

# Algorithm-Level Confidentiality for Average Consensus on Time-Varying Directed Graphs

Huan Gao, *Member, IEEE*, and Yongqiang Wang, *Senior Member, IEEE*

**Abstract**—Average consensus plays a key role in distributed networks, with applications ranging from time synchronization, information fusion, load balancing, to decentralized control. Existing average consensus algorithms require individual agents to exchange explicit state values with their neighbors, which leads to the undesirable disclosure of sensitive information in the state. In this paper, we propose a novel average consensus algorithm for time-varying directed graphs that can protect the confidentiality of a participating agent against other participating agents. The algorithm injects randomness in interaction to obfuscate information on the algorithm-level and can ensure information-theoretic privacy without the assistance of any trusted third party or data aggregator. By leveraging the inherent robustness of consensus dynamics against random variations in interaction, our proposed algorithm can also guarantee the accuracy of average consensus. The algorithm is distinctly different from differential-privacy based average consensus approaches which enable confidentiality through compromising accuracy in obtained consensus value. Numerical simulations confirm the effectiveness and efficiency of our proposed approach.

**Index Terms**—Average consensus, confidentiality, time-varying directed graphs.

## I. INTRODUCTION

Averaging consensus is an important tool in distributed computing. For a network of  $N$  agents interacting on a graph, average consensus can enable the states of all agents to converge to the average of their initial values through local interactions between neighboring agents.

Recently, average consensus is finding increased applications in load balancing [2], [3], network synchronization [4], distributed information fusion [5], [6], and decentralized control [7], [8]. To ensure all agents converge to the average value of their initial values, conventional average consensus approaches require individual agents to exchange explicit state values with their neighbors. This results in the disclosure of sensitive state information, which is sometimes undesirable in terms of confidentiality. In fact, in many applications such as the smart grid, health-care or banking networks, confidentiality is crucial for promoting participation in collaboration since individual agents tend not to trade confidentiality for performance [9]–[11]. For instance, a group of people using average consensus to reach a common opinion may want to

keep their individual opinions secret [12]. Another typical example is power systems in which multiple generators have to reach agreement on cost while maintaining their individual generation information confidential [13].

To achieve confidentiality in average consensus, recently results have started to emerge. A commonly used confidentiality mechanism is differential privacy from the database literature [14]–[20] (and its variants [21], [22]) which injects independent (and hence uncorrelated) noises directly to agents' states in order to achieve confidentiality in average consensus. However, the use of independent noises on the states in these approaches prevents converging to the exact average value [23]. To improve consensus accuracy, which is crucial in cyber-physical systems and sensor networks, [24]–[31] inject carefully calculated *correlated* additive noises to agents' states, instead of independent (and hence uncorrelated) noises used in differential-privacy based approaches. (A similar approach was proposed in [32] to achieve maximum consensus.) However, these prior works only consider average consensus under balanced and static network topologies. Different from injecting noises to agents' states in the aforementioned approaches, [33] employed carefully designed mask maps to protect the actual states. Observability based approaches have also been reported to protect the confidentiality of multi-agent consensus [34]–[36]. Its idea is to design the topology of interactions such that the observability from a compromised agent is minimized, which amounts to minimizing the ability of the compromised agent to infer the initial states of other agents. Recently, encryption based approaches have been proposed to protect the confidentiality by encrypting exchanged messages with the assistance of additive homomorphic encryption [37]–[40], with the price of increasing computation and communication overhead. Another confidentiality approach was proposed in [41] where each agent's confidentiality is protected by decomposing its state into two sub-states. However, [41] relies on undirected interactions and is inapplicable to time-varying directed graphs considered in this paper.

This paper addresses the confidentiality of average consensus under time-varying directed graphs that are not necessarily balanced. Since push-sum based average consensus approaches do not require balanced topologies, we build our confidential average consensus algorithm on the push-sum approach. More specifically, to protect the confidentiality of the initial states of participating agents, in the first several iterations, we let agents send random values instead of their actual states to obfuscate their initial values. Of course, to guarantee the accuracy of average consensus, we have to judiciously design the state-update rule such that the randomness added in the first several iterations does not affect the final convergence result. Different from approaches injecting correlated additive noises directly

Part of the results was presented at 2018 IEEE Conference on Communications and Network Security (CNS) [1]. The work was supported in part by the National Science Foundation under Grants ECCS-1912702 and CCF-2106293.

Huan Gao was with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA. He is now with the School of Automation, Northwestern Polytechnical University, Xi'an 710129, China (email: huangao@nwpu.edu.cn). The work was done when Huan Gao was with Clemson University.

Yongqiang Wang is with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA (email: yongqiw@clemson.edu).

to agents' states, our approach adds independent (and hence uncorrelated in time) randomness directly to the average consensus dynamics, which makes it applicable to time-varying directed graphs. Compared with our prior work in [40] which employs homomorphic encryption to preserve confidentiality and [41] which protects each agent's confidentiality by decomposing its state into two sub-states, this paper proposes a different approach that enables confidentiality by judiciously adding randomness in interaction dynamics. More importantly, both [40] and [41] rely on undirected interactions and hence are inapplicable to time-varying directed graphs considered in this paper. Some of the results here were presented at the 2018 IEEE Conference on Communications and Network Security (CNS) [1]. Compared with the conference version, the journal version has the following significant differences: 1) the journal version extends the results for constant directed graphs in [1] to time-varying directed graphs; 2) the journal version provides formal and rigorous analysis of convergence rate that does not exist in the conference version; 3) the journal version allows multiple adversaries to collude to infer the sensitive value a target agent, which is not addressed in our conference version; and 4) the journal version revises and enhances the proposed confidential average consensus algorithm to guarantee information-theoretic privacy, which is stronger than the confidentiality achieved in the conference version.

## II. PRELIMINARIES AND PROBLEM FORMULATION

### A. Graph Representation

We represent a network of  $N$  agents as a sequence of time-varying directed graphs  $\{\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))\}$  where  $\mathcal{V} = \{1, 2, \dots, N\}$  is the set of agents and  $k = 0, 1, \dots$  is the time index.  $\mathcal{E}(k) \subset \mathcal{V} \times \mathcal{V}$  is the edge set at time  $k$ , whose elements are such that  $(i, j) \in \mathcal{E}(k)$  holds if and only if there exists a directed edge from agent  $j$  to agent  $i$  at time  $k$ , i.e., agent  $j$  can send messages to agent  $i$  at time  $k$ . For notational convenience, we assume that there are no self edges, i.e.,  $(i, i) \notin \mathcal{E}(k)$  for all  $k$  and  $i \in \mathcal{V}$ . At time  $k$ , each edge  $(i, j) \in \mathcal{E}(k)$  has an associated weight,  $p_{ij}(k) > 0$ . The out-neighbor set of agent  $i$  at time  $k$ , which represents the set of agents that can receive messages from agent  $i$  at time  $k$ , is denoted as  $\mathcal{N}_i^{\text{out}}(k) = \{j \in \mathcal{V} \mid (j, i) \in \mathcal{E}(k)\}$ . Similarly, at time  $k$ , the in-neighbor set of agent  $i$ , which represents the set of agents that can send messages to agent  $i$  at time  $k$ , is denoted as  $\mathcal{N}_i^{\text{in}}(k) = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}(k)\}$ . From the above definitions, it can be obtained that  $i \in \mathcal{N}_j^{\text{out}}(k)$  and  $j \in \mathcal{N}_i^{\text{in}}(k)$  are equivalent. Agent  $i$ 's out-degree at time instant  $k$  is represented by  $D_i^{\text{out}}(k) = |\mathcal{N}_i^{\text{out}}(k)|$  and its in-degree is represented by  $D_i^{\text{in}}(k) = |\mathcal{N}_i^{\text{in}}(k)|$ , where  $|\mathcal{S}|$  is the cardinality of the set  $\mathcal{S}$ .

At iteration  $k$ , the incidence matrix  $\mathbf{C}(k) = [c_{il}(k)]_{N \times E(k)}$  for graph  $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$  is defined as

$$c_{il}(k) = \begin{cases} 1 & \text{if the } l\text{-th edge in } \mathcal{E}(k) \text{ is } (i, j) \\ -1 & \text{if the } l\text{-th edge in } \mathcal{E}(k) \text{ is } (j, i) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where  $E(k) = |\mathcal{E}(k)|$  represents the number of edges in  $\mathcal{E}(k)$ . Note that the  $l$ -th column of  $\mathbf{C}(k)$  is corresponding to the  $l$ -th edge in  $\mathcal{E}(k)$ , and the sum of each column of  $\mathbf{C}(k)$  is 0, i.e.,  $\mathbf{1}^T \mathbf{C}(k) = \mathbf{0}^T$ .

For a sequence of time-varying directed graphs  $\{\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))\}$ , we define  $\mathcal{E}_\infty$  as the set of directed edges  $(i, j)$  that exist for infinitely many time instants, i.e.,

$$\mathcal{E}_\infty = \{(i, j) \mid (i, j) \in \mathcal{E}(k) \text{ for infinitely many indices } k\} \quad (2)$$

We focus on time-varying directed graphs which satisfy the following assumptions:

*Assumption 1:* For a sequence of time-varying directed graphs  $\{\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))\}$ , for any  $i, j \in \mathcal{V}$  with  $i \neq j$ , there exists at least one directed path from  $i$  to  $j$  in  $(\mathcal{V}, \mathcal{E}_\infty)$ , i.e.,  $(\mathcal{V}, \mathcal{E}_\infty)$  is strongly connected.

*Assumption 2:* For a sequence of time-varying directed graphs  $\{\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))\}$ , there exists an integer  $T \geq 1$  such that for every  $(i, j) \in \mathcal{E}_\infty$ , agent  $j$  directly communicates with agent  $i$  at least once in every  $T$  consecutive time instants.  $T$  is called intercommunication interval bound.

*Assumption 3:* We assume that each agent  $i$  has access to its out-degree  $D_i^{\text{out}}(k)$  at each iteration  $k$ .

*Remark 1:* Assumption 3 is widely used in existing literature on time-varying directed graphs such as [42]–[44]. In fact, in many directed graphs, it is feasible for a node to know its out-neighbors. For example, in many safety-critical systems such as industrial control systems, the exchange of data occurs in a directed way due to unidirectional gateways (aka data diode) whereas control messages (a special type of messages used to configure network connections) can be exchanged in a bidirectional manner to establish connections [45].

### B. The Conventional Push-Sum

The conventional push-sum considers  $N$  agents interacting on a constant directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , with each agent having an initial state  $x_i^0$  ( $i = 1, 2, \dots, N$ ) [46], [47]. Represent the average value of all initial states as  $\bar{x}^0 = \sum_{j=1}^N x_j^0 / N$ . The conventional push-sum algorithm conducts two iterative computations simultaneously, and allows each agent to obtain the exact average of the initial values  $\bar{x}^0$  in an asymptotic way. This mechanism is summarized in Algorithm 0 below:

---

#### Algorithm 0: The conventional push-sum algorithm

---

- 1)  $N$  agents interact on a constant directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ . Each agent  $i$  is initialized with  $s_i(0) = x_i^0$ ,  $w_i(0) = 1$ , and  $\pi_i(0) = s_i(0)/w_i(0)$ . The weight  $p_{ij}$  associated with the edge  $(i, j) \in \mathcal{E}$  satisfies  $p_{ij} \in (0, 1)$  if  $j \in \mathcal{N}_i^{\text{in}} \cup \{i\}$  is true and  $p_{ij} = 0$  otherwise. For any given  $j = 1, 2, \dots, N$ ,  $p_{ij}$  satisfies  $\sum_{i=1}^N p_{ij} = 1$ .
- 2) At iteration step  $k$ :
  - a) Agent  $i$  calculates  $p_{ji}s_i(k)$  and  $p_{ji}w_i(k)$ , and sends both values to all of its out-neighbors  $j \in \mathcal{N}_i^{\text{out}}$ .

- b) After receiving the values of  $p_{ij}s_j(k)$  and  $p_{ij}w_j(k)$  from all its in-neighbors  $j \in \mathcal{N}_i^{\text{in}}$ , agent  $i$  updates  $s_i$  and  $w_i$  as follows:

$$\begin{cases} s_i(k+1) = \sum_{j \in \mathcal{N}_i^{\text{in}} \cup \{i\}} p_{ij}s_j(k) \\ w_i(k+1) = \sum_{j \in \mathcal{N}_i^{\text{in}} \cup \{i\}} p_{ij}w_j(k) \end{cases} \quad (3)$$

- c) Agent  $i$  uses the ratio  $\pi_i(k+1) = s_i(k+1)/w_i(k+1)$  to estimate the average value  $\bar{x}^0 = \sum_{j=1}^N x_j^0/N$ .

