Information-Theoretic Privacy in Distributed Average Consensus

Nirupam Gupta \(^a\) Jonathan Katz \(^b\) Nikhil Chopra \(^a\)

\(^a\)Department of Mechanical Engineering, University of Maryland, College Park, MD 20742, USA (e-mails: nirupam@umd.edu, nchopra@umd.edu).

\(^b\)Department of Computer Science, University of Maryland, College Park, MD 20742, USA (e-mail: jkatz@cs.umd.edu).

Abstract

We propose an asynchronous distributed average consensus algorithm that guarantees information-theoretic privacy of honest agents’ inputs against colluding semi-honest (passively adversarial) agents, as long as the set of colluding semi-honest agents is not a vertex cut in the underlying communication network. This implies that a network with \((t+1)\)-connectivity guarantees information-theoretic privacy of honest agents’ inputs against any \(t\) colluding semi-honest agents. The proposed protocol is formed by composing a distributed privacy mechanism we provide with any (non-private) distributed average consensus algorithm.

Key words: Privacy; distributed average consensus.

1 Introduction

Algorithms for distributed average consensus allow agents in a peer-to-peer network to compute the average of all the agents’ inputs \([15, 21, 26]\). Some well-known applications of distributed average consensus include sensor fusion in multi-sensor networks \([22, 20]\), distributed computation of support vector machines \([8]\), and solving economic-dispatch problems in smart grids \([27]\). Distributed average consensus can also be used in peer-to-peer networks for voting or monitoring.

Typical distributed average consensus algorithms require agents to share their inputs with their neighbors \([26, 1, 15, 21, 3, 4, 24]\). This infringes agents’ privacy, which is undesirable as certain agents in the network might not be trustworthy.

In this paper, we show how to construct distributed average consensus protocols that ensure privacy of honest agents’ inputs in the presence of passive (aka semi-honest\(^1\)) adversarial agents in the network. Our notion of privacy, adopted from the field of secure multi-party computation (MPC) \([2, 6, 11]\), is a strong one: it ensures that colluding adversarial agents learn nothing, in an information-theoretic sense, about the collective inputs of the honest agents beyond learning the average value of the honest agents’ inputs. (The latter is unavoidable, as it can be deduced from the global average whose computation is the purpose of running the consensus algorithm.)

While privacy can often be achieved by relying on generic completeness theorems for (information-theoretic) secure multi-party computation \([2, 6, 11]\), those results do not immediately apply to our setting because they assume a complete network with a dedicated communication channel between each pair of agents. In contrast, we are interested algorithms that can be used regardless of the underlying network topology. There are few results in that setting. Garay et al. \([9]\) studied secure computation in incomplete networks, and showed that arbitrary functions can be computed with information-theoretic privacy against \(t\) colluding semi-honest agents so long as the communication network is \((t+1)\)-connected. However, their work relies on protocols for secure message transmission \([7]\) that emulate pairwise channels between execution of the protocol to infer something about the inputs of other agents.

---

\(^1\) Semi-honest agents are assumed to follow the prescribed protocol, but may try to use the information learned during
every pair of agents over an incomplete network. In addition to incurring a significant cost in terms of round- and message-complexity, relying on secure message transmission also requires the agents to have complete knowledge of the network topology. The protocols we show here add minimal cost to existing distributed average consensus protocols, and only require agents to be aware of their neighbors. It is nevertheless interesting to observe that our results also require \((t + 1)\)-connectivity in order to guarantee privacy against arbitrary subsets of \(t\) colluding agents.

There have been several previous proposals \([14, 18, 19]\) for achieving privacy by having agents add local noise to intermediate values computed at various points during execution of a distributed average consensus protocol. This added noise induces a loss in accuracy \([19, 5]\); i.e., the agents are only able to compute an approximation to the true average (rather than the exact average), where there is an inherent trade-off between privacy and the achievable accuracy. Our scheme computes the average exactly.

Other works have aimed at ensuring privacy in distributed average consensus without sacrificing accuracy \([17, 23, 12, 25, 10]\). Some of these \([17, 23]\) do not achieve very good privacy guarantees; in particular, they do not ensure privacy in the presence of collusion of multiple agents in the network. The scheme of Gupta et al. \([12]\) assumes a centralized, trusted authority that distributes information to all agents each time they wish to run the consensus algorithm. We note that while protocols based on homomorphic encryption \([25, 16]\) can achieve the strong privacy guarantees we do here, they are orders of magnitude less efficient than our proposed approach since they rely on public-key techniques.

We note also that some of the above solutions \([14, 17, 18]\) require synchronous execution of the agents, whereas our solution can be run asynchronously.

