nu-1}$. Using induction, this also holds for any W^t when $t \ge \nu$. Therefore, for every integer $t \ge 0$, there exists $\tilde{\alpha}_0, \dots, \tilde{\alpha}_{\nu-1}$ such that

$$W^t = \sum_{s=0}^{\nu-1} \tilde{\alpha}_s W^s. \tag{37}$$

Having (13) and (37), we conclude that for any $t \ge 0$ and every $q, j, k, i' \in [n]$ with $j = q$ and $(k, i') \in S$:

$$\gamma_{ji}^q(t) = \sum_{s=0}^{\nu-1} \tilde{\alpha}_s \gamma_{ji}^q(s), \qquad \gamma_{ki'}^q(t) = \sum_{s=0}^{\nu-1} \tilde{\alpha}_s \gamma_{ki'}^q(s). \tag{38}$$

Having the definition of $p_q(t)$ in (14), for every $q \in [n]$ and $t \geq 0$, (38) results in

$$p_q(t) = \sum_{s=0}^{\infty} \tilde{a}_s p_q(s). \tag{39}$$

Note that (39) holds for every $q \in [n]$, particularly when $q \in N_i$. Considering $q \in N_i$, (39) leads to the fact that for very $t \geq 0$ and $\ell \in N_i$,

$$p_\ell(t) \in \text{span}(\{p_q(\tau) : q \in N_i, 0 \leq \tau \leq t-1\}). \tag{40}$$

Hence, $t_k - 1 = \deg(\psi) - 1$. Moreover, due to Cayley-Hamilton theorem, for any $n \times n$ square matrix Ψ, we have $\deg(\psi) \leq n$. ∎

D. Proposition 10

Proof: The statement has two parts and the proof will be enumerated accordingly.

1) For the adjacency matrix A_G of a simple graph G, $[A_G^t]_{ij}$ has a combinatorial meaning. It is equal to the number of walks of length k between the nodes i and j. For a consensus matrix $W \neq A_G$, W can be obtained from A_G by replacing the 1s and (possibly) the diagonal elements of A with some non-negative numbers. Similar to $[A_G^t]_{ij}$, $[W^t]_{ij}$ can also be represented combinatorially. To this aim, define $\Gamma_t(i,j)$ to be the set of walks of length t between (i,j), that is

$$\Gamma_t(i,j) = \{(s_0,\dots,s_t) \in [n]^{t+1} \mid s_0 = i, s_t = j, \forall q \in [t] : \{s_q, s_{q-1}\} \in E\}. \tag{41}$$

Based on the definition of matrix multiplication, we can write

$$[W^t]_{ij} = \sum_{\substack{(s_0,\dots,s_t) \in \\ \Gamma_t(i,j)}} \prod_{q=0}^{t-1} [W]_{s_q s_{q+1}}. \tag{42}$$

Having (42), if $t < d_G(i,m_j)$, then there is no walk of length t between i and m_j, i.e., $\Gamma_t(i,m_j) = \emptyset$; and $\gamma_{im_j}(t) = [W^t]_{im_j} = 0$. Now for $t = d_G(i,m_j)$, there exists at least one positive term (corresponding to each path of length $d_G(i,m_j)$) in (42) which results in $\gamma_{im_j}(t) = [W^t]_{im_j} > 0$. Thus, $\gamma_{im_j}^{(t)}$ exists and equals $d_G(i,m_j)$. This completes the first part.

2) Suppose $j_0 = \arg\max_{s \in [n]} d_G(i,s)$ which leads to $d_G(i,j_0) = \max_{s \in [n]} d_G(i,s) = \text{ecc}(i)$. If $d_G(i,j_0) \leq 2$, the result is trivial. Suppose $d_G(i,j_0) \geq 3$. It suffices to prove that $t_k \geq d_G(i,j_0) - 2$. Define $i_0 = \arg\min_{s \in N_{j_0}} d_G(s,m_{j_0})$. We claim that $t_k \geq d_G(i_0,m_{j_0})$. The claim then leads to $t_k \geq d_G(i_0,m_{j_0}) \geq d_G(i,j_0) - 2$. The claim can be proved as follows: When we run the basis pursuit over the sequence $\{a_s g_s\}$ to obtain (18), consider the j_0-th coordinate. For example, such a coordinate in $p_\ell(t)$ is $\ell_{j_0}(t)$. Note that $p_{i_0}(t)$ with $t = d_G(i_0,m_{j_0})$ lies in the basis because the j_0-th element of $[\ell_{i_0}]_s$ and $[q_{i_0}]_s$ for all s are totally zero and the first time in the basis pursuit that the j_0-th coordinate is not zero occurs at $t = d_G(i_0,m_{j_0})$ in $p_{i_0}(t)$. Due to this, $p_{i_0}(t)$ does not lie in the span of the latest updated basis and must be added to it. Note that t_k is the time when the final vector will be added to the basis, hence, $t_k \geq d_G(i_0,m_{j_0})$. ∎

E. Lemma 11

Proof: Suppose $g_{\setminus j}$ denotes g after removing the j-th coordinate (which corresponds to u_j). Then, we can rewrite (21) in the following way such that for every valid index s:

$$[D_r^i]_s = [R_r^i]_s g = [R_r^i]_{s\setminus j} u_j + R_{r,j}^i [g]_{s\setminus j}. \tag{43}$$

Note that the notation $[\,]_s$ denotes a row vector. Under Case 2 given in (23), there exists a row s_0 of $R_{i,j}^r$ that is a linear combination of other rows of this matrix, i.e., there exist coefficients $\alpha_s \in \mathbb{R}$ such that

$$R_{r,j}^{i}(s_0) = \sum_{s \neq s_0} \alpha_s R_{r,j}^{i}(s). \tag{44}$$

Multiply both sides of (43) by α_s for all $s \neq s_0$, then sum over all these values and subtract the corresponding equation for $s = s_0$ from it. This leads to:

$$\begin{aligned} 0 &= \Big(\sum_{s \neq s_0} \alpha_s [D_r^i]_s A - [D_r^i]_{s_0}\Big) \\ &= \Big(\sum_{s \neq s_0} \alpha_s [R_r^i]_{s\setminus j} - [R_r^i]_{s_0\setminus j}\Big) A\, u_j \\ &\quad + \Big(\sum_{s \neq s_0} \alpha_s R_{r,j}^{i}(s) - R_{r,j}^{i}(s_0)\Big) [g]_{s_0\setminus j}. \end{aligned} \tag{45}$$

(The first term is $=0$ from (44).)

