Charalambous, Themistoklis; Manitara, Nikolas E.; Hadjicostis, Christoforos N.

Privacy-Preserving Average Consensus over Digraphs in the Presence of Time Delays

Published in:
Proceedings of the 57th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2019

DOI:
10.1109/ALLERTON.2019.8919746

Published: 01/09/2019

Please cite the original version:
Abstract—In this paper, we propose a privacy-preserving discrete-time asymptotic average consensus mechanism that allows components of a multi-component system to calculate the exact average of their initial values without revealing to other components their specific value. We assume that components (nodes) interact with other components via possibly directed communication links (edges), forming a generally directed communication topology (digraph). The proposed distributed protocol can be followed by any component that wants to maintain its privacy (i.e., not reveal the initial value it contributes to the average) to possibly multiple curious but not malicious nodes (curious nodes try to identify the initial values of other nodes, and can exchange information with other curious nodes, but do not interfere in the computation in any other way). We devise a distributed mechanism, based on ratio consensus, where each node updates its information state by combining the available information received by its in-neighbors using constant positive weights and by adding an offset (only at one of the two states communicated during the execution of the algorithm). We establish that this privacy-preserving version of ratio consensus, henceforth called the privacy-preserving ratio consensus algorithm, converges to the exact average of the nodes’ initial values, even in the presence of bounded time-varying delays. Illustrative examples demonstrate the validity and performance of our proposed algorithm.

Index Terms—Distributed coordination; privacy preservation; asymptotic exact average consensus.

I. INTRODUCTION

The advancement of smart devices, with embedded sensing, computing and control capabilities, enables the development of smart environments (cities, transportation systems, etc.). Typically, the overall system is spatially distributed and communication between smart devices (i.e., sensors, actuators or controllers) is mainly supported by a communication network. As a result, cooperative distributed control/coordination algorithms and protocols have been extensively studied, especially during the last decade.

When the components (asymptotically) reach agreement to the same value, we say that the distributed system (asymptotically) reaches consensus. One of the most well known consensus problems is the so-called average consensus problem in which components aim to reach the average of their initial values. The problem of convergence of discrete-time consensus algorithms was initially targeted by DeGroot [1] and subsequently by many other researchers (see, for example, [2]–[12]). Average consensus constitutes an emblematic problem, since it is inextricably linked to a wide variety of applications (such as, load balancing in parallel computing, distributed coordination of multi-agent systems, data fusion in sensor networks, the PageRank problem, and power systems generation control) and is an essential primitive of algorithm design in several fields (such as, distributed estimation and control, distributed optimization, and machine learning). A recent thorough review of average consensus in digraphs can be found in [13].

Existing average consensus algorithms require each node to exchange and disclose state information to its neighbors, which may be undesirable in cases where the state is private or contains sensitive information. In many emerging and foreseeable applications including, inter alia, cryptocurrency, healthcare, and opinion forming/agreement in social networks, preserving the privacy of participating components is necessary, so that participation is enabled without requiring the components to disclose any sensitive information.

Several research strands have focused on preserving privacy in average consensus. One of the first and influential works on privacy-preserving average consensus is [14] in which a transformation method is proposed using random offset values in a cooperative wireless network. Specifically, each node that wishes to protect its privacy adds a random offset value to its initial value, thus ensuring that its value will not be revealed to curious nodes that might be observing the exchange of values in the network. The main idea is based upon the observation that, when a large number of nodes employ the protocol, the sum of their offsets will be essentially zero and therefore the values of the nodes converge to the true average value of the network. Another strand of research that received tremendous attention is called differentially private average consensus, in which uncorrelated noise is injected in the exchanged messages so that the data associated to a particular component cannot be inferred by a curious node during the execution of the algorithm. However, the exact average value is not eventually obtained due to the induced trade-off between enabled privacy and computational accuracy; see the analysis and discussion in [15]. To overcome this trade-off and guarantee convergence to the exact average, the injection of correlated (as opposed to uncorrelated) noise is proposed in [16], in which the nodes asymptotically subtract their initial offset values added to the network. Gupta et al. [17] proposed a protocol in which each node masks its initial value with an offset, such that the sum of the offsets of all agents is zero, thus guaranteeing
convergence to the average. In [18], a protocol is proposed in which the state of a node is randomly decomposed into two sub-states, such that the mean remains the same and only one of the sub-states is revealed to the neighboring nodes.

None of the aforementioned approaches is suitable for digraphs that are not balanced. Constructing doubly stochastic weight matrices (and, hence, balanced digraphs) in a distributed fashion is often undesirable though, since it requires a lot of communication overhead (see, for example, [19]–[21]). The only work in the literature so far that considered privacy-preserving average consensus in digraphs that are not balanced, to the best of the authors’ knowledge, is [22]. Therein, the authors enable privacy by adding randomness on the edge weights. As a consequence, this approach requires that each node transmits individual messages to each out-neighbor, instead of just broadcasting a single value, thus increasing the communication overhead considerably. Such a communication overhead can have detrimental effects, especially in big/dense networks, due to possible packet collisions and interference (in wireless networks). Finally, another strand of research guarantees privacy preservation via homomorphic encryption; e.g., [23]. However, this method relies on the existence of a node that is universally trusted.