1.1 Summary of Our Contribution

We show a general approach for achieving privacy in distributed average consensus protocols. Our approach involves two phases:

1. In the first phase, each agent sends correlated random values to its neighbors and then computes a new, “effective input” based on its original input and the random values it received from its neighbors.

2. In the second phase, the agents run an arbitrary distributed average consensus protocol (e.g., flooding or any other protocol from the literature \([15, 21, 26, 4]\)) using their effective inputs computed in the first phase rather than their original inputs.

We show that the above, two-step process correctly computes the average value of the agents’ original inputs as long as the average consensus protocol used in the second phase is correct. This follows from the fact that the first phase is designed to ensure that the average of the agents’ effective inputs is equal to the average of their original inputs. We also show that privacy holds in our approach—in a formal sense and under certain conditions, as discussed below—regardless of the average consensus protocol used in the second phase. We prove this by showing that privacy holds even if all the effective inputs of the honest agents are revealed to the colluding semi-honest parties.

Our notion of privacy is a strong one, adopted from the literature on secure multi-party computation \([11]\). Formally, the guarantee is that the entire view of the colluding agents throughout the execution of our protocol can be simulated by those agents given (1) their original inputs and (2) the average of the original inputs of the honest agents (or, equivalently, the average of the original inputs of all the agents in the network). This holds regardless of the true inputs of the honest agents. As a consequence, this means that the colluding adversarial agents learn nothing (or very little, in a statistical sense) about the collective inputs of the honest agents from an execution of the protocol other than the average of the honest agents’ inputs, and this holds regardless of any prior knowledge the adversarial agents may have about the inputs of (some of) the honest agents, or the distribution of those inputs. We prove that our protocol satisfies this notion of privacy as long as the set of colluding semi-honest agents does not constitute a vertex cut of the network topology.

Our privacy-preserving protocol was previously described in the conference version of our paper \([13]\). However, the privacy definition we use here is stronger, and this version includes full proofs for our privacy claims.

2 Notation and Preliminaries

We let \(\mathbb{Z}\) denote the set of integers, and let \(\mathbb{Z}_q\) denote the set of integers \(\{0, \ldots, q - 1\}\). For a finite set \(S\), we let \(|S|\) denote its cardinality; for an integer \(q\), we let \(|q|\) denote its absolute value. If \(x\) is an \(n\)-dimensional vector, then \(x_i\) denotes its \(i\)th element and \(\sum_i x_i\) simply denotes the sum of all its elements. We use \(1_n\) to denote the \(n\)-dimensional vector all of whose elements is 1.

We consider communication networks represented by simple, undirected graphs. That is, the communication links in a network of \(n\) agents is modeled via a graph \(G = (\mathcal{V}, \mathcal{E})\) where the nodes \(\mathcal{V} \triangleq \{1, \ldots, n\}\) denote the agents, and there is an edge \(\{i, j\} \in \mathcal{E}\) iff there is a direct communication channel between agents \(i\) and \(j\). We let \(\mathcal{N}_i\) denote the set of neighbors of an agent \(i \in \mathcal{V}\), i.e.,
j \in N_i \text{ if and only if } \{i, j\} \in \mathcal{E}. \text{ (Note that } i \not\in N_i \text{ since } \mathcal{G} \text{ is a simple graph.)}

We say two agents \(i, j\) are connected if there is a path from \(i\) to \(j\); since we consider undirected graphs, this notion is symmetric. We let \(p_{ij}\) denote an arbitrary path between \(i\) and \(j\), when one exists. A graph \(\mathcal{G}\) is connected if every distinct pair of nodes is connected; note that a single-node graph is connected.

**Definition 1** (Vertex cut) A set of nodes \(S \subset \mathcal{V}\) is a vertex cut of a graph \(\mathcal{G} = (\mathcal{V}, \mathcal{E})\) if removing the nodes in \(S\) (and the edges incident to those nodes) renders the resulting graph unconnected. In this case, we say that \(S\) cuts \(\mathcal{V} \setminus S\).

A graph is \(k\)-connected if the smallest vertex cut of the graph contains \(k\) nodes.

Let \(\mathcal{G} = (\mathcal{V}, \mathcal{E})\) be a graph. The subgraph induced by \(\mathcal{V}' \subset \mathcal{V}\) is the graph \(\mathcal{G}' = (\mathcal{V}', \mathcal{E}')\) where \(\mathcal{E}' \subset \mathcal{E}\) is the set of edges entirely within \(\mathcal{V}'\) (i.e., \(\mathcal{E}' = \{\{i, j\} \in \mathcal{E} \mid i, j \in \mathcal{V}'\}\)). We say a graph \(\mathcal{G} = (\mathcal{V}, \mathcal{E})\) has \(c\) connected components if its vertex set \(\mathcal{V}\) can be partitioned into disjoint sets \(\mathcal{V}_1, \ldots, \mathcal{V}_c\) such that (1) \(\mathcal{G}\) has no edges between \(\mathcal{V}_i\) and \(\mathcal{V}_j\) for \(i \neq j\) and (2) for all \(i\), the subgraph induced by \(\mathcal{V}_i\) is connected. Clearly, if \(\mathcal{G}\) is connected then it has one connected component.

