Quantized Privacy-Preserving Algorithms for Homogeneous and Heterogeneous Networks With Finite Transmission Guarantees

Apostolos I. Rikos, *Member, IEEE*, Christoforos N. Hadjicostis, *Fellow, IEEE*, and Karl H. Johansson, *Fellow, IEEE*

***Abstract*—Privacy protection is increasingly critical in various applications due to the prevalence of networked systems. In this article, we focus on preserving the privacy of nodes' initial states while computing their average in a network. Curious nodes attempt to identify the initial states of other nodes without interfering in the computation. To address this challenge, we propose two privacy-preserving algorithms. The first algorithm operates over homogeneous networks, i.e., nodes have consistent processing delays and are able to communicate in a synchronous fashion. The second algorithm operates over heterogeneous networks, i.e., nodes have varying processing delays and communicate in an asynchronous fashion. Our algorithms exhibit efficient communication, finite-time convergence, and operation termination after convergence, making them suitable for resource-constrained environments. We also present topological conditions under which our algorithms enable privacy preservation. Finally, we apply our algorithms to a smart grid system and compare their performance against other algorithms, emphasizing their advantages in communication efficiency, convergence rate, and privacy preservation.**

***Index Terms*—Distributed algorithms, finite-time convergence, finite transmission, privacy-preserving average consensus, quantized communication, smart grid.**

Received 10 June 2024; revised 5 August 2024; accepted 8 September 2024. Date of publication 25 September 2024; date of current version 20 March 2025. This work was supported in part by Knut and Alice Wallenberg Foundation and in part by Swedish Research Council. An earlier version of this article was presented in part at the 2022 IEEE 61st Conference on Decision and Control (CDC) [DOI: 10.1109/CDC51059.2022.9993299]. Recommended by Associate Editor Y. Wang. *(Corresponding author: Apostolos I. Rikos.)*

Apostolos I. Rikos is with the Artificial Intelligence Thrust of the Information Hub, Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511453, China, and also with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong (e-mail: apostolosr@hkust-gz.edu.cn).

Christoforos N. Hadjicostis is with the Department of Electrical and Computer Engineering, University of Cyprus, 1678 Nicosia, Cyprus (e-mail: hadjicostis.christoforos@ucy.ac.cy).

Karl H. Johansson is with the Division of Decision and Control Systems, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden, and also with Digital Futures, SE-100 44 Stockholm, Sweden (e-mail: kallej@kth.se).

Digital Object Identifier 10.1109/TCNS.2024.3468990

I. Introduction

NETWORKED systems consist of interconnected nodes, such as sensors and computers, working together to solve specific problems [2]. They are commonly used in applications where fixed communication infrastructure is unavailable. The heterogeneity of nodes (with varying capabilities and characteristics) poses challenges for energy efficiency and system performance. For instance, nodes with higher energy consumption can lead to premature battery depletion, reducing the network's operational lifespan. Extending node lifetime has thus become a significant area of research. Various techniques (e.g., event-triggered operations, transmission stopping guarantees, and quantized processing and communication) have been explored to reduce energy consumption and improve energy efficiency in networked systems [3]. Overall, their aim is to optimize energy usage and ultimately enhance system performance.

The unattended operation of networked systems in hostile environments, while advantageous, poses privacy risks from curious nodes aiming to extract sensitive data. This leads to potential financial, operational, and reputational damages. The heterogeneity of nodes in such systems further complicates security and privacy measures. Thus, researchers have developed mechanisms to mitigate these privacy vulnerabilities in networked systems. One common approach involves injecting noise into message exchanges, making it difficult for curious nodes to discern patterns or extract specific information from the messages [4], [5], [6], [7], [8], [9], [10]. State decomposition [11] and random coupling weights [12] are other strategies to enhance privacy. In addition, hot-pluggable methods [13] and two-phase algorithms for data aggregation [14] have been proposed to protect sensitive information. Encryption techniques offer security by rendering data unreadable without the decryption key [15], [16], [17]. Some recent works include privacy-preserving algorithms with quantized communication [18], [19], [20], [21], which optimize efficiency and convergence. However, most algorithms in the literature operate over homogeneous networks with nodes exchanging real-valued messages without any transmission stopping guarantees. As a result, the development of privacy-preserving algorithms for energy-constrained nodes and heterogeneous networks remains an area largely unexplored.

The main contributions of this article are the following. We present two novel privacy-preserving distributed event-triggered

2325-5870 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on December 05,2025 at 15:14:25 UTC from IEEE Xplore. Restrictions apply.

algorithms that operate with quantized values over homogeneous or heterogeneous networks, and are able to calculate the exact average of the initial states in a privacy-preserving manner (see Algorithms 1 and 2). We show that both our proposed privacy-preserving algorithms converge after a finite number of iterations (for which we provide polynomial upper bounds), and that they exhibit transmission stopping guarantees (see Theorems 1 and 3). We also present sufficient topological conditions that ensure privacy preservation for the nodes that follow our proposed algorithms (see Theorems 2 and 4). Finally, we provide a case study of our algorithms applied to a smart grid system and demonstrate their ability to successfully achieve the desired objectives (see Section VII).

The algorithm discussed in [22] serves as the backbone for our privacy-preserving strategies. However, algorithm in [22] is not directly related to our privacy algorithms as it is not privacy-preserving. Instead, our privacy strategies are designed to leverage the characteristics of [22], such as its finite-time operation, transmission ceasing capabilities, and the use of quantized values. In addition, our privacy strategies adapt the event trigger conditions from [22] to achieve the desired summation while ensuring privacy preservation.

II. Notation and Background

The sets of real, rational, natural, integer, and nonnegative integer numbers are denoted by $\mathbb{R}, \mathbb{Q}, \mathbb{N}, \mathbb{Z}$, and $\mathbb{Z}_+$, respectively.

A. Graph-Theoretic Notions

Consider a network of n ($n \geq 2$) agents communicating only with their immediate neighbors. The communication topology can be captured by a directed graph (digraph) defined as $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} - \{(v_j, v_j) \mid v_j \in \mathcal{V}\}$ is the set of edges (self-edges excluded). A directed edge from node v_i to node v_j is denoted by $m_{ji} \triangleq (v_j, v_i) \in \mathcal{E}$, and captures the fact that node v_j can receive information from node v_i (but not the other way around). We assume that the given digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ is *strongly connected* (i.e., for each pair of nodes $v_j, v_i \in \mathcal{V}$, $v_j \neq v_i$, there exists a directed *path* from v_i to v_j). The subset of nodes that can directly transmit information to node v_j is called the set of in-neighbors of v_j and is represented by $\mathcal{N}_j^- = \{v_i \in \mathcal{V} \mid (v_j, v_i) \in \mathcal{E}\}$. The subset of nodes that can directly receive information from node v_j is called the set of out-neighbors of v_j and is represented by $\mathcal{N}_j^+ = \{v_l \in \mathcal{V} \mid (v_l, v_j) \in \mathcal{E}\}$. The cardinality of $\mathcal{N}_j^-$ is called the *in-degree* of v_j and is denoted by $\mathcal{D}_j^- = |\mathcal{N}_j^-|$. The cardinality of $\mathcal{N}_j^+$ is called the *out-degree* of v_j and is denoted by $\mathcal{D}_j^+ = |\mathcal{N}_j^+|$.

B. Node Variable Notation

With respect to quantization of information flow, we have that at time step $k \in \mathbb{Z}_+$, each node $v_j \in \mathcal{V}$ maintains 1) the state variables $y_j^s[k], z_j^s[k]$, and $q_j^s[k]$ (where $y_j^s[k] \in \mathbb{Z}, z_j^s[k] \in \mathbb{Z}_+$, and $q_j^s[k] = \frac{y_j^s[k]}{z_j^s[k]}$); 2) the mass variables $y_j[k]$ and $z_j[k]$, (where $y_j[k] \in \mathbb{Z}$ and $z_j[k] \in \mathbb{Z}_+$); 3) the substate counter s_j (where $s_j \in \mathbb{N}$); 4) the privacy variables $u_j^y[s_j]$ and $u_j^z[s_j]$ (where $u_j^y[s_j] \in \mathbb{Z}$ and $u_j^z[s_j] \in \mathbb{Z}$); and 5) the transmission variables S_br_j and M_tr_j (where $S\_br_j \in \mathbb{N}$ and $M\_tr_j \in \mathbb{N}$). Note here that for every node v_j, the state variables $y_j^s[k], z_j^s[k]$, and $q_j^s[k]$ are used to store the received messages and calculate the quantized average of the initial values; the mass variables $y_j[k]$ and $z_j[k]$ are used to communicate with other nodes by either transmitting or receiving messages; the substate counter s_j is used to transmit the privacy variables; the privacy variables $u_j^y[s_j]$, $u_j^z[s_j]$, $u_j^{y+}[s_j]$, $u_j^{z+}[s_j]$, and u_{ji}^{y-} for every $v_i \in \mathcal{N}_j^-$ are used to preserve the privacy of the initial state; and the transmission variables S_br_j (or M_tr_j) are used to decide whether the state variables will be broadcasted (or the mass variables will be transmitted).

C. Node Transmission Strategy

In our framework, each node has knowledge of its out-neighbors and can send messages directly to them. However, it cannot necessarily receive directly messages from them. In our proposed algorithms, each node v_j assigns a unique order in the set $0, 1, \ldots, \mathcal{D}_j^+ - 1$ to each of its outgoing edges m_{lj}, where $v_l \in \mathcal{N}_j^+$. This predetermined order P_{lj} is utilized during the execution of our distributed algorithms to enable each node v_j to directly transmit messages to its out-neighbors in a *round-robin* fashion. This means that each node v_j transmits to its out-neighbors one at a time, following the predetermined order. The next time it performs a transmission, it continues from the outgoing edge it stopped the previous time and cycles through the edges in a round-robin fashion according to the predetermined order P_{lj}.

III. Problem Formulation

Let us consider a strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$. At time step $k = 0$, each node $v_j \in \mathcal{V}$ has an initial quantized value $y_j[0]$ (for simplicity $y_j[0] \in \mathbb{Z}$). The node set $\mathcal{V}$ is partitioned into three subsets: 1) the subset of nodes $v_j \in \mathcal{V}_p \subset \mathcal{V}$ that wish to preserve their privacy by not revealing their initial states $y_j[0]$ to other nodes, 2) the subset of nodes $v_c \in \mathcal{V}_c \subset \mathcal{V}$ that are curious and try to identify the initial states $y[0]$ of all (or a subset of) nodes in the network, and 3) the rest of the nodes $v_i \in \mathcal{V}_n \subset \mathcal{V}$ that neither wish to preserve their privacy nor to identify the states of other nodes. We assume that curious nodes in $\mathcal{V}_c$ can collaborate arbitrarily in order to identify the initial states of other nodes.