In this work, building on the concepts introduced in [24], we propose a distributed mechanism in which each node that would like to protect its privacy, at each time step and for a finite period of time, adds an arbitrary offset to the state it broadcasts, thus avoiding to reveal its own initial value as well as the initial values of other nodes. Once this period of time ends, each node ensures that the total (accumulated sum of) offset that has been added is cancelled out. While [24] uses a similar protocol, their distributed algorithm was only limited for undirected graphs or digraphs with balanced weight matrices. In this work, we devise a distributed mechanism, based on ratio consensus, where each node updates its information state by combining the available information received by its in-neighbors using constant positive weights and by adding an offset (only at one of the two states broadcasted during the execution of the algorithm). Then, we extend our proposed distributed algorithm to handle bounded time-varying delays. We assume the worst-case scenario in which the curious nodes may know the predefined consensus strategy as well as the topology of the network (and the delays when they are present), but not the actual offset values used by the nodes that want to preserve their privacy. We establish that the privacy-preserving ratio consensus algorithm converges to the exact average of the initial values of all the nodes, even in the presence of time-varying delays, without allowing the curious nodes to infer the initial values of the nodes that follow the privacy-preserving protocol, at least subject to some mild conditions on the communication topology.

The remainder of the paper is organized as follows. In Section II we review necessary notation and background. Sections III and IV present our main results in the delay-free and delayed cases, respectively, with illustrative examples. Finally, Section V presents concluding remarks.

II. NOTATION AND PRELIMINARIES

A. Notation

The set of real (integer) numbers is denoted by \( \mathbb{R} \) (\( \mathbb{Z} \)) and the set of nonnegative numbers (integers) is denoted by \( \mathbb{R}_+ \) (\( \mathbb{Z}_+ \)). \( \mathbb{R}_+^n \) denotes the nonnegative orthant of the \( n \)-dimensional real space \( \mathbb{R}^n \). Vectors are denoted by small letters whereas matrices are denoted by capital letters. The transpose of a matrix \( A \) is denoted by \( A^\top \). For \( A \in \mathbb{R}^{n \times n} \), \( A_{ij} \) denotes the entry at row \( i \) and column \( j \). By \( I \) we denote the all-ones vector and by \( I \) we denote the identity matrix (of appropriate dimensions).

In multi-component systems with fixed communication links (edges), the exchange of information between components (nodes) can be conveniently captured by a directed graph (digraph) \( G(\mathcal{N}, \mathcal{E}) \) of order \( n \) (\( n \geq 2 \)), where \( \mathcal{N} = \{v_1, v_2, \ldots, v_n\} \) is the set of nodes and \( \mathcal{E} \subseteq \mathcal{N} \times \mathcal{N} \) is the set of edges. A directed edge from node \( v_i \) to node \( v_j \) is denoted by \( e_{ij} := (v_j, v_i) \in \mathcal{E} \) and represents a communication link that allows node \( v_j \) to receive information from node \( v_i \).

A graph is said to be undirected if and only if \( e_{ij} \in \mathcal{E} \) implies \( e_{ji} \in \mathcal{E} \). In this paper, links are not required to be bidirectional, i.e., we deal with digraphs; for this reason, we use the terms “graph” and “digraph” interchangeably.

A digraph is called strongly connected if there exists a path from each vertex \( v_i \) in the graph to each vertex \( v_j \) (\( v_j \neq v_i \)). In other words, for any \( v_j, v_i \in \mathcal{N}, v_j \neq v_i \), one can find a sequence of nodes \( v_i = v_{i_1}, v_{i_2}, v_{i_3}, \ldots, v_{i_t} = v_j \) such that link \( (v_{i_{m+1}}, v_{i_m}) \in \mathcal{E} \) for all \( m = 1, 2, \ldots, t - 1 \). In such a case, we say that a path of length \( \ell \) exists from node \( v_i \) to node \( v_j \). The eccentricity \( \epsilon(v_i) \) of a node \( v_i \) is the greatest distance from \( v_i \) to any other vertex, i.e., it shows how far node \( v_i \) is from the node most distant from it in the graph. [Here, the distance from node \( v_i \) to node \( v_j \) is the length of the shortest path from node \( v_i \) to node \( v_j \).] The diameter \( d \) of a graph is the maximum eccentricity of any node in the graph, i.e., \( d = \max_{v_i \in \mathcal{N}} \epsilon(v_i) \).

All nodes that can transmit information to node \( v_j \) directly are said to be in-neighbors of node \( v_j \) and belong to the set \( \mathcal{N}_j^- = \{v_i \in \mathcal{N} \mid e_{ji} \in \mathcal{E}\} \). The cardinality of \( \mathcal{N}_j^- \), is called the in-degree of \( v_j \) and is denoted by \( D_j^- = |\mathcal{N}_j^-| \). The nodes that receive information from node \( v_j \) belong to the set of out-neighbors of node \( v_j \), denoted by \( \mathcal{N}_j^+ = \{v_i \in \mathcal{N} \mid e_{ij} \in \mathcal{E}\} \). The cardinality of \( \mathcal{N}_j^+ \), is called the out-degree of \( v_j \) and is denoted by \( D_j^+ = |\mathcal{N}_j^+| \).