Note that the first term is definitely not zero because otherwise Case 1 (22) holds rather than Case 2 (23). From (45), u_j can be obtained by the observed data points of node i up to $t = t_r$ as the following:

$$u_j = \frac{\sum_{s \neq s_0} \alpha_s [R_i^r]_{s\setminus j} - [R_i^r]_{s_0\setminus j}}{\sum_{s \neq s_0} \alpha_s [D_i^r]_s - [D_i^r]_{s_0}}. \tag{46}$$

Moreover, note that in (46), R_i^r only depends to the underlying graph G and matrix consensus W both of which are known to all nodes. Hence, u_j leaks before time exceeds t. ∎

F. Theorem 12

For the sake of clarity and ease of computation, we need to state and prove two intermediate lemmas before proving Theorem 12. The first lemma is just a formal computing of a combination of normal random variables.

Lemma 20: Suppose X is an arbitrary multi-dimensional non-degenerate random variable i.e., Σ_X is invertible and $A \in \mathbb{R}^{n \times n}$ is a constant matrix. Then $\mathbb{E}[(X - \mathbb{E}[X])^\top A (X - \mathbb{E}[X])] = \text{tr}(A\Sigma_X)$.

Proof: Letting $Y = \Sigma_X^{-\frac{1}{2}}(X - E[X])$ gives $E[Y] = 0$ and $\Sigma_Y = \Sigma_X^{-\frac{1}{2}} \Sigma_X \Sigma_X^{-\frac{1}{2}} = I_n$. Hence,

$$\begin{aligned} E[(X - E[X])^\top A (X - E[X])] &= E\left[Y^\top \Sigma_X^{\frac{1}{2}} A \Sigma_X^{\frac{1}{2}} Y\right] \\ &= E\left[Y^\top \underbrace{\Sigma_X^{\frac{1}{2}} A \Sigma_X^{\frac{1}{2}}}_{B} Y\right] \\ &= E\left[\sum_{i,j} B_{ij} Y_i Y_j\right] \\ &= \operatorname{tr}(B) = \operatorname{tr}\left(\Sigma_X^{\frac{1}{2}} A \Sigma_X^{\frac{1}{2}}\right) = \operatorname{tr}(A\Sigma_X). \end{aligned} \tag{47}$$

■

Lemma 20 eases the proof of the following lemma that computes the mutual information with specific form of arguments that will appear later in the proof of Theorem 12.

Lemma 21: Suppose $(X, Y_1, \dots, Y_k)$ is a multivariate normal where $X \perp Y = (Y_1, \dots, Y_k)^\top$ and assume Σ_Y is invertible. Then

$$I(X + a_1 Y_1, \dots, X + a_k Y_k; X) = \frac{1}{2} \log\left(1 + \Sigma_X a^\top \Sigma_Y^{-1} a\right),$$

where $a = [a_1, \dots, a_k]^\top$.

Proof: Denote $I = I(X + a_1 Y_1, \dots, X + a_k Y_k; X)$ and $Z = X + a^\top Y = [Z_1, \dots, Z_k]^\top$ and note that $Z_i = X + a_i Y_i$ for $i \in \{1, \dots, k\}$. By definition,

$$I = \int f_{X,Z}(x, z_1, \dots, z_k) \log \frac{f_{X,Z}(x, z_1, \dots, z_k)}{f_X(x) f_Z(z_1, \dots, z_k)} \, dx \, dz_1 \dots dz_k. \tag{48}$$

To obtain the joint distribution of (X, Z), the following transformation is useful:

$$\begin{bmatrix} X \\ Z \end{bmatrix} = J \begin{bmatrix} X \\ Y \end{bmatrix}, \quad J = \begin{bmatrix} 1 & 0 \\ a & I_k \end{bmatrix}, \quad \det(J) = 1, \tag{49}$$

which results in

$$f_{X,Z}(x, z) = f_{X,Y}(x, z - ax) = f_X(x) f_Y(z - ax). \tag{50}$$

By replacing (50) into (48) and cancelling out the term f_X, we get

$$I = \int f_X(x) f_Y(z - ax) \log \frac{f_Y(z - ax)}{f_Z(z)} \, dx \, dz. \tag{51}$$

To compute the terms in (51), we need a quick discussion. Letting $Y \sim \mathcal{N}(\mu_Y, \Sigma_Y)$ and $Z \sim \mathcal{N}(\mu_Z, \Sigma_Z)$, we know that

$$Z = Y + aX, \tag{52}$$

and we can relate Σ_Z and Σ_Y as follows:

$$\begin{aligned} \Sigma_Z &= J \operatorname{Cov}\begin{bmatrix} X \\ Y \end{bmatrix} J^\top \\ &= J \begin{bmatrix} \sigma_X^2 & 0 \\ 0 & \Sigma_Y \end{bmatrix} J^\top = \begin{bmatrix} \sigma_X^2 & \sigma_X^2 a^\top \\ \sigma_X^2 a & \Sigma_Y + \sigma_X^2 a a^\top \end{bmatrix} \\ \Rightarrow \Sigma_Z &= \Sigma_Y + \sigma_X^2 a a^\top. \end{aligned} \tag{53}$$

Since $\Sigma_Y \succ 0$, from (53) we conclude that $\Sigma_Z \succ 0$ and thus Σ_Z is invertible. Therefore, Σ_Y^{-1} and Σ_Z^{-1} exist. Having this, we can write

$$\begin{aligned} \frac{f_Y(z - ax)}{f_Z(z)} = & \\ \frac{|\Sigma_Z|^{\frac{1}{2}}}{|\Sigma_Y|^{\frac{1}{2}}} & \exp\left(\frac{1}{2}(z - ax - \mu_Y)^\top \Sigma_Y^{-1}(z - ax - \mu_Y) \right. \\ & \left. + \frac{1}{2}(z - \mu_Z)^\top \Sigma_Z^{-1}(z - \mu_Z)\right). \end{aligned} \tag{54}$$

Replacing (54) into (51) leads to the computation of (51) in three terms as follows:

$$I = I_1 + I_2 + I_3; \tag{55}$$

where the first term I_1 equals

$$\int f_X(x) f_Y(z - ax) \log \frac{|\Sigma_Z|^{\frac{1}{2}}}{|\Sigma_Y|^{\frac{1}{2}}} dx dz = \frac{1}{2} \log \frac{|\Sigma_Z|}{|\Sigma_Y|}. \tag{56}$$