For a graph \(\mathcal{G} = (\mathcal{V}, \mathcal{E})\), we define its incidence matrix \(\nabla \in \{-1, 0, 1\}^{\mathcal{V} \times |\mathcal{E}|}\) (see [10]) to be the matrix with \(|\mathcal{V}|\) rows and \(|\mathcal{E}|\) columns in which

\[
\nabla_{i, e} = \begin{cases} 
1 & \text{if } e = \{i, j\} \text{ and } i < j \\
-1 & \text{if } e = \{i, j\} \text{ and } i > j \\
0 & \text{otherwise.}
\end{cases}
\]

Note that \(\mathbf{1}_n^T \nabla = 0\). We use \(\nabla_{*, e}\) to denote the column of \(\nabla\) corresponding to the edge \(e \in \mathcal{E}\).

We rely on the following result [10] Theorem 8.3.1]:

**Lemma 1** Let \(\mathcal{G}\) be an \(n\)-node graph with incidence matrix \(\nabla\). Then \(\text{rank}(\nabla) = n - c\), where \(c\) is the number of connected components of \(\mathcal{G}\).

### 2.1 Problem Formulation

As noted above, we consider a network of \(n\) agents where the communication network between agents is represented by an undirected, simple, connected graph \(\mathcal{G} = (\mathcal{V}, \mathcal{E})\); that is, agents \(i\) and \(j\) have a direct communication link between them iff \(\{i, j\} \in \mathcal{E}\). The communication channel between two nodes is assumed to be both private and authentic; equivalently, in our adversarial model we do not consider an adversary who can eavesdrop on communications between honest agents, or tamper with their communication. (Alternately, private and authentic communication can be ensured using standard cryptographic techniques.)

Each agent \(i\) holds a (private) input \(s_i\). By discretizing and scaling appropriately, we may assume without loss of generality that \(s_i \in \{0, \ldots, q - 1\}\) for some publicly known, integer bound \(q > 1\). We let \(\bar{s} = [s_1, \ldots, s_n]^T\).

A distributed average consensus algorithm is an interactive protocol allowing the agents in the network to each compute the average of the agents’ inputs, i.e., after execution of the protocol each agent outputs the value \(\bar{s} = \frac{1}{n} \sum_i s_i\).

We are interested in distributed average consensus algorithms that ensure privacy against an attacker who controls some fraction of the agents in the network. We let \(\mathcal{C} \subset \mathcal{V}\) denote the set of corrupted agents controlled by the attacker, and let \(\mathcal{H} = \mathcal{V} \setminus \mathcal{C}\) denote the remaining honest agents. As stated earlier, we assume the attacker is semi-honest and thus runs the prescribed protocol. Privacy requires that the entire view of the attacker—i.e., the inputs of the corrupted agents as well as their internal states and all the protocol messages they received throughout execution of the protocol—does not leak (significant) information about the original inputs of the honest agents. Note that, by definition, the attacker learns \(\bar{s}\) (assuming it corrupts at least one agent) from which it can compute the sum of the inputs of the honest agents, and so our privacy definition requires that the attacker does not learn anything more than this.

Before giving our formal definition of privacy, we introduce some notation. Let \(s_C\) denote a set of inputs held by the corrupted agents, and \(s_H\) a set of inputs held by the honest agents. Fixing some protocol, we let \(\text{View}_C(s)\) be a random variable denoting the view of the corrupted agents in an execution of the protocol when the agents all begin holding inputs \(s\). Then:

**Definition 2** A distributed average consensus protocol is (perfectly) \(C\)-private if for all \(s, s'\) such that \(s_C = s'_C\) and \(\sum_{i \in \mathcal{H}} s_i = \sum_{i \in \mathcal{H}} s'_i\), the distributions of \(\text{View}_C(s)\) and \(\text{View}_C(s')\) are identical.

We remark that this definition makes sense even if \(|\mathcal{C}| = n - 1\), though in that case the definition is vacuous since \(s_H = \sum_{i \in \mathcal{H}} s_i\) and so revealing the sum of the honest agents’ inputs reveals the (single) honest agent’s input!