In the algorithms we consider, we associate a positive weight \( p_{ji} \) with each edge \( e_{ji} \in \mathcal{E} \cup \{(v_j, v_i) \mid v_j \in \mathcal{N}\} \). The nonnegative matrix \( P = \{p_{ji}\} \in \mathbb{R}_+^{n \times n} \) (with \( p_{ji} \) as the entry at its \( j \)th row, \( i \)th column position) is a weighted adjacency matrix (also referred to as weight matrix) that has zero entries at locations that do not correspond to directed edges (or self-edges) in the graph. In other words, apart from the main diagonal, the zero-nonzero structure of the weighted matrix \( P \) matches exactly the given set of links in the graph.

We use \( x_j[k] \in \mathbb{R}, k \in \mathbb{Z}_+ \), to denote the state of node \( v_j \) at time \( t_k \). In the synchronous setting we consider, each node \( v_j \) updates and sends its information to its neighbors...
at discrete times $t_0, t_1, t_2, \ldots$. We index nodes’ information states and any other information at time $t_k$ by $k$. Hence, we use $x_j[k]$ to denote the state of node $v_j$ at time $t_k$.

### B. Distributed linear iterations

Each node updates its information state $x_j[k]$ by combining the available information received by its in-neighbors $x_i[k]$ ($v_i \in \mathcal{N}_j^-$) using the positive weights $p_{ji}[k]$, that capture the weight of the information inflow from agent $v_i$ to agent $v_j$ at time $k$. In this work, we assume that each node $v_j$ can choose its self-weight $p_{jj}$ and the weights $p_{ij}$ on its outgoing links $(v_i, v_j)$, $v_i \in \mathcal{N}_j^+$ (e.g., by sending $p_{ij}v_j$ to its out-neighbour $v_i \in \mathcal{N}_j^+$. Hence, in its general form, each node updates its information state according to the following relation:

$$x_j[k+1] = p_{jj}x_j[k] + \sum_{v_i \in \mathcal{N}_j^-} p_{ji}[k]x_i[k], \quad (1)$$

for $k \geq 0$, where $x_j[0] \in \mathbb{R}$ is the initial state of node $v_j$. Let $x[k] = (x_1[k], x_2[k], \ldots, x_n[k])^T$ and $P = [p_{ji}] \in \mathbb{R}^n \times n$, with $p_{ji} = 0$ if $v_i \notin \mathcal{N}_j^- \cup \{v_j\}$. Then, (1) can be expressed in matrix form as

$$x[k+1] = Px[k], \quad (2)$$

where $x[0] = (x_1[0], x_2[0], \ldots, x_n[0])^T := x_0$. We say that the nodes asymptotically reach average consensus if

$$\lim_{k \to \infty} x_j[k] = \frac{1}{n} \sum_{v_i \in \mathcal{N}_j^-} x_i[0], \quad \forall v_j \in \mathcal{N}.$$

The necessary and sufficient conditions for (2) to reach average consensus are the following [25]: (a) $P$ has a simple eigenvalue at 1 with left eigenvector $\mathbb{1}^T$ and right eigenvector $\mathbb{1}$, and (b) all other eigenvalues of $P$ have magnitude less than 1. If $P \geq 0$ (as in our case), the necessary and sufficient condition is that $P$ is a primitive doubly stochastic matrix $P$. However, in a digraph, it is not possible to set up a doubly stochastic weight matrix without having the nodes in the network exchange information among them [13], [27].

### III. DELAY-FREE PRIVACY-PRESERVING RATIO CONSENSUS

In this section, we present the delay-free privacy-preserving ratio consensus algorithm, whose objective is to compute the average of the initial values of the nodes in the digraph, while at the same time preserving the privacy of, at least, the nodes that adopt the algorithm. As aforementioned, we assume that curious nodes try to identify the initial values of other nodes but do not interfere in the computation in any other way. We also assume that curious nodes may know the predefined algorithm and topology of the network, but not the actual values used by the nodes that want to preserve their privacy.

In [28], an algorithm is suggested that solves the average consensus problem in a directed graph in which each node $v_j$ distributively sets the weights on its self-link and outgoing-links to be $p_{jj} = \frac{1}{1+D_j}$ for $(v_i, v_j) \in \mathcal{E}$, so that the resulting weight matrix $P = [p_{ji}]$ is column stochastic, but not necessarily row stochastic. Asymptotic average consensus is reached by using this weight matrix to run two iterations with appropriately chosen initial conditions. The algorithm is stated below for a specific choice of weights on each link that assumes that each node knows its out-degree; note, however, the algorithm works for any set of weights that adhere to the graph structure and form a primitive column stochastic weight matrix.

**Proposition 1 ([28]):** Consider a strongly connected di-graph $G(\mathcal{N}, \mathcal{E})$. Let $y_j[k]$ and $x_j[k]$ (for all $v_j \in \mathcal{N}$ and $k = 0, 1, 2, \ldots$) be the result of the iterations

$$y_j[k+1] = p_{jj}y_j[k] + \sum_{v_i \in \mathcal{N}_j^-} p_{ji}[k]y_i[k], \quad (3a)$$

$$x_j[k+1] = p_{jj}x_j[k] + \sum_{v_i \in \mathcal{N}_j^-} p_{ji}[k]x_i[k], \quad (3b)$$

where $p_{ij} = \frac{1}{1+D_j}$ for $v_i \in \mathcal{N}_j^+ \cup \{v_j\}$ (zeros otherwise), and the initial conditions are $y[0] \in \mathbb{R}^n$ and $x[0] = 1$. Then, the solution to the average consensus problem can be asymptotically obtained as

$$\lim_{k \to \infty} \mu_j[k] = \frac{\sum_{v_i \in \mathcal{N}_j^-} y_i[0]}{|\mathcal{N}|}, \quad \forall v_j \in \mathcal{N}.$$

**Remark 1:** Proposition 1 proposes a decentralized algorithm with which the exact average is asymptotically reached, even if the directed graph is not balanced.