The second term is

$$\begin{aligned} I_2 &= \frac{1}{2} \int f_X(x) f_Y(z - ax)(z - ax - \mu_Y)^\top \Sigma_Y^{-1}(z - ax - \mu_Y) \, dz \, dx \\ &= \frac{1}{2} \int f_X(x) E_Y\left[(Y - \mu_Y)^\top \Sigma_Y^{-1}(Y - \mu_Y)\right] dx \\ &= \frac{1}{2} \int f_X(x) \operatorname{tr}\left(\Sigma_Y^{-1} \Sigma_Y\right) dx \quad \text{due to Lemma 20} \\ &= \frac{\operatorname{tr}(I_k)}{2} \int f_X(x) \, dx = \frac{k}{2}. \end{aligned} \tag{57}$$

And the third term can be obtained as follows.

$$\begin{aligned} I_3 &= \frac{1}{2} \int f_X(x) f_Y(z - ax)(z - \mu_Z)^\top \Sigma_Z^{-1}(z - \mu_Z) \, dz \, dx \\ &= \frac{1}{2} \int f_X(x) E_Y[(Y + ax - \mu_Z)^\top \Sigma_Z^{-1}(Y + ax - \mu_Z)] dx. \end{aligned} \tag{58}$$

For each given $x \in \mathbb{R}$, we can compute the argument of the integration in (58) as follows.

$$\begin{aligned} & E_Y\left[(Y + ax - \mu_Z)^\top \Sigma_Z^{-1}(Y + ax - \mu_Z)\right] \\ &= E_Y\left[(Y - \mu_Y)^\top \Sigma_Z^{-1}(Y - \mu_Y)\right] && (59) \\ &+ E_Y\left[2(Y - \mu_Y)^\top \Sigma_Z^{-1}(ax + \mu_Y - \mu_Z)\right] && (60) \\ &+ (ax + \mu_Y - \mu_Z)^\top \Sigma_Z^{-1}(ax + \mu_Y - \mu_Z). && (61) \end{aligned}$$

To compute the right-hand side of the latest equation, note that (59) can be obtained using Lemma 20 as follows:

$$\begin{aligned} & E_Y\left[(Y - \mu_Y)^\top \Sigma_Z^{-1}(Y - \mu_Y)\right] \\ &= \operatorname{tr}\left(\Sigma_Z^{-1} \Sigma_Y\right) \\ &= \operatorname{tr}\left(\Sigma_Z^{-1}(\Sigma_Z - \sigma_X^2 a a^\top)\right) \quad \text{due to (53)} \\ &= \operatorname{tr}(I_k) - \sigma_X^2 \operatorname{tr}\left(\Sigma_Z^{-1} a a^\top\right) \\ &= k - \sigma_X^2 a^\top \Sigma_Z^{-1} a. \end{aligned} \tag{62}$$

Having (62) together with the fact that (60) is zero and replacing $Y - Z = aX$ in (61), we can put together the three terms and write

$$\begin{aligned} &E_{Y,Z}\left[(Y+aX-Z)^{\top}\Sigma_Y^{-1}(Y+aX-Z)\right] \\ &= k - \Sigma_X a^{\top}\Sigma_Z^{-1}a + (x-\mu_X)^{\top}a^{\top}\Sigma_Z^{-1}a. \end{aligned} \tag{63}$$

Replacing (63) in (58) leads to

$$\begin{aligned} I_3 &= \frac{1}{2}\left(k - \Sigma_X a^{\top}\Sigma_Z^{-1}a\right) \\ &\quad + \frac{1}{2}a^{\top}\Sigma_Z^{-1}a \int f_X(x)(x-\mu_X)^2\,dx \\ &= \frac{k}{2} - \frac{1}{2}\Sigma_X a^{\top}\Sigma_Z^{-1}a + \frac{1}{2}a^{\top}\Sigma_Z^{-1}a\,\Sigma_X = \frac{k}{2}. \end{aligned} \tag{64}$$

Putting the values of I_1, I_2, and I_3 from (56), (57), and (64) into (55) results in

$$I = I_1 + I_2 + I_3 = \frac{1}{2}\log\frac{|Z|}{|Y|} + \frac{k}{2} - \frac{k}{2} = \frac{1}{2}\log\frac{|Z|}{|Y|}. \tag{65}$$

To simplify further, note that from (53) and the fact that Σ_Y is invertible, we have

$$|Z| = |\Sigma_Y|\left(1 + \Sigma_X a^{\top}\Sigma_Y^{-1}a\right). \tag{66}$$

Therefore,

$$I = \frac{1}{2}\log\frac{|Z|}{|\Sigma_Y|} = \frac{1}{2}\log\left(1 + \Sigma_X a^{\top}\Sigma_Y^{-1}a\right). \tag{67}$$

Finally, the proof of Theorem 12 can be provided as follows using Lemma 21. ∎

Proof of Theorem 12: Suppose g_{-j} denotes g after removing the j-th coordinate (which corresponds to u_j). Then, we can rewrite (21) in the following way such that for every valid index s:

$$[D_i^r]_s = [R_i^r]_s\, g = [R_i^r]_{sj}\, u_j + R_{i;-j,s}^r\, g_{-j}. \tag{68}$$

Note that the notation $[]_s$ denotes a row vector. For ease of notation, suppose s_{max} is the number of rows of D_i^r. We directly apply Lemma 21 by considering $X = u_j$, $Y_s = R_{i;-j,s}^r\, g_{-j}$, $Y = [Y_1;\dots;Y_{s_{max}}]^{\top}$, and $a_s = [R_i^r]_{sj}$ which also gives $a = [a_1;\dots;a_{s_{max}}]^{\top} = [R_i^r]_j$. With these replacements, the conditions of Lemma 21 hold because due to Assumption 2, u_j is independent of all other randomness sources, i.e., $u_j \perp g_{-j}$ which leads to $X \perp Y$ and also we assumed in the statement of Theorem 8 that all randomnesses are Gaussian, hence, $(X;Y)$ is multivariate normal. Further, with this assignment of Y, we need to compute the covariance matrix $\mathrm{Cov}(Y)$ and show that it is invertible. Note that

$$\begin{aligned} Y = R_i^r{}_{;-j}\, g_{-j}) \quad \Sigma_Y &= \mathrm{Cov}(Y) \\ &= R_i^r{}_{;-j}\,\mathrm{Cov}(g_{-j})\,R_i^{r\top}{}_{;-j} \\ &= R_i^r{}_{;-j}\,S_{-j}\,R_i^{r\top}{}_{;-j}; \end{aligned} \tag{69}$$

where S_{-j} is defined in (24). Note that R_i^r is a full row-rank matrix and under Case 1 (i.e., (22)), $R_{i;-j}^r$ is also full row-rank. Hence,

$$\mathrm{rank}(\Sigma_Y) = \mathrm{rank}(S_{-j}) = 2m-1. \tag{70}$$

Since Σ_Y is a $(2m-1)\times(2m-1)$ matrix, (70) means Σ_Y is invertible. So far, we showed that all the conditions of Lemma 21 hold under the above assignments. Therefore, the result also holds and we have

$$I(D_i^r; u_j) = \frac{1}{2}\log\left(1 + \sigma_j^2\, a^{\top}\Sigma_Y^{-1}a\right); \tag{71}$$

