2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

Finite-Time Privacy-Preserving Quantized Average Consensus with Transmission Stopping

Apostolos I. Rikos, Christoforos N. Hadjicostis, and Karl H. Johansson

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9993299

Abstract— **Due to their flexibility, battery powered or energy-harvesting wireless networks are deployed in diverse applications. Securing data transmissions between wireless devices is of critical importance in order to avoid privacy-sensitive user data leakage. In this paper, we focus on the scenario where some nodes are curious (but not malicious) and try to identify the initial states of one (or more) other nodes, while some other nodes aim to preserve the privacy of their initial states from these curious nodes. We present a privacy-preserving finite transmission event-triggered quantized average consensus algorithm. Its operation is suitable for battery-powered or energy-harvesting wireless network since it guarantees (i) efficient (quantized) communication, and (ii) transmission ceasing (which allows preservation of available energy). Furthermore, we present topological conditions under which the proposed algorithm allows nodes to preserve their privacy. We conclude with a comparison of our algorithm against other algorithms in the existing literature.**

I. INTRODUCTION

Wireless control networks (WCN) play a major role in important applications due to their deployment flexibility, which allows them to operate unattended in hostile environments with a limited energy budget [1]. Security and privacy of WCN is a challenging issue since, during their operation in a potentially hostile environment, they are exposed to a variety of privacy attacks. Specifically, distributed coordination algorithms require exchange of collected data between neighboring nodes. In many occasions, there might be curious nodes in the network that aim to extract private and/or sensitive data, such as the state of a node. Efficient (quantized) communication between nodes is another desirable feature since (i) it is suitable for the available network resources, and (ii) it exhibits advantages and applicability to public-key cryptosystems. For these reasons, several strategies have been proposed for distributed coordination via quantized average consensus [2], [3].

Existing Literature. There have been different approaches for dealing with the problem of calculating the quantized average of the initial states with privacy preservation guarantees. In [4], [5] the authors present approaches based on differential privacy, in which nodes inject uncorrelated noise into the exchanged messages. The injection of correlated noise at each time step and for a finite period of time was proposed in [6]. In [7] the nodes asymptotically subtract the initial offset values they added in the computation. The problem of calculating the average of the initial states in a privacy-preserving manner is also discussed in [8] for a continuous time weight balanced system. In [9] the average of the initial states is calculated in a privacy-preserving manner via a state-decomposition-based approach, whereas [10] discusses the problem under certain topological conditions. Homomorphic encryption [11], [12] is another strategy which guarantees privacy preservation, but requires the existence of trusted nodes and imposes heavier computational requirements. In [13] the authors present an event-based offset adding algorithm. This strategy allows the calculation of the exact quantized average in a finite number of time steps, but requires a large number of time steps for convergence. Finally, in [14] the authors present an initial zero-sum offset algorithm. This strategy leads to fast finite-time convergence to the exact average, but requires multiple simultaneous transmissions, which increase significantly the header of the transmitted message.

Main Contributions. In this paper, we present a novel privacy-preserving event-triggered distributed algorithm which (i) achieves average consensus under privacy constraints with quantized communication, (ii) converges after a finite number of time steps, and (iii) relies on event-driven operation and ceases transmissions once convergence has been achieved (which makes it suitable for battery powered or energy-harvesting wireless networks). The main contributions of our paper are the following.

A. We present a novel privacy-preserving distributed event-triggered algorithm, which operates with quantized values and calculates the exact average of the initial states under privacy constraints; see Algorithm 1 in Section IV-B.

B. We show that our proposed privacy-preserving algorithm converges after a finite number of iterations for which we provide a polynomial upper bound. Furthermore, we establish our algorithm's transmission stopping capabilities; see Theorem 1 in Section IV-C.

C. We present topological conditions that ensure privacy preservation for the nodes that follow the proposed algorithm; see Proposition 1 in Section IV-D.

D. We demonstrate our algorithm's operation and compare its performance against other finite-time privacy-preserving algorithms from the current literature; see Section V.

The proposed privacy-preserving algorithm relies on *multiple state decomposition*. Specifically, at initialization, each node decomposes its initial state into multiple substate values, so

Apostolos I. Rikos and K. H. Johansson are with the Division of Decision and Control Systems, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden. They are also affiliated with Digital Futures, SE-100 44 Stockholm, Sweden. E-mails:{rikos,kallej}@kth.se.

C. N. Hadjicostis is with the Department of Electrical and Computer Engineering, University of Cyprus, 1678 Nicosia, Cyprus: E-mail: chadjic@ucy.ac.cy.

This work was supported by the Knut and Alice Wallenberg Foundation and the Swedish Research Council.