An alternate, perhaps more natural, way to define privacy is to require that for any distribution \(S\) (known to the attacker) over the honest agents’ inputs, the distribution of the honest agents’ inputs conditioned on the attacker’s view is identical to the distribution of the honest agents’ inputs conditioned on their sum. It is not hard to see that this is equivalent to the above definition.
3 Private Distributed Average Consensus

As described previously, our protocol has a two-phase structure. In the first phase, each agent $i$ computes an “effective input” $\tilde{s}_i$ based on its original input $s_i$ and random values it sends to its neighbors; this is done while ensuring that $\sum_{j} \tilde{s}_i \mod p$ is equal to $\sum_i s_i$ for some publicly known integer $p$ (see below). In the second phase, the agents use any (correct) distributed average consensus protocol $\Pi$ to compute $\sum_{i} \tilde{s}_i$, reduce that result modulo $p$, and then divide by $n$. This clearly gives the correct average $\frac{1}{n} \sum_{i} \tilde{s}_i$, and thus all that remains is to analyze privacy.

It may at first seem strange that we can prove privacy of our algorithm without knowing anything about the distributed average consensus protocol $\Pi$ used in the second phase of our protocol. We do this by making a “worst-case” assumption about $\Pi$, namely, that it simply reveals all the agents’ inputs to all the agents! Such an algorithm is, of course, not at all private; for our purposes, however, this does not immediately violate privacy because $\Pi$ is run on the agents’ inputs rather than their true inputs $s_i$.

From now on, then, we let the view of the attacker consist of the initial inputs of the corrupted agents, their internal states and all the protocol messages they receive throughout execution of the first phase of our protocol, and the vector $\tilde{s} = [\tilde{s}_1, \ldots, \tilde{s}_n]^T$ of all agents’ effective inputs at the end of the first phase. Our definition of privacy (cf. Definition 2) remains unchanged.

Before continuing with an analysis of privacy, we describe our first-phase algorithm. Let $p$ be an integer such that $p > n \cdot (q - 1) \geq \sum_i s_i$. The first phase of our protocol proceeds as follows:

1. Each agent $i \in \mathcal{V}$ chooses independent, uniform values $r_{ij} \in \mathbb{Z}_p$ for all $j \in \mathcal{N}_i$, and sends $r_{ij}$ to agent $j$.
2. Each agent $i \in \mathcal{V}$ computes a mask
   \[ a_i = \sum_{j \in \mathcal{N}_i} (r_{ji} - r_{ij}) \mod p, \] (1)
   where $a_i \in \mathbb{Z}_p$.
3. Each agent $i \in \mathcal{V}$ computes effective input
   \[ \tilde{s}_i = (s_i + a_i) \mod p. \] (2)

Note that
\[ \sum_i \tilde{s}_i = \sum_i s_i + \sum_i a_i \mod p. \]

Moreover,
\[ \sum_i a_i = \sum_i \sum_{j \in \mathcal{N}_i} (r_{ji} - r_{ij}) \mod p = 0 \mod p, \]

since $\mathcal{G}$ is undirected. Thus, $\sum_i \tilde{s}_i = \sum_i s_i \mod p$. Since $\sum_i s_i < p$ by choice of $p$, this implies that $\sum_i \tilde{s}_i \mod p$ is equal to $\sum_i s_i$ over the integers, and hence correctness of our overall algorithm (i.e., including the second phase) follows.

Our algorithm is illustrated by example in Section 4.

3.1 Privacy Analysis

We show here that $C$-privacy holds as long as $C$ is not a vertex cut of $\mathcal{G}$.

For an edge $e = \{i, j\}$ in the graph with $i < j$, define
\[ b_{e} = r_{ji} - r_{ij} \mod p. \]

Let $b = [b_{e_1}, \ldots]$ be the collection of such values for all the edges in $\mathcal{G}$. If we let $a = [a_1, \ldots, a_n]^T$ denote the masks used by the agents, then we have
\[ a = \nabla \cdot b \mod p. \]

Since the $r_{ij}$ are uniform and independent in $\mathbb{Z}_p$, it is easy to see that the values $\{b_{e}\}_{e \in \mathcal{E}}$ are uniform and independent in $\mathbb{Z}_p$ as well.\footnote{If $x$ and $y$ are two independent random variables in $\mathbb{Z}_p$ with at least one of them being uniformly distributed (in $\mathbb{Z}_q$), then $z = x + y \mod p$ is uniformly distributed in $\mathbb{Z}_p$.} Thus, $a$ is uniformly distributed over the vectors in the span (over $\mathbb{Z}_p$) of the columns of $\nabla$, which we denote by $L(\nabla)$. The following is easy to prove using the fact that rank($\nabla$) = $n - 1$ when $\mathcal{G}$ is connected (cf. Lemma 1):

**Lemma 2** If $\mathcal{G}$ is connected then $a$ is uniformly distributed over $\mathbb{Z}_p^{n-1}$ possibilities.

(A full proof of Lemma 2 is given in Section 7.1.) Since $1_n^T \cdot \nabla = 0$, the above implies that if $\mathcal{G}$ is connected then $a$ is uniformly distributed over $\mathbb{Z}_p^n$ subject to the constraint that $\sum_i a_i = 0 \mod p$.