Finally, due to Theorem 8, we know that

$$I(D_i(t); u_j) = I(D_i^r; u_j). \tag{72}$$

With (71) and (72) together, we conclude that

$$I(D_i(t); u_j) = \frac{1}{2}\log\left(1 + \sigma_j^2\, a^{\top}\Sigma_Y^{-1}a\right). \tag{73}$$

∎

G. Theorem 14

Proof: Suppose G does not contain any generalized leaf. We want to show that Case 1 (Equation (22)) holds. To do so is, the main idea is that we show that under such a condition, the j-th column (the column corresponding to u_j) in R_i^r, i.e., the column to be removed to obtain $R_{i;-j}^r$ is a linear combination of other columns of R_i^r. Therefore, removing it does not change the rank. To this aim, we need to setup a notation that we only use throughout this proof. Recall from (21) that R_i^r is a concatenation of three matrices, Γ_i, Λ_i, and P_i^r as follows:

$$P_i^r = \begin{bmatrix} p_{j_1}^{\top}(t_1) \\ \vdots \\ p_{j_r}^{\top}(t_r) \end{bmatrix}; \quad R_i^r = \begin{bmatrix} \Gamma_i \\ \Lambda_i \\ P_i^r \end{bmatrix}; \tag{74}$$

Thus, each column of R_i^r consists of three sub-columns in each of these matrices. A set of columns of R_i^r are linearly dependent if and only if the exact same linear combination holds for the sub-columns in each of Γ_i, Λ_i, and P_i. Notation Setup. All the aforementioned matrices, i.e., R_i^r, Γ_i, Λ_i, and P_i^r have $2m$ columns. Each column corresponds to one randomness source in the model that are u_p for $p \in [n]$ and g_{ps} for $(p;s) \in \mathcal{S}$. We use index u_p or g_{ps} to refer to those columns. For example, $[R_i^r]_{g_{ps}}$ denotes the column of R_i^r that corresponds to g_{ps}. To refer to rows, we setup a notation for each sub-matrices Γ_i, Λ_i, and P_i^r. The rows of Γ_i correspond to u_i and γ_{ip} for $(i;p) \in \mathcal{S}$. We also use these as row-indices. For example, $[\Gamma_i]_{\gamma_{ip}}$ equals the coefficient of u_q when we write γ_{ip} in terms of u_p, g_{pq}. This coefficient is zero since γ_{ip} and u_q are independent noises. The rows of Λ_i correspond to $\lambda_{i\ell}$ for $\ell \in \mathcal{N}_i$. Note that here $(\ell;i)$ may or may not lie in $\mathcal{S}$. The quantity $\lambda_{i\ell}$ is what neighbor ℓ sends to i in the preparation phase. It may be a pure noise (when $(\ell;i) \in \mathcal{S}$) or in the form of $u_\ell + \sum_{s\in\mathcal{N}_\ell\cap\mathcal{N}_i} g_{\ell s}$. We also use these $\lambda_{i\ell}$ for $\ell \in \mathcal{N}_i$ as row-indices. Finally, the rows of P_i^r are characterized by pairs $(j_1;t_1);\dots;(j_k;t_k)$, e.g., the row of P_i^r corresponding to $(\ell;t) \in \{(j_1;t_1);\dots;(j_k;t_k)\}$ is $p_\ell^{\top}(t)$ and is denoted by $[P_i^r]_{(\ell;t)}$. When we want to refer to the element u_b in this row, we write $[P_i^r]_{(\ell;t)u_b}$. As we will see in the remaining of the proof, our discussion here holds for each row identically. Hence, we refer to rows by general

pair $(\ell; t)$ and use this pair as a row-index.

Using the notation explained above, we can now state the proof. Recall that node i is the attacker and node j is the victim and we want to show that $[R_i^r]_{u_j}$ can be generated as a linear combination of other columns of R_i^r. Note that the following arguments hold regardless of r which means it holds for all $r \in \{1; \dots; k\}$. Therefore, we omit the superscript r for simplicity. Suppose the connected graph G does not have any generalized leaf. Then j must have a neighbor $s \neq i$. Considering all such neighbors, there exists nodes $b; s$ such that $s \in N_j$, $s \neq i$, $b \in N_s$, and $b \notin \{i; j\}$ because otherwise either the graph is not connected or we have a generalized leaf in the graph. Now, we consider four cases based on whether $s = m_j$ versus $s = m_j$ and $s = m_b$ versus $s = m_b$. The proof in each case are provided in the following. Note that in obtaining the elements of P_i, we are basically using (13). Again, note that the superscript r will be omitted for simplicity and we use $(\ell; t)$ subscript to refer to the rows of P_i where the choice of $\ell \in N_i$ and t does not matter. Moreover, 0 denotes a vector with all zero elements.

1) Case $s = m_j$ and $s = m_b$:

$$\left.\begin{array}{l} [P_i]_{(\ell;t)u_j} = [W^t]_{\ell,m_j} = [W^t]_{\ell,s} \\ [P_i]_{(\ell;t)u_b} = [W^t]_{\ell,m_b} = [W^t]_{\ell,s} \end{array}\right\} \Rightarrow [P_i]_{u_j} = [P_i]_{u_b}; \quad (75)$$

$$[i]_{u_j} = [i]_{u_b} = 0; \quad (76)$$

$$[i]_{u_j} = [i]_{u_b} = 0: \quad (77)$$

Hence,

$$[R_i]_{u_j} = [R_i]_{u_b}: \quad (78)$$

2) Case $s = m_j$ and $s = m_b$:
Note that m_b might be i, j, or any other node other than s. Since $s = m_b$, we have $(b; s) \in S$ meaning that $_{bs}$ is a pure noise. This means the relevant coefficient is computable based on $_{bs}(t)$ in (13).

$$\left.\begin{array}{l} [P_i]_{(\ell;t)u_j} = [W^t]_{\ell,m_j} = [W^t]_{\ell,s} \\ [P_i]_{(\ell;t)u_b} = [W^t]_{\ell,m_b} \\ [P_i]_{(\ell;t)\,{}_{bs}} = [W^t]_{\ell,s} \quad [W^t]_{\ell,m_b} \end{array}\right\} \Rightarrow [P_i]_{u_j} = [P_i]_{u_b} + [P_i]_{\,{}_{bs}}; \quad (79)$$