In what follows, we propose an algorithm that is based on ratio consensus with some modifications (as it has already been shown in [22] that ratio consensus is not privacy-preserving, even for time-varying weights). More specifically, the algorithm makes use of the ratio consensus iterations, as in (3), the main difference being that node $v_j$ following the privacy-preserving consensus algorithm sets its initial value to $\tilde{y}_j[0] = y_j[0] + \alpha_j$, where $y_j[0] = S_j$ and $\alpha_j$ is an arbitrary offset, and in the following iterations it updates its information value as

$$\tilde{y}_j[k+1] = p_{jj}\tilde{y}_j[k] + \sum_{v_i \in \mathcal{N}_j^-} p_{ji}\tilde{y}_i[k] + \alpha_j[k], \quad k \in \mathbb{Z}_+, \quad (4)$$

where $\alpha_j[k]$ is a pseudo-random value chosen by node $v_j$ at time step $k$. Note that, for ratio consensus, at the same time, node $v_j$ runs iteration (3b) with initial condition $x[0] = 1$, without any changes. At time-step $k = L_j$ (for some $L_j$ known only to node $v_j$), node $v_j$ effectively cancels the pseudo-random values it has added during the information
exchange in the digraph up to that point, i.e.,
\[ \alpha_j[L_j] = - \sum_{k=0}^{L_j-1} \alpha_j[k] - \alpha_j, \]  
and for \( k > L_j \), \( \alpha_j[k] = 0 \) in the iteration described in \([4]\).

Note that \( L_j \) is an integer number of steps, randomly chosen by node \( v_j \). This number can be decided even after the start of the iteration and at a step, say \( k^* \), at which the offset occurs, is not noticeable. The nodes not close to zero, so that the correction, at the last state for which the offset occurs, is not noticeable. The nodes not following the privacy-preserving ratio consensus algorithm run the ratio consensus algorithm with zero offsets in \([4]\), i.e., as it is in \([3]\).

The procedure is summarized in Algorithm 1.

**Algorithm 1** Delay-free privacy-preserving ratio consensus

**Input:** A strongly connected digraph \( \mathcal{G}(\mathcal{N}, \mathcal{E}) \) with \( |\mathcal{N}| \) nodes and \( |\mathcal{E}| \) edges.

**Initialization:** Each node \( v_j \in \mathcal{N} \), that follows the protocol, with initial conditions \( y_j[0] = S_j + \alpha_j \) (where \( \alpha_j \) is a random offset) and \( x_j[0] = 1 \), does the following:

- **If** \( k < L_j \) **then:** It chooses a random offset \( \alpha_j[k] \) and executes iterations \([3b]\) and \([4]\) as

  \[ \tilde{y}_j[k+1] = p_{jj} \tilde{y}_j[k] + \sum_{v_i \in \mathcal{N}_j^-} p_{ji} \tilde{y}_i[k] + \alpha_j[k], \]

  where \( \alpha_j,_{\text{total}} = \alpha_j,_{\text{total}} + \alpha_j[k] \).

- **If** \( k = L_j \) **then:** It executes iterations \([3b]\) and \([4]\) as

  \[ \tilde{y}_j[k+1] = p_{jj} \tilde{y}_j[k] + \sum_{v_i \in \mathcal{N}_j^-} p_{ji} \tilde{y}_i[k] - \alpha_j,_{\text{total}}, \]

  using the current value of \( \alpha_j,_{\text{total}} \).

- **If** \( k > L_j \) **then:** It executes iterations \([3a]\) and \([3b]\), starting with values \( \tilde{y}_j[L_j] \) and \( x_j[L_j] \), respectively.

**Output:** The ratio \( \mu_j[k] := \tilde{y}_j[k]/x_j[k] \), which satisfies

\[ \lim_{k \to \infty} \mu_j[k] = \lim_{k \to \infty} \tilde{y}_j[k]/x_j[k] = \bar{\mu}, \]

where \( \bar{\mu} \) is the exact average, i.e., \( \bar{\mu} = \frac{\sum_{v_i \in \mathcal{N}} S_i}{|\mathcal{N}|} \).

**Lemma 1:** Consider a strongly connected digraph \( \mathcal{G}(\mathcal{N}, \mathcal{E}) \). Under Algorithm 1 the solution to the privacy-preserving average consensus problem can be asymptotically obtained as

\[ \lim_{k \to \infty} \mu_j[k] = \frac{\sum_{v_i \in \mathcal{N}} y_j[0]}{|\mathcal{N}|}, \forall v_j \in \mathcal{N}, \]

where \( \mu_j[k] = \frac{\tilde{y}_j[k]}{x_j[k]} \).