978-1-6654-6761-2/22/$31.00 ©2022 IEEE

that the average of the substate values is equal to the initial state. The node utilizes one substate as its initial state. Then, it injects the other substates to its state at specific instances and transmits them to a different neighboring node each time. This ensures that each neighboring node receives at least one substate, which helps preserve the privacy of each node's initial state (as long as at least one neighboring node is not colluding with curious nodes).

II. NOTATION AND BACKGROUND

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

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), called *communication digraph* $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$, 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*[1] 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 $\mathcal{N}_j^- = \{v_i \in \mathcal{V} \mid (v_j, v_i) \in \mathcal{E}\}$, while 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 $\mathcal{D}_j^- = |\mathcal{N}_j^-|$, while the cardinality of $\mathcal{N}_j^+$ is called the *out-degree* of v_j and is denoted by $\mathcal{D}_j^+ = |\mathcal{N}_j^+|$.

Node Operation. 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 (i) the state variables $y_j^s[k], z_j^s[k], q_j^s[k]$ (where $y_j^s[k] \in \mathbb{Z}$, $z_j^s[k] \in \mathbb{Z}_+$, $q_j^s[k] = \frac{y_j^s[k]}{z_j^s[k]}$), (ii) the mass variables $y_j[k], z_j[k]$, (where $y_j[k] \in \mathbb{Z}$ and $z_j[k] \in \mathbb{Z}_+$), (iii) the substate counter s_j (where $s_j \in \mathbb{N}$), (iv) the privacy variables $u_j^y[s_j]$, $u_j^z[s_j]$ (where $u_j^y[s_j] \in \mathbb{Z}$, $u_j^z[s_j] \in \mathbb{Z}$), (v) the transmission variables S_br_j and M_tr_j (where $S\_br_j \in \mathbb{N}$ and $M\_tr_j \in \mathbb{N}$).

For every node v_j, the state variables $y_j^s[k], z_j^s[k], q_j^s[k]$ are used to store the received messages, calculate the quantized average of the initial values, and communicate with other nodes; the mass variables $y_j[k], z_j[k]$ are used to communicate with other nodes; the substate counter s_j is used to transmit the privacy variables; the privacy variables $u_j^y[s_j]$, $u_j^z[s_j]$ are used to preserve the privacy of the initial state; and the transmission variables S_br_j, M_tr_j are used to decide whether the state variables will be broadcasted or the mass variables will be transmitted.

Node Transmission Strategy. We assume that each node is aware of its out-neighbors and can directly transmit messages to each out-neighbor; however, it cannot necessarily receive messages (at least not directly) from them. In the proposed distributed algorithm, each node v_j assigns a round-robin *unique order* in the set $\{0, 1, ..., \mathcal{D}_j^+ - 1\}$ to each of its outgoing edges m_{lj}, where $v_l \in \mathcal{N}_j^+$. More specifically, the order of link (v_l, v_j) for node v_j is denoted by P_{lj} (such that $\{P_{lj} \mid v_l \in \mathcal{N}_j^+\} = \{0, 1, ..., \mathcal{D}_j^+ - 1\}$). This unique predetermined order is used during the execution of the proposed distributed algorithm as a way of allowing node v_j to transmit messages to its out-neighbors in a *round-robin* fashion. This means that each node v_j transmits directly to one out-neighbor at a time, following a predetermined order. The next time it transmits to an out-neighbor, 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 ordering.

III. PROBLEM FORMULATION

Consider a strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$, where each node $v_j \in \mathcal{V}$ has an initial (i.e., for $k = 0$) quantized value $y_j[0]$ (for simplicity, we take $y_j[0] \in \mathbb{Z}$). Furthermore, consider that the node set $\mathcal{V}$ is partitioned into three disjoint subsets. Specifically, we have (i) 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, (ii) the subset of nodes $v_c \in \mathcal{V}_c \subset \mathcal{V}$ that are curious and try to identify the initial states of all or a subset of nodes in the network, and (iii) the rest of the nodes $v_i \in \mathcal{V}_n \subset \mathcal{V}$ that neither wish to preserve their privacy nor aim to identify the states of any other nodes. Note that $\mathcal{V}_p \cap \mathcal{V}_c = \emptyset$, which means that curious nodes in $\mathcal{V}_c$ do not worry about preserving the privacy of their initial states. In fact, since curious nodes collaborate arbitrarily among themselves (in order to identify the initial states of other nodes in the network), we assume that they are willing to communicate to other curious nodes their states and their received messages over side channels (effectively, curious nodes operate like a single entity). An example is shown in Fig. 1 (from [14]).