In this article, we develop two distributed algorithms that allow nodes to address the Problems **P1** and **P2**, while processing and transmitting *quantized* information.

P1—*Every node:*

a. $v_j \in \mathcal{V}$ obtains, after a finite number of steps, a fraction q_j^s, which is equal to the *exact* average q of the initial states of the nodes, given as

$$q = \frac{\sum_{l=1}^{n} y_l[0]}{n}. \tag{1}$$

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on December 05,2025 at 15:14:25 UTC from IEEE Xplore. Restrictions apply.

Specifically, we argue that there exists k_0 so that for every $k \geq k_0$ we have

$$y_j^s[k] = \frac{\sum_{l=1}^n y_l[0]}{\alpha} \text{ and } z_j^s[k] = \frac{n}{\alpha} \tag{2}$$

where $\alpha \in \mathbb{N}$. This means that

$$q_j^s[k] = \frac{(\sum_{l=1}^n y_l[0])/\alpha}{n/\alpha} := q \tag{3}$$

for every $v_j \in \mathcal{V}$, i.e., for $k \geq k_0$, every node v_j has calculated q as the ratio of two integer values (and there is no quantization error).

b. $v_j \in \mathcal{V}_p$ preserves the privacy of its initial state $y_j[0]$ (i.e., curious nodes cannot determine a finite range in which the initial state $y_j[0]$ lies; see Definition 1) when it exchanges quantized information with neighboring nodes, while calculating q in (1) [so that eventually its state variables y_j^s, z_j^s, and q_j^s fulfill (2) and (3), respectively].
c. $v_j \in \mathcal{V}$ stops performing transmissions toward its out-neighbors $v_l \in \mathcal{N}_j^+$ soon after its state variables y_j^s, z_j^s, and q_j^s fulfill (2) and (3), respectively.
d. $v_j \in \mathcal{V}$ is able to perform transmissions in a *synchronous* manner with every other node in the network, i.e., the network is homogeneous.

P2—*Every node:*

a. Same as **P1–a.**
b. Same as **P1–b.**
c. Same as **P1–c.**
d. $v_j \in \mathcal{V}$ performs transmissions in an *asynchronous* manner with other nodes, i.e., the network is heterogeneous.

Concept of privacy: During the developments in this article, we define privacy as the protection of critical information that is stored, processed, or transmitted by a particular node within the network. We consider that the information of interest for each node $v_j \in \mathcal{V}$ is its initial state $y_j[0]$. The notion of privacy that we adopt aims to ensure that value $y_j[0]$ cannot be inferred by curious nodes with any given accuracy. This definition of privacy relates to notions of possible innocence in theoretical computer science [23], [24] in the sense that there is some uncertainty about $y_j[0]$.

Definition 1 *(Privacy definition [19]):* A node $v_j \in \mathcal{V}_p$ preserves the privacy of its initial state $y_j[0] \in \mathbb{Z}$ if $y_j[0]$ cannot be inferred by curious nodes $v_c \in \mathcal{V}_c$ at any point during the operation of the algorithm. More specifically, curious nodes in $\mathcal{V}_c$ cannot determine a finite range $[\alpha, \beta]$ (where $\alpha < \beta$ and $\alpha, \beta \in \mathbb{R}$) in which the initial state $y_j[0]$ lies.

Remark 1: Definition 1 is similar to the notion of privacy in [11]. It emphasizes that a curious node cannot even determine a finite range of values when attempting to estimate the states of other nodes. This definition is stronger compared to the one proposed in [4] and [7], where privacy is defined as the inability of curious nodes to uniquely identify the protected value. In addition, following Definition 1, the privacy algorithms proposed in the remainder of this article are robust to privacy leaks because curious nodes have to consider unbounded intervals when they attempt to estimate a finite range in which the initial states of the private nodes belong. This uncertainty makes it impossible for curious nodes to gain any useful information, effectively preventing any information leakage (at least in the absence of any probabilistic description of how initial values and other algorithmic parameters are chosen). Note here that an extended analysis on information leakage in the presence of probabilistic information and/or bounded ranges of legitimate initial values is not within the scope of this article, but will be explored in future work.

IV. Synchronous Privacy-Preserving Algorithm

In this section, we present a distributed algorithm that addresses problem **P1**, presented in Section III.

Assumption 1: Each node $v_j \in \mathcal{V}$ has knowledge of the maximum out-degree in the network $\mathcal{D}_{\max}^+ = \max_{v_i \in \mathcal{V}} \mathcal{D}_j^+$.

Assumption 1 is important for preserving privacy and guaranteeing convergence to the average of the initial states. In case Assumption 1 does not hold, then our algorithm may fail to preserve privacy and the nodes may converge to a value that is not equal to the average of the initial states (i.e., nodes simply achieve consensus). Note that nodes can obtain knowledge of the maximum out-degree in the network by executing a max-consensus algorithm for a finite number of time steps [25].

A. Privacy-Preserving Strategy for Synchronous Operation

Our strategy is based on [22] with some modifications (since the strategy in [22] is not privacy-preserving). Our privacy strategy requires each node $v_j \in \mathcal{V}_p$ to decompose its initial state $y_j[0]$ into $\mathcal{D}_{\max}^+ + 2$ substates, whose average is equal to v_j's initial state. One substate is utilized as the node's initial state. Then, the remaining substates are transmitted each to a different out-neighbor at a different time step, effectively preserving the privacy of the initial state $y_j[0]$. Furthermore, each node v_j maintains its substate counter $s_j \in \mathbb{N}$, and its privacy variables $u_j^y[s_j] \in \mathbb{Z}$ and $u_j^z[s_j] \in \mathbb{Z}$. At initialization, each node $v_j \in \mathcal{V}_p$ chooses the privacy variables $u_j^y[s_j] \in \mathbb{Z}$, $u_j^z[s_j] \in \mathbb{Z}$, for $s_j \in \{0, 1, 2, \ldots, \mathcal{D}_{\max}^+ + 1\}$, to satisfy the following constraints:

$$u_j^y[s_j] \in \mathbb{Z} \quad \forall\, s_j \in [0, \mathcal{D}_{\max}^+ + 1] \tag{4a}$$

$$u_j^y[s_j] = 0 \quad \forall\, s_j > \mathcal{D}_{\max}^+ + 1 \tag{4b}$$

$$u_j^z[s_j] = 1 \quad \forall\, s_j \in [0, \mathcal{D}_{\max}^+ + 1] \tag{4c}$$

$$u_j^z[s_j] = 0 \quad \forall\, s_j > \mathcal{D}_{\max}^+ + 1 \tag{4d}$$

$$y_j[0] = \frac{\sum_{s_j=0}^{\mathcal{D}_{\max}^+ + 1} u_j^y[s_j]}{\mathcal{D}_{\max}^+ + 2}. \tag{4e}$$

Constraints (4a)–(4e) are explicitly described as follows.

1) In (4a), each substate $u_j^y[s_j]$ needs to have a quantized value.
2) In (4b), each node v_j stops injecting nonzero substates after $\mathcal{D}_{\max}^+ + 2$ time steps in order not to intervene with the calculation of the quantized average. This allows each node to calculate the exact quantized average of the initial states without any error.
3) In (4c), the substate $u_j^z[s_j]$, which is injected to the network by node v_j, needs to be equal to 1 so that a) the event-triggered conditions of the presented algorithm hold and

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on December 05,2025 at 15:14:25 UTC from IEEE Xplore. Restrictions apply.

b) the operation of the algorithm leads to the calculation of the exact average.

4) In (4d), each node v_j stops injecting nonzero substates after $\mathcal{D}^+_{\max}+1$ time steps, which allows the calculation of the quantized average without any error.
5) In (4e), the average of the total injected substates in the network by node v_j needs to be equal to node v_j's initial state $y_j[0]$. This means that each node v_j creates $\mathcal{D}^+_{\max}+2$ substates of its initial state, which have arbitrary $u^y_j[s_j]$ values. These substates allow the calculation of the exact quantized average of the initial states without any error.

The above choices imply that each node $v_j \in \mathcal{V}_p$ generates $\mathcal{D}^+_{\max}+2$ substates $u^y_j[s_j], u^z_j[s_j]$ of its initial state $y_j[0]$. Then, during each time step $s_j \in [0, \mathcal{D}^+_{\max}+1]$, node v_j injects in the network the substates $u^y_j[s_j]$ and $u^z_j[s_j]$. This leads to the calculation of the exact quantized average in a privacy-preserving manner. Note that each node $v_i \notin \mathcal{V}_p$, which does not wish to preserve its privacy sets $u^y_j[s_j] = y_i[0]$, for every $s_i \in [0, \mathcal{D}^+_{\max}+1]$, and follows the same operation.

B. Synchronous Privacy-Preserving Algorithm With Multiple State Decomposition and Finite Transmissions

The details of the synchronous privacy-preserving distributed algorithm can be seen in Algorithm 1.

The intuition behind Algorithm 1 is the following. Initially, each node decomposes its initial state into multiple substates according to (4a)–(4e). The number of substates of each node is $\mathcal{D}^+_{\max}+2$. Each node uses the first substate as its initial state, and performs obligatory directed transmissions for $\mathcal{D}^+_{\max}+1$ time steps. During these transmissions, the remaining substates are being transmitted, each to a different out-neighbor. After the obligatory directed transmissions have been performed, each node executes the underlying averaging algorithm [22]. During the execution of the algorithm in [22], the initial states of nodes are collected and aggregated to one node (or some nodes) in the network. Then, this node (or nodes) informs the other nodes of the exact average. Communication and processing of nodes rely on the event-triggered conditions in Algorithm 1.A. For more details, refer to [22].

C. Convergence Analysis of Algorithm 1

Before analyzing the deterministic convergence of Algorithm 1, we consider the following setup.

Setup: Consider a strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes and $m = |\mathcal{E}|$ edges. During the execution of Algorithm 1, at time step k_0, there is at least one node $v_{j'} \in \mathcal{V}$, for which

$$z_{j'}[k_0] \geq z_i[k_0] \quad \forall v_i \in \mathcal{V}. \tag{5}$$

Then, among the nodes $v_{j'}$ for which (5) holds, there is at least one node v_j for which

$$y_j[k_0] \geq y_l[k_0], \text{ where } v_j, v_l \in \{v_{j'} \in \mathcal{V} \mid (5) \text{ holds}\}. \tag{6}$$

For notational convenience, we will call the mass variables of node v_j for which (5) and (6) hold as the "leading mass" (or "leading masses").

Algorithm 1: Synchronous Privacy-Preserving Algorithm With Multiple State Decomposition and Finite Transmissions.