**Proof:** Denote \( L_{\text{max}} := \max_i \{L_i\} \). Then, it is easy to see that the total “mass” of the initial states is preserved, i.e.,

\[ \sum_{v_j \in \mathcal{N}} \tilde{y}_j[L_{\text{max}} + 1] = \sum_{v_j \in \mathcal{N}} y_j[0], \]

\[ \sum_{v_j \in \mathcal{N}} x_j[L_{\text{max}} + 1] = \sum_{v_j \in \mathcal{N}} x_j[0]. \]

As a result, the limit of the ratio of the values of the two iterations, \( \mu_j[k] \), at a node \( v_j \in \mathcal{N} \) is given by

\[ \lim_{k \to \infty} \mu_j[k] = \lim_{k \to \infty} \frac{\tilde{y}_j[k]}{x_j[k]} = \frac{c(j)[(1^T \tilde{y}[L_{\text{max}} + 1])]}{c(j)(1^T x[L_{\text{max}} + 1])} = \frac{\sum_{v_i \in \mathcal{N}} y_i[0]}{\sum_{v_i \in \mathcal{N}} x_i[0]} \]

where \( c \) is the right eigenvector of \( P \) that corresponds to eigenvalue 1. Note that \( c(j) \neq 0 \), from the fact that matrix \( P \) is Stochastic Irreducible and Aperiodic (SIA) and the digraph is strongly connected and, therefore, all the entries in vector \( c \) are nonzero. \( \Box \)

**Remark 2:** Each node will converge to the average with a minimum rate \( \gamma \) given by [29, Proposition 1(b)]

\[ \gamma = \left( 1 - \frac{1}{\left( 1 + \max_{v_i \in \mathcal{N}} D^+_i \right)^{|\mathcal{N}|-1}} \right)^{1/(|\mathcal{N}|-1)}. \]

In [22], the authors allow the weights of nodes to change arbitrarily in the range \([\varepsilon, 1] \). In order to be able to allow all the out-neighbors to have different weights, \( \varepsilon < 1/(1 + \max_{v_i \in \mathcal{N}} D^+_i) \). As a consequence, the worst-case rate of convergence of the algorithm proposed in [22] is smaller than \( \gamma \). \( \Box \)

**Remark 3:** Unlike [24], in this work we do not need to assume that the graph is undirected or that the weight matrix \( P \) is primitive doubly stochastic. Note that [22] also leverages on the ratio consensus concept for achieving average consensus while preserving privacy, as we do in this paper; however, with our approach, nodes need not send individual messages to different out-neighbors, thus enhancing the applicability of the proposed privacy-preserving mechanism without additional communication overhead and with very limited additional computational overhead. Furthermore, the rate of convergence of our approach can be faster, as discussed in Remark 2. \( \Box \)

**Example 1:** We consider a digraph consisting of 6 nodes, as shown in Fig. 1, in which all the nodes follow the privacy-preserving consensus algorithm proposed in Algorithm 1.
Each node $v_j$ chooses its self-weight and the weight of its outgoing links to be $(1 + D_j^+)^{-1}$ (such that the sum of all weights assigned by each node $v_j$ is equal to 1). In matrix form, $P$ is given by

$$
P = \begin{pmatrix}
1/4 & 1/2 & 1/4 & 0 & 1/2 & 0 \\
1/4 & 1/2 & 1/4 & 0 & 0 & 0 \\
1/4 & 0 & 1/4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1/2 & 1/3 \\
0 & 0 & 1/4 & 1/2 & 0 & 1/3 \\
0 & 0 & 0 & 0 & 1/2 & 1/3 \\
\end{pmatrix}.
$$

We set $\tilde{y}(0) = \begin{bmatrix} -1 & 0 & 1 & 3 & 4 & 5 \end{bmatrix}^T$, and with the initial offset $\alpha_j$ which, in this example, is obtained from a normal distribution with mean equal to zero and variance equal to 5, $\tilde{y}(0)$ becomes

$$
\tilde{y}(0) = \begin{bmatrix} -2.38 & -9.29 & 1.20 & 4.41 & 4.32 & 7.17 \end{bmatrix}^T.
$$

When each node $v_j$ updates its information states $\tilde{y}_j[k]$ and $x_j[k]$ using equations (4) (with $L_j = L_{\text{max}} = 10$) and (5b), respectively, the evolution of the network and the convergence of the ratios are shown in Fig. 2.

**Remark 4:** From the point of view of the curious nodes, what they can do essentially amounts to observability analysis of the network. Let $Q = \{i_1, i_2, \ldots, i_q\}$ be the set of nodes following the proposed privacy-preserving consensus algorithm. Then, (4) can be written in matrix form as follows

$$
z[k+1] = C_\ell \tilde{y}[k] + E_\ell \alpha[k],
$$

where $E_\ell = [e_{i_1} e_{i_2} \ldots e_{i_q}]$ and $e_i \in \mathbb{R}^{|N|}$ is an $|N|$-dimensional column vector with a single nonzero entry of value 1 at its $i^{th}$ component, and $\alpha[k] := [\alpha_{i_1}[k] \alpha_{i_2}[k] \ldots \alpha_{i_q}[k]]^T$. At each time step, curious node $v_\ell$ observes its in-neighbors and this can be expressed as

$$
z[k] = C_\ell \tilde{y}[k] = C_\ell P^k \tilde{y}[0] + \sum_{s=0}^{k-1} C_\ell P^s E_\ell \alpha_{q}[k - 1 - s],
$$