Moreover,

$$[i]_{u_j} = 0; \quad (80)$$

$$\text{if } b \in N_i: \begin{cases} [i]_{\,{}_{bu_b}} = 1_{m_b=i}; \\ [\,]_{i\,{}_{b\,{}_{bs}}} = 1_{m_b=i}; \\ \text{Other elements of } [i]_{u_b} \\ \text{and } [i]_{\,{}_{bs}} \text{ are zero;} \end{cases} \quad (81)$$

$$\text{if } b \notin N_i: [i]_{u_b} = [i]_{\,{}_{bs}} = 0: \quad (82)$$

Hence, we always get

$$[i]_{u_j} = [i]_{u_b} + [i]_{\,{}_{bs}}: \quad (83)$$

Further,

$$\left.\begin{array}{l} [i]_{u_j} = [i]_{u_b} = [i]_{\,{}_{bs}} = 0 \end{array}\right\} \Rightarrow [i]_{u_j} = [i]_{u_b} + [i]_{\,{}_{bs}}: \quad (84)$$

Putting (79), (83), and (84), we conclude that

$$[R_i]_{u_j} = [R_i]_{u_b} + [R_i]_{\,{}_{bs}}: \quad (85)$$

3) Case $s = m_j$ and $s = m_b$:
Since $s = m_j$, we have $(j; s) \in S$ meaning that $_{js}$ is a pure noise. This means the relevant coefficient is computable based on $_{js}(t)$ in (13).

$$\left.\begin{array}{l} [P_i]_{(\ell;t)u_j} = [W^t]_{\ell,m_j} \\ [P_i]_{(\ell;t)u_b} = [W^t]_{\ell,m_b} = [W^t]_{\ell,s} \\ [P_i]_{(\ell;t)\,{}_{js}} = [W^t]_{\ell,s} \quad [W^t]_{\ell,m_j} \end{array}\right\} \Rightarrow [P_i]_{u_j} = [P_i]_{u_b} \quad [P_i]_{\,{}_{js}}: \quad (86)$$

Now, for $_i$ we have

$$[i]_{u_b} = 0; \quad (87)$$

$$\text{if } j \in N_i: \begin{cases} [i]_{ju_j} = 1_{m_j=i}; \\ [i]_{j\,{}_{js}} = 1_{m_j=i}; \\ \text{Other elements of } [i]_{u_j} \\ \text{and } [i]_{\,{}_{js}} \text{ are zero;} \end{cases} \quad (88)$$

$$\text{if } j \notin N_i: [i]_{u_j} = [i]_{\,{}_{js}} = 0: \quad (89)$$

Hence, we always get

$$[i]_{u_j} = [i]_{u_b} \quad [i]_{\,{}_{js}}: \quad (90)$$

Further,

$$\left.\begin{array}{l} [i]_{u_j} = [i]_{u_b} = [i]_{\,{}_{js}} = 0 \end{array}\right\} \Rightarrow [i]_{u_j} = [i]_{u_b} \quad [i]_{\,{}_{js}}: \quad (91)$$

Putting (86), (90), and (91), we conclude that

$$[R_i]_{u_j} = [R_i]_{u_b} \quad [R_i]_{\,{}_{js}}: \quad (92)$$

4) Case $s = m_j$ and $s = m_b$:
First note that since $s = m_j$ and $s = m_b$, we have $(j; s); (b; s) \in S$ meaning that $_{js}$ and $_{bs}$ are pure noises. This means the relevant coefficients are computable based on $_{js}(t)$ and $_{bs}(t)$ in (13).

$$\left.\begin{array}{l} [P_i]_{(\ell;t)u_j} = [W^t]_{\ell,m_j} \\ [P_i]_{(\ell;t)u_b} = [W^t]_{\ell,m_b} \\ [P_i]_{(\ell;t)\,{}_{js}} = [W^t]_{\ell,s} \quad [W^t]_{\ell,m_j} \\ [P_i]_{(\ell;t)\,{}_{bs}} = [W^t]_{\ell,s} \quad [W^t]_{\ell,m_b} \end{array}\right\} \Rightarrow [P_i]_{u_j} = [P_i]_{u_b} \quad [P_i]_{\,{}_{js}} + [P_i]_{\,{}_{bs}}: \quad (93)$$

Now, for $_i$ we have

$$\text{if } j \in N_i: \begin{cases} [i]_{ju_j} = 1_{m_j=i}; \\ [\,]_{i\,{}_{j\,{}_{js}}} = 1_{m_j=i}; \\ \text{Other elements of } [i]_{u_j} \\ \text{and } [i]_{\,{}_{js}} \text{ are zero;} \end{cases} \quad (94)$$

$$\text{if } j \notin N_i: [i]_{u_j} = [i]_{\,{}_{js}} = 0; \quad (95)$$

$$\text{if } b \in N_i: \begin{cases} [i]_{\,{}_{bu_b}} = 1_{m_b=i}; [i] \\ = {}_{b}1_{m_b=i}; \\ \text{Other elements of } [i]_{u_b} \\ \text{and } [i]_{\,{}_{bs}} \text{ are zero;} \end{cases} \quad (96)$$

$$\text{if } b \notin N_i: [i]_{u_b} = [i]_{\,{}_{bs}} = 0: \quad (97)$$

Hence, we always get

$$[i]_{u_j} = [i]_{u_b} \quad [i]_{js} + [i]_{bs}: \tag{98}$$

Further,

$$[i]_{u_j} = [i]_{u_b} = [i]_{js} = [i]_{bs} = 0$$
$$=) \quad [i]_{u_j} = [i]_{u_b} \quad [i]_{js} + [i]_{bs}: \tag{99}$$

Putting (93), (98), and (99), we conclude that

$$[R_i]_{u_j} = [R_i]_{u_b} \quad [R_i]_{js} + [R_i]_{bs}: \tag{100}$$

The equations (78), (85), (92), and (100) show that in all cases the column $[R_i]_{u_j}$ can be written as a linear combination of other columns of R_i. Hence, removing it does not change the rank, that is, for all $r \in \{1,\dots,k\}$.

$$\text{rank } R_{i_{r;\ j}} = \text{rank}(R_{i_r}): \tag{101}$$

This means Case 1, i.e., (22) holds for all $r \in \{1,\dots,k\}$. ■

H. Theorem 16

We start with a simple lemma that eases the computation of t.