Input: A strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes and $m = |\mathcal{E}|$ edges. Each node $v_j \in \mathcal{V}$ has an initial state $y_j[0] \in \mathbb{Z}$ and Assumption 1 holds.
Output: (3) holds for every $v_j \in \mathcal{V}$.
Initialization: Each node $v_j \in \mathcal{V}$ does the following:

1) Assigns to each outgoing edge $v_l \in \mathcal{N}^+_j$ a unique order P_{lj} in the set $\{0, 1, \ldots, \mathcal{D}^+_j - 1\}$.
2) Chooses $u^y_j[s_j], u^z_j[s_j]$ according to (4a)–(4e).
3) Sets counter $s_j = 0$.
4) Sets $y_j[0] = u^y_j[s_j]$, $z_j[0] = u^z_j[s_j]$, $z^s_j[0] = z_j[0]$, $y^s_j[0] = y_j[0]$, $q^s_j[0] = y^s_j[0]/z^s_j[0]$, $s_j = s_j + 1$ and $S\_br_j = 0$, $M\_tr_j = 0$.
5) Broadcasts $z^s_j[0], y^s_j[0]$ to every $v_l \in \mathcal{N}^+_j$.

Iteration: For $k = 0, 1, 2, \ldots$, each node $v_j \in \mathcal{V}$:

1) Receives $y^s_i[k], z^s_i[k]$ from every $v_i \in \mathcal{N}^-_j$ (if no message is received it sets $y^s_i[k] = 0$, $z^s_i[k] = 0$).
2) Receives $y_i[k], z_i[k]$ from each $v_i \in \mathcal{N}^-_j$ and sets

$$y_j[k+1] = y_j[k] + \sum_{v_i \in \mathcal{N}^-_j} w_{ji}[k] y_i[k],$$

$$z_j[k+1] = z_j[k] + \sum_{v_i \in \mathcal{N}^-_j} w_{ji}[k] z_i[k],$$

where $w_{ji}[k] = 1$ if a message with $y_i[k], z_i[k]$ is received from in-neighbor v_i, otherwise $w_{ji}[k] = 0$.
3) **If** $w_{ji}[k] \neq 0$ or $z^s_i[k] \neq 0$ for some $v_i \in \mathcal{N}^-_j$ **then** sets $z^s_j[k+1] = z^s_j[k]$, $y^s_j[k+1] = y^s_j[k]$, and calls Algorithm (1.A).
4) Sets $M\_tr_j = \max\{M\_tr_j, u^z_j[s_j]\}$.
5) **If** $M\_tr_j = 1$ **then** (i) sets $y_j[k+1] = y_j[k+1] + u^y_j[s_j]$, $z_j[k+1] = z_j[k+1] + u^z_j[s_j]$, and (ii) chooses $v_l \in \mathcal{N}^+_j$ according to P_{lj} (in a round-robin fashion) and transmits $y_j[k+1], z_j[k+1]$. Then, sets $y_j[k+1] = 0$, $z_j[k+1] = 0$, $M\_tr_j = 0$, $s_j = s_j + 1$.
6) **If** $S\_br_j = 1$ **then** broadcasts $z^s_j[k+1], y^s_j[k+1]$ to every $v_l \in \mathcal{N}^+_j$; then, sets $S\_br_j = 0$.
7) Repeats (increases k to $k+1$ and goes to Step 1).

Lemma 1 (See [22]): If, during time step k_0 of Algorithm 1, the mass variables of node v_j fulfill (5) and (6), then the state variables of every node $v_i \in \mathcal{V}$ satisfy

$$z^s_i[k_0] \leq z_j[k_0] \tag{7}$$

or

$$z^s_i[k_0] = z_j[k_0] \text{ and } y^s_i[k_0] \leq y_j[k_0]. \tag{8}$$

Lemma 2 (See [22]): If, during time step k_0 of Algorithm 1, the mass variables of *each* node v_j with nonzero mass variables fulfill (5) and (6), then we have only "leading masses" and no "follower masses." This means that the "Event Trigger

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on December 05,2025 at 15:14:25 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1.A: Event-Triggered Conditions for Algorithm 1 (For Each Node v_j).

Input: $y_j^s[k]$, $z_j^s[k]$, $q_j^s[k]$, $y_j^s[k+1]$, $z_j^s[k+1]$, $y_j[k+1]$, $z_j[k+1]$, S_br_j, M_tr_j and the received $y_i^s[k]$, $z_i^s[k]$ from every $v_i \in \mathcal{N}_j^-$.
Output: $y_j^s[k+1]$, $z_j^s[k+1]$, $q_j^s[k+1]$, S_br_j, M_tr_j.
Execution: Node v_j checks:
1) Event Trigger Conditions 1: **If**
Condition (i): $z_i^s[k] > z_j^s[k]$, or
Condition (ii): $z_i^s[k] = z_j^s[k]$ and $y_i^s[k] > y_j^s[k]$,
then sets

$$z_j^s[k+1] = \max_{v_i \in \mathcal{N}_j^-} z_i^s[k], \text{ and}$$

$$y_j^s[k+1] = \max_{v_i \in \{v_{i'} \in \mathcal{N}_j^- | z_{i'}^s[k] = z_j^s[k+1]\}} y_i^s[k],$$

and sets $q_j^s[k+1] = \frac{y_j^s[k+1]}{z_j^s[k+1]}$, and $S\_br_j = 1$.
2) Event Trigger Conditions 2: **If**
Condition (i): $z_j[k+1] > z_j^s[k+1]$, or
Condition (ii): $z_j[k+1] = z_j^s[k+1]$ and $y_j[k+1] > y_j^s[k+1]$,
then sets $z_j^s[k+1] = z_j[k+1]$, $y_j^s[k+1] = y_j[k+1]$ and sets $q_j^s[k+1] = \frac{y_j^s[k+1]}{z_j^s[k+1]}$ and $S\_br_j = 1$.
3) Event Trigger Conditions 3: **If**
Condition (i): $0 < z_j[k+1] < z_j^s[k+1]$ or
Condition (ii): $z_j[k+1] = z_j^s[k+1]$ and $y_j[k+1] < y_j^s[k+1]$,
then sets $M\_tr_j = 1$.

Conditions 2" will never hold again for future time steps $k \geq k_0$. As a result, the transmissions that (may) take place will only be via broadcasting (from "Event Trigger Conditions 1 and 3") for at most $n-1$ time steps and then they will cease.

The following theorem characterizes the convergence of Algorithm 1. Its proof can be found in Appendix A.

Theorem 1: Consider a strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes and $m = |\mathcal{E}|$ edges. The execution of Algorithm 1 allows each node $v_j \in \mathcal{V}$ to calculate the exact average of the initial states after a finite number of time steps k_0 upper bounded by $1 + \mathcal{D}_{\max}^+ + n^2 + (n-1)m^2$, where $\mathcal{D}_{\max}^+$ is the maximum out-degree in the network. Furthermore, each node stops transmitting once quantized average consensus is reached.

D. Topological Conditions for Privacy Preservation of Algorithm 1

Theorem 2: Consider a fixed strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes. Assume that a subset of nodes $v_j \in \mathcal{V}_p$ follow Algorithm 1, where they choose the set of substates as in (4a)–(4e). Curious nodes $v_c \in \mathcal{V}_c$ will not be able to determine a finite range in which the initial state $y_j[0]$ lies as long as v_j has connected to it at least one in- or out-neighbor $v_\ell \in \mathcal{V}_p$ that aims to preserve its privacy.

Proof: Our proof establishes sufficient topological conditions to ensure privacy preservation.

A. Suppose one out-neighbor of node v_j, say $v_{l'}$, is following the privacy-preserving strategy (i.e., $v_{l'} \in \mathcal{V}_p$) and (as a worst-case assumption) all other in- and out-neighbors of both nodes v_j and $v_{l'}$ are curious (i.e., $v_i \in \mathcal{V}_c \, \forall v_i \in (\mathcal{N}_j^- \cup \mathcal{N}_{l'}^-) \setminus \{v_j, v_{l'}\}$, and $v_l \in \mathcal{V}_c \, \forall v_l \in (\mathcal{N}_j^+ \cup \mathcal{N}_{l'}^+) \setminus \{v_j, v_{l'}\}$). During the Iteration procedure, curious nodes will not be able to infer the substates transmitted from node v_j to node $v_{l'}$. Furthermore, curious nodes will not be able to infer the substates of node $v_{l'}$ that are summed with the substates received from node v_j (and then submitted to $v_{l'}$'s out-neighbors). As a result, curious nodes will not be able to infer the initial state of node v_j *or* the initial state of node $v_{l'}$. Thus, in this case, node v_j preserves the privacy of its initial state.

B. The case where only one in-neighbor of v_j, say $v_{i'}$, is following the privacy-preserving strategy and all other in- and out-neighbors of v_j and $v_{i'}$ are curious can be analyzed as Case **A**.

From Cases **A** and **B**, we have that a node $v_j \in \mathcal{V}_p$ is able to preserve its privacy if it has at least one in- or out-neighbor (say $v_{i'}$ or $v_{l'}$) who also wants to preserve its privacy and follows the proposed privacy-preserving strategy. Furthermore, curious nodes will not be able to determine 1) the values of the messages transmitted from $v_{i'}$ to v_j, and 2) the values of the messages transmitted from v_j to $v_{l'}$. This means that curious nodes will not be able to determine a finite range $[\alpha, \beta]$ (where $\alpha < \beta$ and $\alpha, \beta \in \mathbb{R}$) in which the initial state $y_j[0]$ lies (as already mentioned in Definition 1).

Remark 2 (Inability of Algorithm 1 to Preserve Privacy): If the conditions in Theorem 2 do not hold, then privacy may not be preserved. Two topological instances where nodes are unable to preserve their privacy are the following.