For the sake of notational simplicity, we rewrite (3) in the following more compact form:

$$\begin{cases} \mathbf{s}(k+1) = \mathbf{P}\mathbf{s}(k) \\ \mathbf{w}(k+1) = \mathbf{P}\mathbf{w}(k) \end{cases} \quad (4)$$

where  $\mathbf{s}(k) = [s_1(k), s_2(k), \dots, s_N(k)]^T$  and  $\mathbf{w}(k) = [w_1(k), w_2(k), \dots, w_N(k)]^T$ , and  $\mathbf{P} = [p_{ij}]$ . From Algorithm 0, we have  $\mathbf{s}(0) = [x_1^0, x_2^0, \dots, x_N^0]^T$  and  $\mathbf{w}(0) = \mathbf{1}$ . We can also obtain that the matrix  $\mathbf{P}$  is column-stochastic, i.e.,  $\sum_{i=1}^N p_{ij} = 1$  holds for  $j = 1, 2, \dots, N$ .

At iteration step  $k$ , each agent computes the ratio  $\pi_i(k+1) = s_i(k+1)/w_i(k+1)$  to estimate the average value  $\bar{x}^0 = \sum_{j=1}^N x_j^0/N$ . Since  $\mathcal{G}$  is assumed to be a strongly connected directed graph,  $\mathbf{P}^k$  will converge to a rank-1 matrix exponentially fast [48], [49]. Defining  $\mathbf{P}^\infty$  as the limit of  $\mathbf{P}^k$  as  $k \rightarrow \infty$ , we can obtain the form of  $\mathbf{P}^\infty$  as  $\mathbf{P}^\infty = \mathbf{v}\mathbf{1}^T$  where  $\mathbf{v} = [v_1, v_2, \dots, v_N]^T$ . Using the facts  $\mathbf{s}(k) = \mathbf{P}^k\mathbf{s}(0)$  and  $\mathbf{w}(k) = \mathbf{P}^k\mathbf{w}(0)$ , we can further have [50]:

$$\pi_i(\infty) = \frac{s_i(\infty)}{w_i(\infty)} = \frac{[\mathbf{P}^\infty\mathbf{s}(0)]_i}{[\mathbf{P}^\infty\mathbf{w}(0)]_i} = \frac{v_i \sum_{j=1}^N s_j(0)}{v_i \sum_{j=1}^N w_j(0)} = \bar{x}^0 \quad (5)$$

where  $[\mathbf{P}^\infty\mathbf{s}(0)]_i$  and  $[\mathbf{P}^\infty\mathbf{w}(0)]_i$  represent the  $i$ -th element of vector  $\mathbf{P}^\infty\mathbf{s}(0)$  and vector  $\mathbf{P}^\infty\mathbf{w}(0)$ , respectively. Hence, all estimates  $\pi_1(k), \pi_2(k), \dots, \pi_N(k)$  will asymptotically converge to the average  $\bar{x}^0 = \sum_{j=1}^N x_j^0/N$ .

### C. Problem Formulation

In this paper, we will address average consensus under time-varying directed graphs while protecting the confidentiality of participating agents against adversaries. To this end, we first present some assumptions and definitions.

*Assumption 4:* We assume that all agents' initial states  $x_i^0$  are bounded. Without loss of generality, the lower bound and upper bound are denoted as  $a$  and  $b$ , respectively. Both  $a$  and  $b$  are assumed known to all agents.

*Remark 2:* It is worth noting that although the bounds  $a$  and  $b$  are assumed known to all agents, this does not mean that the minimum and maximum of all agents' initial states are known to all agents. In fact,  $a$  (resp.  $b$ ) can be arbitrarily small (resp. large) compared with the actual minimum (resp. maximum) of agents' initial states.

*Definition 1:* We define an honest-but-curious adversary as an agent who follows all protocol steps correctly but collects received messages in an attempt to infer the initial value of other participating agents.

*Assumption 5:* We assume that agents can collude, i.e., a set of honest-but-curious agents  $\mathcal{A}$  can share information with each other to infer the initial value  $x_i^0$  of a target agent  $i \notin \mathcal{A}$ .

*Definition 2:* We define that confidentiality (privacy) of the initial value  $x_i^0$  of agent  $i$  is preserved if  $x_i^0$  is indistinguishable from the viewpoint of honest-but-curious adversaries  $\mathcal{A}$ . By "indistinguishable," we mean that the probability distribution of information set accessible to  $\mathcal{A}$  does not change when agent  $i$ 's initial state  $x_i^0$  is altered to any  $\tilde{x}_i^0 \neq x_i^0$  under the constraint that the sum of the initial states of all nodes not in  $\mathcal{A}$  (i.e.,  $\sum_{j \in \mathcal{V} \setminus \mathcal{A}} x_j^0$ ) is unchanged.

Our definition of confidentiality requires perfect indistinguishability of a target agent's different initial states from the viewpoint of honest-but-curious adversaries  $\mathcal{A}$ , and, therefore, is more stringent than the confidentiality definition in [31], [32], [51]–[53] which defines confidentiality as the inability of an adversary to *uniquely* determine the sensitive value.

We next show that the conventional push-sum is not confidential. From (3) and (4), an honest-but-curious agent  $i$  can receive  $p_{ij}s_j(0)$  and  $p_{ij}w_j(0)$  from its in-neighbor agent  $j$  after the first iteration step  $k = 0$ . Then agent  $i$  is able to uniquely determine  $x_j^0$  by  $x_j^0 = s_j(0) = \frac{p_{ij}s_j(0)}{p_{ij}w_j(0)}$  using the fact  $w_j(0) = 1$ . Therefore, an honest-but-curious agent can always infer the initial values of all its in-neighbors, and hence the conventional push-sum algorithm cannot provide confidentiality against honest-but-curious adversaries. It is worth noting that using a similar argument, we can also obtain that the conventional push-sum is not confidential even when the weight is allowed to be time-varying (e.g., [47].)

## III. THE CONFIDENTIALITY ALGORITHM AND PERFORMANCE ANALYSIS

In this section, we will propose a confidential average consensus algorithm for time-varying directed graphs, and then provide rigorous analysis of its convergence rate and enabled strength of confidentiality.

### A. Confidential Average Consensus Algorithm

The analysis above reveals that using the same weight  $p_{ij}$  for both  $p_{ij}s_j(0)$  and  $p_{ij}w_j(0)$  discloses the initial state value. Motivated by this observation and the work in [29], here we introduce a novel confidential average consensus algorithm which injects randomness in the dynamics of interactions in iterations  $k = 0, \dots, K$ . Note that here  $K$  is a non-negative integer and is known to every agent. Its influence will be discussed in detail in Remark 11 and Remark 12.

---

#### Algorithm 1: Confidential average consensus algorithm

---

- 1)  $N$  agents interact on a sequence of time-varying directed graphs  $\{\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))\}$ . Each agent  $i$  is initialized with  $s_i(0) = \frac{1}{N^2} + \frac{(N-2)(x_i^0 - a)}{(b-a)N^2} \in [\frac{1}{N^2}, \frac{N-1}{N^2}]$ ,  $w_i(0) = 1$ , and  $\pi_i(0) = \frac{b-a}{N-2} [N \times \text{frac}(\frac{Ns_i(0)}{w_i(0)}) - 1] + a$  where the function  $\text{frac}(x) = x - \lfloor x \rfloor$  denotes the fractional part of a real number  $x$  (here  $\lfloor x \rfloor$  represents the largest integer not greater than  $x$ ).
- 2) At iteration step  $k$ :

- Agent  $i$  generates a set of random weights  $\{p_{ji}(k) \in (\varepsilon, 1) \mid j \in \mathcal{N}_i^{\text{out}}(k) \cup \{i\}\}$  with the sum of this set equal to 1, and sets  $\Delta w_{ji}(k) = p_{ji}(k)w_i(k)$  for  $j \in \mathcal{N}_i^{\text{out}}(k) \cup \{i\}$ .
- If  $k \leq K$ , agent  $i$  independently generates uniformly distributed values  $\Delta s_{ji}(k) \in [0, 1)$  for its out-neighbors  $j \in \mathcal{N}_i^{\text{out}}(k)$ , and sets  $\Delta s_{ii}(k) = \text{frac}(s_i(k) - \sum_{j \in \mathcal{N}_i^{\text{out}}(k)} \Delta s_{ji}(k))$ ; otherwise, agent  $i$  sets  $\Delta s_{ji}(k) = p_{ji}(k)s_i(k)$  for  $j \in \mathcal{N}_i^{\text{out}}(k) \cup \{i\}$ .
- Agent  $i$  sends  $\Delta s_{ji}(k)$  and  $\Delta w_{ji}(k)$  to its out-neighbors  $j \in \mathcal{N}_i^{\text{out}}(k)$ .
- After receiving  $\Delta s_{ij}(k)$  and  $\Delta w_{ij}(k)$  from its in-neighbors  $j \in \mathcal{N}_i^{\text{in}}(k)$ , agent  $i$  updates  $s_i$  and  $w_i$  as

$$s_i(k+1) = \begin{cases} \text{frac}\left(\sum_{j \in \mathcal{N}_i^{\text{in}}(k) \cup \{i\}} \Delta s_{ij}(k)\right) & \text{for } k \leq K \\ \sum_{j \in \mathcal{N}_i^{\text{in}}(k) \cup \{i\}} \Delta s_{ij}(k) & \text{for } k \geq K+1 \end{cases} \quad (6)$$

and

$$w_i(k+1) = \sum_{j \in \mathcal{N}_i^{\text{in}}(k) \cup \{i\}} \Delta w_{ij}(k) \quad \text{for } k \geq 0 \quad (7)$$

respectively.

- Agent  $i$  uses the ratio  $\pi_i(k+1) = \frac{b-a}{N-2}[N \times \text{frac}(\frac{Ns_i(k+1)}{w_i(k+1)} - 1) + a]$  to estimate the average value  $\bar{x}^0 = \sum_{j=1}^N x_j^0/N$ .

*Remark 3:* Compared to the conventional confidentiality-violating push-sum algorithm which broadcasts messages, Algorithm 1 needs agent  $i$  to send different random numbers to different out-neighbors in iterations  $k \leq K$ . This is a price of obtaining confidentiality without losing accuracy in the time-varying directed topology case.

*Remark 4:* The way of injecting randomness in  $\Delta s_{ji}(k)$  is different in iterations  $k \leq K$  from  $k > K$ . In fact, in iterations  $k \leq K$ ,  $\Delta s_{ji}(k)$  can be nonzero even when  $s_i(k)$  is zero. This is crucial in enabling strong confidentiality as receiving  $\Delta s_{ji}(k)$  of a value zero will not allow the recipient to infer information about  $s_i(k)$ .

*Remark 5:* In Algorithm 1, the coupling weights are randomly chosen at every iteration. Compared with the commonly used deterministic setting in which the weights are set as  $p_{ji}(k) = 1/(D_i^{\text{out}}(k)+1)$  for all  $j \in \mathcal{N}_i^{\text{out}}(k) \cup \{i\}$ , our setting is more general since it includes the commonly used setting as a special case by fixing  $p_{ji}(k)$  to deterministic values. Furthermore, the random weights beyond step  $K$  in Algorithm 1 can provide additional confidentiality protection for intermediate states  $s_i(k)$  after iteration  $K$ . Given  $\Delta s_{ji}(k) = p_{ji}(k)s_i(k)$  for  $k \geq K+1$ , we can see that using random weights  $p_{ji}(k)$  makes the intermediate state  $s_i(k)$  more difficult to infer than using deterministic weights  $p_{ji}(k)$ .