where $C_\ell \in [0, 1]^{(D_\ell+1)\times|N|}$ (a matrix with a single 1 in each row corresponding to an in-neighbor of curious node $v_\ell$ or node $v_\ell$ itself). Since we assumed that curious nodes can arbitrarily exchange information among themselves, we can consider a single curious node with all the information from all curious nodes. What curious node $v_\ell$ observes during the first $k + 2$ steps is given by [30]

$$
z[k+1] = C_\ell \tilde{y}[k] + [\alpha_{i_1}[k] \alpha_{i_2}[k] \ldots \alpha_{i_q}[k]]^T,
$$

where $z[k] = [z[0 : k + 1]] = [z[k][0] z[k][1] \ldots z[k][k+1]]^T$ and $\alpha[k] := [\alpha_{i_1}[k] \alpha_{i_2}[k] \ldots \alpha_{i_q}[k]]^T$. Matrices $C_\ell$ and $M_\ell$ are expressed recursively as

$$
C_\ell = \begin{bmatrix} C_\ell & 0 \\
0 & M_\ell \end{bmatrix},
$$

where $C_\ell$ is the empty matrix [30]. The ability of the curious node to identify the initial values $y[0]$ as well as the injected offsets $\alpha_q$ are described by (11). For a similar setup as in this paper, it was shown in [24, Theorem 2] that, even under the worst-case assumption that curious node $v_\ell$ knows exactly which nodes follow the protocol as well as the weights used in iteration (4), curious node $v_\ell$ will not be able to identify $y[0] \in Q$ as long as node $v_j$ has at least one other node $v_i$ connected to it for which all paths from $v_i$ to curious node $v_\ell$ are through a node $v_f \in Q$. In other words, if the topological conditions hold, then the network will reach average consensus and the privacy of the initial values of the nodes following the privacy-preserving protocol will be preserved.

**IV. PRIVACY-PRESERVING RATIO CONSSENSUS UNDER TIME-VARYING DELAYS**

In this section, we address the privacy-preserving average consensus problem over directed graphs in the presence of bounded delays in the communication links, arising mainly due to propagation and computational delays. Towards this end, we propose an asynchronous operation of Algorithm 7 where each node updates its own value by using delayed information from its in-neighboring nodes.
We use the integer $\tau_{ji}[k] \geq 0$ to represent the delay of a message sent from node $v_i$ to node $v_j$ at time instant $k$. We require that $0 \leq \tau_{ji}[k] \leq \tilde{\tau} \leq \bar{\tau}$ for all $k \geq 0$ for some finite $\tilde{\tau} = \max \{ \tau_{ji} \}, \bar{\tau} \in \mathbb{Z}_+$. We make the reasonable assumption that $\tau_{ji}[k] = 0, \forall v_j \in \mathcal{N}$, at all time instances $k$ (i.e., the own value of a node is always available without delay). A protocol is employed where each node updates its information state $w_j[k+1]$ by combining the available (possibly delayed) information received by its neighbors $w_i[s]$ ($s \in \mathbb{Z}_+, s \leq k, v_i \in \mathcal{N}_j$) using constant positive weights $p_{ji}$. Proposition 2 presents the decentralized algorithm proposed in [31] with which the exact average is asymptotically reached in the presence of bounded delays, even if the digraph is not balanced.

**Proposition 2 ([31]):** Consider a strongly connected digraph $G(\mathcal{N}, \mathcal{E})$, where each node $v_j \in \mathcal{N}$ has some initial value $y_j[0] = S_j$. Let $y_j[k]$ and $x_j[k]$ (for all $v_j \in \mathcal{N}$ and $k = 0, 1, 2, \ldots$) be the result of the iterations

\[
y_j[k+1] = p_{ji}y_j[k] + \sum_{v_i \in \mathcal{N}_j^-} \sum_{r=0}^{\tilde{\tau}} p_{ji}y_i[k-r]I_{k-r,j}[r],
\]

(12a)

\[
x_j[k+1] = p_{ji}x_j[k] + \sum_{v_i \in \mathcal{N}_j^-} \sum_{r=0}^{\tilde{\tau}} p_{ji}x_i[k-r]I_{k-r,j}[r],
\]

(12b)

where $y[0] = (S_1, S_2, \ldots, S_{|\mathcal{N}|})^T$ and $x[0] = 1$, and $I_{k,j}[\tau]$ is an indicator function that captures the bounded delay $\tau_{ji}[k] \leq \bar{\tau}$ ($\tilde{\tau} < \infty$) on link $(v_j, v_i)$ at iteration $k$, defined as

\[
I_{k,j}[\tau] = \begin{cases} 1, & \text{if } \tau_{ji}[k] = \tau, \\ 0, & \text{otherwise}. \end{cases}
\]

(13)

(note that $y[k]$ and $x[k]$ are taken to be zero for negative $k$).