Since $\tilde{s}_i = s_i + a_i \mod p$, we have

**Lemma 3** If $\mathcal{G}$ is connected, then the effective inputs $\tilde{s}$ are uniformly distributed in $\mathbb{Z}_p^n$ subject to the constraint that $\sum_i \tilde{s}_i = \sum_i s_i \mod p$. 
The proof of Lemma 3 is given in Section 7.2.

The above implies privacy for the case when $C = \emptyset$, i.e., when there are no corrupted agents. In that case, the view of the adversary consists only of the effective inputs $\tilde{s}_i$, and Lemma 3 shows that the distribution of those values depends only on the sum of the agents’ true inputs. Below, we extend this line of argument to the case of nonempty $C$.

Fix some set $C$ of corrupted agents, and recall that $H = \mathcal{V} \setminus C$. Let $E_C$ denote the set of edges incident to $C$, and let $E_H = \mathcal{E} \setminus E_C$ be the edges incident only to honest agents. Note that now the attacker’s view contains (information that allows it to compute) $\{b_{ij}\}_{i \in E_C}$ in addition to the honest agents’ effective inputs $\{\tilde{s}_i\}_{i \in H}$.

The key observation enabling a proof of privacy is that the values $\{b_{ij}\}_{i \in E_C}$ are uniform and independent in $\mathbb{Z}_p$ even conditioned on the values $\{b_{ij}\}_{i \in E_C}$. Thus, as long as $C$ is not a vertex cut of $\mathcal{G}$, an argument as earlier implies that the masks $\{a_i\}_{i \in H}$ are uniformly distributed in $\mathbb{Z}_p^{\mathcal{H}}$ subject to $\sum_{i \in E_C} b_{ij} = - \sum_{i \in C} a_i \mod p$ (even conditioned on knowledge of the values $\{b_{ij}\}_{i \in E_C}$), and hence the effective inputs $\{\tilde{s}_i\}_{i \in H}$ are uniformly distributed in $\mathbb{Z}_p^{\mathcal{H}}$ subject to

$$
\sum_{i \in H} \tilde{s}_i = \sum_{i \in V} s_i - \sum_{i \in C} \tilde{s}_i \mod p
$$

(again, even conditioned on knowledge of the $\{b_{ij}\}_{i \in E_C}$).

Since the right-hand side of the above equation can be computed from the effective inputs of the corrupted agents, the $\{b_{ij}\}_{i \in E_C}$, and the sum of the honest agents’ inputs, this implies:

**Theorem 4** If $C$ is not a vertex cut of $\mathcal{G}$, then our proposed distributed average consensus protocol is perfectly $C$-private.

A formal proof of this theorem is given in Section 7.3.

As a corollary, we have

**Corollary 1** If $\mathcal{G}$ is $(t+1)$-connected, then for any $C$ with $|C| \leq t$ our proposed distributed average consensus protocol is perfectly $C$-private.

### 4 Illustration

To demonstrate our proposed distributed average consensus protocol we consider a simple network of 3 agents with $\mathcal{V} = \{1, 2, 3\}$ and $\mathcal{E} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$, as shown in Fig. 1. Let the values of $q$ and $p$ be 10 and 30, respectively.

Consider an instance where $s_1 = 4$, $s_2 = 7$ and $s_3 = 3$.

![Diagram](image_url)

**Fig. 1.** Arrows (in blue) show the flow of information over an edge.

In first phase, the agents execute the following steps

1. As shown in Fig. 1, all pair of adjacent agents $i$ and $j$ exchange the respective values of $r_{ij}$ and $r_{ji}$ (chosen independently and uniformly in $\mathbb{Z}_p$) with each other. Consider a particular instance where 

   - $r_{12} = 14$, $r_{21} = 11$, $r_{23} = 17$, $r_{32} = 5$, $r_{31} = 3$, $r_{13} = 8$

2. The agents compute their respective masks,

   - $a_1 = ((r_{21} - r_{12}) + (r_{31} - r_{13})) \mod p$
   - $= ((11 - 14) + (3 - 8)) \mod 30 = 22$
   - $a_2 = ((r_{12} - r_{21}) + (r_{32} - r_{23})) \mod p$
   - $= ((14 - 11) + (5 - 17)) \mod 30 = 21$
   - $a_3 = ((r_{13} - r_{31}) + (r_{23} - r_{32})) \mod p$
   - $= ((8 - 3) + (17 - 5)) \mod 30 = 17$

(one can verify that $(a_1 + a_2 + a_3) \mod 30 = 0$)

3. The agents compute their respective effective inputs,

   - $\tilde{s}_1 = (s_1 + a_1) \mod p = (4 + 22) \mod 30 = 26$
   - $\tilde{s}_2 = (s_2 + a_2) \mod p = (7 + 21) \mod 30 = 28$
   - $\tilde{s}_3 = (s_3 + a_3) \mod p = (3 + 17) \mod 30 = 20$

After the first phase, each agent uses a (non-private) distributed average consensus protocol $\Pi$ (an instance shown in Fig. 1) in the second phase to compute $(1/3) \sum_i \tilde{s}_i$ (it can be easily verified that $\sum_i \tilde{s}_i \mod 30 = \sum_i s_i = 14$).