Setting  $\Delta s_{ji}(k)$ ,  $\Delta w_{ji}(k)$ , and  $p_{ji}(k)$  to 0 for  $j \notin \mathcal{N}_i^{\text{out}}(k) \cup \{i\}$ , we can rewrite the update rules of  $\mathbf{s}(k)$  for

$k \geq K+1$  and  $\mathbf{w}(k)$  for  $k \geq 0$  as

$$\begin{cases} s_i(k+1) = \sum_{j=1}^N \Delta s_{ij}(k) = \sum_{j=1}^N p_{ij}(k)s_j(k) & \text{for } k \geq K+1 \\ w_i(k+1) = \sum_{j=1}^N \Delta w_{ij}(k) = \sum_{j=1}^N p_{ij}(k)w_j(k) & \text{for } k \geq 0 \end{cases} \quad (8)$$

Denoting  $\mathbf{s}(k)$ ,  $\mathbf{w}(k)$ , and  $\mathbf{P}(k)$  as  $\mathbf{s}(k) = [s_1(k) \cdots s_N(k)]^T$ ,  $\mathbf{w}(k) = [w_1(k) \cdots w_N(k)]^T$ , and  $\mathbf{P}(k) = [p_{ij}(k)]_{N \times N}$ , we can further rewrite (8) into a matrix form

$$\begin{cases} \mathbf{s}(k+1) = \mathbf{P}(k)\mathbf{s}(k) & \text{for } k \geq K+1 \\ \mathbf{w}(k+1) = \mathbf{P}(k)\mathbf{w}(k) & \text{for } k \geq 0 \end{cases} \quad (9)$$

For iteration  $k = 0$ , we have  $\mathbf{s}(0) = [x_1^0, x_2^0, \dots, x_N^0]^T$  and  $\mathbf{w}(0) = \mathbf{1}$ . From Algorithm 1, we know that  $\mathbf{P}(k)$  in (9) is time-varying and column-stochastic for  $k \geq 0$ .

Defining the transition matrix as follows

$$\Phi(k:t) = \mathbf{P}(k) \cdots \mathbf{P}(t) \quad (10)$$

for all  $k$  and  $t$  with  $k \geq t$ , where  $\Phi(k:k) = \mathbf{P}(k)$ , we can rewrite (9) as

$$\begin{cases} \mathbf{s}(k+1) = \Phi(k:K+1)\mathbf{s}(K+1) & \text{for } k \geq K+1 \\ \mathbf{w}(k+1) = \Phi(k:0)\mathbf{w}(0) & \text{for } k \geq 0 \end{cases} \quad (11)$$

## B. Convergence Analysis

Next we prove that Algorithm 1 can guarantee that the estimates of all agents converge to the exact average value of initial values. We will also analyze the rate of convergence of Algorithm 1. Using the convergence definition in [43] and [54], we define the rate of convergence to be at least  $\gamma \in (0, 1)$  if there exists a positive constant value  $C$  such that  $\|\pi(k) - \bar{x}^0 \mathbf{1}\| \leq C\gamma^k$  is true for all  $k$ , where  $\pi(k) = [\pi_1(k), \dots, \pi_N(k)]^T$  and  $\bar{x}^0 = \sum_{j=1}^N x_j^0/N$  is the average value. Note that this definition means a smaller  $\gamma$  corresponding to a faster convergence. To analyze the convergence rate of Algorithm 1, we first introduce Lemma 1 below:

*Lemma 1:* For a network of  $N$  agents represented by a sequence of time-varying directed graphs  $\{\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))\}$  which satisfy Assumptions 1, 2, and 3, under Algorithm 1, each agent  $i$  has  $w_i(k) \geq \varepsilon^{T(N-1)}$  for  $k \geq 1$  where  $T$  is defined in Assumption 2.

*Proof:* For  $k \geq 1$ , from (11) we have

$$\mathbf{w}(k) = \Phi(k-1:0)\mathbf{1} \quad (12)$$

Represent  $\delta(k)$  as

$$\delta(k) \triangleq \min_{1 \leq i \leq N} w_i(k) = \min_{1 \leq i \leq N} [\Phi(k-1:0)\mathbf{1}]_i \quad (13)$$

for  $k \geq 1$ . To prove  $w_i(k) \geq \varepsilon^{T(N-1)}$  for  $k \geq 1$ , it is sufficient to prove  $\delta(k) \geq \varepsilon^{T(N-1)}$  for  $k \geq 1$ . We divide our proof into two parts:  $1 \leq k \leq T(N-1)$  and  $k \geq T(N-1)+1$ .

**Part 1:**  $\delta(k) \geq \varepsilon^{T(N-1)}$  for  $1 \leq k \leq T(N-1)$ . One can verify that the following relationship holds

$$\begin{aligned} & [\Phi(k-1:0)]_{ii} \\ &= [\mathbf{P}(k-1) \cdots \mathbf{P}(0)]_{ii} \\ &\geq [\mathbf{P}(k-1)]_{ii} [\mathbf{P}(k-2) \cdots \mathbf{P}(0)]_{ii} \\ &\geq \varepsilon [\Phi(k-2:0)]_{ii} \end{aligned} \quad (14)$$

Given  $[\Phi(0:0)]_{ii} = [\mathbf{P}(0)]_{ii} \geq \varepsilon$ , one can obtain  $[\Phi(k-1:0)]_{ii} \geq \varepsilon^k$ . Therefore, it follows that

$$\begin{aligned} & [\Phi(k-1:0)\mathbf{1}]_i \geq [\Phi(k-1:0)]_{ii} \\ &\geq \varepsilon^k \geq \varepsilon^{T(N-1)} \end{aligned} \quad (15)$$

is true for  $i = 1, \dots, N$  and  $1 \leq k \leq T(N-1)$ , implying  $\delta(k) \geq \varepsilon^{T(N-1)}$  for  $1 \leq k \leq T(N-1)$ .

**Part 2:**  $\delta(k) \geq \varepsilon^{T(N-1)}$  for  $k \geq T(N-1) + 1$ . Under Assumptions 1 and 2, and the requirements on weights  $p_{ij}(k)$  in Algorithm 1, and following the arguments in Lemma 2 in [55], we can obtain

$$[\Phi(k-1:k-T(N-1))]_{ij} \geq \varepsilon^{T(N-1)} \quad (16)$$

for  $1 \leq i, j \leq N$ . Since  $k \geq T(N-1) + 1$  holds and  $\mathbf{P}(k)$  is a column-stochastic matrix,

$$\Phi(k-T(N-1)-1:0) = \mathbf{P}(k-T(N-1)-1) \cdots \mathbf{P}(0) \quad (17)$$

should also be a column-stochastic matrix. Further using the fact  $\Phi(k-1:0) = \Phi(k-1:k-T(N-1))\Phi(k-T(N-1)-1:0)$  leads to  $[\Phi(k-1:0)]_{ij} \geq \varepsilon^{T(N-1)}$  for  $1 \leq i, j \leq N$ . Therefore, we have

$$[\Phi(k-1:0)\mathbf{1}]_i \geq N\varepsilon^{T(N-1)} \geq \varepsilon^{T(N-1)} \quad (18)$$

for  $i = 1, \dots, N$ , meaning  $\delta(k) \geq \varepsilon^{T(N-1)}$  for  $k \geq T(N-1) + 1$ .

Based on  $\delta(k) \geq \varepsilon^{T(N-1)}$  for  $k \geq 1$ , we can obtain  $w_i(k) \geq \varepsilon^{T(N-1)}$  for  $k \geq 1$ . In summary, we always have  $w_i(k) \geq \varepsilon^{T(N-1)}$  for  $k \geq 1$ . ■

*Theorem 1:* For a network of  $N$  agents represented by a sequence of time-varying directed graphs  $\{\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))\}$  satisfying Assumptions 1, 2, 3, and 4, under Algorithm 1, the estimate  $\pi_i(k) = \frac{b-a}{N-2} [N \times \text{frac}(\frac{Ns_i(k)}{w_i(k)}) - 1] + a$  of each agent  $i$  will converge to the average  $\bar{x}^0 = \sum_{j=1}^N x_j^0 / N$ . More specifically, the rate of convergence of Algorithm 1 is at least  $\gamma = (1 - \varepsilon^{T(N-1)})^{\frac{1}{T(N-1)}} \in (0, 1)$ , meaning that there exists a positive constant value  $C$  satisfying  $\|\pi(k) - \bar{x}^0 \mathbf{1}\| \leq C\gamma^k$  for all  $k$ .

*Proof:* From (6), we have

$$\begin{aligned} & \text{frac}\left(\sum_{i=1}^N s_i(k+1)\right) \\ &= \text{frac}\left(\sum_{i=1}^N \text{frac}\left(\sum_{j \in \mathcal{N}_i^{\text{in}}(k) \cup \{i\}} \Delta s_{ij}(k)\right)\right) \\ &= \text{frac}\left(\sum_{i=1}^N \sum_{j \in \mathcal{N}_i^{\text{in}}(k)} \Delta s_{ij}(k) + \sum_{i=1}^N \Delta s_{ii}(k)\right) \\ &= \text{frac}\left(\sum_{i=1}^N \sum_{j \in \mathcal{N}_i^{\text{in}}(k)} \Delta s_{ij}(k) \right. \\ &\quad \left. + \sum_{i=1}^N \text{frac}\left(s_i(k) - \sum_{j \in \mathcal{N}_i^{\text{out}}(k)} \Delta s_{ji}(k)\right)\right) \\ &= \text{frac}\left(\sum_{i=1}^N s_i(k)\right) \end{aligned} \quad (19)$$

for  $k \leq K$  where in the derivation we used the property  $\text{frac}(x+y) = \text{frac}(x + \text{frac}(y)) = \text{frac}(\text{frac}(x) + y) = \text{frac}(\text{frac}(x) + \text{frac}(y))$  for any  $x, y \in \mathbb{R}$ . According to Assumption 4,  $x_i^0 \in [a, b]$  holds for each agent  $i$ . Given  $s_i(0) = \frac{1}{N^2} + \frac{(N-2)(x_i^0 - a)}{(b-a)N^2}$ , one can obtain  $s_i(0) \in [\frac{1}{N^2}, \frac{N-1}{N^2}]$ , leading to

$$\text{frac}\left(\sum_{i=1}^N s_i(0)\right) = \sum_{i=1}^N s_i(0) \in \left[\frac{1}{N}, \frac{N-1}{N}\right] \subset (0, 1) \quad (20)$$

Further combining (19) and (20) yields

$$\text{frac}\left(\sum_{i=1}^N s_i(K+1)\right) = \cdots = \text{frac}\left(\sum_{i=1}^N s_i(0)\right) = \sum_{i=1}^N s_i(0) \quad (21)$$

Since  $\mathbf{P}(k)$  is column stochastic, from (9) we have

$$\mathbf{1}^T \mathbf{w}(k+1) = \mathbf{1}^T \mathbf{w}(k) = \cdots = \mathbf{1}^T \mathbf{w}(0) = N \quad (22)$$

for  $k \geq 0$ . Then we rewrite (11) as

$$\begin{cases} \mathbf{s}(K+l+1) = \Phi(K+l:K+1)\mathbf{s}(K+1) \\ \mathbf{w}(K+l+1) = \Phi(K+l:K+1)\mathbf{w}(K+1) \end{cases} \quad (23)$$

for  $l \geq 1$ . Under Assumptions 1 and 2, and the requirements on weights  $p_{ij}(k)$  in Algorithm 1, following Proposition 1(b) in [56], we know that the transition matrix  $\Phi(K+l:K+1)$  will converge to a stochastic vector  $\varphi(K+l)$  with a geometric rate for all  $i$  and  $j$ , i.e., for all  $i, j = 1, \dots, N$  and  $l \geq 1$ , we have

$$|[\Phi(K+l:K+1)]_{ij} - \varphi_i(K+l)| \leq C_0 \gamma^{l-1} \quad (24)$$

with  $C_0 = 2(1 + \varepsilon^{-T(N-1)})/(1 - \varepsilon^{T(N-1)})$  and  $\gamma = (1 - \varepsilon^{T(N-1)})^{\frac{1}{T(N-1)}}$ . Defining  $\mathbf{M}(K+l:K+1)$  as

$$\mathbf{M}(K+l:K+1) \triangleq \Phi(K+l:K+1) - \varphi(K+l) \mathbf{1}^T \quad (25)$$

we have

$$|[\mathbf{M}(K+l:K+1)]_{ij}| \leq C_0 \gamma^{l-1} \quad (26)$$

for all  $i, j = 1, \dots, N$  and  $l \geq 1$ . Further combining (25) with (23) leads to

$$\begin{cases} \mathbf{s}(K+l+1) = \mathbf{M}(K+l:K+1)\mathbf{s}(K+1) \\ \quad + \boldsymbol{\varphi}(K+l)\mathbf{1}^T\mathbf{s}(K+1) \\ \mathbf{w}(K+l+1) = \mathbf{M}(K+l:K+1)\mathbf{w}(K+1) \\ \quad + N\boldsymbol{\varphi}(K+l) \end{cases} \quad (27)$$

where in the derivation we used  $\mathbf{1}^T\mathbf{w}(K+1) = N$  from (22). Then from (27), we have

$$\begin{aligned} & \frac{s_i(K+l+1)}{w_i(K+l+1)} - \frac{\sum_{j=1}^N s_j(K+1)}{N} \\ &= \frac{s_i(K+l+1)}{w_i(K+l+1)} - \frac{\mathbf{1}^T\mathbf{s}(K+1)}{N} \\ &= \frac{s_i(K+l+1)}{w_i(K+l+1)} - \frac{\mathbf{1}^T\mathbf{s}(K+1)w_i(K+l+1)}{Nw_i(K+l+1)} \\ &= \frac{[\mathbf{M}(K+l:K+1)\mathbf{s}(K+1)]_i + \varphi_i(K+l)\mathbf{1}^T\mathbf{s}(K+1)}{w_i(K+l+1)} \\ &\quad - \frac{\mathbf{1}^T\mathbf{s}(K+1)[\mathbf{M}(K+l:K+1)\mathbf{w}(K+1)]_i}{Nw_i(K+l+1)} \\ &\quad - \frac{\mathbf{1}^T\mathbf{s}(K+1)N\varphi_i(K+l)}{Nw_i(K+l+1)} \\ &= \frac{[\mathbf{M}(K+l:K+1)\mathbf{s}(K+1)]_i}{w_i(K+l+1)} \\ &\quad - \frac{\mathbf{1}^T\mathbf{s}(K+1)[\mathbf{M}(K+l:K+1)\mathbf{w}(K+1)]_i}{Nw_i(K+l+1)} \end{aligned} \quad (28)$$