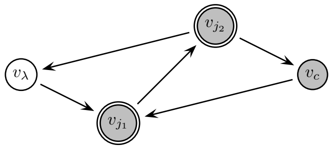

Fig. 1. Example of a digraph with the different types of nodes in the network: nodes $v_{j_1}, v_{j_2} \in \mathcal{V}_p$ wish to preserve their privacy, node $v_c \in \mathcal{V}_c$ is curious and wishes to identify the initial states of other nodes in the network, and node $v_\lambda \in \mathcal{V}_n$ is neither curious nor wishes to preserve its privacy.

We consider that the private information for each node is its initial state $y_j[0]$. We adopt the following notion of privacy, which aims to ensure that the state $y_j[0]$ cannot be inferred exactly by curious nodes and relates to notions of possible innocence in theoretical computer science [15], [16] in the sense that there is some uncertainty about $y_j[0]$.

[1] A directed *path* from v_i to v_j exists if we can find a sequence of vertices $v_i \equiv v_{l_0}, v_{l_1}, \ldots, v_{l_t} \equiv v_j$ such that $(v_{l_{\tau+1}}, v_{l_\tau}) \in \mathcal{E}$ for $\tau = 0, 1, \ldots, t-1$.

Definition 1. *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. This means that curious nodes in $\mathcal{V}_c$ cannot determine a finite range $[\alpha, \beta]$ (where $-\infty < \alpha < \beta < \infty$ and $\alpha, \beta \in \mathbb{R}$) in which the initial state $y_j[0]$ lies in.*

In this paper, we develop a distributed algorithm that allows nodes to address problems **P1**, **P2** and **P3** presented below, while processing and transmitting *quantized* information via available communication links.

P1. Every node v_j 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 (i.e., there is no quantization error), where

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

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

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

for some $\alpha \in \mathbb{N}$. 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 calculates q as the ratio of two integer values).

P2. Every node $v_j \in \mathcal{V}_p$ preserves the privacy of its initial state $y_j[0]$ (i.e., it does not reveal its initial state $y_j[0]$ to other nodes) when it exchanges quantized information with neighboring nodes while calculating q in (1) (i.e., its state variables y_j^s, z_j^s, q_j^s fulfill (2) and (3), respectively).

P3. Every node v_j stops performing transmissions towards its out-neighbors $v_l \in \mathcal{N}_j^+$ once its state variables y_j^s, z_j^s, q_j^s fulfill (2) and (3), respectively.

IV. PRIVACY-PRESERVING EVENT-TRIGGERED QUANTIZED AVERAGE CONSENSUS ALGORITHM WITH FINITE TRANSMISSION CAPABILITIES

Assumption 1. *We assume that all nodes have knowledge of the maximum out-degree in the network $\mathcal{D}_{max}^+ = \max_{v_j \in \mathcal{V}} \mathcal{D}_j^+$ (or a common upper bound of it).*

Assumption 1 is important for guaranteeing convergence to the average of the initial states. In case Assumption 1 does not hold (or if different nodes use different upper bounds), then our algorithm may converge to a value that is not equal to the average of the initial states (i.e., our algorithm will simply achieve consensus). Specifically, in (2) the α may not be the same for y^s and z^s. Thus, q^s in (3) may not be equal to q; the reason for this will become clear later in the paper. It is worth pointing out that nodes can run a simple max consensus algorithm [17] on their out-degrees to easily obtain $\mathcal{D}_{max}^+$ needed in Assumption 1.

A. Initialization for Privacy-Preserving Algorithm with Multiple State Decomposition

Our strategy is based on the event-triggered deterministic algorithm in [18] with some modifications (since the algorithm in [18] is not privacy-preserving). The main difference is the deployment of a mechanism that decomposes the initial state $y_j[0]$ of each node $v_j \in \mathcal{V}_p$ into $\mathcal{D}_{max}^+ + 2$ substates. The average of the $\mathcal{D}_{max}^+ + 2$ substates is equal to the initial state $y_j[0]$. Subsequently, each substate is transmitted to a different out-neighbor at a different time step.

In previous works (e.g., [6], [7], [13], [14], [19]), each node $v_j \in \mathcal{V}_p$ injects a nonzero offset u_j to its initial state. This means that it sets $\tilde{y}_j[0] = y_j[0] + u_j$, where $u_j \neq 0$. However, in our case we require 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 the initial state. Furthermore, each node v_j maintains its substate counter $s_j \in \mathbb{N}$, and its privacy variables $u_j^y[s_j] \in \mathbb{Z}$, $u_j^z[s_j] \in \mathbb{Z}$. At initialization, each node $v_j \in \mathcal{V}_p$ chooses the privacy variables $u_j^y[s_j] \in \mathbb{Z}$, $u_j^z[s_j] \in \mathbb{Z}$, to satisfy the following constraints:

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

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

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

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

Constraints (4a)–(4d) are explicitly analyzed below:

1. In (4a) each node v_j stops injecting nonzero offsets 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.

2. In (4b) the average of the total injected offset 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 in a way that allows the calculation of the exact quantized average of the initial states without any error.