Let $C = \{3\}$ and so, $E_C = \{\{1, 3\}, \{2, 3\}\}$. It is easy to see that $C$ does not cut the graph $\mathcal{G}$ and therefore, for
any pair of inputs \( s_1 \in \mathbb{Z}_{10} \) and \( s_2 \in \mathbb{Z}_{10} \) that satisfy \( s_1 + s_2 = 11 \), \( \delta_1 \) and \( \delta_2 \) are uniformly distributed over \( \mathbb{Z}_{30} \) subject to \( \delta_1 + \delta_2 = 24 \) mod \( 30 \) (cf. Lemma 3).

5 Extension

In this section, we present an extension of Theorem 4 for the case when \( \mathcal{C} \) is a vertex cut, by relaxing our definition of \( C \)-privacy to \((\mathcal{C},\mathcal{H})\)-privacy as following. Here, the set \( \mathcal{H} \) is a subset of \( \mathcal{V}\setminus \mathcal{C} \), i.e. we are now interested in privacy of some of the honest agents instead of all honest agents. (Apart from \( \mathcal{H} \), all other notations remain the same.)

Definition 3. A distributed average consensus protocol is (perfectly) \((\mathcal{C},\mathcal{H})\)-private if for all \( s, s' \) such that \( s_{\mathcal{V}\setminus \mathcal{H}} = s'_{\mathcal{V}\setminus \mathcal{H}} \) and \( \sum_{i \in \mathcal{H}} s_i = \sum_{i \in \mathcal{H}} s'_i \), the distributions of \( \text{View}_\mathcal{C}(s) \) and \( \text{View}_\mathcal{C}(s') \) are identical.

Similarly, we remark that this definition makes sense even if \( |\mathcal{H}| = 1 \), though in that case the definition is vacuous since \( |\mathcal{H}| = 1 \).

From Theorem 4 we infer that - our proposed distributed average consensus protocol is \((\mathcal{C},\mathcal{H})\)-private if \( \mathcal{C} \) does not cut \( \mathcal{H} \). (In other words, if the set of agents \( \mathcal{H} \) are connected in the sub-graph \( \bar{G}_\mathcal{C} = (\mathcal{C},\mathcal{E}_\mathcal{C}) \) then our proposed distributed average consensus protocol is \((\mathcal{C},\mathcal{H})\)-private.)

6 Conclusion

In this paper, we propose a private (asynchronous) distributed average consensus protocol that guarantees (perfect) privacy of honest agents against a semi-honest attacker if the set of corrupted agents is not a vertex cut of the underlying communication network. The only information that semi-honest corrupted agents can get on the inputs of honest agents is their sum (or average). This reduces to having a network of \((t+1)\)-connectivity for guaranteed privacy of honest agents against if the attacker corrupts at most \( t \) number of arbitrary agents in the network. In an obvious extension of this result, we conclude that our proposed distributed average consensus protocol can preserve of any subset of honest agents (in the network) as long as that subset of honest agents are not cut by the set of corrupted agents.
as $G'$ is connected\footnote{It follows easily from the fact that there exists a path in $G'$ between the terminal nodes of the edge $e'$ as $G'$ is connected.} where $\mu_e \in \{-1, 0, 1\}$ for all $e \in E'$. Define $E'' = E' \cup \{e'\}$. From (3), each point $a''$ of $L(N'')$ (span of the columns of the incidence matrix $N''$ of $G'' = \{V, E''\}$ over $\mathbb{Z}_p$) takes the following form

$$a'' = \sum_{e \in E'} \nabla_{\ast,e} \cdot b_e + \nabla_{\ast,e'} \cdot b_{e'} \mod p$$

As the values $\{b_e\}_{e \in E'}$ are independent and uniform over $\mathbb{Z}_p$, thus $\{b_e + \mu_e b_{e'} \mod p\}_{e \in E'}$ is uniformly distributed over all the values in $\mathbb{Z}_p$, i.e. $p^{n-1}$ possibilities. Hence, $a''$ is uniformly distributed over $L(N'')$ (and span of $N''$ is same as the span of $N'$ in $\mathbb{Z}_p$).