1) Let us suppose that all in- and out-neighbors of node v_j are curious (i.e., $v_i \in \mathcal{V}_c \, \forall v_i \in \mathcal{N}_j^-$, and $v_l \in \mathcal{V}_c \, \forall v_l \in \mathcal{N}_j^+$). Since the curious in- and out-neighbors can communicate with each other, node $v_j \in \mathcal{V}_p$ will not be able to preserve its privacy. At Initialization, curious nodes will know $u_j^y[0]$. Then, during the Iteration procedure, curious nodes will know the messages v_j has received and transmitted. This means that they will be able to determine the values of $u_j^y[s_j] \in \mathbb{Z}$, for $s_j \in \{0, 1, \ldots, \mathcal{D}_{\max}^+ + 1\}$. Note that the average of every $u_j^y[s_j]$, for $s_j \in \{0, 1, 2, \ldots, \mathcal{D}_{\max}^+ + 1\}$, is equal to v_j's initial state $y_j[0]$. This means that curious nodes will be able to determine the initial state $y_j[0]$.
2) Let us suppose that one out-neighbor of node v_j, say $v_{l'}$, is neither curious nor following the privacy-preserving strategy (i.e., $v_{l'} \in \mathcal{V}_n$), and all other in- and out-neighbors of both nodes v_j and $v_{l'}$ are curious (i.e., $v_i \in \mathcal{V}_c \, \forall v_i \in (\mathcal{N}_j^- \cup \mathcal{N}_{l'}^-) \setminus \{v_j, v_{l'}\}$, and $v_l \in \mathcal{V}_c \, \forall v_l \in (\mathcal{N}_j^+ \cup \mathcal{N}_{l'}^+) \setminus \{v_j, v_{l'}\}$). During the Initialization procedure, curious nodes will know $y_{l'}[0]$, and the messages v_j has received and transmitted (with the exception of the transmissions to $v_{l'}$). However, curious nodes can infer the input that node $v_{l'}$ received from v_j based on its output. Then, they will be able to extract the messages of node v_j (as if a curious node was directly connected to node v_j). Thus,

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on December 05,2025 at 15:14:25 UTC from IEEE Xplore. Restrictions apply.

in this case, v_j does not preserve the privacy of its initial state.

Remark 3: Note here that decomposing the initial state of every $v_j \in \mathcal{V}_p$ into $\mathcal{D}^+_{\max}+2$ substates is essential for privacy preservation. For every node, the first substate is used as the initial state. Then, every node transmits the remaining $\mathcal{D}^+_{\max}+1$ substates toward its out-neighbors. This means that v_j transmits at least one set of privacy variables to each out-neighbor. As a result, if $v_{l'} \in \mathcal{V}_p$ (where $v_{l'} \in \mathcal{N}^+_j$), then $v_{l'}$ receives (and sums with its own mass variables) at least one set of v_j's private substates before it transmits every set of its own private variables toward its out-neighbors.

V. Asynchronous Privacy-Preserving Algorithm

In this section, we present a distributed algorithm that addresses problem **P2**, presented in Section III.

Assumption 2: At each time step k, each node $v_j \in \mathcal{V}$ undergoes an a priori unknown processing delay $\tau_j[k]$ for which we have $0 \leq \tau_j[k] \leq \overline{\tau} < \infty \forall v_j \in \mathcal{V}$ (i.e., delays are bounded).

Assumption 2 implies that each node's state is obtained at a random future time step, accounting for delays introduced by the necessary processing operations. This assumption is important for guaranteeing convergence to the average of the initial states. In case Assumption 2 does not hold, this means that node v_l may process for infinite time, never transmitting its subsets in the network. In this case, our algorithm may converge to a value that is not equal to the average of the initial states.

A. Main Idea of Asynchronous Privacy-Preserving Algorithm

Realistic computer networks often consist of interconnected nodes and links of various types, and are known as heterogeneous networks [26], [27]. These networks connect devices where the operating capabilities and protocols have significant differences. In distributed computing over heterogeneous networks, synchronizing the operation of multiple nodes may be inefficient. For this reason, there is a growing need for asynchronous algorithms for distributed networks, as they offer greater flexibility, scalability, and efficiency compared to synchronous algorithms.

The main idea of this section is to design a privacy preservation strategy that can operate over heterogeneous networks. In heterogeneous networks, nodes may have different processing capabilities and thus they may perform transmissions in an *asynchronous* fashion. The operation of Algorithm 1 and its privacy-preserving strategy (described in Section IV-A) assumes that nodes perform transmissions in a synchronous fashion, and may fail when Algorithm 1 operates over heterogeneous networks. To justify this statement, we examine the following scenario: let us consider node $v_j \in \mathcal{V}_p$ and an out-neighbor of node v_j, say $v_{l'}$, that is following the privacy-preserving strategy (i.e., $v_{l'} \in \mathcal{V}_p$), whereas all other in- and out-neighbors of both nodes v_j and $v_{l'}$ are curious (i.e., $v_i \in \mathcal{V}_c \ \forall v_i \in (\mathcal{N}^-_j \cup \mathcal{N}^-_{l'}) \setminus \{v_j, v_{l'}\}$, and $v_l \in \mathcal{V}_c \ \forall v_l \in (\mathcal{N}^+_j \cup \mathcal{N}^+_{l'}) \setminus \{v_j, v_{l'}\}$). If both nodes v_j and $v_{l'}$ do not suffer from processing delays, then according to Theorem 2, node v_j will be able to preserve the privacy of its initial state. However, let us now consider that, during every time step k, the operation of nodes v_j and $v_{l'}$ undergoes a priori unknown processing delays $\tau_j[k]$ and $\tau_{l'}[k]$, respectively, and $\tau_j[k] \gg \tau_{l'}[k]$. In this case, if nodes v_j and $v_{l'}$ execute Algorithm 1, then node $v_{l'}$ may transmit all its $\mathcal{D}^+_{\max}+1$ subsets before it receives a message from node v_j. From the proof of Theorem 2, Case A, let us recall that node v_j is able to preserve its privacy if curious nodes are not able to infer the substate of node $v_{l'}$ that is summed with the substate received from node v_j (and then submitted to $v_{l'}$'s out-neighbors). However, when $\tau_j[k] \gg \tau_{l'}[k]$, node $v_{l'}$ may transmit all its substates before receiving any message from v_j (i.e., $v_{l'}$ will not be able to sum its substates with the substates received from node v_j). As a result, in this case, node v_j will not be able to preserve the privacy of its initial state (see Remark 2).

Considering the above limitation of Algorithm 1, we propose a privacy-preserving algorithm for asynchronous node operation over heterogeneous networks. The main idea is the following: each node $v_{l'} \in \mathcal{V}_p$ decomposes its initial state into multiple substates. One substate is used as the initial state, and the other substates are assigned to each *incoming and outgoing* edge (a key difference from Algorithm 1). It transmits the substates assigned to its outgoing edges toward its out-neighbors. The substates assigned to its incoming edges are summed with the messages received from the corresponding edge. In this way, node $v_{l'}$ is able to sum its substates with the substates received from in-neighbor v_j regardless of the time step at which v_j performs a transmission toward $v_{l'}$ (i.e., regardless of the processing delays). Our privacy algorithm is designed for heterogeneous networks provided that the processing delays of every node are bounded (see Assumption 2).

B. Privacy-Preserving Strategy for Asynchronous Operation

Our strategy is again based on the event-triggered deterministic algorithm in [22]. In our strategy, we require each node $v_j \in \mathcal{V}_p$ to decompose its initial state $y_j[0]$ into $\mathcal{D}^+_j + \mathcal{D}^-_j + 1$ substates whose sum is equal to the initial state. Each of the $\mathcal{D}^+_j$ substates is transmitted to a different out-neighbor at a different time step. Each node v_j maintains a substate counter $s_j \in \mathbb{N}$, and its privacy variables $u^{y+}_j[s_j], u^{z+}_j[s_j] \in \mathbb{Z}$, $v_l \in \mathcal{N}^+_j$ and $u^{y-}_{ji} \in \mathbb{Z}$ for every $v_i \in \mathcal{N}^-_j$. At initialization, each node $v_j \in \mathcal{V}_p$ chooses the privacy variables $u^{y+}_j[s_j], u^{z+}_j[s_j]$, and u^{y-}_{ji} to satisfy the following constraints:

$$u^{y+}_j[s_j] \in \mathbb{Z} \quad \forall s_j \in [0, \mathcal{D}^+_j] \tag{9a}$$

$$u^{y-}_{ji} \in \mathbb{Z} \quad \forall v_i \in \mathcal{N}^-_j \tag{9b}$$

$$u^{y+}_j[s_j] = 0 \quad \forall s_j > \mathcal{D}^+_j, \text{ and } u^{y-}_{ji} = 0, \ \forall v_i \notin \mathcal{N}^-_j \tag{9c}$$

$$u^{z+}_j[0] = 1 \tag{9d}$$

$$u^{z+}_j[s_j] = 0 \quad \forall s_j > 0 \tag{9e}$$

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on December 05,2025 at 15:14:25 UTC from IEEE Xplore. Restrictions apply.

$$y_j[0] = \sum_{s_j=0}^{\mathcal{D}_j^+} u_j^{y+}[s_j] + \sum_{i=1}^{n} u_{ji}^{y-}. \tag{9f}$$

Constraints (9a)–(9f) are explicitly analyzed as follows.

1) In (9a), each substate $u_j^{y+}[s_j]$ has a quantized value.
2) In (9b), each substate u_{ji}^{y-} has a quantized value.
3) In (9c), each node v_j stops injecting nonzero substates after $\mathcal{D}_j^+$ time steps in order not to intervene with the calculation of the quantized average.
4) In (9d), each substate $u_j^{z+}[s_j]$ is used to inject a nonzero substate $u_j^{y+}[s_j]$ for the first time step.
5) In (9e), each $u_j^{z+}[s_j]$ becomes equal to zero for $s_j > 0$ because the injection of the nonzero substates $u_j^{y+}[s_j]$ has been completed.
6) In (9f), the sum of the total injected substates in the network by node v_j needs to be equal to node v_j's initial state $y_j[0]$.

According to the above choices, each node $v_j \in \mathcal{V}_p$ that wishes to preserve its privacy generates $\mathcal{D}_j^+ + 1$ substates $u_j^{y+}[s_j]$ and $\mathcal{D}_j^-$ substates u_{ji}^{y-} of its initial state $y_j[0]$, for which (9a)–(9f) hold. One substate is used as the initial state of v_j. Then, v_j injects the substates 1) $u_j^{y+}[s_j]$, $s_j \in [0, \mathcal{D}_j^+]$, to the transmitted messages, and 2) u_{ji}^{y-} $\forall v_i \in \mathcal{N}_j^-$, to the received messages (each u_{ji}^{y-} is injected to the message received from node v_i). Furthermore, each node $v_i \notin \mathcal{V}_p$, which does not wish to preserve its privacy sets $u_i^{y+}[s_i] = \frac{y_i[0]}{\mathcal{D}_j^+ + \mathcal{D}_j^- + 1}$, for every $s_i \in [0, \mathcal{D}_i^+]$, and $u_{ii'}^{y-} = \frac{y_i[0]}{\mathcal{D}_j^+ + \mathcal{D}_j^- + 1}$, $\forall v_i' \in \mathcal{N}_i^-$.

C. Asynchronous Privacy-Preserving Algorithm With Multiple State Decomposition and Finite Transmissions

The details of the asynchronous privacy-preserving distributed algorithm can be seen in Algorithm 2.