3. In (4c) the substate $u_j^z[s_j]$ which is injected in the network by node v_j needs to be equal to 1 so that (i) the event-triggered conditions of the presented algorithm hold and (ii) the operation of the algorithm leads to the calculation of the exact average.

4. In (4d) each node v_j stops injecting nonzero offsets after $\mathcal{D}_{max}^+ + 1$ time steps which allows the calculation of the quantized average without any error.

We will argue that the above choices lead to the calculation of the exact quantized average in a privacy-preserving manner.

Remark 1. *Note here that each node $v_j \in \mathcal{V}_p$ that wishes to preserve its privacy chooses its privacy variables $u_j^y[s_j] \in \mathbb{Z}$, $u_j^z[s_j] \in \mathbb{Z}$ according to* (4a)–(4d). *Each node $v_i \notin \mathcal{V}_p$ that does not wish to preserve its privacy can simply set $u_i^y[s_j] = y_i[0]$, for every $s_i \in [0, \mathcal{D}_{max}^+ + 1]$. Then, each $v_i \notin \mathcal{V}_p$ simply executes the proposed algorithm. Note here* *that curious nodes execute the proposed algorithm with parameters chosen as described previously, but may communicate and reveal to other curious nodes their states and their received messages. However, they do not interfere with the computation of the quantized average in any way (e.g., they do not transmit corrupted messages to their neighbors).*

B. Privacy-Preserving Finite Transmission Event-Triggered Algorithm

```
Algorithm 1 Finite-Time Privacy-Preserving Event-Triggered Quantized Average Consensus

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 has knowledge of $\mathcal{D}^+_{max}$.
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 $\{0, 1, ..., \mathcal{D}^+_j - 1\}$.
2) Chooses $u^y_j[s_j]$ such that $(\sum_{s_j=0}^{\mathcal{D}^+_{max}+1} u^y_j[s_j])/(\mathcal{D}^+_{max} + 2) = y_j[0]$; $u^y_j[s'_j] = 0$ for $s'_j > \mathcal{D}^+_{max} + 1$; then, sets counter $s_j = 0$.
3) Chooses $u^z_j[s_j] = 1$ for $s_j \in [0,\ \mathcal{D}^+_{max} + 1]$, and $u^z_j[s'_j] = 0$ for $s'_j > \mathcal{D}^+_{max} + 1$.
4) Sets $y_j[0] = u^y_j[s_j]$, $z_j[0] = u^z_j[s_j]$, $z^s_j[0] = z_j[0]$, $y^s_j[0] = y_j[0]$, $q^s_j[0] = y^s_j[0]/z^s_j[0]$, $s_j = s_j + 1$, $S\_br_j = 0$, $M\_tr_j = 0$.
5) Broadcasts $z^s_j[0]$, $y^s_j[0]$ to every $v_l \in \mathcal{N}^+_j$.
Iteration: For $k = 0, 1, 2, \dots$, 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 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 $y_j[k] = y_j[k] + u^y_j[s_j]$, $z_j[k] = z_j[k] + u^z_j[s_j]$, and (ii) chooses $v_l \in \mathcal{N}^+_j$ according to $P_{lj}$ (in a round-robin fashion) and transmits $y_j[k]$, $z_j[k]$; then, sets $y_j[k] = 0$, $z_j[k] = 0$, $M\_tr_j = 0$, $s_j = s_j + 1$.
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).
Output: (3) holds for every $v_j \in \mathcal{V}$.
```

```
Algorithm 1.A Event-Triggered Conditions for Algorithm 1 (for each node $v_j$)

Input
$y^s_j[k]$, $z^s_j[k]$, $q^s_j[k]$, $y_j[k+1]$, $z_j[k+1]$, $S\_br_j$, $M\_tr_j$ and the received $y^s_i[k]$, $z^s_i[k]$ from every $v_i \in \mathcal{N}^-_j$.
Execution
1) Event Trigger Conditions 1: If
   Condition $(i)$: $z^s_i[k] > z^s_j[k]$, or
   Condition $(ii)$: $z^s_i[k] = z^s_j[k]$ and $y^s_i[k] > y^s_j[k]$,
   then sets
       $z^s_j[k+1] = \max_{v_i \in \mathcal{N}^-_j} z^s_i[k],$ and
       $y^s_j[k+1] = \max_{v_i \in \{v_{i'} \in \mathcal{N}^-_j \mid z^s_{i'}[k] = z^s_j[k+1]\}} y^s_i[k],$
   and sets $q^s_j[k+1] = \frac{y^s_j[k+1]}{z^s_j[k+1]}$, and $S\_br_j = 1$. If neither Condition (i) nor Condition (ii) hold, node $v_j$ sets $z^s_j[k+1] = z^s_j[k]$, $y^s_j[k+1] = y^s_j[k]$, $q^s_j[k+1] = q^s_j[k]$.
2) Event Trigger Conditions 2: If
   Condition $(i)$: $z_j[k+1] > z^s_j[k+1]$, or
   Condition $(ii)$: $z_j[k+1] = z^s_j[k+1]$ and $y_j[k+1] > y^s_j[k+1]$,
   then sets $z^s_j[k+1] = z_j[k+1]$, $y^s_j[k+1] = y_j[k+1]$, and sets $q^s_j[k+1] = \frac{y^s_j[k+1]}{z^s_j[k+1]}$ and $S\_br_j = 1$.
3) Event Trigger Conditions 3: If
   Condition $(i)$: $0 < z_j[k+1] < z^s_j[k+1]$ or
   Condition $(ii)$: $z_j[k+1] = z^s_j[k+1]$ and $y_j[k+1] < y^s_j[k+1]$,
   then sets $M\_tr_j = 1$.
Output
$y^s_j[k]$, $z^s_j[k]$, $q^s_j[k]$, $S\_br_j$, $M\_tr_j$.
```