Therefore, for  $i = 1, \dots, N$  and  $l \geq 1$ , we can obtain

$$\begin{aligned} & \left| N \frac{s_i(K+l+1)}{w_i(K+l+1)} - \sum_{j=1}^N s_j(K+1) \right| \\ &\leq \frac{N|[\mathbf{M}(K+l:K+1)\mathbf{s}(K+1)]_i|}{w_i(K+l+1)} \\ &\quad + \frac{N|\mathbf{1}^T\mathbf{s}(K+1)[\mathbf{M}(K+l:K+1)\mathbf{w}(K+1)]_i|}{Nw_i(K+l+1)} \\ &\leq \frac{N}{\varepsilon^{T(N-1)}} \left( \max_j |[\mathbf{M}(K+l:K+1)]_{ij}| \right) \|\mathbf{s}(K+1)\|_1 \\ &\quad + \frac{N}{\varepsilon^{T(N-1)}} |\mathbf{1}^T\mathbf{s}(K+1)| \left( \max_j |[\mathbf{M}(K+l:K+1)]_{ij}| \right) \end{aligned} \quad (29)$$

where in the derivation we used  $w_i(K+l+1) \geq \varepsilon^{T(N-1)}$  from Lemma 1 and  $\|\mathbf{w}(K+1)\|_1 = \sum_{i=1}^N |w_i(K+1)| = \mathbf{1}^T\mathbf{w}(K+1) = N$  from (22). Further using the relationship  $|\mathbf{1}^T\mathbf{s}(K+1)| \leq \|\mathbf{s}(K+1)\|_1$  and (26) yields

$$\left| N \frac{s_i(k)}{w_i(k)} - \sum_{j=1}^N s_j(K+1) \right| \leq C_1 \gamma^k \quad (30)$$

for  $k \geq K+2$  with  $C_1$  given by

$$C_1 = 2NC_0 \|\mathbf{s}(K+1)\|_1 \varepsilon^{-T(N-1)} \gamma^{-K-2} \quad (31)$$

Given  $s_i(0) = \frac{1}{N^2} + \frac{(N-2)(x_i^0 - a)}{(b-a)N^2}$ , we have

$$\begin{aligned} \bar{x}^0 &= \frac{1}{N} \sum_{j=1}^N x_j^0 = \frac{b-a}{N-2} \left( N \sum_{j=1}^N s_j(0) - 1 \right) + a \\ &= \frac{b-a}{N-2} \left( N \times \text{frac} \left( \sum_{j=1}^N s_j(K+1) \right) - 1 \right) + a \end{aligned} \quad (32)$$

where in the derivation we used  $\text{frac}(\sum_{j=1}^N s_j(K+1)) = \sum_{j=1}^N s_j(0)$  from (21). Combining (32) with  $\pi_i(k) = \frac{b-a}{N-2} [N \times \text{frac}(\frac{Ns_i(k)}{w_i(k)}) - 1] + a$  leads to

$$\begin{aligned} & \pi_i(k) - \bar{x}^0 \\ &= \frac{b-a}{N-2} N \left( \text{frac} \left( \frac{Ns_i(k)}{w_i(k)} \right) - \text{frac} \left( \sum_{j=1}^N s_j(K+1) \right) \right) \end{aligned} \quad (33)$$

From (20) and (21), one can obtain  $\text{frac}(\sum_{j=1}^N s_j(K+1)) = \sum_{j=1}^N s_j(0) \in [\frac{1}{N}, \frac{N-1}{N}]$ . Defining  $\eta$  as  $\eta \triangleq \sum_{j=1}^N s_j(0) \in [\frac{1}{N}, \frac{N-1}{N}]$ , then there must exist an integer  $Q$  such that  $\sum_{j=1}^N s_j(K+1) = Q + \eta$  holds. From (30), we can see that there must exist a positive integer  $K_1 \geq K+2$  such that

$$\left| N \frac{s_i(k)}{w_i(k)} - \sum_{j=1}^N s_j(K+1) \right| \leq C_1 \gamma^k < \frac{1}{N} \quad (34)$$

holds for  $k \geq K_1$ . Then it follows naturally one has

$$Q < Q + \eta - C_1 \gamma^k \leq N \frac{s_i(k)}{w_i(k)} \leq Q + \eta + C_1 \gamma^k < Q + 1 \quad (35)$$

for  $k \geq K_1$ , which leads to

$$0 < \eta - C_1 \gamma^k \leq \text{frac} \left( N \frac{s_i(k)}{w_i(k)} \right) \leq \eta + C_1 \gamma^k < 1 \quad (36)$$

for  $k \geq K_1$ . Thus, we have

$$\begin{aligned} |\pi_i(k) - \bar{x}^0| &= \frac{b-a}{N-2} N \left| \text{frac} \left( \frac{Ns_i(k)}{w_i(k)} \right) - \eta \right| \\ &\leq \frac{b-a}{N-2} NC_1 \gamma^k \end{aligned} \quad (37)$$

and further

$$\|\boldsymbol{\pi}(k) - \bar{x}^0 \mathbf{1}\| \leq C_2 \gamma^k \quad (38)$$

for  $k \geq K_1$  with  $C_2 \triangleq \frac{b-a}{N-2} N \sqrt{N} C_1$ .

For  $k \leq K_1 - 1$ , from (33), we have

$$|\pi_i(k) - \bar{x}^0| < \frac{b-a}{N-2} N \quad (39)$$

which further implies

$$\|\boldsymbol{\pi}(k) - \bar{x}^0 \mathbf{1}\| < \frac{b-a}{N-2} N \sqrt{N} \quad (40)$$

for  $k \leq K_1 - 1$ . Defining  $C$  as

$$C \triangleq \max \left\{ C_2, \frac{b-a}{N-2} N \sqrt{N} \gamma^{-k} \mid 0 \leq k \leq K_1 - 1 \right\} \quad (41)$$

one can obtain

$$\|\boldsymbol{\pi}(k) - \bar{x}^0 \mathbf{1}\| \leq C \gamma^k \quad (42)$$

for all  $k$ . Therefore, each agent  $i$  will converge to the average value  $\bar{x}^0 = \sum_{j=1}^N x_j^0/N$  with the rate of convergence of at least  $\gamma = (1 - \varepsilon^{T(N-1)})^{\frac{1}{T(N-1)}} \in (0, 1)$ . ■

From Theorem 1, we can see that a smaller  $\gamma$  means a faster convergence. Under the relationship  $\gamma = (1 - \varepsilon^{T(N-1)})^{\frac{1}{T(N-1)}}$ , to expedite the convergence, i.e., a smaller  $\gamma$ , it is sufficient to increase  $\varepsilon$ , which amounts to reducing the width of the range  $(\varepsilon, 1)$  for the random selection of  $p_{ji}(k)$ . Note that although a reduced range  $(\varepsilon, 1)$  enables an honest-but-curious adversary to obtain a better estimation of the range of agent  $i$ 's intermediate states  $s_i(k)$  and  $w_i(k)$  for  $k \geq K+1$  from received  $p_{ji}(k)s_i(k)$  and  $p_{ji}(k)w_i(k)$ , it does not affect the confidentiality of agent  $i$ 's initial state  $x_i^0$ , as will be shown in the following subsection. It is also worth noting that to meet the requirement of randomly selecting weights in our algorithm,  $\varepsilon$  cannot be arbitrarily close to 1. In fact,  $\varepsilon$  must satisfy  $\varepsilon < 1/\max_{i,k}(D_i^{\text{out}}(k) + 1)$ . An easy way to select  $\varepsilon$  is to set  $0 < \varepsilon < 1/N$  since  $D_i^{\text{out}}(k) \leq N-1$  is true for all  $k$  and  $i \in \mathcal{V}$ .

*Remark 6:* Theorem 1 provides a detailed analysis of the rate of convergence under time-varying directed graphs, the results on which are sparse in the literature on Push-Sum under time-varying random weights.

### C. Confidentiality Analysis

Before presenting our main confidentiality result, we first introduce the following lemma.

*Lemma 2:* Consider a network of  $N$  agents represented by a sequence of time-varying directed graphs  $\{\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))\}$  which satisfy Assumptions 1, 2, 3, 4, and 5. Under the proposed Algorithm 1, if at some time instant  $0 \leq k^* \leq K$ , agent  $i \notin \mathcal{A}$  has an in-neighbor or out-neighbor  $l$  not belonging to  $\mathcal{A}$ , then under  $\mathcal{I}_s^* \triangleq \{\Delta s_{mn}(k) \mid (m, n) \in \mathcal{E}(k), (m, n) \neq (i, l), (m, n) \neq (l, i), k = 0, 1, \dots, K\}$ , the joint probability distributions of  $s_i(K+1)$  and  $s_l(K+1)$  are identical for any two different initial states  $\mathbf{x}^0, \tilde{\mathbf{x}}^0 \in [a, b]^N$  subject to  $x_i^0 + x_l^0 = \tilde{x}_i^0 + \tilde{x}_l^0$  and  $x_j^0 = \tilde{x}_j^0$  for  $j \in \mathcal{V} \setminus \{i, l\}$ .

*Proof:* According to (6), we can rewrite the update rule of  $s_i(k)$  as follows

$$\begin{aligned} s_i(k+1) &= \text{frac} \left( \sum_{j \in \mathcal{N}_i^{\text{in}}(k)} \Delta s_{ij}(k) + s_i(k) - \sum_{j \in \mathcal{N}_i^{\text{out}}(k)} \Delta s_{ji}(k) \right) \end{aligned} \quad (43)$$

for  $k \leq K$ . We stack all variables  $\Delta s_{mn}(k)$  into a vector  $\Delta \mathbf{s}(k)$  according to indices of edges in  $\mathcal{E}(k)$ , namely, the index of  $\Delta s_{mn}(k)$  is determined by the index of the edge  $(m, n)$  in  $\mathcal{E}(k)$ . Then we can further rewrite (43) in the following more compact form:

$$\mathbf{s}(k+1) = \text{frac}(\mathbf{s}(k) + \mathbf{C}(k)\Delta \mathbf{s}(k)) \quad (44)$$

for  $k \leq K$ , where  $\mathbf{C}(k)$  is the incidence matrix for graph  $\mathcal{G}(k)$  at iteration  $k$ .