The intuition of Algorithm 2 is the following. Initially, each node decomposes its initial state into multiple substates according to (9a)–(9f). The number of substates is equal to one plus the number of outgoing and incoming edges of the node. The first substate is used as the initial state, and the rest are assigned to every outgoing and every incoming edge. Each node performs obligatory directed transmissions for a specific number of time steps equal to the node's outgoing edges. During these transmissions, the substates assigned to the outgoing edges are sent. If a node receives a transmission from an in-neighbor, it adds the substate assigned to the corresponding incoming edge. After the obligatory transmissions, each node executes underlying averaging algorithm [22].

D. Convergence Analysis of Algorithm 2

The following theorem characterizes the convergence of Algorithm 2. Its proof can be found in Appendix B.

Theorem 3: Consider a strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes and $m = |\mathcal{E}|$ edges for which Assumption 2 holds. The execution of Algorithm 2 allows each node $v_j \in \mathcal{V}$ to calculate the exact average of the initial states after a finite number of time steps k_0, upper bounded by $\overline{\tau}(\mathcal{D}_{\max}^+ + n^2 + (n-1)m^2)$, where $\mathcal{D}_{\max}^+$ is the maximum out-degree in the network. Furthermore, each node stops transmitting toward its out-neighbors once quantized average consensus is reached.

Algorithm 2: Asynchronous Privacy-Preserving Algorithm With Multiple State Decomposition and Finite Transmissions.

Input: Same as Input of Algorithm 1, Assumption 2 holds, Assumption 1 need not to hold.
Output: (3) holds for every $v_j \in \mathcal{V}$.
Initialization: Each node $v_j \in \mathcal{V}$ does the following:
1) Same as Initialization-Step 1 of Algorithm 1.
2) Chooses $u_j^{y+}[s_j]$, $u_j^{z+}[s_j]$, u_{ji}^{y-} according to (9a)–(9f).
3) Same as Initialization-Step 3 of Algorithm 1.
4) Sets $y_j[0] = u_j^{y+}[s_j]$, $z_j[0] = 1$, $z_j^s[0] = z_j[0]$, $y_j^s[0] = y_j[0]$, $q_j^s[0] = y_j^s[0]/z_j^s[0]$, $s_j = s_j + 1$ and $S\_br_j = 0$, $M\_tr_j = 0$.
5) Same as Initialization-Step 5 of Algorithm 1.

Iteration: For $k = 0, 1, 2, \ldots$, each node $v_j \in \mathcal{V}$:
1) Same as Iteration-Step 1 of Algorithm 1.
2) Same as Iteration-Step 2 of Algorithm 1.
3) **If** $w_{ji}[k] \neq 0$ for one (or more) $v_i \in \mathcal{N}_i^-$, then sets $y_j[k+1] = y_j[k+1] + u_{ji}^{y-}$, and $u_{ji}^{y-} = 0$, for every $v_i \in \mathcal{N}_i^-$ for which $w_{ji}[k] \neq 0$.
4) Same as Iteration-Step 3 of Algorithm 1.
5) Sets $M\_tr_j = \max\{M\_tr_j, u_j^{z+}[s_j]\}$.
6) **If** $M\_tr_j = 1$ **then** (i) sets $y_j[k+1] = y_j[k+1] + u_j^{y+}[s_j]$, and (ii) chooses $v_l \in \mathcal{N}_j^+$ according to P_{lj} (in a round-robin fashion) and transmits $y_j[k+1]$, $z_j[k+1]$. Then, sets $y_j[k+1] = 0$, $z_j[k+1] = 0$, $M\_tr_j = 0$, $s_j = s_j + 1$.
7) Same as Iteration-Step 6 of Algorithm 1.
8) Repeats (increases k to $k+1$ and goes to Step 1).

E. Topological Conditions for Privacy Preservation of Algorithm 2

Theorem 4: Consider a fixed strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes for which Assumption 2 holds. Assume that a subset of nodes $v_j \in \mathcal{V}_p$ follow Algorithm 2 where they choose the set of substates as in (9a)–(9f). Curious nodes $v_c \in \mathcal{V}_c$ will not be able to determine a finite range in which the initial state $y_j[0]$ lies, as long as v_j has connected to it at least one in- or out-neighbor $v_\ell \in \mathcal{V}_p$ that aims to preserve its privacy.

Proof: Our proof examines various topological cases for Algorithm 2. By summarizing, we derive sufficient topological conditions that ensure privacy preservation. Without loss of generality, for the following scenarios, we consider the case $\tau_j[k] \ll \tau_{l'}[k]$ $\forall v_{l'} \in \mathcal{N}_j^+$, and $\tau_j[k] \ll \tau_{i'}[k]$ $\forall v_{i'} \in \mathcal{N}_j^-$, for all k. Note that the cases: 1) $\tau_j[k] \geq \tau_{l'}[k]$ and $\tau_j[k] \geq \tau_{i'}[k]$, 2) $\tau_j[k] \leq \tau_{l'}[k]$ and $\tau_j[k] \leq \tau_{i'}[k]$, 3) $\tau_j[k] \gg \tau_{l'}[k]$ and $\tau_j[k] \gg$

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on December 05,2025 at 15:14:25 UTC from IEEE Xplore. Restrictions apply.

$\tau_{i'}[k]$, 4) $\tau_j[k] \geq \tau_{l'}[k]$ and $\tau_j[k] \leq \tau_{i'}[k]$, etc., for one or more $v_{l'} \in \mathcal{N}_j^+$ and $v_{i'} \in \mathcal{N}_j^-$, can be proven identically.

A. Suppose that one in-neighbor of node v_j, say $v_{i'}$, is following the privacy-preserving strategy (i.e., $v_{i'} \in \mathcal{V}_p$) and all other in- and out-neighbors of both nodes v_j and $v_{l'}$ are curious (i.e., $v_i \in \mathcal{V}_c \; \forall v_i \in (\mathcal{N}_j^- \cup \mathcal{N}_{l'}^-) \setminus \{v_j, v_{l'}\}$, and $v_l \in \mathcal{V}_c$ $\forall v_l \in (\mathcal{N}_j^+ \cup \mathcal{N}_{l'}^+) \setminus \{v_j, v_{l'}\}$). During the Iteration procedure, curious nodes will not be able to infer the substate $u_{i'}^{y+}[s_{i'}]$, $s_{i'} \in [0, \mathcal{D}_{i'}^+]$, transmitted from node $v_{i'}$ to node v_j. Furthermore, curious nodes will not be able to infer v_j's substate $u_{ji'}^{y-}$ that is summed with the substate received from node v_j. As a result, curious nodes will not be able to infer the initial state of node v_j *or* the initial state of node $v_{i'}$. Thus, in this case, node v_j (as well as node $v_{i'}$) preserves the privacy of its initial state.

B. The case where one out-neighbor of v_j, say $v_{l'}$, is following the privacy-preserving strategy and all other in- and out-neighbors of these two nodes are curious can be analyzed in a manner similar to Case **A**.

From Cases **A** and **B**, we have that a node $v_j \in \mathcal{V}_p$ is able to preserve its privacy if it has at least one in- or out-neighbor (say $v_{i'}$ or $v_{l'}$) that also wants to preserve its privacy and follows the proposed privacy-preserving strategy. Note here that the existence of at least one in- or out-neighbor that also wants to preserve its privacy and follows the proposed privacy-preserving strategy is a sufficient condition regardless of the processing delays that affect the operation of nodes in the network. Furthermore, it is important to note that curious nodes will not be able to determine 1) the values of the messages transmitted from $v_{i'}$ to v_j, and/or 2) the values of the messages transmitted from v_j to $v_{l'}$. This means that curious nodes will not be able to determine a finite range $[\alpha, \beta]$ (where $\alpha < \beta$ and $\alpha, \beta \in \mathbb{R}$) in which the initial state $y_j[0]$ lies (as we mentioned in Definition 1). □

Remark 4 (Inability of Algorithm 2 to preserve privacy): If the conditions in Theorem 4 do not hold, then privacy may not be preserved. Two examples where nodes are unable to preserve their privacy are the following.

1) Suppose all in- and out-neighbors of node v_j are curious (i.e., $v_i \in \mathcal{V}_c \; \forall v_i \in \mathcal{N}_j^-$, and $v_l \in \mathcal{V}_c \; \forall v_l \in \mathcal{N}_j^+$). Since the curious in- and out-neighbors can communicate with each other, node $v_j \in \mathcal{V}_p$ will not be able to preserve its privacy. Specifically, at Initialization, curious nodes will know $u_j^{y+}[0]$. Then, during the Iteration procedure, curious nodes will know the messages v_j has received and the messages v_j has transmitted. This means that they will be able to determine the values of $u_j^{y+}[s_j] \in \mathbb{Z}$, for $s_j \in \{1, 2, \ldots, \mathcal{D}_{\max}^+\}$, and $u_{ji}^{y-} \; \forall v_i \in \mathcal{N}_j^-$. Note that the sum of every $u_j^{y+}[s_j]$, and u_{ji}^{y-}, is equal to v_j's initial state $y_j[0]$ (see (9f)). This means that curious nodes will be able to determine the initial state $y_j[0]$. As a result, for the case when all in- and out-neighbors of node v_j are curious, node v_j is not able to preserve the privacy of its initial state.
2) One out-neighbor of node v_j, say $v_{l'}$, is neither curious nor following the privacy-preserving strategy (i.e., $v_{l'} \in \mathcal{V}_n$), and all other in- and out-neighbors of both nodes v_j and $v_{l'}$ are curious (i.e., $v_i \in \mathcal{V}_c \; \forall v_i \in (\mathcal{N}_j^- \cup \mathcal{N}_{l'}^-) \setminus \{v_j, v_{l'}\}$, and $v_l \in \mathcal{V}_c \; \forall v_l \in (\mathcal{N}_j^+ \cup \mathcal{N}_{l'}^+) \setminus \{v_j, v_{l'}\}$). During the Initialization procedure, curious nodes will know $y_{l'}[0]$. Also, during the Iteration procedure, curious nodes will know the messages v_j has received and the messages v_j has transmitted, with the exception of the message sent to node $v_{l'}$. However, curious nodes can infer the input of node $v_{l'}$ from its output. Then, they will be able to extract the messages of node v_j (as if a curious node was directly connected to node v_j). As a result, for the case where one out-neighbor of node v_j is neither curious nor following the privacy-preserving protocol and all other in- and out-neighbors of node v_j are curious, v_j does not preserve the privacy of its initial state.