Definition 2. *During the execution of Algorithm 1, at time step k_0, there is at least one node $v_{j'} \in \mathcal{V}$, for which*

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

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

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

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").

The intuition behind Algorithm 1 is as follows. Each node $v_j \in \mathcal{V}_p$ that would like to preserve its privacy performs the steps described below. Note that a node $v_i \in \mathcal{V}_n$ performs similar steps but with different values for the substate variables as discussed earlier.

Initialization.
A. Node v_j assigns to each outgoing edge a unique order P_{lj} in order to perform transmissions in a round-robin fashion.
B. Node v_j initializes the substate counter s_j to zero (i.e., $s_j = 0$) and the set of $(\mathcal{D}^+_{max} + 2)$ privacy variables $u^y_j[s_j]$, $u^z_j[s_j]$ according to (4a)–(4d) for $s_j \in [0, \mathcal{D}^+_{max} + 1]$.

C. Node v_j utilizes the substates $u_j^y[0]$, $u_j^z[0]$ as its initial state (i.e., it sets $y_j[0] = u_j^y[0]$ and $z_j[0] = u_j^z[0]$). Then, it considers its set of stored mass variables $y_j[0]$, $z_j[0]$ to be the "leading mass." For this reason, it sets its state variables $z_j^s[0]$, $y_j^s[0]$, $q_j^s[0]$ to be equal to the stored mass variables $y_j[0]$, $z_j[0]$, and then broadcasts the values of its state variables.

Iteration.
A. Node v_j receives the (possibly) transmitted state variables from its in-neighbors and, receives and stores the (possibly) transmitted mass variables from its in-neighbors.
B. If node v_j received a set of state variables and/or a set of mass variables from its in-neighbors, then it executes Algorithm 1.A. During Algorithm 1.A each node v_j checks:
B – Event Trigger Conditions 1**:** It checks whether the received set of state variables is equal to the "leading mass." If it receives messages from multiple in-neighbors it checks which set of state variables is the "leading mass." If Event Trigger Conditions 1 hold, it sets its state variables to be equal to the received set of state variables which is the "leading mass" and decides to broadcast its updated state variables (i.e., it sets $S\_br_j = 1$).
B – Event Trigger Conditions 2**:** It checks whether the set of mass variables it stored is the "leading mass." If this condition holds, it sets its state variables to be equal to the stored set of mass variables and decides to broadcast its updated state variables (i.e., it sets $S\_br_j = 1$).
B – Event Trigger Conditions 3**:** It checks whether the set of mass variables it stored is not the "leading mass" (i.e., it checks whether its state variables are equal to the "leading mass"). If this condition holds, this means that a pair of mass variables of another node in the network is the "leading mass" (and the state variables of node v_j became equal to the "leading mass" from Event Trigger Conditions 1). This means the stored pair of mass variables is not the "leading mass" and thus v_j decides to transmit its stored mass variables (i.e., it sets its transmission variable $M\_tr_j = 1$).
C. Node v_j sets its transmission variable M_tr_j to be equal to the maximum value of M_tr_j and the substate $u_j^z[s_j]$. This step is important for the privacy-preserving mechanism. Note that the value of the substate $u_j^z[s_j]$ is equal to 1 for $s_j \in [0, \mathcal{D}_{max}^+ + 1]$. This means that the value of the transmission variable M_tr_j will become equal to 1 for the first $\mathcal{D}_{max}^+ + 1$ time steps (since s_j becomes equal to 1 during the Initialization procedure). Thus, node v_j will perform transmissions of the substates $u_j^y[s_j]$, $u_j^z[s_j]$ towards its out-neighbors.
D. If M_tr_j is equal to 1, node v_j (i) injects the substates $u_j^y[s_j]$, $u_j^z[s_j]$ to its mass variables, (ii) transmits its mass variables towards an out-neighbor according to the unique order P_{lj}, and (iii) increases the substate counter s_j.
E. If S_br_j is equal to 1, node v_j broadcasts its state variables towards every out-neighbor. Then, it repeats the procedure.