Lemma 22: Let $\{x(t)\}_{t=0}^{1}$ be an arbitrary sequence of vectors over the integer t that converges to x and suppose for every $t \geq 0$

$$\|x(t+1) - x\|_2 \leq \|x(t) - x\|_2; \tag{102}$$

for some real number $\in (0,1)$. Moreover, define the quantity $t = \min\{t \geq 0 \mid \|x(t) - x\|_2 \leq \}$. Then

$$t \leq \frac{1}{\log\frac{1}{}} \log\frac{1 + \|x(0) - x\|_2}{}. \tag{103}$$

Proof: Note that

$$\|x(t) - x\|_2 \leq {}^{t}\|x(0) - x\|_2. \tag{104}$$

When one is not considering $t = 0$, we can let $\tilde{t}$ be the minimum real number t such that the right-hand side of (104) holds, i.e.,

$$\tilde{t} = \min\{t \in \mathbb{R} \mid {}^{t}\|x(0) - x\|_2 \leq \}. \tag{105}$$

Then by taking the logarithm of both sides of ${}^{t}\|x(0) - x\|_2 = $, we have

$$\tilde{t}\log + \log\|x(0) - x\|_2 = \log$$
$$=) \quad \tilde{t} = \frac{1}{\log\frac{1}{}} \log\frac{\|x(0) - x\|_2}{}. \tag{106}$$

Note that small $\|x(0) - x\|_2$ may lead to a negative $\tilde{t}$. Hence, for a minimum non-negative time, we can define

$$t = \min\{t \geq 0 \mid \|x(t) - x\|_2 \leq \}. \tag{107}$$

Having $t \geq 0$ constraint and $\|x(t) - x\|_2$ instead of ${}^{t}\|x(0) - x\|_2$ in the definition of t, we have $t \leq \tilde{t}$ if $\tilde{t} \geq 0$ and $t = 0$ otherwise. This together with (106) gives

$$t \leq \frac{1}{\log\frac{1}{}} \log\frac{1 + \|x(0) - x\|_2}{}. \tag{108}$$

■

Proof of Theorem 16:

1) Define $u = [u_1,\dots,u_n]^\top$ and note that by construction, Algorithm 1 preserves the summation of all values over all iterations. In other words, $1_n^\top v(t) = 1_n^\top u = nu$ for all $t \geq 0$. Due to [8], we know that the conditions $W 1_n = W^\top 1_n = 1_n$ and $\|(W - 1/n 1_n 1_n^\top)\|_2 < 1$ lead to (4). Therefore,

$$\lim_{t\to\infty} v(t) = \lim_{t\to\infty} W^t v(0) = \frac{1}{n} 1_n 1_n^\top v(0) = 1_n^\top u = u:$$

2) Since Algorithm 1 only changes $v(0)$ compared to an ordinary consensus, (5) trivially holds.

3) We use Lemma 22 to compute t for our problem. Note that for $t \geq 0$, we have

$$v(t+1) - u$$
$$= W v(t) - W u - \frac{1}{n} 1_n 1_n^\top v(t) + \frac{1}{n} 1_n 1_n^\top u$$
$$= \left(W - \frac{1}{n} 1_n 1_n^\top\right)(v(t) - u): \tag{109}$$

In obtaining (109), we used the following:

$$W u = W \frac{1}{n} 1_n 1_n^\top u = \frac{1}{n} W 1_n 1_n^\top u$$
$$= \frac{1}{n} 1_n 1_n^\top u = u; \tag{110}$$

and the fact that $1_n^\top v(t) = 1_n^\top u$ for all $t \geq 0$. From (109), we conclude that for all $t \geq 0$

$$\|v(t+1) - u\|_2 \leq \left\|W - \frac{1}{n} 1_n 1_n^\top\right\|_2 \|v(t) - u\|_2: \tag{111}$$

Comparing (111) with (102), by putting

$$= \left\|W - \frac{1}{n} 1_n 1_n^\top\right\|_2: \tag{112}$$

Note that < 1 (because $\|(W - 1/n 1_n 1_n^\top)\|_2 < 1$). Thus, the quantity t can be obtained from (103) as follows:

$$t \leq \frac{1}{\log\frac{1}{}} \log\frac{1 + \|v(0) - u\|_2}{} + 1$$
$$\leq \frac{1}{\log\frac{1}{}} \log\left(\frac{2}{} \|v(0) - u\|_2^2\right) + 1: \tag{113}$$

Here, the quantity $v(0) - u$ is a random variable. We take the expectation of both sides of (113) and apply the Jensen inequality due to the concavity of the logarithm function (Jensen inequality states that for a concave function f and random variable X, $E[f(x)] \leq f(E[X])$).

Hence,

$$E[t] \leq \frac{1}{\log\frac{1}{}} \log\left(\frac{2}{} \left(1 + E\left[\|v(0) - u\|_2^2\right]\right)\right) + 1:$$
$$\tag{114}$$

Next, we compute $E\left[\|v(0) - u\|_2^2\right]$ and replace it into (114). To do so, we start by writing

$$E\left[\|v(0) - u\|_2^2\right] \leq n \max_{i\in[n]} E\left[(v_i(0) - u)^2\right] \tag{115}$$
$$= n \max_{i\in[n]} \left(E[v_i(0) - u]\right)^2 + \text{Var}(v_i(0) - u):$$

To compute the right-hand side of (115), we denote the maximum degree of a node in G by d_{max}, the indicator function by $\mathbb{1}$, and the number of elements of a set A by $|A|$ and continue as follows:

$$\begin{aligned}
&|\mathbb{E}[v_i(0) - u]| \\
&= \Big|\mathbb{E}\Big[\sum_{j\in N_i} w_{ij}\Big(\sum_{j'\in N_j\setminus\{m_j\}} x_{j'}\mathbb{1}_{i=m_j} + \sum_{j'\in N_j\setminus\{m_j\}} x_{j'}\mathbb{1}_{i\neq m_j} - \frac{1}{n}(u_1+\cdots+u_n)\Big)\Big]\Big| \\
&\le |\{j\in N_i : i=m_j\}|\max_{j\in N_i} w_{ij}\, d_{max}\max_{j\in N_i} w_{ij} \\
&\quad + |\{j\in N_i : i=m_j\}|\max_{j\in N_i} w_{ij}\, d_{max}\max_{j\in N_i} w_{ij} + \max_{j\in N_i} w_{ij}\, d_{max}\max_{j\in N_i} w_{ij} + \max_{j\in N_i} w_{ij} \\
&\le d_{max}^2\max_{j\in N_i} w_{ij}^2 + 1\max_{j\in N_i} w_{ij}.
\end{aligned} \tag{116}$$