F. Main Differences of Algorithm 1 and Algorithm 2

The operation of both Algorithms 1 and 2 is adjusted to the nature of the underlying network, resulting in important differences between the two algorithms. Algorithm 1 is designed for situations where each node needs only knowledge of its out-neighbors and the maximum number of out-neighbors in the network. Knowledge of the maximum number of out-neighbors in the network can be obtained at every node in finite time via a max-consensus algorithm (see Assumption 1). This configuration allows for a quick implementation of the proposed privacy strategy requiring $\mathcal{D}_{\max}^+ + 1$ time steps. However, strong privacy guarantees are applied only in homogeneous networks where nodes have similar processing capabilities. Conversely, Algorithm 2 is designed to overcome the limitations of Algorithm 1 in heterogeneous networks. Here, nodes do not require knowledge of global parameters, such as the maximum number of out-neighbors. Instead, they only require knowledge regarding their own number of in- and out-neighbors. During Algorithm 2, nodes can obtain knowledge of their in-degree by sending a 1-bit signal at Initialization. Subsequently, each node counts the number of 1-bit signals it receives. In this way, each node v_j can calculate the number of its in-neighbors. During Algorithm 2, each node must have a unique ID as this ensures that the incoming signals at each node are distinguishable. The ability to operate over heterogeneous networks comes at the cost of a longer implementation time, as it depends on the maximum processing delay in the network. However, Algorithm 2 can provide robust privacy guarantees in heterogeneous networks as long as the processing delay of each node remains bounded (see Assumption 2).

VI. Operational Advantages

There have been different approaches for dealing with the problem of calculating the average of the initial states with privacy preservation guarantees. In [5] and [6], the authors explore differential privacy (DP), involving uncorrelated noise injection into exchanged messages. However, this approach sacrifices the exact average of initial states for the sake of privacy [6]. In contrast, Manitara and Hadjicostis [4] proposed injecting correlated noise for a finite period, ensuring both

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on December 05,2025 at 15:14:25 UTC from IEEE Xplore. Restrictions apply.

privacy and convergence to the exact average. The work in [7] asymptotically subtracts the injected initial offsets, the work in [9] addresses privacy-preserving average computation over a continuous-time weight-balanced network, and the work in [12] utilizes random coupling weights for privacy in directed graphs. In [14], a two-phase algorithm collaboratively aggregates private data, and Charalambous et al. [8] proposed an asymptotic privacy-preserving average consensus mechanism considering offset injection. The work in [11] calculates the average of initial states privately through state decomposition, while the work in [13] discusses the problem under specific topological conditions with a hot-pluggable strategy. Liu et al. [10] presented a differentially private algorithm for continuous-time heterogeneous systems. Homomorphic encryption [15], [16], [17] guarantees privacy but requires trusted nodes, intensive computational resources, and is susceptible to single points of failure. In [20], the privacy-preserving algorithm combines the benefits of both secure multiparty computation and DP while leveraging quantized communication. The work in [21] extends the application of DP strategies to quantized communication environments. In [18], an event-based offset algorithm employs quantized communication to calculate the exact quantized average in a finite number of steps but requires many steps for full convergence. Finally, in [19], an initial zero-sum offset algorithm leads to fast finite-time convergence to the exact average, but its multiple simultaneous transmissions increase the message overhead.

Algorithms 1 and 2 differ significantly from existing privacy algorithms in the literature. Compared to [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], and [14], both Algorithms 1 and 2 exhibit more efficient operation as nodes are processing and transmitting quantized values. Note that quantization reduces the amount of data that need to be transmitted between nodes. This reduces communication overhead and improves the system performance. Compared to [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [20], and [21], Algorithms 1 and 2 exhibit finite-time convergence to the exact average of the nodes' initial states, which enhances the reliability for our algorithms and is important for applications where accuracy and consistency are critical. Compared to [15], [16], [17], [18], [19], [20], and [21], both Algorithms 1 and 2 exhibit finite transmission guarantees. More specifically, the privacy protocols in Algorithms 1 and 2 are adjusted to the finite transmission nature of the underlying averaging operation. The protocols in [18], [19], [20], and [21] do not allow termination of transmissions once the average of the initials states is calculated. Furthermore, the finite transmission characteristic leads to the reduction of network congestion and improves the network's operational efficiency. Compared to [15], [16], and [17], our algorithms do not require encryption of the messages transmitted between nodes, which can be power consuming (especially in resource-constrained environments). Therefore, compared to [15], [16], and [17], our algorithms are more efficient in terms of power consumption and computational resources. Finally, compared to [8], [10], and [14], Algorithm 2 focuses on the presence of heterogeneous nodes in the network. Heterogeneity of a network introduces processing delays and asynchronicity. Algorithm 2 is able to operate over heterogeneous networks and provides sufficient privacy guarantees while maintaining efficient operation (i.e., quantized communication and processing) and convergence to the exact result in finite time.

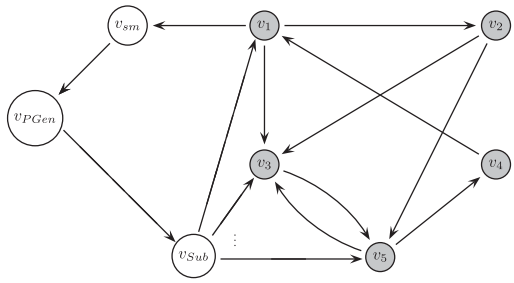

Fig. 1. Example of a digraph representing a smart grid consisting of a neighborhood of five households (nodes) v_1–v_5, a smart meter v_{sm}, a substation v_{Sub}, and a power generator v_{PGen}.

VII. Application: Privacy-Preserving Power Request in Smart Grids

We now present an application of Algorithms 1 and 2. Our application is borrowed from [19], but is adjusted in the context of (possibly) heterogeneous networks, which require finite transmission guarantees. We focus on the scenario where a neighborhood of interconnected households need the total demanded power (or offered power for the case where households produce electricity, e.g., via photovoltaic panels) from a smart meter in a privacy-preserving manner [28]. In our application, each household has its own embedded computing system and uses it for electricity demand forecasting, e.g., by using machine learning techniques [29]. We analyze the following two cases: 1) all households utilize homogeneous computing systems for demand forecasting, and 2) all households utilize heterogeneous computing systems for demand forecasting. In the first case, the processing delay is expected to be consistent across all devices, and households are able to perform transmissions to their neighboring households in a synchronous fashion. In the second case, the processing delay can vary significantly between different devices or components, and households perform transmissions to their neighboring households in an asynchronous fashion. We will show that Algorithms 1 and 2 allow households to make power requests in a privacy-preserving manner for the two aforementioned scenarios, respectively, and then terminate their operations.

Our example smart grid network is shown in Fig. 1. It consists of five households $\mathcal{V} = \{v_1, \ldots, v_5\}$, a smart meter v_{sm}, a substation v_{Sub}, and a power generator v_{PGen}. In our scenario, we assume that any household may become a curious entity and attempt to infer the initial state (i.e., demanded power) of one or more other households. The use of Algorithm 1 or Algorithm 2 enables nodes to calculate the total requested power in a privacy-preserving manner, effectively securing their

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on December 05,2025 at 15:14:25 UTC from IEEE Xplore. Restrictions apply.

privacy against any household acting as a curious entity. During the smart grid operation, households 1) utilize their embedded computing systems to calculate their local power demand, and 2) calculate the average electricity demand per household in the neighborhood in a privacy-preserving manner by utilizing Algorithm 1 or Algorithm 2 (for each household, the result of the embedded computing system serves as that household's initial state when executing the chosen algorithm). After these two steps, the smart meter v_{sm} collects the daily demands (or offers) through v_1 (since the state of v_1 is equal to the average of the daily demanded/offered power from all households in the neighborhood). Then, v_{sm} multiplies the state of v_1 with the number of houses in the neighborhood in order to calculate the total demanded/offered power. The smart meter v_{sm} transmits the total demanded/offered power to the power generator v_{PGen}. Finally, the power generator v_{PGen} produces and delivers the demanded electricity to the substation v_{Sub}, and the electricity is claimed/offered from/to the substation to the households.

In our application, we utilize Algorithms 1 and 2 to compute the total requested/offered power during a specific day by the set of interconnected households in a privacy-preserving manner. For the neighborhood in Fig. 1, let us consider the amount of requested power as $\mathcal{C} = \{68, 73, 69, 79, 36\}$. Note that the requested power is represented using integer values to simplify the application of our algorithms in the smart grid scenario. If the requested power is expressed as decimal values, these can be converted to rational numbers (i.e., as fractions of two integers). Nodes can then perform a max-consensus to determine the maximum denominator (to ensure that all nodes utilize the same denominator for representing the rational numbers). Subsequently, nodes can execute our proposed algorithms using these rational numbers as inputs. The average power demand is equal to 65 and the total power demand is equal to 325. For the case where households are utilizing homogeneous computing systems, we assume that each household v_j undergoes a processing delay $\tau_j[k] = 0$, during every time step of the executed algorithm (i.e., since computing systems are homogeneous, they experience similar processing delays). Furthermore, for the case where households are utilizing heterogeneous computing systems, we assume that each household v_j undergoes a processing delay $\tau_j[k]$ randomly chosen with uniform probability from the set $\{1, 2, 3\}$, during every time step of the executed algorithm. For the case where households utilize homogeneous(/heterogeneous) computing systems, we show the execution Algorithm 1(/Algorithm 2) in Fig. 2(/Fig. 3). In Figs. 2 and 3, we can see that both Algorithms 1 and 2 are able to calculate the average requested power per household in the neighborhood. Furthermore, we can see that Algorithm 1 in Fig. 2 converges faster compared to Algorithm 2 in Fig. 3. However, during Algorithm 2, households perform fewer transmissions (i.e., 26 total transmissions) compared to Algorithm 1 (i.e., 36 total transmissions). This is due to the fact that during Algorithm 2, each household performs $\mathcal{D}_j^+$ transmissions for executing the privacy protocol, while during Algorithm 1, each household performs $\mathcal{D}_{\max}^+ + 1$ transmissions for the privacy protocol.

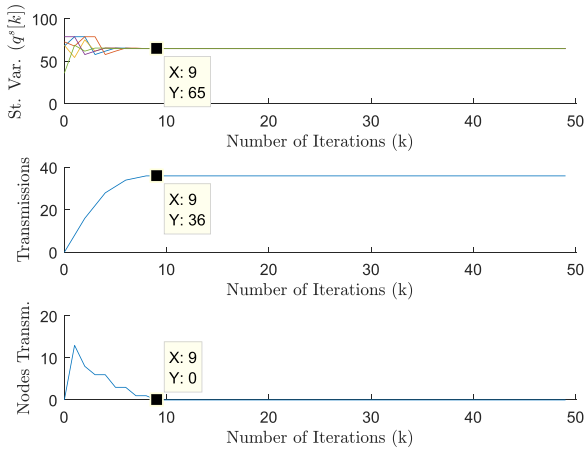

Fig. 2. Execution of Algorithm 1 for the household network in Fig. 1. *Top figure:* Average requested power per household with privacy preservation for day 1 plotted against the number of iterations. *Middle Figure:* Average total number of transmissions plotted against the number of iterations. *Bottom Figure:* Average number of households performing transmissions plotted against the number of iterations.