Then, we have $\lim_{k \to \infty} \mu_j[k] = \frac{\sum_{v_i \in \mathcal{N}} S_i}{|\mathcal{N}|}, \forall v_j \in \mathcal{N}$, where

\[
\mu_j[k] = \frac{y_j[k]}{x_j[k]}.
\]

An augmented graph representation is employed in [31] by adding extra “virtual” nodes and using them to model the delays. The maximum number of “virtual” nodes for each original node is bounded by the maximum delay $\bar{\tau}$. In particular, for each node $v_j \in \mathcal{N}$ we introduce $\bar{\tau}$ “virtual” nodes $v_j^{(1)}, v_j^{(2)}, \ldots, v_j^{(\bar{\tau})}$ (node $v_j^{(d)}$ holds information that is destined to arrive at node $v_j$ after $d$ steps). The augmented digraph has $(\bar{\tau} + 1)|\mathcal{N}|$ nodes and $(1 + 2\bar{\tau})|\mathcal{E}|$ edges. In the general case, in a network of $n = |\mathcal{N}|$ nodes, we introduce $\bar{\tau} n$ nodes (for a total of $(\bar{\tau} + 1)n$ nodes) so that $\psi[k+1] = \Xi[k] \psi[k]$, where \[
\Xi[k] := \begin{pmatrix} P_0[k] & I_{n \times n} & 0 & \cdots & 0 \\ P_1[k] & 0 & I_{n \times n} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ P_{\bar{\tau}-1}[k] & 0 & 0 & \cdots & I_{n \times n} \\ P_{\bar{\tau}}[k] & 0 & 0 & \cdots & 0 \end{pmatrix},
\]

(14)

with $\psi[k] = (w^{(r)}[k] \ldots w^{(1)}[k])^T$ and $w^{(r)}[k] = \left( w^{(r)}_0[k] \ldots w^{(r)}_{\bar{\tau}}[k] \right), r = 1, 2, \ldots, \bar{\tau}$, with $w[k]$ being a generic information state vector (e.g., $y[k]$ or $x[k]$). Note that $P_0[k], P_1[k], \ldots, P_{\bar{\tau}}[k]$ are appropriately defined nonnegative matrices that depend on the link delays that are experienced by messages sent at time $k$. Specifically, $P_r[k]$ is a matrix associated only with the links of the graph for which the message is being delayed by $r$ steps at time step $k$, and satisfies

\[
P_r[k](j,i) = \begin{cases} P(j,i), & \text{if } \tau_{ji}[k] = r, (j,i) \in \mathcal{E}, \\ 0, & \text{otherwise}. \end{cases}
\]

In what follows, we propose a privacy-preserving average consensus problem over directed graphs in the presence of bounded delays. More specifically, the algorithm makes use of iterations (12a) and (12b), but as before, node $v_j$ following the privacy-preserving algorithm sets its initial value to $\tilde{y}_j[0] = y_j[0] + \alpha_j$ (where $y_j[0] = S_j$ and $\alpha_j$ is an arbitrary offset), and in the following iterations it updates its information value as

\[
\tilde{y}_j[k+1] = p_{ji}\tilde{y}_j[k] + \sum_{v_i \in \mathcal{N}_j^-} \sum_{r=0}^{\tilde{\tau}} p_{ji}\tilde{y}_i[k-r]I_{k-r,j}[r] + \alpha_j[k], k \in \mathbb{Z}_+, \tag{15}
\]

where $\alpha_j[k]$ is a pseudo-random value chosen by node $v_j$ at time step $k$. Note that, for ratio consensus, at the same time, node $v_j$ runs iteration (12b) with initial condition $x[0] = 1$, without any offsets. At time-step $k = L_j$ (for some $L_j$ known only to node $v_j$), node $v_j$ effectively cancels the pseudo-random values it has added during the information exchange in the digraph up to that point as described in [15], irrespective of whether the messages have reached the out-neighbors or not. For $k > L_j$, $\alpha_j[k] = 0$ in the iteration described in (15).

The procedure is summarized in Algorithm 2.

**Lemma 2:** Consider a strongly connected digraph $G(\mathcal{N}, \mathcal{E})$. Under Algorithm 2, the solution to the average consensus problem can be asymptotically obtained as

\[
\lim_{k \to \infty} \mu_j[k] = \frac{\sum_{v_j \in \mathcal{N}} \tilde{y}_j[0]}{|\mathcal{N}|}, \forall v_j \in \mathcal{N},
\]

where $\mu_j[k] = \frac{\tilde{y}_j[k]}{x_j[k]}$.

**Proof:** For each $(v_j, v_i) \in \mathcal{E}$, only one of $P_0[k](j,i), P_1[k](j,i), \ldots, P_{\bar{\tau}}[k](j,i)$ is nonzero and is equal to $P(j,i)$. Hence, it is deduced that $\Xi[\cdot]$ is a column-stochastic, non-negative matrix. As a result, once the added pseudo-random values during the information exchange have been cancelled out, the total “mass” of the initial states is preserved. If we use the augmented graph representation with initial conditions $\tilde{y}[0] = [y^T[0] \ 0 \ 0 \ \ldots \ 0]^T$ and $\tilde{x}[0] = [1 \ 0 \ 0 \ \ldots \ 0]^T$, the following holds:

Algorithm 2 Privacy-preserving ratio consensus under delays

**Input:** A strongly connected digraph $G(N, E)$ with $n = |N|$ nodes and $m = |E|$ edges.

**Initialization:** Each node $v_j \in N$, that follows the protocol, with initial conditions $y_j[0] = S_j$ and $x_j[0] = 1$, does the following:

a) Sets its initial values $\bar{y}_j[0] = S_j + \alpha_j$ (where $\alpha_j$ is a random offset) and $x_j[0] = 1$.

b) Selects an integer $L_j > 0$.

c) Sets $\alpha_{j, \text{total}} = \alpha_j$.