Same as above, if $E'' = E$, the proof concludes. Otherwise, repeat this procedure by considering another edge from $E' \setminus E''$, which will lead to a similar conclusion. This process is repeated until all the edges of $E$ have been considered. Finally, it leads to the conclusion that $a$ is uniformly distributed over $p^n - 1$ points in $L(N)$ (and it axiomatic implies that $L(N)$ is same as $L(N')$).

7.2 Proof of Lemma 3

Let $\hat{S}$, $S$ and $A$ represent the random vectors of the agents’ effective inputs, true inputs and masks, respectively.

As $\hat{s}_i = s_i + a_i \mod p$ and $s_i, a_i$ are independent, we have

$$Pr(\hat{S} = \hat{s} | S = s) = Pr(A = (\hat{s} - s) \mod p)$$

If $\sum \hat{s}_i = \sum s_i \mod p$ then $(\hat{s} - s) \mod p$ belongs to $L(N)$ when $G$ is connected, and so,

$$Pr(\hat{S} = \hat{s} | S = s) = 1/p^{n-1}$$

for all the values $\hat{s}$ (effective inputs) in $\mathbb{Z}_p$ that satisfy $\sum \hat{s}_i = \sum s_i \mod p$, when $G$ is connected (cf. Lemma 2).

7.3 Proof of Theorem 4

Let $G_H = \{H, E_H\}$ be the graph of honest agents (and edges incident to only honest agents) and $\nabla_H$ be its incidence matrix.

The attacker’s view essentially consists of the honest agents’ effective inputs $\hat{s}_H$ after the first phase of our protocol (considering the “worst-case” scenario where agents can acquire all the inputs through their internal states in II) and the values $\{b_e\}_{e \in E_c}$. Therefore, all that remains to show is that the joint distribution of $\hat{s}_H$ and $\{b_e\}_{e \in E_c}$ remains the same for any two sets of true inputs $s, s'$, that satisfy $s_c = s'_c$ and $\sum_{e \in V} s_i = \sum_{e \in V} s'_i$ when $G_H$ is connected. ($G_H$ being connected is equivalent to $C$ not being a vertex cut of $G$).

Each $a_i$ can be dissected as following

$$a_i = \sum_{e \in H} \nabla_{i,e} \cdot b_e + \sum_{e \in E_c} \nabla_{i,e} \cdot b_e \mod p$$

Note that the values $\{b_e\}_{e \in E_c}$ are uniformly and independently distributed in $\mathbb{Z}_p$ (given the values $\{b_e\}_{e \in E_c}$).

The values $\{\sum_{e \in H} \nabla_{i,e} b_e \mod p\}_{i \in H}$ lie in the span of $\nabla_H$ (over $\mathbb{Z}_p$), represented as $L(\nabla_H)$. Therefore, $\{\sum_{e \in H} \nabla_{i,e} b_e \mod p\}_{i \in H}$ is uniformly distributed over $\mathbb{Z}_p$ subject to $\sum_{i \in H} (\sum_{e \in E_c} \nabla_{i,e} b_e) = 0 \mod p$ when $G_H$ is connected (cf. Lemma 2).

Thus, it is clear that the masks $\{a_i\}_{i \in H}$ are uniformly distributed in $\mathbb{Z}_p$ subject to $\sum_{i \in H} a_i = -\sum_{i \in C} a_i \mod p$, when $G_H$ is connected (given the values of $\{b_e\}_{e \in E_c}$).

Combining the above with Lemma 3 implies, $(\tilde{S}_H$ represents the random vector of the honest agents’ effective inputs $\hat{s}_H$)

$$Pr(\tilde{S}_H = \hat{s}_H | s_H, \{b_e\}_{e \in E_c}) = 1/p^{|H|-1}$$

for all the values $\hat{s}_H$ in $\mathbb{Z}_p$ that satisfy

$$\sum_{i \in H} \hat{s}_i = \sum_{i \in C} s_i - \sum_{i \in V} a_i = \sum_{i \in C} s_i - \sum_{i \in C} \hat{s}_i \mod p$$

when $G_H$ is connected. ($a_i = \sum_{e \in E_c} \nabla_{i,e} b_e \mod p$ for every $i \in C$).

Combining (4) with the fact that all the values of $\{b_e\}_{e \in E}$ are independent to each other and the inputs $s$ simply implies that

$$Pr(\text{View}_{C}(s) = \{\hat{s}_H, \{b_e\}_{e \in E_c}\} = \Pr(\text{View}_{\tilde{C}}(s') = \{\tilde{s}_H, \{b_e\}_{e \in E_c}\}$$

for any two inputs $s, s'$ such that $s_c = s'_c$ and $\sum_{i \in V} s_i = \sum_{i \in V} s'_i$ when $G_H$ is connected. ($\text{View}_{\tilde{C}}(s)$ is the random vector representing both the effective inputs of the honest agents $s_H$ and the values $\{b_e\}_{e \in E_c}$ given the inputs $s$).