C. Convergence Analysis of Algorithm 1

Due to space limitations, we omit the proofs of the two lemmas and the theorem below; they will be available in an extended version of our paper.

Lemma 1. *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*

$$z_i^s[k_0] \leq z_j[k_0], \tag{7}$$

or

$$z_i^s[k_0] = z_j[k_0] \;\; \text{and} \;\; y_i^s[k_0] \leq y_j[k_0]. \tag{8}$$

Lemma 2. *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 "Event Trigger 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.*

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 reach quantized average consensus after a finite number of time steps k_0 upper bounded by $1 + \mathcal{D}_{max}^+ + n^2 + (n-1)m^2$, where n is the number of nodes and m is the number of edges in the network, and $\mathcal{D}_{max}^+$ is the maximum out-degree in the network. Furthermore, each node stops transmitting towards its out-neighbors once quantized average consensus is reached.*

D. Topological Conditions for Privacy Preservation

Proposition 1. *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)–(4d)*. Curious nodes $v_c \in \mathcal{V}_c$ will not be able to identify the initial state $y_j[0]$ of $v_j \in \mathcal{V}_p$, as long as v_j has at least one in- or out-neighbor that aims to preserve its privacy, i.e., in-neighbor $v_i \in \mathcal{V}_p \cap \mathcal{N}_j^-$ or out-neighbor $v_l \in \mathcal{V}_p \cap \mathcal{N}_j^+$.*

Proof: We consider the following cases regarding the topological conditions for privacy preservation during the execution of Algorithm 1. Then, we summarize the results and derive the necessary and sufficient topological conditions for privacy preservation.
A. Every in- and out-neighbor of node v_j is curious (i.e., $v_i \in \mathcal{V}_c$, $\forall v_i \in \mathcal{N}_j^-$, and $v_l \in \mathcal{V}_c$, $\forall v_l \in \mathcal{N}_j^+$). In this case, the curious in- and out-neighbors communicate with each other and node $v_j \in \mathcal{V}_p$ will not be able to maintain 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 $s_j \in [1,\; \mathcal{D}_{max}^+]$.

Note that the average of every $u_j^y[s_j]$, for $s_j \in [0,\ \mathcal{D}_{max}^+]$, is equal to v_j's initial state $y_j[0]$. This means that curious nodes will be able to determine the initial state $y_j[0]$. As a result, for the case where every in- and out-neighbor of node v_j is curious, node v_j does not preserve the privacy of its initial state.

B. 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, $v_{l'}$ are curious (i.e., $v_i \in \mathcal{V}_c$, $\forall v_i \in \mathcal{N}_j^- \cup \mathcal{N}_{l'}^-$, and $v_l \in \mathcal{V}_c$, $\forall v_l \in (\mathcal{N}_j^+ \setminus \{v_{l'}\}) \cup \mathcal{N}_{l'}^+$). 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. Furthermore, 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, say $v_{l'}$, is neither curious nor following the privacy-preserving protocol and every other in- and out-neighbor of node v_j and $v_{l'}$ is curious, node v_j does not preserve the privacy of its initial state.

C. 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 all other in- and out-neighbors of both nodes v_j, $v_{l'}$ are curious (i.e., $v_i \in \mathcal{V}_c$, $\forall v_i \in \mathcal{N}_j^- \cup \mathcal{N}_{l'}^-$, and $v_l \in \mathcal{V}_c$, $\forall v_l \in (\mathcal{N}_j^+ \setminus \{v_{l'}\}) \cup \mathcal{N}_{l'}^+$). During the Iteration procedure, curious nodes will not be able to infer the substates transmitted from node v_j to node $v_{l'}$. This means that the inferred substates do not reveal the initial state of node v_j. As a result, curious nodes will not be able to infer the initial state of node v_j and the initial state of node $v_{l'}$. Thus, in this case node v_j preserves the privacy of its initial state.