For $k = 0, 1, 2, \ldots$, each node $v_j \in N$, that follows the protocol, does the following:

- **if** $k < L_j$ **then:** It chooses a random offset $\alpha_j[k]$ and executes iterations (12b) and (15) as

$$\bar{y}_j[k+1] = p_{jj} \bar{y}_j[k] + \sum_{v_i \in N_j^-} \sum_{r=0}^{\tau} p_{ji} \bar{y}_i[k-r] I_{k-r, ji}[r] + \alpha_j[k], \quad k \in \mathbb{Z}_+,$$

$$\alpha_{j, \text{total}} = \alpha_{j, \text{total}} + \alpha_j[k].$$

- **if** $k = L_j$ **then:** It executes iteration (12b) and (15) as

$$\bar{y}_j[k+1] = p_{jj} \bar{y}_j[k] + \sum_{v_i \in N_j^-} \sum_{r=0}^{\tau} p_{ji} \bar{y}_i[k-r] I_{k-r, ji}[r]$$

$$- \alpha_{j, \text{total}},$$

using the current value of $\alpha_{j, \text{total}}$.

- **if** $k > L_j$ **then:** It executes iterations (12a) and (12b), with initial conditions $\bar{y}_j[L_j]$ and $x_j[L_j]$, respectively.

**Output:** The ratio $\mu_j[k] := \bar{y}_j[k]/x_j[k]$, which satisfies

$$\lim_{k \to \infty} \mu_j[k] = \lim_{k \to \infty} \bar{y}_j[k]/x_j[k] = \bar{\mu},$$

where $\bar{\mu}$ is the exact average, i.e., $\bar{\mu} = \sum_{v_i \in N} S_i/|N|$.

then

$$\sum_{v_j \in N} \bar{y}_j[L_{\text{max}} + \tau + 1] = \sum_{v_j \in N} \bar{y}_j[0],$$

$$\sum_{v_j \in N} x_j[L_{\text{max}} + \tau + 1] = \sum_{v_j \in N} x_j[0] = |N|.$$  

As a result, the ratio of the values of the two iterations at an original (non-virtual) node $v_j \in N$ is given by

$$\lim_{l \to \infty} \mu_j[l] = \lim_{l \to \infty} \frac{\bar{y}_j[l]/x_j[l]}{c_{B_i}(j)(1^T \bar{y}_j[l])/c_{B_i}(j)(1^T x_j[l])} = \frac{\sum_{v_i \in N} \bar{y}_i[0]}{\sum_{v_i \in N} x_i[0]} = \frac{|N|}{x_i[0]}.$$

as long as $c_{B_i}(j) \neq 0$. This is guaranteed from the fact that the matrix product $\Xi[l] \ldots \Xi[m] \Xi[m+1] |N|$ is SIA and the directed graph of the original (non-virtual) nodes is strongly connected [31]. As a result, all the entries in vector $c_{B_i}$ that correspond to the original nodes are (strictly) positive.

**Example 2:** We consider the same digraph as in Example 1 depicted in Fig. 1. For this specific example, the upper bound on the delay is 5 steps (i.e., $\tau = 5$). When each node $v_j$ updates its information states $\bar{y}_j[k]$ and $x_j[k]$ using equations (15) (with $L_j = L_{\text{max}} = 10$) and (12b), respectively, the evolution of the network and the convergence are shown in Fig. 3.

![Fig. 3. Average consensus is asymptotically reached for the ratio $\bar{y}_j[k]/x_j[k]$ of each node $v_j$ for a digraph of size 6 for $L_j = L_{\text{max}} = 10$ and upper bound on the delay $\tau = 5$ for all nodes $v_j$.](image)

As it can be deduced from Fig. 3 it is not possible for any node in the network (in case it becomes curious) to infer the initial state for any other node in the network and average consensus is reached, even when there are time-varying delays in the network.

**Remark 5:** From the perspective of curious nodes things are getting more complicated, since packets are arriving to their destinations with delays. If we assume the extreme scenario that curious nodes also have knowledge of the delays (at each link, at each time step), then we can do a linear time-varying observability analysis (more complicated but similar to the linear time-invariant case of Remark 4) to fully understand the capability of the curious nodes.
V. CONCLUSIONS AND FUTURE DIRECTIONS

A. Conclusions

In this paper, we proposed a privacy-preserving discrete-time asymptotic average consensus mechanism that allows components of a multi-component system to calculate the exact average of their initial values without revealing to other components their specific value. Specifically, we propose a protocol based on ratio consensus, where each node updates its information state by combining the available information received by its in-neighbors using constant positive weights and by adding an offset. We establish that the proposed protocol converges to the exact average of the nodes’ initial values. We also extend the algorithm to account for bounded time-varying delays. This is the first privacy-preserving consensus algorithm proposed that handles delays. We have demonstrated the performance of our proposed protocol via illustrative examples.

B. Future Directions

Part of ongoing research is focusing on the development and analysis of distributed average consensus algorithms that also determine when to terminate the process [33]–[36], while ensuring at the same time that their privacy is preserved.

REFERENCES