References

[1] Ayasal, T. C., Yildiz, M. E., Sarwate, A. D., and Scaglione, A. (2009). Broadcast gossip algorithms for
consensus. *IEEE Transactions on Signal Processing*, 57(7):2748–2761.

[2] Ben-Or, M., Goldwasser, S., and Wigderson, A. (1988). Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc. 20th Annual ACM Symposium on Theory of Computing*, pages 1–10. ACM.

[3] Bénédit, F., Blondel, V., Thiran, P., Tsitsiklis, J., and Vetterli, M. (2010). Weighted gossip: Distributed averaging using non-doubly stochastic matrices. In *Information Theory Proceedings (ISIT)*, pages 1753–1757. IEEE.

[4] Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006). Randomized gossip algorithms. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2508–2530.

[5] Braca, P., Lazzeretti, R., Marano, S., and Matta, V. (2016). Learning with privacy in consensus + obfuscation. *IEEE Signal Processing Letters*, 23(9):1174–1178.

[6] Chaum, D., Crépeau, C., and Damgard, I. (1988). Multiparty unconditionally secure protocols. In *Proc. 20th Annual ACM Symposium on Theory of Computing*, pages 11–19. ACM.

[7] Dolev, D., Dwork, C., Waarts, O., and Yung, M. (1993). Perfectly secure message transmission. *J. ACM*, 40(1):17–47.

[8] Forero, P. A., Cano, A., and Giannakis, G. B. (2010). Consensus-based distributed support vector machines. *Journal of Machine Learning Research*, 11(5):1663–1707.

[9] Garay, J. and Ostrovsky, R. (2008). Almost-everywhere secure computation. In *Advances in Cryptology—Eurocrypt 2008*, Lecture Notes in Computer Science, pages 307–323. Springer.

[10] Godsil, C. and Royle, G. (2001). *Algebraic Graph Theory*. Springer.

[11] Goldreich, O. (2004). *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press.

[12] Gupta, N. and Chopra, N. (2016). Confidentiality in distributed average information consensus. In *55th IEEE Conf. on Decision and Control*, pages 6709–6714. IEEE.

[13] Gupta, N., Katz, J., and Chopra, N. (2017). Privacy in distributed average consensus. *IFAC-PapersOnLine*, 50(1):9515–9520.

[14] Huang, Z., Mitra, S., and Dullerud, G. (2012). Differentially private iterative synchronous consensus. In *Proc. ACM Workshop on Privacy in the Electronic Society*, pages 81–90. ACM.

[15] Jadbabaie, A., Lin, J., and Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001.

[16] Lazzeretti, R., Horn, S., Braca, P., and Willett, P. (2014). Secure multi-party consensus gossip algorithms. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 7406–7410. IEEE.

[17] Manitara, N. E. and Hadjicostis, C. N. (2013). Privacy-preserving asymptotic average consensus. In *European Control Conference*, pages 760–765. IEEE.

[18] Mo, Y. and Murray, R. M. (2017). Privacy preserving average consensus. *IEEE Transactions on Automatic Control*, 62(2):753–765.

[19] Nozari, E., Tallapragada, P., and Cortés, J. (2017). Differentially private average consensus: obstructions, trade-offs, and optimal algorithm design. *Automatica*, 81:221–231.

[20] Olfati-Saber, R. (2005). Distributed kalman filter with embedded consensus filters. In *44th IEEE Conference on Decision and Control*, pages 8179–8184. IEEE.

[21] Olfati-Saber, R., Fax, A., and Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233.

[22] Olfati-Saber, R. and Shammas, J. S. (2005). Consensus filters for sensor networks and distributed sensor fusion. In *44th IEEE Conference on Decision and Control*, pages 6698–6703. IEEE.

[23] Pequito, S., Kar, S., Sundaram, S., and Aguiar, A. P. (2014). Design of communication networks for distributed computation with privacy guarantees. In *53rd IEEE Conference on Decision and Control*, pages 1370–1376. IEEE.

[24] Ren, W., Beard, R. W., et al. (2005). Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5):655–661.

[25] Ruan, M., Ahmad, M., and Wang, Y. (2017). Secure and privacy-preserving average consensus. In *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and Privacy*, pages 123–129. ACM.

[26] Xiao, L. and Boyd, S. (2004). Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78.

[27] Yang, S., Tan, S., and Xu, J.-X. (2013). Consensus based approach for economic dispatch problem in a smart grid. *IEEE Transactions on Power Systems*, 28(4):4416–4426.