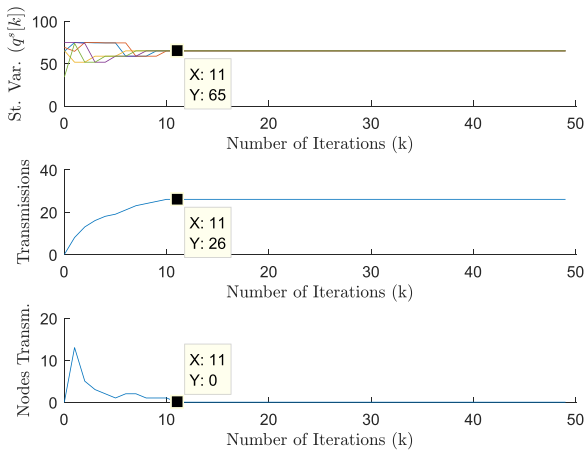

Fig. 3. Execution of Algorithm 2 for the household network in Fig. 1. *Top figure:* Average requested power per household with privacy preservation for day 1 plotted against the number of iterations. *Middle figure:* Average total number of transmissions plotted against the number of iterations. *Bottom figure:* Average number of households performing transmissions plotted against the number of iterations.

VIII. Comparison With Literature

In Fig. 4, we compare Algorithms 1 and 2 against [4], [7], [11], [19, Algorithm 1], and [19, Algorithm 2]. We plot the error $e[k] := (\sum_{v_j \in \mathcal{V}} |q_j^s[k] - \overline{y}|)/|\mathcal{V}|$ against the number of iterations. The error is averaged over 50 random digraphs. The average of the nodes' initial states is $\overline{y} = 23.625$. The random digraphs were generated using the Erdos–Renyi model, and the initial state of each node was kept constant across all 50 digraphs (resulting in a fixed $\overline{y}$ equal to 23.625). The parameters for each algorithm (e.g., substate values) were randomly chosen, leading to different initial error values $e[0]$ in Fig. 4. To ensure a fair comparison across all algorithms, we eliminated the impact of processing delays for Algorithm 2 by setting $\tau_j[k] = 0$ for every node $v_j \in \mathcal{V}$. From Fig. 4, we observe that Algorithms 1 and 2 exhibit the fastest convergence speed among the

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on December 05,2025 at 15:14:25 UTC from IEEE Xplore. Restrictions apply.

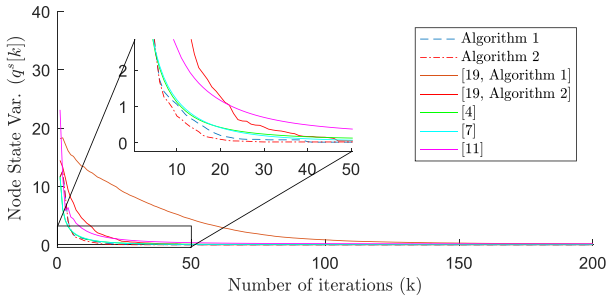

Fig. 4. Comparison between Algorithm 1, Algorithm 2, [4], [7], [11], [19, Algorithm 1], and [19, Algorithm 2].

compared algorithms, with Algorithm 2 being slightly faster than Algorithm 1. In addition, the algorithms in [4] and [7] require an identical number of time steps for convergence, and the same holds also for [11] and [19, Algorithm 2]. The largest number of time steps for convergence is required by [19, Algorithm 1] due to its event-triggered conditions for communication among nodes. In general, Algorithms 1 and 2 offer significant advantages over existing algorithms from the literature. Also they are the first in the literature to achieve three key properties simultaneously: 1) ensuring privacy preservation, 2) calculating the exact quantized average in finite time, and 3) ceasing transmissions after convergence. Algorithm 2 has also the capability to handle node processing delays (which is a critical feature as delays can potentially compromise privacy guarantees, as discussed in Section V-A).

IX. Conclusion

We proposed two privacy-preserving event-triggered quantized averaging algorithms for homogeneous and heterogeneous networks. Key features of our algorithms include ensuring privacy preservation through topological conditions, calculating the exact quantized average in finite time, and automatically terminating transmissions after convergence, with one algorithm being also able to handle node processing delays. We analyzed our algorithms' operation, derived an upper bound on convergence time, and provided necessary and sufficient topological conditions for achieving privacy preservation. Finally, we presented an application over smart grids and compared their performance against the literature.

In our future work, we intend to conduct an extended study on analyzing and preventing possible information leakage from our algorithms when probabilistic information about the initial states and the choice of algorithmic parameters is available to the curious nodes. In addition, we plan to analyze and extend our privacy strategies for the case where the communication network is dynamically changing over time.

Appendix A
Proof of Theorem 1

In this proof, we will show that there exists $k_0 \in \mathbb{Z}_+$, for which the mass variables of *every* node v_j (for which $z_j[k] > 0$) fulfill (5) and (6), for every $k \geq k_0$. This means that for $k \geq k_0$, we have only leading masses. Note here that, during the execution of Algorithm 1, the leading mass will not fulfill the Event Trigger Conditions 3 (in Execution Step 3 of Algorithm (1.A)). This means that the corresponding node (say v_j) will not transmit its mass variables to its out-neighbors $v_l \in \mathcal{N}_j^+$ according to its predetermined priority. Furthermore, from Lemma 2, we will also show that there exists $k_1 > k_0$ ($k_1 \in \mathbb{Z}_+$), where for every $k \geq k_1$, the state variables of every $v_j \in \mathcal{V}$ fulfill (2) and (3) for $\alpha \in \mathbb{Z}_+$ (i.e., every node has reached quantized consensus) and thus transmissions cease.

During the Initialization steps of Algorithm (1.A), each node v_j will 1) decompose its initial state $y_j[0]$ into $\mathcal{D}_{\max}^+ + 2$ substates $u_j^y[s_j] \in \mathbb{Z}$, for $s_j \in \{0, 1, 2, \ldots, \mathcal{D}_{\max}^+ + 1\}$, 2) set its initial state $y_j[0]$ to be equal to $u_j^y[0]$, 3) increase the substate counter s_j, and 4) broadcast its state variables to every out-neighbor. Then, during Iteration Step 1, each node will 1) receive and update its state variables, 2) receive and update its mass variables, and 3) call Algorithm (1.A) to check Event Trigger Conditions 1, Event Trigger Conditions 2, and Event Trigger Conditions 3. However, note here that regardless of the output of Algorithm (1.A), each node v_j will utilize the privacy-preserving strategy for the first $\mathcal{D}_{\max}^+ + 1$ time steps (i.e., for $k = 0, 1, \ldots, \mathcal{D}_{\max}^+$). This means that for the first $\mathcal{D}_{\max}^+ + 1$ time steps of the Iteration procedure, each node v_j will inject to its mass variables and the set of substates $u_j^y[s_j]$ and $u_j^z[s_j]$, for $s_j \in \{1, 2, \ldots, \mathcal{D}_{\max}^+ + 1\}$. Then, it will transmit its mass variables to an out-neighbor according to the order P_{lj}. Thus, after $\mathcal{D}_{\max}^+ + 1$ time steps, each node v_j will have injected in the network every set of substates $u_j^y[s_j]$ and $u_j^z[s_j]$. For the analysis of the execution of Algorithm 1 for time steps $k \geq \mathcal{D}_{\max}^+ + 1$, we can use steps similar as [22, Thm. 1]. As a result, combining $1 + \mathcal{D}_{\max}^+$ (which represents the number of iterations required for completing the privacy protocol) and $n^2 + (n-1)m^2$ (which is the upper bound on the number of iterations for the algorithm proposed in [22] to converge), we can conclude that the required number of iterations for Algorithm 2 is upper bounded by $1 + \mathcal{D}_{\max}^+ + n^2 + (n-1)m^2$.

Let us now highlight our algorithm's exact convergence. In Algorithm 1, the sum of $u_j^y[s_j]$ values for every node v_j (where $s_j \in [0, \mathcal{D}_{\max}^+ + 1]$) is equal to $(\mathcal{D}_{\max}^+ + 2)$ times the sum of $y_j[0]$ values of every node v_j. Furthermore, the sum of $u_j^z[s_j]$ values for every node v_j (where $s_j \in [0, \mathcal{D}_{\max}^+ + 1]$) is equal to $(\mathcal{D}_{\max}^+ + 2)$ times the total number of nodes in the network n. Dividing these two sums yields the correct average. Therefore, Algorithm 1 is able to converge to the exact average.

Remark 5 (Analysis of operational advantages): Let us note that the upper bound on the convergence time is determined by network parameters, such as diameter, number of edges, maximum out-degree, and number of nodes. Changing these parameters affects the time steps needed for convergence (e.g., increasing the number of nodes in the network means our algorithms may require more time steps to converge). However, the convergence time also relies on the network structure, since nodes are transmitting data between each other. In this article, our aim is to utilize the network parameters to derive an upper bound on the required number of time steps. While an in-depth analysis of how the aforementioned parameters affect

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on December 05,2025 at 15:14:25 UTC from IEEE Xplore. Restrictions apply.

our proposed algorithms' performance is beyond this article's scope, it will be considered in future research. In addition, the usage of quantized values enhances operational efficiency but does not affect finite-time convergence. Specifically, during the execution of our algorithms, nodes are able to calculate the exact average even for the case when they exchange real-valued messages (an advantage of particular importance as most algorithms in the literature exhibit asymptotic convergence). Finally, our algorithms ensure finite-time convergence and operation termination, irrespective of network parameters, as long as the network is strongly connected.