Define the subsets of agents  $\mathcal{H}$  and  $\mathcal{R}$  as  $\mathcal{H} = \{i, l\}$  and  $\mathcal{R} = \mathcal{V} \setminus (\mathcal{H} \cup \mathcal{A})$ , respectively. Let  $N_{\mathcal{A}} = |\mathcal{A}|$  and  $N_{\mathcal{R}} = |\mathcal{R}|$  represent the number of agents in  $\mathcal{A}$  and  $\mathcal{R}$ , respectively. It is clear that  $\mathcal{H}$ ,  $\mathcal{A}$ , and  $\mathcal{R}$  are disjoint, and

$\mathcal{H} \cup \mathcal{A} \cup \mathcal{R} = \mathcal{V}$  holds. For graph  $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$ , we define the subgraph  $\mathcal{G}_{\mathcal{H}}(k)$  as  $\mathcal{G}_{\mathcal{H}}(k) = (\mathcal{H}, \mathcal{E}_{\mathcal{H}}(k))$  where  $\mathcal{E}_{\mathcal{H}}(k) \subset \mathcal{E}(k)$  is the set of edges entirely within  $\mathcal{H}$ . The union of subgraphs  $\mathcal{G}_{\mathcal{H}}(k)$  for iterations  $k = 0, 1, \dots, K$  is further denoted as  $\mathcal{G}_{\mathcal{H}}^* = \bigcup_{k=0}^K \mathcal{G}_{\mathcal{H}}(k) = (\mathcal{H}, \mathcal{E}_{\mathcal{H}}^*)$ , where

$\mathcal{E}_{\mathcal{H}}^* = \bigcup_{k=0}^K \mathcal{E}_{\mathcal{H}}(k)$  is the union of edge sets  $\mathcal{E}_{\mathcal{H}}(k)$  for iterations  $k = 0, 1, \dots, K$ . Denote the edge set  $\mathcal{E}_{\mathcal{A}}(k)$  as the collection of edges associated with all agents in  $\mathcal{A}$  at iteration  $k$ , i.e.,  $\mathcal{E}_{\mathcal{A}}(k) = \{(m, n) \mid (m, n) \in \mathcal{E}(k), m \text{ or } n \in \mathcal{A}\}$ . Then the set of remaining edges not belonging to  $\mathcal{E}_{\mathcal{H}}(k)$  or  $\mathcal{E}_{\mathcal{A}}(k)$  can be denoted as  $\mathcal{E}_{\mathcal{R}}(k) = \mathcal{E}(k) \setminus (\mathcal{E}_{\mathcal{H}}(k) \cup \mathcal{E}_{\mathcal{A}}(k))$ , which is a collection of edges that are either entirely within  $\mathcal{R}$  or between  $\mathcal{R}$  and  $\mathcal{H}$ . Let  $E_{\mathcal{H}}(k) = |\mathcal{E}_{\mathcal{H}}(k)|$ ,  $E_{\mathcal{A}}(k) = |\mathcal{E}_{\mathcal{A}}(k)|$ , and  $E_{\mathcal{R}}(k) = |\mathcal{E}_{\mathcal{R}}(k)|$  represent the number of edges in  $\mathcal{E}_{\mathcal{H}}(k)$ ,  $\mathcal{E}_{\mathcal{A}}(k)$ , and  $\mathcal{E}_{\mathcal{R}}(k)$ , respectively. Note that  $\mathcal{E}_{\mathcal{H}}(k)$ ,  $\mathcal{E}_{\mathcal{A}}(k)$ , and  $\mathcal{E}_{\mathcal{R}}(k)$  are disjoint, and we always have  $\mathcal{E}_{\mathcal{H}}(k) \cup \mathcal{E}_{\mathcal{A}}(k) \cup \mathcal{E}_{\mathcal{R}}(k) = \mathcal{E}(k)$ . Without loss of generality, we can partition the state vector  $\mathbf{s}(k)$  as  $\mathbf{s}(k) = [\mathbf{s}_{\mathcal{H}}(k)^T \ \mathbf{s}_{\mathcal{A}}(k)^T \ \mathbf{s}_{\mathcal{R}}(k)^T]^T$  where  $\mathbf{s}_{\mathcal{H}}(k)$ ,  $\mathbf{s}_{\mathcal{A}}(k)$ , and  $\mathbf{s}_{\mathcal{R}}(k)$  denote the state vectors of agents in  $\mathcal{H}$ ,  $\mathcal{A}$ , and  $\mathcal{R}$ , respectively. Then we can further rewrite the update rule of  $\mathbf{s}(k)$  for  $k \leq K$  as

$$\begin{aligned} \begin{bmatrix} \mathbf{s}_{\mathcal{H}}(k+1) \\ \mathbf{s}_{\mathcal{A}}(k+1) \\ \mathbf{s}_{\mathcal{R}}(k+1) \end{bmatrix} &= \text{frac} \left( \begin{bmatrix} \mathbf{s}_{\mathcal{H}}(k) \\ \mathbf{s}_{\mathcal{A}}(k) \\ \mathbf{s}_{\mathcal{R}}(k) \end{bmatrix} \right. \\ &\quad \left. + \begin{bmatrix} \mathbf{C}_{\mathcal{H}}(k) & \mathbf{C}_{\mathcal{A}}^1(k) & \mathbf{C}_{\mathcal{R}}^1(k) \\ \mathbf{0}_{N_{\mathcal{A}} \times E_{\mathcal{H}}(k)} & \mathbf{C}_{\mathcal{A}}^2(k) & \mathbf{0}_{N_{\mathcal{A}} \times E_{\mathcal{R}}(k)} \\ \mathbf{0}_{N_{\mathcal{R}} \times E_{\mathcal{H}}(k)} & \mathbf{C}_{\mathcal{A}}^3(k) & \mathbf{C}_{\mathcal{R}}^2(k) \end{bmatrix} \begin{bmatrix} \Delta \mathbf{s}_{\mathcal{H}}(k) \\ \Delta \mathbf{s}_{\mathcal{A}}(k) \\ \Delta \mathbf{s}_{\mathcal{R}}(k) \end{bmatrix} \right) \end{aligned} \quad (45)$$

where  $\mathbf{C}_{\mathcal{H}}(k)$  is the incidence matrix for subgraph  $\mathcal{G}_{\mathcal{H}}(k) = (\mathcal{H}, \mathcal{E}_{\mathcal{H}}(k))$ , and  $\Delta \mathbf{s}_{\mathcal{H}}(k)$ ,  $\Delta \mathbf{s}_{\mathcal{A}}(k)$ , and  $\Delta \mathbf{s}_{\mathcal{R}}(k)$  are vectors stacking  $\Delta s_{mn}(k)$  corresponding to edge sets  $\mathcal{E}_{\mathcal{H}}(k)$ ,  $\mathcal{E}_{\mathcal{A}}(k)$ , and  $\mathcal{E}_{\mathcal{R}}(k)$ , respectively. It is obvious that  $\Delta \mathbf{s}_{\mathcal{A}}(k)$  is completely known to agents in  $\mathcal{A}$  since every element in  $\Delta \mathbf{s}_{\mathcal{A}}(k)$  is either sent or received by the agents in  $\mathcal{A}$ . From (45), we can further obtain

$$\begin{aligned} \begin{bmatrix} \mathbf{s}_{\mathcal{H}}(K+1) \\ \mathbf{s}_{\mathcal{A}}(K+1) \\ \mathbf{s}_{\mathcal{R}}(K+1) \end{bmatrix} &= \text{frac} \left( \begin{bmatrix} \mathbf{s}_{\mathcal{H}}(0) \\ \mathbf{s}_{\mathcal{A}}(0) \\ \mathbf{s}_{\mathcal{R}}(0) \end{bmatrix} \right. \\ &\quad \left. + \begin{bmatrix} \mathbf{C}_{\mathcal{H}}^* & \mathbf{C}_{\mathcal{A}}^{1*} & \mathbf{C}_{\mathcal{R}}^{1*} \\ \mathbf{0}_{N_{\mathcal{A}} \times E_{\mathcal{H}}^*} & \mathbf{C}_{\mathcal{A}}^{2*} & \mathbf{0}_{N_{\mathcal{A}} \times E_{\mathcal{R}}^*} \\ \mathbf{0}_{N_{\mathcal{R}} \times E_{\mathcal{H}}^*} & \mathbf{C}_{\mathcal{A}}^{3*} & \mathbf{C}_{\mathcal{R}}^{2*} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{s}_{\mathcal{H}}^* \\ \Delta \mathbf{s}_{\mathcal{A}}^* \\ \Delta \mathbf{s}_{\mathcal{R}}^* \end{bmatrix} \right) \end{aligned} \quad (46)$$

where  $E_{\mathcal{H}}^* = \sum_{k=0}^K E_{\mathcal{H}}(k)$ ,  $E_{\mathcal{R}}^* = \sum_{k=0}^K E_{\mathcal{R}}(k)$ , and

$$\begin{aligned} \mathbf{C}_{\mathcal{H}}^* &= [\mathbf{C}_{\mathcal{H}}(0) \ \cdots \ \mathbf{C}_{\mathcal{H}}(K)] \\ \mathbf{C}_{\mathcal{A}}^{t*} &= [\mathbf{C}_{\mathcal{A}}^t(0) \ \cdots \ \mathbf{C}_{\mathcal{A}}^t(K)], \ t = 1, 2, 3 \\ \mathbf{C}_{\mathcal{R}}^{t*} &= [\mathbf{C}_{\mathcal{R}}^t(0) \ \cdots \ \mathbf{C}_{\mathcal{R}}^t(K)], \ t = 1, 2 \\ \Delta \mathbf{s}_{\mathcal{H}}^* &= [\Delta \mathbf{s}_{\mathcal{H}}(0)^T \ \cdots \ \Delta \mathbf{s}_{\mathcal{H}}(K)^T]^T \\ \Delta \mathbf{s}_{\mathcal{A}}^* &= [\Delta \mathbf{s}_{\mathcal{A}}(0)^T \ \cdots \ \Delta \mathbf{s}_{\mathcal{A}}(K)^T]^T \\ \Delta \mathbf{s}_{\mathcal{R}}^* &= [\Delta \mathbf{s}_{\mathcal{R}}(0)^T \ \cdots \ \Delta \mathbf{s}_{\mathcal{R}}(K)^T]^T \end{aligned} \quad (47)$$

It is worth noting that  $\mathbf{C}_{\mathcal{H}}^*$  is the incidence matrix for graph

$\mathcal{G}_{\mathcal{H}}^* = \bigcup_{k=0}^K \mathcal{G}_{\mathcal{H}}(k)$ . From (46), we have

$$\begin{aligned} & \mathbf{s}_{\mathcal{H}}(K+1) \\ &= \text{frac}(\mathbf{s}_{\mathcal{H}}(0) + \mathbf{C}_{\mathcal{H}}^* \Delta \mathbf{s}_{\mathcal{H}}^* + \mathbf{C}_{\mathcal{A}}^{1*} \Delta \mathbf{s}_{\mathcal{A}}^* + \mathbf{C}_{\mathcal{R}}^{1*} \Delta \mathbf{s}_{\mathcal{R}}^*) \end{aligned} \quad (48)$$

Denoting  $\mathbf{v} = [v_1 \ v_2]^T$  as

$$\mathbf{v} \triangleq \text{frac}(\mathbf{C}_{\mathcal{H}}^* \Delta \mathbf{s}_{\mathcal{H}}^*) \quad (49)$$

next we prove that  $\mathbf{v}$  is uniformly distributed in  $[0, 1) \times [0, 1)$  subject to  $\text{frac}(v_1 + v_2) = 0$  if at some time instant  $0 \leq k^* \leq K$ , agent  $i$  has an in-neighbor or out-neighbor  $l$  not belonging to  $\mathcal{A}$ .

Given  $\mathcal{H} = \{i, l\}$ , if at some time instant  $0 \leq k^* \leq K$ , agent  $i$  has an in-neighbor or out-neighbor  $l$  not belonging to  $\mathcal{A}$ , the columns of  $\mathbf{C}_{\mathcal{H}}^*$  are either  $[1 \ -1]^T$  ( $l$  is an in-neighbor of agent  $i$ ) or  $[-1 \ 1]^T$  ( $l$  is an out-neighbor of agent  $i$ ). Denote the  $j$ -th column of  $\mathbf{C}_{\mathcal{H}}^*$  as  $\mathbf{c}_j^*$ , then  $\mathbf{c}_j^*$  can be expressed as

$$\mathbf{c}_j^* = r_j \mathbf{c}_1^* \quad (50)$$

where  $r_j \in \{1, -1\}$  is the corresponding coefficient. Defining  $\mathbf{r}$  as  $\mathbf{r} = [r_1 \ \dots \ r_{E_{\mathcal{H}}}]$ , one can obtain

$$\begin{aligned} \mathbf{v} &= \text{frac}(\mathbf{C}_{\mathcal{H}}^* \Delta \mathbf{s}_{\mathcal{H}}^*) = \text{frac}(\mathbf{c}_1^* \mathbf{r} \Delta \mathbf{s}_{\mathcal{H}}^*) \\ &= \text{frac}(\mathbf{c}_1^* \cdot \text{frac}(\mathbf{r} \Delta \mathbf{s}_{\mathcal{H}}^*)) \end{aligned} \quad (51)$$

Since all the elements in  $\Delta \mathbf{s}_{\mathcal{H}}^*$  are independent of each other and uniformly distributed in  $[0, 1)$ , we have that  $\text{frac}(\mathbf{r} \Delta \mathbf{s}_{\mathcal{H}}^*)$  is uniformly distributed in  $[0, 1)$ . As  $\mathbf{c}_1^*$  is either  $[1 \ -1]^T$  or  $[-1 \ 1]^T$ , we have that  $\mathbf{v}$  is uniformly distributed in  $[0, 1) \times [0, 1)$  subject to  $\text{frac}(v_1 + v_2) = 0$ .