To compute $\mathrm{Var}(v_i(0) - u)$, we write

$$\begin{aligned}
&\mathrm{Var}(v_i(0) - u) \\
&= \mathrm{Var}\Big[\sum_{j\in N_i} w_{ij}\Big(\sum_{j'\in N_j\setminus\{m_j\}} x_{j'}\mathbb{1}_{i=m_j} + \sum_{j'\in N_j\setminus\{m_j\}} x_{j'}\mathbb{1}_{i\neq m_j} - \frac{1}{n}(u_1+\cdots+u_n)\Big)\Big] \\
&\le |\{j\in N_i : i=m_j\}|\Big(1-\frac{1}{n}\Big)^2\max_{j\in N_i} w_{ij}^2 + (d_{max}-1)\max_{j\in N_i} w_{ij}^2 \\
&\quad + |\{j\in N_i : i=m_j\}|\max_{j\in N_i} w_{ij}^2 + |\{j\in[n] : i=m_j\}|\frac{\max_{j\in N_i} w_{ij}^2}{n^2} \\
&\le |\{j\in N_i : i=m_j\}|\, d_{max}\max_{j\in N_i} w_{ij}^2 + |\{j\in[n] : i=m_j\}|\frac{\max_{j\in N_i} w_{ij}^2}{n^2} \\
&\le d_{max}^2\max_{j\in N_i} w_{ij}^2 + (n-d_{max})\frac{\max_{j\in N_i} w_{ij}^2}{n^2} \\
&\le d_{max}^2\max_{j\in N_i} w_{ij}^2 + 1\max_{j\in N_i} w_{ij}^2
\end{aligned} \tag{117}$$

Putting (115), (116), and (117) together results in

$$\begin{aligned}
&\mathbb{E}\big[\|v(0) - u\|^2\big] \\
&\le n\big(d_{max}^2\max_{j\in N_i} w_{ij}^2 + 1\max_{j\in N_i} w_{ij}^2 + \big(d_{max}\max_{j\in N_i} w_{ij}^2 + 1\max_{j\in N_i} w_{ij}\big)^2\big) \\
&\le n\big(d_{max}^2\max_{j\in N_i} w_{ij}^2 + 1\max_{j\in N_i} w_{ij}^2 + \max_{j\in N_i} w_{ij}^2\big).
\end{aligned} \tag{118}$$

Finally, replace (112) and (118) into (114) to obtain

$$\begin{aligned}
\mathbb{E}[t] &\le \frac{1}{\log\frac{1}{\|W - \frac{1}{n}\mathbb{1}\mathbb{1}^T\|_2}}\log\Big(\frac{1+n\big(d_{max}^2\max_{j\in N_i} w_{ij}^2 + 1\max_{j\in N_i} w_{ij}^2 + \max_{j\in N_i} w_{ij}^2\big)}{\varepsilon^2}\Big) + 1.
\end{aligned} \tag{119}$$

Acknowledgment

The work of M. Fereydounian and H. Hassani is funded by DCIST, NSF CPS-1837253, and NSF CIF-1943064 and NSF CAREER award CIF-1943064, and Air Force Office of Scientific Research Young Investigator Program (AFOSR-YIP) under award FA9550-20-1-0111. The research of A. Mokhtari is supported in part by NSF Grants 2019844 and 2112471, ARO Grant W911NF2110226, the Machine Learning Lab (MLL) at UT Austin, and the Wireless Networking and Communications Group (WNCG) Industrial Affiliates Program. The work of R. Pedarsani is funded by NSF Grant 2003035 and 2236483.

References

[1] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in Proceedings of the 44th IEEE Conference on Decision and Control. IEEE, 2005, pp. 6698–6703.

[2] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005. IEEE, 2005, pp. 63–70.

[3] J. He, L. Duan, F. Hou, P. Cheng, and J. Chen, "Multiperiod scheduling for wireless sensor networks: A distributed consensus approach," IEEE Transactions on Signal Processing, vol. 63, no. 7, pp. 1651–1663, 2015.

[4] C. Zhao, J. He, P. Cheng, and J. Chen, "Consensus-based energy management in smart grid with transmission losses and directed communication," IEEE Transactions on smart grid, vol. 8, no. 5, pp. 2049–2061, 2016.

[5] J. Chen, K. Hu, Q. Wang, Y. Sun, Z. Shi, and S. He, "Narrowband internet of things: Implementations and applications," IEEE Internet of Things Journal, vol. 4, no. 6, pp. 2309–2314, 2017.

[6] R. Bekkerman, M. Bilenko, and J. Langford, Scaling up Machine Learning: Parallel and Distributed Approaches. Cambridge University Press, 2011.

[7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Artificial intelligence and statistics. PMLR, 2017, pp. 1273–1282.

[8] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," Systems & Control Letters, vol. 53, no. 1, pp. 65–78, 2004.

[9] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," IEEE Transactions on automatic control, vol. 49, no. 9, pp. 1520–1533, 2004.

[10] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," IEEE Transactions on automatic control, vol. 48, no. 6, pp. 988–1001, 2003.

[11] Y. Mo and R. M. Murray, "Privacy preserving average consensus," IEEE Transactions on Automatic Control, vol. 62, no. 2, pp. 753–765, 2016.

[12] M. H. DeGroot, "Reaching a consensus," Journal of the American Statistical Association, vol. 69, no. 345, pp. 118–121, 1974.

[13] J. Cortés, G. E. Dullerud, S. Han, J. Le Ny, S. Mitra, and G. J. Pappas, "Differential privacy in control and network systems," in 2016 IEEE 55th Conference on Decision and Control (CDC). IEEE, 2016, pp. 4252–4272.

[14] C. Dwork, A. Roth et al., "The algorithmic foundations of differential privacy." Found. Trends Theor. Comput. Sci., vol. 9, no. 3-4, pp. 211–407, 2014.

[15] 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, 2012, pp. 81–90.

[16] 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.

[17] V. Feldman and D. Xiao, "Sample complexity bounds on differentially private learning via communication complexity," in Conference on Learning Theory. PMLR, 2014, pp. 1000–1019.

[18] K. Chaudhuri and S. A. V interbo, "A stability-based validation procedure for differentially private machine learning," in Advances in Neural Information Processing Systems 26, December 2013.

[19] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in Proceedings of the 55th Annual Symposium on the Foundations of Computer Science (FOCS '14), October 18–21 2014.

[20] G. Kamath, J. Li, V. Singhal, and J. Ullman, "Privately learning high-dimensional distributions," in Conference on Learning Theory. PMLR, 2019, pp. 1853–1902.