Appendix B
Proof of Theorem 3

During the Initialization steps of Algorithm 2, each node v_j will 1) decompose its initial state $y_j[0]$ into $\mathcal{D}_j^+ + \mathcal{D}_j^- + 1$ substates (specifically, into $\mathcal{D}_j^+ + 1$ substates $u_j^{y+}[s_j] \in \mathbb{Z}$, for $s_j \in \{0, 1, 2, \ldots, \mathcal{D}_j^+\}$, and $\mathcal{D}_j^-$ substates u_{ji}^{y-} $\forall v_i \in \mathcal{N}_j^-$), 2) set its initial state $y_j[0]$ to be equal to $u_j^{y+}[0]$, 3) increase the substate counter s_j, and 4) broadcast its state variables to every out-neighbor. Then, during Iteration Step 1, each node will 1) receive and update its state variables, 2) receive and update its mass variables, and 3) call Algorithm (1.A) to check Event Trigger Conditions 1, Event Trigger Conditions 2, and Event Trigger Conditions 3. However, note here that regardless of the output of Algorithm (1.A), each node v_j will utilize the privacy-preserving strategy for at most $\overline{\tau}\mathcal{D}_j^+$ time steps. Specifically, during the time interval $[0, \overline{\tau}\mathcal{D}_j^+]$, it will perform at least $\mathcal{D}_j^+$ transmissions, one transmission toward each outgoing link. This means that for the first $\overline{\tau}\mathcal{D}_j^+$ time steps of the Iteration procedure, each node v_j will inject 1) to the mass variables it transmits the set of substates $u_j^y[s_j]$ and $u_j^z[s_j]$, for $s_j \in \{0, 1, 2, \ldots, \mathcal{D}_{\max}^+ - 1\}$, and 2) to the mass variables it receives the set of substates u_{ji}^{y-}, for every $v_i \in \mathcal{N}_j^-$. Then, it will transmit its mass variables to an out-neighbor according to the order P_{lj}. Thus, after $\overline{\tau}\mathcal{D}_{\max}^+$ time steps, each node v_j will have injected in the network every set of substates $u_j^y[s_j]$, for $s_j \in \{0, 1, 2, \ldots, \mathcal{D}_j^+\}$, and u_{ji}^{y-} $\forall v_i \in \mathcal{N}_j^-$.

For the analysis of the execution of Algorithm 2 for time steps $k \geq \overline{\tau}\mathcal{D}_{\max}^+$, we can use steps similar as [22, Thm. 1], adjusted to the heterogeneous nature of the network (i.e., each transmission is performed after at most $\overline{\tau}$ time steps). As a result, combining $\overline{\tau}\mathcal{D}_{\max}^+$ (which represents the upper bound on the number of iterations for completing the privacy protocol) and $\overline{\tau}(n^2 + (n-1)m^2)$ (which is the upper bound on the number of iterations for the algorithm proposed in [22] to converge), we can conclude that the required number of iterations for Algorithm 2 is upper bounded by $\overline{\tau}(\mathcal{D}_{\max}^+ + n^2 + (n-1)m^2)$.

Finally, regarding the operation of Algorithm 2, we observe that the sum of $u_j^{y+}[s_j]$ and u_{ji}^{y-} values for every node v_j (where $s_j \in [0, \mathcal{D}_j^+]$) is equal to the sum of $y_j[0]$ values of every node v_j. Furthermore, the sum of $u_j^{z+}[s_j]$ values for every node v_j (where $s_j \in [0, \mathcal{D}_j^+]$) is equal to the total number of nodes n. Dividing these two sums provides the average, which means that Algorithm 2 is able to converge to the exact average.

References

[1] A. I. Rikos, C. N. Hadjicostis, and K. H. Johansson, "Finite-time privacy-preserving quantized average consensus with transmission stopping," in *Proc. 61st IEEE Conf. Decis. Control*, 2022, pp. 6762–6768.

[2] P. Park, S. C. Ergen, C. Fischione, C. Lu, and K. H. Johansson, "Wireless network design for control systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 978–1013, 2018.

[3] S. Knorn, S. Dey, A. Ahlen, and D. E. Quevedo, "Optimal energy allocation in multisensor estimation over wireless channels using energy harvesting and sharing," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4337–4344, Oct. 2019.

[4] N. Manitara and C. N. Hadjicostis, "Privacy-preserving asymptotic average consensus," in *Proc. Eur. Control Conf.*, 2013, pp. 760–765.

[5] J. Cortés, G. E. Dullerud, S. Han, J. L. Ny, S. Mitra, and G. J. Pappas, "Differential privacy in control and network systems," in *Proc. 55th IEEE Conf. Decis. Control*, 2016, pp. 4252–4272.

[6] 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.

[7] Y. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 753–765, Feb. 2017.

[8] T. Charalambous, N. E. Manitara, and C. N. Hadjicostis, "Privacy-preserving average consensus over digraphs in the presence of time delays," in *Proc. Allerton Conf. Commun. Control, Comput.*, 2019, pp. 238–245.

[9] N. Rezazadeh and S. S. Kia, "Privacy preservation in a continuous-time static average consensus algorithm over directed graphs," in *Proc. Amer. Control Conf.*, 2018, pp. 5890–5895.

[10] X.-K. Liu, J.-F. Zhang, and J. Wang, "Differentially private consensus algorithm for continuous-time heterogeneous multi-agent systems," *Automatica*, vol. 122, 2020, Art. no. 109283.

[11] Y. Wang, "Privacy-preserving average consensus via state decomposition," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4711–4716, Nov. 2019.

[12] H. Gao, C. Zhang, M. Ahmad, and Y. Wang, "Privacy-preserving average consensus on directed graphs using push-sum," in *Proc. IEEE Conf. Commun. Netw. Secur.*, 2018, pp. 1–9.

[13] I. L. D. Ridgley, R. A. Freeman, and K. M. Lynch, "Private and hot-pluggable distributed averaging," *IEEE Contr. Syst. Lett.*, vol. 4, no. 4, pp. 988–993, Oct. 2020.

[14] X. Wang, J. He, P. Cheng, and J. Chen, "Privacy preserving collaborative computing: Heterogeneous privacy guarantee and efficient incentive mechanism," *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 221–233, Jan. 2019.

[15] C. N. Hadjicostis and A. D. Dominguez-Garcia, "Privacy-preserving distributed averaging via homomorphically encrypted ratio consensus," *IEEE Trans. Autom. Control*, vol. 65, no. 9, pp. 3887–3894, Sep. 2020.

[16] W. Chen, L. Liu, and G.-P. Liu, "Privacy-preserving distributed economic dispatch of microgrids: A dynamic quantization-based consensus scheme with homomorphic encryption," *IEEE Trans. Smart Grid*, vol. 14, no. 1, pp. 701–713, Jan. 2023.

[17] Y. Shi and E. Nekouei, "Quantization and event-triggered policy design for encrypted networked control," *IEEE/CAA J. Automatica Sinica*, vol. 11, no. 4, pp. 946–955, Apr. 2024.

[18] A. I. Rikos, T. Charalambous, K. H. Johansson, and C. N. Hadjicostis, "Privacy-preserving event-triggered quantized average consensus," in *Proc. 59th IEEE Conf. Decis. Control*, 2020, pp. 6246–6253.

[19] A. I. Rikos, T. Charalambous, K. H. Johansson, and C. N. Hadjicostis, "Distributed event-triggered algorithms for finite-time privacy-preserving quantized average consensus," *IEEE Trans. Control Netw. Syst.*, vol. 10, no. 1, pp. 38–50, Mar. 2023.

[20] Q. Li, J. S. Gundersen, M. Lopuhaä-Zwakenberg, and R. Heusdens, "Adaptive differentially quantized subspace perturbation (ADQSP): A unified framework for privacy-preserving distributed average consensus," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 1780–1793, 2024.

[21] L. Gao, S. Deng, W. Ren, and C. Hu, "Differentially private consensus with quantized communication," *IEEE Trans. Cybern.*, vol. 51, no. 8, pp. 4075–4088, Aug. 2021.

[22] A. I. Rikos, C. N. Hadjicostis, and K. H. Johansson, "Finite time quantized average consensus with transmission stopping guarantees and no quantization error," *Automatica*, vol. 163, 2024, Art. no. 111522.

[23] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for web transactions," *ACM Trans. Inf. Syst. Secur.*, vol. 1, pp. 66–92, 1998.

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on December 05,2025 at 15:14:25 UTC from IEEE Xplore. Restrictions apply.

[24] K. Chatzikokolakis and C. Palamidessi, "Probable innocence revisited," *Theor. Comput. Sci.*, vol. 367, no. 1-2, pp. 123–138, 2006.
[25] S. Giannini, D. Di Paola, A. Petitti, and A. Rizzo, "On the convergence of the max-consensus protocol with asynchronous updates," in *Proc. 52nd IEEE Conf. Decis. Control*, 2013, pp. 2605–2610.
[26] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.
[27] W. Y. B. Lim et al., "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, Thirdquarter 2020.
[28] C.-M. Yu, C.-Y. Chen, S.-Y. Kuo, and H.-C. Chao, "Privacy-preserving power request in smart grid networks," *IEEE Syst. J.*, vol. 8, no. 2, pp. 441–449, Jun. 2014.
[29] M. Massaoudi, H. Abu-Rub, S. S. Refaat, I. Chihi, and F. S. Oueslati, "Deep learning in smart grid technology: A review of recent advancements and future prospects," *IEEE Access*, vol. 9, pp. 54558–54578, 2021.

Apostolos I. Rikos (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus, in 2010, 2012, and 2018, respectively.

He is currently an Assistant Professor with the Artificial Intelligence Thrust of the Information Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. He is also with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong. In 2018, he joined the KIOS Research and Innovation Center of Excellence, Aglandjia, Cyprus, where he was a Research Lecturer. In 2020, he joined the Division of Decision and Control Systems of KTH Royal Institute of Technology as a Postdoctoral Researcher. In 2023, he joined the Department of Electrical and Computer Engineering, Division of Systems Engineering, Boston University, Boston, MA, USA, as a Postdoctoral Associate. His research interests include distributed optimization and learning, distributed network control and coordination, privacy and security, and algorithmic design.

Christoforos N. Hadjicostis (Fellow, IEEE) received the B.S. degrees in electrical engineering, computer science and engineering, and in mathematics, the M.Eng. degree in electrical engineering and computer science, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1993, 1995, and 1999, respectively.

In 1999, he joined the Faculty at the University of Illinois at Urbana-Champaign, Champaign, IL, USA, where he was an Assistant Professor and then an Associate Professor with the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Information Trust Institute. Since 2007, he has been with the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus, where he is currently a Professor.

Karl H. Johansson (Fellow, IEEE) received the M.Sc. degree in electrical engineering and the Ph.D. degree in automatic control from Lund University, Lund, Sweden, in 1992 and 1997, respectively.

He is currently a Professor with the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden, and the Director of Digital Futures, Stockholm. He has held visiting positions at UC Berkeley, Caltech, NTU, HKUST Institute of Advanced Studies, and NTNU.

Dr. Johansson is the President of the European Control Association and a Member of the IFAC Council. He was on the IEEE Control Systems Society Board of Governors and the Swedish Scientific Council for Natural Sciences and Engineering Sciences. He was the recipient of several best paper awards and other distinctions from IEEE, IFAC, and ACM. He is Fellow of the Royal Swedish Academy of Engineering Sciences.

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on December 05,2025 at 15:14:25 UTC from IEEE Xplore. Restrictions apply.