From (48), we have  $\mathbf{s}_{\mathcal{H}}(K+1) = \text{frac}(\mathbf{s}_{\mathcal{H}}(0) + \mathbf{v} + \mathbf{C}_{\mathcal{A}}^{1*} \Delta \mathbf{s}_{\mathcal{A}}^* + \mathbf{C}_{\mathcal{R}}^{1*} \Delta \mathbf{s}_{\mathcal{R}}^*)$ . Further taking into account the fact that  $\mathbf{v}$  is uniformly distributed in  $[0, 1) \times [0, 1)$  subject to  $\text{frac}(v_1 + v_2) = 0$  if at some time instant  $0 \leq k^* \leq K$ , agent  $i$  has an in-neighbor or out-neighbor  $l$  not belonging to  $\mathcal{A}$ , we can obtain that  $\mathbf{s}_{\mathcal{H}}(K+1)$  is also uniformly distributed in  $[0, 1) \times [0, 1)$  subject to  $\text{frac}(\mathbf{1}^T \mathbf{s}_{\mathcal{H}}(K+1)) = \text{frac}(\mathbf{1}^T \mathbf{s}_{\mathcal{H}}(0) + \mathbf{1}^T \mathbf{C}_{\mathcal{A}}^{1*} \Delta \mathbf{s}_{\mathcal{A}}^* + \mathbf{1}^T \mathbf{C}_{\mathcal{R}}^{1*} \Delta \mathbf{s}_{\mathcal{R}}^*)$ .

Now we analyze the probability distributions of  $\mathbf{s}_{\mathcal{H}}(K+1)$  under different initial conditions of agent  $i$ . For any two different initial conditions  $\mathbf{x}^0, \tilde{\mathbf{x}}^0 \in [a, b]^N$  subject to  $x_i^0 + x_l^0 = \tilde{x}_i^0 + \tilde{x}_l^0$  and  $x_j^0 = \tilde{x}_j^0$  for  $j \in \mathcal{V} \setminus \{i, l\}$ , one can obtain  $s_i(0) + s_l(0) = \tilde{s}_i(0) + \tilde{s}_l(0)$ . Note that all the elements in  $\Delta \mathbf{s}_{\mathcal{A}}^*$  and  $\Delta \mathbf{s}_{\mathcal{R}}^*$  belong to the set  $\mathcal{I}_s^* = \{\Delta s_{mn}(k) \mid (m, n) \in \mathcal{E}(k), (m, n) \neq (i, l), (m, n) \neq (l, i), k = 0, 1, \dots, K\}$ . Thus, given  $\mathcal{I}_s^*$ , both  $\mathbf{s}_{\mathcal{H}}(K+1)$  and  $\tilde{\mathbf{s}}_{\mathcal{H}}(K+1)$  are uniformly distributed in  $[0, 1) \times [0, 1)$  subject to  $\text{frac}(\mathbf{1}^T \mathbf{s}_{\mathcal{H}}(K+1)) = \text{frac}(\mathbf{1}^T \mathbf{s}_{\mathcal{H}}(0) + \mathbf{1}^T \mathbf{C}_{\mathcal{A}}^{1*} \Delta \mathbf{s}_{\mathcal{A}}^* + \mathbf{1}^T \mathbf{C}_{\mathcal{R}}^{1*} \Delta \mathbf{s}_{\mathcal{R}}^*) = \text{frac}(\mathbf{1}^T \tilde{\mathbf{s}}_{\mathcal{H}}(0) + \mathbf{1}^T \mathbf{C}_{\mathcal{A}}^{1*} \Delta \mathbf{s}_{\mathcal{A}}^* + \mathbf{1}^T \mathbf{C}_{\mathcal{R}}^{1*} \Delta \mathbf{s}_{\mathcal{R}}^*) = \text{frac}(\mathbf{1}^T \tilde{\mathbf{s}}_{\mathcal{H}}(K+1))$ . Therefore, given  $\mathcal{I}_s^*$ , the probability distributions of  $\mathbf{s}_{\mathcal{H}}(K+1)$  under different  $\mathbf{x}^0$  and  $\tilde{\mathbf{x}}^0$  are identical when  $x_j^0 = \tilde{x}_j^0$  for  $j \in \mathcal{V} \setminus \{i, l\}$  and  $x_i^0 + x_l^0 = \tilde{x}_i^0 + \tilde{x}_l^0$  hold for some agent  $l$  that is an in-neighbor or out-neighbor of agent  $i$  at some time instant  $0 \leq k^* \leq K$  but does not belong to  $\mathcal{A}$ . ■

Now we are in position to present our main confidentiality result.

**Theorem 2:** Consider a network of  $N$  agents represented by a sequence of time-varying directed graphs  $\{\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))\}$  which satisfy Assumptions 1, 2, 3, 4, and 5. Under the proposed Algorithm 1, the confidentiality of agent  $i \notin \mathcal{A}$  can be preserved against  $\mathcal{A}$  if at some time instant  $0 \leq k^* \leq K$ , agent  $i$  has an in-neighbor or out-neighbor  $l$  not belonging to  $\mathcal{A}$ .

*Proof:* To show that the confidentiality of agent  $i$  can be protected, we have to show that under different initial values of agent  $i$ , the probability distributions are identical for honest-but-curious agents  $\mathcal{A}$ ' information set. It suffices to prove that the probability distributions of information sets accessible to  $\mathcal{A}$  are identical for all  $\mathbf{x}^0, \tilde{\mathbf{x}}^0 \in [a, b]^N$  subject to  $x_i^0 + x_l^0 = \tilde{x}_i^0 + \tilde{x}_l^0$  and  $x_j^0 = \tilde{x}_j^0$  for  $j \in \mathcal{V} \setminus \{i, l\}$  where agent  $l$  is an in-neighbor or out-neighbor of agent  $i$  at some time instant  $0 \leq k^* \leq K$  and does not belong to  $\mathcal{A}$ . Note that under such  $\mathbf{x}^0$  and  $\tilde{\mathbf{x}}^0$ , the sum of the initial states of all nodes not in  $\mathcal{A}$  keeps unchanged, i.e.,  $\sum_{j \in \mathcal{V} \setminus \mathcal{A}} x_j^0 = \sum_{j \in \mathcal{V} \setminus \mathcal{A}} \tilde{x}_j^0$ . Given  $s_i(0) = \frac{1}{N^2} + \frac{(N-2)(x_i^0 - a)}{(b-a)N^2}$ , this is equivalent to proving that the probability distributions of information sets accessible to  $\mathcal{A}$  are identical for all  $\mathbf{s}(0), \tilde{\mathbf{s}}(0) \in [\frac{1}{N^2}, \frac{N-1}{N^2}]^N$  subject to  $s_i(0) + s_l(0) = \tilde{s}_i(0) + \tilde{s}_l(0)$  and  $s_j(0) = \tilde{s}_j(0)$  for  $j \in \mathcal{V} \setminus \{i, l\}$ . Denoting  $\mathcal{I}_s(k)$  and  $\mathcal{I}_w(k)$  as

$$\begin{aligned} \mathcal{I}_s(k) &= \{\mathbf{s}_{\mathcal{A}}(k), \Delta \mathbf{s}_{\mathcal{A}}(k)\} \\ \mathcal{I}_w(k) &= \{\mathbf{w}_{\mathcal{A}}(k), \Delta \mathbf{w}_{\mathcal{A}}(k)\} \end{aligned} \quad (52)$$

where  $\mathbf{s}_{\mathcal{A}}(k)$  and  $\mathbf{w}_{\mathcal{A}}(k)$  are state vectors of agents in  $\mathcal{A}$ , and  $\Delta \mathbf{s}_{\mathcal{A}}(k)$  and  $\Delta \mathbf{w}_{\mathcal{A}}(k)$  are augmented vectors of  $\Delta s_{mn}(k)$  and  $\Delta w_{mn}(k)$  corresponding to edge set  $\mathcal{E}_{\mathcal{A}}(k)$ , respectively. We can see that agents in  $\mathcal{A}$  have access to both  $\mathcal{I}_s(k)$  and  $\mathcal{I}_w(k)$  at each iteration  $k$ . Further denote  $\mathcal{I}_1^s, \mathcal{I}_1^w, \mathcal{I}_2^s, \mathcal{I}_2^w, \mathcal{I}_1$ , and  $\mathcal{I}_2$  as

$$\begin{aligned} \mathcal{I}_1^s &= \bigcup_{k=0}^K \mathcal{I}_s(k) & \mathcal{I}_1^w &= \bigcup_{k=0}^K \mathcal{I}_w(k) \cup \{\mathbf{w}(0)\} \\ \mathcal{I}_2^s &= \bigcup_{k=K+1}^{\infty} \mathcal{I}_s(k) & \mathcal{I}_2^w &= \bigcup_{k=K+1}^{\infty} \mathcal{I}_w(k) \\ \mathcal{I}_1 &= \mathcal{I}_1^s \cup \mathcal{I}_1^w & \mathcal{I}_2 &= \mathcal{I}_2^s \cup \mathcal{I}_2^w \end{aligned} \quad (53)$$

According to Algorithm 1,  $\mathcal{I} \triangleq \mathcal{I}_1 \cup \mathcal{I}_2$  represents all the information accessible to  $\mathcal{A}$ . We denote the conditional probability density of  $\mathcal{I}$  given  $\mathbf{s}(0)$  as  $f(\mathcal{I} | \mathbf{s}(0))$ . Thus, to show that the confidentiality of agent  $i$  can be protected, it is equivalent to proving  $f(\mathcal{I} | \mathbf{s}(0)) = f(\mathcal{I} | \tilde{\mathbf{s}}(0))$  for all  $\mathbf{s}(0), \tilde{\mathbf{s}}(0) \in [\frac{1}{N^2}, \frac{N-1}{N^2}]^N$  subject to  $s_i(0) + s_l(0) = \tilde{s}_i(0) + \tilde{s}_l(0)$  and  $s_j(0) = \tilde{s}_j(0)$  for  $j \in \mathcal{V} \setminus \{i, l\}$ .

Since

$$f(\mathcal{I} | \mathbf{s}(0)) = f(\mathcal{I}_1, \mathcal{I}_2 | \mathbf{s}(0)) = f(\mathcal{I}_1 | \mathbf{s}(0)) f(\mathcal{I}_2 | \mathcal{I}_1, \mathbf{s}(0)) \quad (54)$$

holds, to show  $f(\mathcal{I} | \mathbf{s}(0)) = f(\mathcal{I} | \tilde{\mathbf{s}}(0))$ , it suffices to prove that

$$f(\mathcal{I}_1 | \mathbf{s}(0)) = f(\mathcal{I}_1 | \tilde{\mathbf{s}}(0)) \quad (55)$$

and

$$f(\mathcal{I}_2 | \mathcal{I}_1, \mathbf{s}(0)) = f(\mathcal{I}_2 | \mathcal{I}_1, \tilde{\mathbf{s}}(0)) \quad (56)$$



hold for any  $\mathbf{s}(0), \tilde{\mathbf{s}}(0) \in [\frac{1}{N^2}, \frac{N-1}{N^2}]^N$  subject to  $s_i(0) + s_l(0) = \tilde{s}_i(0) + \tilde{s}_l(0)$  and  $s_j(0) = \tilde{s}_j(0)$  for  $j \in \mathcal{V} \setminus \{i, l\}$ .