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

From the four cases **A – D** we considered, 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, it is important to note that curious nodes will not be able to determine (i) the values of the messages transmitted from $v_{i'}$ to v_j, or (ii) the values of the messages transmitted from v_j to $v_{l'}$. Since the substate values can be arbitrary, 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 in (as already mentioned in Definition 1). □

Remark 2. *Note here that decomposing the initial state of every $v_j \in \mathcal{V}_p$ into $\mathcal{D}_{max}^+ + 2$ sets of substates is essential for privacy preservation. One set of substates is used as v_j's initial state. The remaining $\mathcal{D}_{max}^+ + 1$ substates are transmitted towards the out-neighbors of v_j. This means that v_j transmits at least one set of privacy variables (or substates) 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 at least one set of v_j's privacy variables and sums it with its own mass variables and privacy variables. Then $v_{l'}$ transmits its own privacy variables towards its out-neighbors (which are summed with the received privacy variables from v_j).*

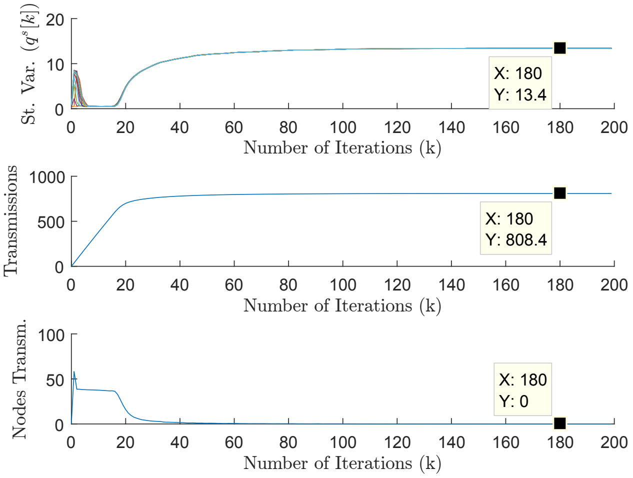

Fig. 2. Execution of Algorithm 1 averaged over 1000 random digraphs of 20 nodes. *Top figure:* Average values of node state variables plotted against the number of iterations. *Middle Figure:* Average total number of transmissions plotted against the number of iterations. *Bottom Figure:* Average number of nodes performing transmissions plotted against the number of iterations.

V. SIMULATION RESULTS

In this section, we illustrate the behavior of Algorithm 1 and the advantages of its operation. We analyze the scenario of 1000 randomly generated digraphs of 20 nodes each where, the initial quantized state of each node remained the same (for each one of the 1000 randomly generated digraphs); the average of the chosen initial states happened to be $q = 13.4$.

In Fig. 2, we illustrate Algorithm 1 over 1000 random digraphs of 20 nodes; we show the average number of time steps needed for quantized average consensus to be reached, the average number of transmissions accumulated until each time step, and the average number of nodes performing transmissions at each time step. We observe that Algorithm 1 converges after 180 time steps, with the average total number of transmissions performed until time step 180 being equal to 808.4. Additionally, we observe that the average number of nodes performing transmissions at each iteration becomes almost equal to zero after 50 time steps, and becomes eventually equal to zero after 180 time steps.

In Fig. 3 we plot the node state variables averaged over 1000 randomly generated digraphs of 20 nodes each, where the average of the initial states is $q = 13.4$. We compare Algorithm 1 against (i) the event-based offset algorithm in [13] (see middle of Fig. 3), and (ii) the initial zero-sum offset algorithm in [14] (see bottom of Fig. 3). In [13] (case (i)), the initial offset for every node v_j is $u_j \in [-100, -50]$ and the offset adding steps are $L_j \in [20, 40]$ during the execution. In [14] (case (ii)), the initial offset for every node v_j is $u_j \in [-100, 100]$ and the offsets are $u_j^{(l)} \in [-20, 20]$, for

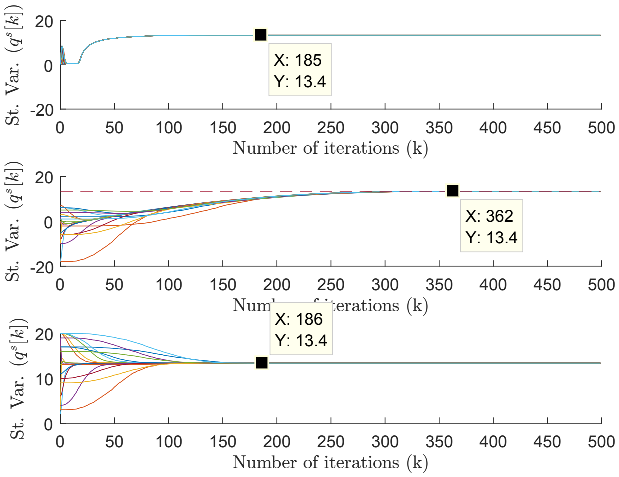

Fig. 3. Comparison between Algorithm 1, the event-based offset algorithm in [13], and the initial zero-sum offset algorithm in [14] (averaged over 1000 random digraphs of 20 nodes). *Top figure:* Average values of node state variables plotted against the number of iterations for Algorithm 1. *Middle Figure:* Average values of node state variables plotted against the number of iterations for the event-based offset algorithm in [13]. *Bottom Figure:* Average values of node state variables plotted against the number of iterations for the initial zero-sum offset algorithm in [14].