[21] M. Heikkila; E. Lagerspetz, S. Kaski, K. Shimizu, S. Tarkoma, and A. Honkela, "Differentially private bayesian learning on distributed data," Advances in neural information processing systems, vol. 30, 2017.
[22] J. He, L. Cai, C. Zhao, P. Cheng, and X. Guan, "Privacy-preserving average consensus: Privacy analysis and algorithm design," IEEE Transactions on Signal and Information Processing over Networks, vol. 5, no. 1, pp. 127–138, 2018.
[23] J. Le Ny and G. J. Pappas, "Differentially private filtering," IEEE Transactions on Automatic Control, vol. 59, no. 2, pp. 341–354, 2013.
[24] S. Gade and N. H. Vaidya, "Private learning on networks," arXiv preprint arXiv:1612.05236, 2016.
[25] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25. Springer, 2006, pp. 486–503.
[26] C. Altafini, "A system-theoretic framework for privacy preservation in continuous-time multiagent dynamics," Automatica, vol. 122, p. 109253, 2020.
[27] Y. Wang, "Privacy-preserving average consensus via state decomposition," IEEE Transactions on Automatic Control, vol. 64, no. 11, pp. 4711–4716, 2019.
[28] S. S. Kia, J. Cortés, and S. Martinez, "Dynamic average consensus under limited control authority and privacy requirements," International Journal of Robust and Nonlinear Control, vol. 25, no. 13, pp. 1941–1966, 2015.
[29] S. Pequito, S. Kar, S. Sundaram, and A. P. Aguiar, "Design of communication networks for distributed computation with privacy guarantees," in 53rd IEEE Conference on Decision and Control. IEEE, 2014, pp. 1370–1376.
[30] N. Gupta, J. Katz, and N. Chopra, "Information-theoretic privacy in distributed average consensus," arXiv preprint arXiv:1809.01794, 2018.
[31] Y. Xiong and Z. Li, "Privacy-preserved average consensus algorithms with edge-based additive perturbations," Automatica, vol. 140, p. 110223, 2022.
[32] C. N. Hadjicostis, "Privary preserving distributed average consensus via homomorphic encryption," in 2018 IEEE Conference on Decision and Control (CDC). IEEE, 2018, pp. 1258–1263.
[33] T. Yin, Y. Lv, and W. Yu, "Accurate privacy preserving average consensus," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 4, pp. 690–694, 2019.
[34] M. Ruan, H. Gao, and Y. Wang, "Secure and privacy-preserving consensus," IEEE Transactions on Automatic Control, vol. 64, no. 10, pp. 4035–4049, 2019.
[35] P. Cuff and L. Yu, "Differential privacy as a mutual information constraint," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 43–54.
[36] E. A. Abbe, A. E. Khandani, and A. W. Lo, "Privacy-preserving methods for sharing financial risk exposures," American Economic Review, vol. 102, no. 3, pp. 65–70, 2012.
[37] I. Issa, A. B. Wagner, and S. Kamath, "An operational approach to information leakage," IEEE Transactions on Information Theory, vol. 66, no. 3, pp. 1625–1657, 2019.
[38] S. Roman, Advanced Linear Algebra, ser. Graduate Texts in Mathematics. Springer New York, 2013.

Mohammad Fereydounian (Graduate Student Member, IEEE) received a B.Sc. degree in pure mathematics as well as a B.Sc. degree in electrical engineering in 2014, and following that, an M.Sc. degree in pure mathematics in 2016, all from Sharif University of Technology, Tehran, Iran. Subsequently, in 2021, he earned an A.M. degree in statistics from the Wharton School, University of Pennsylvania, where he is currently pursuing his PhD degree in the electrical and systems engineering department. His research interests include information and coding theory, optimization, and machine learning.

Aryan Mokhtari received the B.Sc. degree in elec- trical engineering from the Sharif University of Technology, Tehran, Iran, in 2011, the M.Sc. and Ph.D. degrees in electrical and systems engineer- ing from the University of Pennsylvania in 2014 and 2017, respectively, and the A.M. degree in statistics from Wharton School in 2017. He is an Assistant Professor with the Electrical and Computer Engineering Department, the University of Texas at Austin (UT Austin) where he holds the Fellow of Texas Instruments/Kilby. Before joining UT Austin, he was a Postdoctoral Associate with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology from January 2018 to July 2019. Before that, he was a Research Fellow with the Simons Institute for the Theory of Computing, University of California at Berkeley, Berkeley, for the program on Bridging Continuous and Discrete Optimization, from August to December 2017. Prior to that, he was a graduate student with the University of Pennsylvania. During his graduate study, he was a Research Intern with the Big-Data Machine Learning Group, Yahoo!, Sunnyvale, CA, USA, from June to August 2016. His research interests include the areas of optimization, machine learning, and artificial intelligence. His current research focuses on the theory and applications of convex and non-convex optimization in large-scale machine learning and data science problems. He has received several awards and fellowships, including the Army Research Office Early Career Program Award, the Simons-Berkeley Research Fellowship, and the Penns Joseph and Rosaline Wolf Award for Best Doctoral Dissertation in electrical and systems engineering.

Ramtin Pedarsani (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 2009, the M.Sc. degree in communication systems from the Swiss Federal Institute of Technology, Lausanne, Switzerland, in 2011, and the Ph.D. degree from the University of California at Berkeley, Berkeley, in 2015. He is an Associate Professor with the ECE Department, University of California at Santa Barbara, Santa Barbara. His research interests include machine learning, information and coding theory, networks, and transportation systems. He is a recipient of the Communications Society and Information Theory Society Joint Paper Award in 2020, the best paper award in the IEEE International Conference on Communications in 2014, and the NSF CRII award in 2017.

Hamed Hassani (Member, IEEE) received the Ph.D. degree in computer and communication sciences from the EPFL, Lausanne. He is currently an Assistant Professor with the Electrical and Systems Engineering Department as well as the Computer and Information Systems Department, and the Statis- tics Department, University of Pennsylvania. Prior to that, he was a Research Fellow at the Simons Institute for the Theory of Computing (UC Berkeley) affiliated with the Program of Foundations of Machine Learning, and a Post-Doctoral Researcher with the Institute of Machine Learning, ETH Zrich. He was a recipient of the 2014 IEEE Information Theory Society Thomas M. Cover Dissertation Award, the 2015 IEEE International Symposium on Information Theory Student Paper Award, the 2017 Simons-Berkeley Fellowship, the 2018 NSF-CRII Research Initiative Award, the 2020 Air Force Office of Scientific Research (AFOSR) Young Investigator Award, the 2020 National Science Foundation (NSF) CAREER Award, and the 2020 Intel Rising Star Award. He has recently been selected as a Distinguished Lecturer of the IEEE Information Theory Society during 2022-2023.