We first show  $f(\mathcal{I}_1 | \mathbf{s}(0)) = f(\mathcal{I}_1 | \tilde{\mathbf{s}}(0))$ . Since  $\mathcal{I}_1^w$  is independent of  $\mathcal{I}_1^s$  and  $\mathbf{s}(0)$ , one can obtain  $f(\mathcal{I}_1 | \mathbf{s}(0)) = f(\mathcal{I}_1^s, \mathcal{I}_1^w | \mathbf{s}(0)) = f(\mathcal{I}_1^s) f(\mathcal{I}_1^w | \mathbf{s}(0))$ . From (45), we can see that given  $\mathbf{s}_A(0)$ ,  $\mathcal{I}_1^s$  is independent of  $\mathbf{s}_H(0)$  and  $\mathbf{s}_R(0)$ , where  $\mathcal{H} = \{i, l\}$  and  $\mathcal{R} = \mathcal{V} \setminus (\mathcal{H} \cup \mathcal{A})$ . So  $f(\mathcal{I}_1^s | \mathbf{s}(0)) = f(\mathcal{I}_1^s | \mathbf{s}_A(0), \mathbf{s}_H(0), \mathbf{s}_R(0)) = f(\mathcal{I}_1^s | \mathbf{s}_A(0))$  holds. Therefore, we have

$$\begin{aligned} f(\mathcal{I}_1 | \mathbf{s}(0)) &= f(\mathcal{I}_1^w) f(\mathcal{I}_1^s | \mathbf{s}_A(0)) = f(\mathcal{I}_1^w) f(\mathcal{I}_1^s | \tilde{\mathbf{s}}_A(0)) \\ &= f(\mathcal{I}_1 | \tilde{\mathbf{s}}(0)) \end{aligned} \quad (57)$$

where we used  $\mathbf{s}_A(0) = \tilde{\mathbf{s}}_A(0)$  in the derivation.

To show  $f(\mathcal{I}_2 | \mathcal{I}_1, \mathbf{s}(0)) = f(\mathcal{I}_2 | \mathcal{I}_1, \tilde{\mathbf{s}}(0))$ , it suffices to prove

$$\begin{aligned} &f(\mathcal{I}_2, \mathbf{s}(K+1), \mathbf{w}(K+1) | \mathcal{I}_1, \mathbf{s}(0)) \\ &= f(\mathcal{I}_2, \mathbf{s}(K+1), \mathbf{w}(K+1) | \mathcal{I}_1, \tilde{\mathbf{s}}(0)) \end{aligned} \quad (58)$$

Given  $\mathbf{s}(K+1)$  and  $\mathbf{w}(K+1)$ ,  $\mathcal{I}_2$  is independent of  $\mathcal{I}_1$ , which further leads to

$$\begin{aligned} &f(\mathcal{I}_2, \mathbf{s}(K+1), \mathbf{w}(K+1) | \mathcal{I}_1, \mathbf{s}(0)) \\ &= f(\mathcal{I}_2 | \mathbf{s}(K+1), \mathbf{w}(K+1), \mathcal{I}_1, \mathbf{s}(0)) \\ &\quad \times f(\mathbf{s}(K+1), \mathbf{w}(K+1) | \mathcal{I}_1, \mathbf{s}(0)) \\ &= f(\mathcal{I}_2 | \mathbf{s}(K+1), \mathbf{w}(K+1)) \\ &\quad \times f(\mathbf{s}(K+1), \mathbf{w}(K+1) | \mathcal{I}_1, \mathbf{s}(0)) \\ &= f(\mathcal{I}_2 | \mathbf{s}(K+1), \mathbf{w}(K+1)) \\ &\quad \times f(\mathbf{s}(K+1) | \mathbf{w}(K+1), \mathcal{I}_1, \mathbf{s}(0)) f(\mathbf{w}(K+1) | \mathcal{I}_1, \mathbf{s}(0)) \end{aligned} \quad (59)$$

Further taking into account the facts that 1)  $\mathbf{s}(K+1)$  is conditionally independent of  $\mathbf{w}(K+1)$  and  $\mathcal{I}_1^w$  given  $\mathcal{I}_1^s$  and  $\mathbf{s}(0)$ ; and 2)  $\mathbf{w}(K+1)$  is conditionally independent of  $\mathcal{I}_1^s$  and  $\mathbf{s}(0)$  given  $\mathcal{I}_1^w$ , one can obtain

$$\begin{aligned} &f(\mathcal{I}_2, \mathbf{s}(K+1), \mathbf{w}(K+1) | \mathcal{I}_1, \mathbf{s}(0)) \\ &= f(\mathcal{I}_2 | \mathbf{s}(K+1), \mathbf{w}(K+1)) f(\mathbf{s}(K+1) | \mathcal{I}_1^s, \mathbf{s}(0)) \\ &\quad \times f(\mathbf{w}(K+1) | \mathcal{I}_1^w) \end{aligned} \quad (60)$$

Similarly, one can also obtain

$$\begin{aligned} &f(\mathcal{I}_2, \mathbf{s}(K+1), \mathbf{w}(K+1) | \mathcal{I}_1, \tilde{\mathbf{s}}(0)) \\ &= f(\mathcal{I}_2 | \mathbf{s}(K+1), \mathbf{w}(K+1)) f(\mathbf{s}(K+1) | \mathcal{I}_1^s, \tilde{\mathbf{s}}(0)) \\ &\quad \times f(\mathbf{w}(K+1) | \mathcal{I}_1^w) \end{aligned} \quad (61)$$

Therefore, to show  $f(\mathcal{I}_2 | \mathcal{I}_1, \mathbf{s}(0)) = f(\mathcal{I}_2 | \mathcal{I}_1, \tilde{\mathbf{s}}(0))$ , it suffices to prove  $f(\mathbf{s}(K+1) | \mathcal{I}_1^s, \mathbf{s}(0)) = f(\mathbf{s}(K+1) | \mathcal{I}_1^s, \tilde{\mathbf{s}}(0))$ . Denote  $\mathcal{I}_R^s$  as  $\mathcal{I}_R^s \triangleq \{\Delta \mathbf{s}_R(k) | k = 0, 1, \dots, K\}$ . To prove  $f(\mathbf{s}(K+1) | \mathcal{I}_1^s, \mathbf{s}(0)) = f(\mathbf{s}(K+1) | \mathcal{I}_1^s, \tilde{\mathbf{s}}(0))$ , we only need to prove

$$f(\mathbf{s}(K+1), \mathcal{I}_R^s | \mathcal{I}_1^s, \mathbf{s}(0)) = f(\mathbf{s}(K+1), \mathcal{I}_R^s | \mathcal{I}_1^s, \tilde{\mathbf{s}}(0)) \quad (62)$$

From (46), we can obtain the following facts: 1)  $\mathbf{s}_H(K+1)$  is conditionally independent of  $\mathbf{s}_A(K+1)$  and  $\mathbf{s}_R(K+1)$  given  $\mathcal{I}_R^s$ ,  $\mathcal{I}_1^s$ , and  $\mathbf{s}(0)$ ; and 2)  $\mathbf{s}_A(K+1)$  and  $\mathbf{s}_R(K+1)$

are conditionally independent of  $\mathbf{s}_H(0)$  given  $\mathcal{I}_R^s$ ,  $\mathcal{I}_1^s$ ,  $\mathbf{s}_A(0)$ , and  $\mathbf{s}_R(0)$ . Taking into account these facts, we have

$$\begin{aligned} &f(\mathbf{s}(K+1), \mathcal{I}_R^s | \mathcal{I}_1^s, \mathbf{s}(0)) \\ &= f(\mathbf{s}_H(K+1), \mathbf{s}_A(K+1), \mathbf{s}_R(K+1), \mathcal{I}_R^s | \mathcal{I}_1^s, \mathbf{s}(0)) \\ &= f(\mathbf{s}_H(K+1) | \mathbf{s}_A(K+1), \mathbf{s}_R(K+1), \mathcal{I}_R^s, \mathcal{I}_1^s, \mathbf{s}(0)) \\ &\quad \times f(\mathbf{s}_A(K+1), \mathbf{s}_R(K+1) | \mathcal{I}_R^s, \mathcal{I}_1^s, \mathbf{s}(0)) f(\mathcal{I}_R^s | \mathcal{I}_1^s, \mathbf{s}(0)) \\ &= f(\mathbf{s}_H(K+1) | \mathcal{I}_R^s, \mathcal{I}_1^s, \mathbf{s}(0)) \\ &\quad \times f(\mathbf{s}_A(K+1), \mathbf{s}_R(K+1) | \mathcal{I}_R^s, \mathcal{I}_1^s, \mathbf{s}_A(0), \mathbf{s}_R(0)) f(\mathcal{I}_R^s) \end{aligned} \quad (63)$$

where in the derivation we used the independence between  $\mathcal{I}_R^s$  and  $\{\mathcal{I}_1^s, \mathbf{s}(0)\}$ . Similarly, one can obtain

$$\begin{aligned} &f(\mathbf{s}(K+1), \mathcal{I}_R^s | \mathcal{I}_1^s, \tilde{\mathbf{s}}(0)) \\ &= f(\mathbf{s}_H(K+1) | \mathcal{I}_R^s, \mathcal{I}_1^s, \tilde{\mathbf{s}}(0)) \\ &\quad \times f(\mathbf{s}_A(K+1), \mathbf{s}_R(K+1) | \mathcal{I}_R^s, \mathcal{I}_1^s, \tilde{\mathbf{s}}_A(0), \tilde{\mathbf{s}}_R(0)) f(\mathcal{I}_R^s) \end{aligned} \quad (64)$$

From Lemma 2, we have that if at some time instant  $0 \leq k^* \leq K$ , agent  $i$  has an in-neighbor or out-neighbor  $l$  not belonging to  $\mathcal{A}$ ,  $f(\mathbf{s}_H(K+1) | \mathcal{I}_s^*, \mathbf{s}(0)) = f(\mathbf{s}_H(K+1) | \mathcal{I}_s^*, \tilde{\mathbf{s}}(0))$  holds, where  $\mathcal{I}_s^* = \{\Delta s_{mn}(k) | (m, n) \in \mathcal{E}(k), (m, n) \neq (i, l), (m, n) \neq (l, i), k = 0, 1, \dots, K\} = \{\Delta s_{mn}(k) | (m, n) \in \mathcal{E}_A(k) \cup \mathcal{E}_R(k), k = 0, 1, \dots, K\}$  is the collection of all elements  $\Delta s_{mn}(k)$  in  $\Delta \mathbf{s}_A(k)$  and  $\Delta \mathbf{s}_R(k)$  from iteration 0 to iteration  $K$ . Given that  $\Delta \mathbf{s}_A(k) \in \mathcal{I}_1^s$  and  $\Delta \mathbf{s}_R(k) \in \mathcal{I}_R^s$  hold for  $k = 0, 1, \dots, K$ , we further have  $f(\mathbf{s}_H(K+1) | \mathcal{I}_R^s, \mathcal{I}_1^s, \mathbf{s}(0)) = f(\mathbf{s}_H(K+1) | \mathcal{I}_R^s, \mathcal{I}_1^s, \tilde{\mathbf{s}}(0))$ . Based on  $\mathbf{s}_A(0) = \tilde{\mathbf{s}}_A(0)$  and  $\mathbf{s}_R(0) = \tilde{\mathbf{s}}_R(0)$ , we have  $f(\mathbf{s}(K+1), \mathcal{I}_R^s | \mathcal{I}_1^s, \mathbf{s}(0)) = f(\mathbf{s}(K+1), \mathcal{I}_R^s | \mathcal{I}_1^s, \tilde{\mathbf{s}}(0))$ , implying  $f(\mathcal{I}_2 | \mathcal{I}_1, \mathbf{s}(0)) = f(\mathcal{I}_2 | \mathcal{I}_1, \tilde{\mathbf{s}}(0))$  if at some time instant  $0 \leq k^* \leq K$ , agent  $i$  has an in-neighbor or out-neighbor  $l$  not belonging to  $\mathcal{A}$ .

Therefore, we have  $f(\mathcal{I} | \mathbf{s}(0)) = f(\mathcal{I} | \tilde{\mathbf{s}}(0))$  for any  $\mathbf{s}(0), \tilde{\mathbf{s}}(0) \in [\frac{1}{N^2}, \frac{N-1}{N^2}]^N$  subject to  $s_i(0) + s_l(0) = \tilde{s}_i(0) + \tilde{s}_l(0)$  and  $s_j(0) = \tilde{s}_j(0)$  for  $j \in \mathcal{V} \setminus \{i, l\}$ , meaning that the confidentiality of agent  $i$  can be preserved if at some time instant  $0 \leq k^* \leq K$ , agent  $i$  has an in-neighbor or out-neighbor  $l$  that does not belong to  $\mathcal{A}$ . ■

*Remark 7:* Compared with our previous work [1] which defines privacy as the positivity of probability that adversaries' accessible information set being the same under two different initial states, in this work we significantly improved our confidential results by proving that the probability distributions of information sets are identical under different initial states, meaning that the initial states are perfectly indistinguishable from the viewpoint of adversaries.