every $v_l \in \mathcal{N}_j^+$. We observe that Algorithm 1 converges after 185 time steps and again significantly outperforms the event-based offset algorithm in [13] which converges after 362 time steps. Furthermore, it is interesting to note that Algorithm 1 requires almost the same number of time steps as the initial zero-sum offset algorithm [14] which converges after 186 time steps. However, note that in [14] each node performs multiple simultaneous transmissions of different quantized values during the Initialization operation. Finally, note that neither [13] nor [14] exhibit finite transmission capabilities. This makes Algorithm 1 the first algorithm in the current literature in which each node (i) achieves the exact quantized average of the initial states, (ii) terminates its transmission operation, and (iii) preserves the privacy of its initial state.

VI. Conclusions

In this paper, we presented a privacy-preserving event-triggered quantized average consensus algorithm. The algorithm allows each node in the network to calculate the exact quantized average of the initial states in the form of a quantized fraction without revealing its initial quantized state to other nodes. The privacy-preserving strategy takes full advantage of the algorithm's event-based nature and finite transmission capabilities and allows each node to cease transmissions once convergence has been achieved without knowledge of any global parameter (e.g., network diameter). We also analyzed the algorithm's finite-time convergence and presented an upper bound on the required number of time steps. Then, we presented necessary and sufficient topological conditions under which the proposed algorithm allows nodes to preserve their privacy. Finally, we demonstrated the performance of our proposed algorithm and compared it against other algorithms in the existing literature.

References

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

[2] M. E. Chamie, J. Liu, and T. Basar, "Design and analysis of distributed averaging with quantized communication," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3870–3884, 2016.

[3] A. I. Rikos and C. N. Hadjicostis, "Event-triggered quantized average consensus via ratios of accumulated values," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 4035–4049, 2020.

[4] J. Cortés, G. E. Dullerud, S. Han, J. L. Ny, S. Mitra, and G. J. Pappas, "Differential privacy in control and network systems," *in IEEE Conference on Decision and Control*, pp. 4252–4272, 2016.

[5] 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.

[6] N. Manitara and C. N. Hadjicostis, "Privacy-preserving asymptotic average consensus," *in European Control Conference*, pp. 760–765, 2013.

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

[8] N. Rezazadeh and S. S. Kia, "Privacy preservation in a continuous-time static average consensus algorithm over directed graphs," *in American Control Conference (ACC)*, pp. 5890–5895, 2018.

[9] Y. Wang, "Privacy-preserving average consensus via state decomposition," *IEEE Transactions on Automatic Control*, vol. 64, no. 11, pp. 4711–4716, 2019.

[10] I. L. D. Ridgley, R. A. Freeman, and K. M. Lynch, "Private and hot-pluggable distributed averaging," *IEEE Control Systems Letters*, vol. 4, no. 4, pp. 988–993, 2020.

[11] C. N. Hadjicostis, "Privary preserving distributed average consensus via homomorphic encryption," *in IEEE Conference on Decision and Control*, pp. 1258–1263, 2018.

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

[13] A. I. Rikos, T. Charalambous, K. H. Johansson, and C. N. Hadjicostis, "Privacy-preserving event-triggered quantized average consensus," *in IEEE Conference on Decision and Control*, pp. 6246–6253, 2020.

[14] ——, "Distributed event-triggered algorithms for finite-time privacy-preserving quantized average consensus," *arXiv preprint arXiv:2102.06778*, 2021.

[15] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for web transactions," *ACM Transactions on Information and System Security (TISSEC)*, vol. 1, pp. 66–92, 1998.

[16] K. Chatzikokolakis and C. Palamidessi, "Probable innocence revisited," *Theoretical Computer Science*, vol. 367, no. 1-2, pp. 123–138, 2006.

[17] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726–737, 2008.

[18] A. I. Rikos, C. N. Hadjicostis, and K. H. Johansson, "Finite time exact quantized average consensus with limited resources and transmission stopping for energy-aware networks," *arXiv preprint arXiv:2003.14183*, 2021.

[19] T. Charalambous, N. E. Manitara, and C. N. Hadjicostis, "Privacy-preserving average consensus over digraphs in the presence of time delays," *in Allerton Conference on Communication, Control, and Computing*, pp. 238–245, 2019.