*Remark 8:* It is worth noting that although the confidential approach in [30] looks similar to ours (both protocols employ random values in the first several iterations), they are in fact significantly different. More specifically, to guarantee the accuracy of average consensus, not only does the confidential protocol in [30] require pairwise local averaging of exchanged random values in the first several iterations, but it also needs to compensate the errors induced by the random values immediately after the first several iterations, which is equivalent to introducing time-correlated additive random noises to agents' states. To the contrary, our confidential approach exchanges

time-uncorrelated random values for iterations  $k \leq K$ . To ensure consensus accuracy, each agent  $i$  uses a carefully-designed  $\Delta s_{ii}(k)$  to compensate the errors induced by the random values at each iteration  $k \leq K$ . Therefore, the random values used in our approach are space-correlated. As shown in Theorem 2, the space-correlated randomness can make the probability distributions of information sets accessible to adversaries identical under different initial states and hence achieves information-theoretic privacy, which is stronger than the confidentiality achieved using time-correlated noises in [30].

Next we proceed to show that if the conditions in Theorem 2 are not met, then the confidentiality of agent  $i$  may be breached.

**Theorem 3:** Consider a network of  $N$  agents represented by a sequence of time-varying directed graphs  $\{\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))\}$  which satisfy Assumptions 1, 2, 3, 4, and 5. Under the proposed Algorithm 1, the confidentiality of agent  $i \notin \mathcal{A}$  cannot be preserved against  $\mathcal{A}$  if all in-neighbors and out-neighbors of agent  $i$  belong to  $\mathcal{A}$ , i.e.,  $\{\mathcal{N}_i^{in}(k) \cup \mathcal{N}_i^{out}(k) \mid k \geq 0\} \subset \mathcal{A}$ . In fact, when  $\{\mathcal{N}_i^{in}(k) \cup \mathcal{N}_i^{out}(k) \mid k \geq 0\} \subset \mathcal{A}$  is true, the initial value  $x_i^0$  of agent  $i$  can be uniquely determined by honest-but-curious agents in  $\mathcal{A}$ .

*Proof:* From (43) we have

$$\begin{aligned} & \text{frac}(s_i(k+1) - s_i(k)) \\ &= \text{frac}\left(\sum_{n \in \mathcal{N}_i^{in}(k)} \Delta s_{in}(k) - \sum_{m \in \mathcal{N}_i^{out}(k)} \Delta s_{mi}(k)\right) \end{aligned} \quad (65)$$

for  $k \leq K$  and further

$$\begin{aligned} & \text{frac}(s_i(K+1) - s_i(0)) \\ &= \text{frac}\left(\sum_{k=0}^K \left[ \sum_{n \in \mathcal{N}_i^{in}(k)} \Delta s_{in}(k) - \sum_{m \in \mathcal{N}_i^{out}(k)} \Delta s_{mi}(k) \right]\right) \end{aligned} \quad (66)$$

Since  $s_i(0) \in [\frac{1}{N^2}, \frac{N-1}{N^2}]$  holds, one can obtain

$$\begin{aligned} s_i(0) &= \text{frac}\left(s_i(K+1) \right. \\ & \quad \left. - \sum_{k=0}^K \left[ \sum_{n \in \mathcal{N}_i^{in}(k)} \Delta s_{in}(k) - \sum_{m \in \mathcal{N}_i^{out}(k)} \Delta s_{mi}(k) \right]\right) \end{aligned} \quad (67)$$

According to the requirements in Algorithm 1, we have  $\Delta s_{ii}(k) + \sum_{m \in \mathcal{N}_i^{out}(k)} \Delta s_{mi}(k) = s_i(k)$  for  $k \geq K+1$  and  $\Delta w_{ii}(k) + \sum_{m \in \mathcal{N}_i^{out}(k)} \Delta w_{mi}(k) = w_i(k)$  for  $k \geq 0$ . Plugging these relationships into (6) and (7), we can obtain

$$\begin{aligned} s_i(k+1) - s_i(k) &= \\ & \sum_{n \in \mathcal{N}_i^{in}(k)} \Delta s_{in}(k) - \sum_{m \in \mathcal{N}_i^{out}(k)} \Delta s_{mi}(k) \end{aligned} \quad (68)$$

for  $k \geq K+1$  and

$$\begin{aligned} w_i(k+1) - w_i(k) &= \\ & \sum_{n \in \mathcal{N}_i^{in}(k)} \Delta w_{in}(k) - \sum_{m \in \mathcal{N}_i^{out}(k)} \Delta w_{mi}(k) \end{aligned} \quad (69)$$

for  $k \geq 0$ , which further lead to

$$\begin{aligned} s_i(k) - s_i(K+1) &= \\ & \sum_{l=K+1}^{k-1} \left[ \sum_{n \in \mathcal{N}_i^{in}(l)} \Delta s_{in}(l) - \sum_{m \in \mathcal{N}_i^{out}(l)} \Delta s_{mi}(l) \right] \end{aligned} \quad (70)$$

for  $k \geq K+1$  and

$$\begin{aligned} w_i(k) - w_i(0) &= \\ & \sum_{l=0}^{k-1} \left[ \sum_{n \in \mathcal{N}_i^{in}(l)} \Delta w_{in}(l) - \sum_{m \in \mathcal{N}_i^{out}(l)} \Delta w_{mi}(l) \right] \end{aligned} \quad (71)$$

for  $k \geq 0$ .

Under Assumption 5, if  $\{\mathcal{N}_i^{out}(k) \cup \mathcal{N}_i^{in}(k) \mid k \geq 0\} \subset \mathcal{A}$  is true, then all terms on the right-hand side of (70) and (71) are known to the honest-but-curious agents in  $\mathcal{A}$ . Further taking into account  $w_i(0) = 1$ , we have that agents in  $\mathcal{A}$  can uniquely determine  $w_i(k)$  for all  $k$ . Under Assumption 1 and  $\{\mathcal{N}_i^{out}(k) \cup \mathcal{N}_i^{in}(k) \mid k \geq 0\} \subset \mathcal{A}$ , there must exist at least one agent  $j \in \mathcal{A}$  such that  $(j, i) \in \mathcal{E}_\infty$  is true. This is because otherwise graph  $(\mathcal{V}, \mathcal{E}_\infty)$  is not strongly connected, which does not satisfy Assumption 1. According to Assumption 2, agent  $i$  directly communicates with agent  $j \in \mathcal{A}$  at least once in every  $T$  consecutive time instants. So there must exist  $k' \geq K+1$  at which agent  $i$  directly communicates with agent  $j$ , i.e., agent  $i$  sends  $\Delta s_{ji}(k')$  and  $\Delta w_{ji}(k')$  to agent  $j$  at iteration  $k'$ . As  $j \in \mathcal{A}$ , every honest-but-curious agent in  $\mathcal{A}$  has access to  $\Delta s_{ji}(k')$  and  $\Delta w_{ji}(k')$ . So agents in  $\mathcal{A}$  can easily infer  $s_i(k')$  by using the following relationship

$$s_i(k') = \frac{\Delta s_{ji}(k')}{\Delta w_{ji}(k')} w_i(k') = \frac{p_{ji}(k') s_i(k')}{p_{ji}(k') w_i(k')} w_i(k') \quad (72)$$

and then determine the value of  $s_i(K+1)$  using (70). Further making use of (67), agents in  $\mathcal{A}$  can infer the value of  $s_i(0)$ , and then uniquely determine the initial value of agent  $i$  using  $x_i^0 = \frac{b-a}{N-2}(N^2 s_i(0) - 1) + a$ . Therefore, the confidentiality of agent  $i \notin \mathcal{A}$  cannot be preserved against  $\mathcal{A}$  if all in-neighbors and out-neighbors of agent  $i$  belong to  $\mathcal{A}$ . ■

**Remark 9:** It is worth noting that in confidential average consensus, topology requirements such as the ones in Theorem 2 are widely used. In fact, to guarantee both accuracy and confidentiality, [26]–[31], [33]–[35], [39]–[41] all rely on similar topology requirements.

**Remark 10:** Our algorithm can protect the confidentiality of an agent even when all its neighbors interact (at least one does not collude) with adversaries, which is not allowed in [27] and [31].

**Remark 11:** From the above analysis, we know that introducing randomness into interaction dynamics by each agent  $i$  for  $k \leq K$  is key to protect confidentiality against honest-but-curious agents. It is worth noting that compared with the conventional push-sum approach which does not take confidentiality into consideration, the introduced randomness in our approach has no influence on the convergence rate  $\gamma$ . However, the randomness does delay the convergence process and hence leads to a trade-off between confidentiality preservation and convergence time. This is confirmed in our

numerical simulations in Fig. 2, which shows that convergence only initiates after  $k = K + 1$ .

*Remark 12:* If an adversary can obtain side information, then a larger  $K$  protects the confidentiality of more intermediate states  $s_i(k)$  for  $1 \leq k \leq K$ . This is because for  $k \geq K + 1$ ,  $s_i(k)$  can be easily obtained by its out-neighbor  $j$  due to the relationship  $s_i(k) = w_i(k)\Delta s_{ji}(k)/\Delta w_{ji}(k)$  if side information about  $w_i(k)$  is available to the adversary  $j$ . Therefore, although a larger  $K$  leads to more delay in the convergence process, as discussed in Remark 11, it can protect more intermediate states ( $s_i(k)$  for  $1 \leq k \leq K$ ) when an adversary can obtain side information. Of course, if side information is not of concern, a smaller  $K$  is preferable to minimize the delay in the convergence process.

*Remark 13:* Our algorithm can be extended to preserve confidentiality against external eavesdroppers wiretapping all communication links without compromising algorithmic accuracy by patching partially homomorphic encryption. More specifically, using public-key cryptosystems (e.g., Paillier [57], RSA [58], and ElGamal [59]), each agent generates and floods its public key before the consensus iteration starts. Then in decentralized implementation, an agent encrypts its messages to be sent, which can be decrypted by a legitimate recipient without the help of any third party. Note that since public-key cryptosystems can only deal with integers, the final consensus result would be subject to a quantization error. However, as indicated in our previous work [40], the quantization error can be made arbitrarily small in implementation.

#### IV. RESULTS VALIDATION

We conducted numerical simulations to verify the correctness and the effectiveness of our proposed approach.

We first evaluated our proposed Algorithm 1 under a network of  $N = 5$  agents interacting on a time-varying directed graph. More specifically, we used the interaction graph in Fig. 1(a) when  $k$  is even and Fig. 1(b) when  $k$  is odd. It can be verified that this time-varying directed graph satisfies Assumptions 1 and 2. Parameter  $\varepsilon$  was set to 0.05. The initial values  $x_i^0$  for  $i = 1, \dots, N$  were randomly chosen from  $(-50, 50)$ . We used  $e(k)$  to measure the estimation error between the estimate  $\pi_i(k) = s_i(k)/w_i(k)$  and the true average value  $\bar{x}^0 = \sum_{j=1}^N x_j^0/N$  at iteration  $k$ , i.e.,

$$e(k) = \|\pi(k) - \bar{x}^0 \mathbf{1}\| = \left( \sum_{i=1}^N (\pi_i(k) - \bar{x}^0)^2 \right)^{1/2} \quad (73)$$

Three experiments were conducted with parameter  $K$  being 10, 20, and 30, respectively. The evolution of  $e(k)$  is shown in Fig. 2. It can be seen that  $e(k)$  approached 0, meaning that every agent converged to the average value  $\bar{x}^0 = \sum_{j=1}^N x_j^0/N$ . From Fig. 2, we can also see that Algorithm 1 did not start to converge in the first  $K + 1$  iterations due to the introduced randomness, which confirms our analysis in Remark 11.

We also evaluated the influence of parameter  $\varepsilon$  on the convergence rate  $\gamma$ . The interaction graph was the same as above.  $K$  was set to 10. The simulation results are given in Fig. 3 where the mean and variance of  $\gamma$  from 1,000 runs of the algorithm are shown under different values of  $\varepsilon$ . Fig. 3

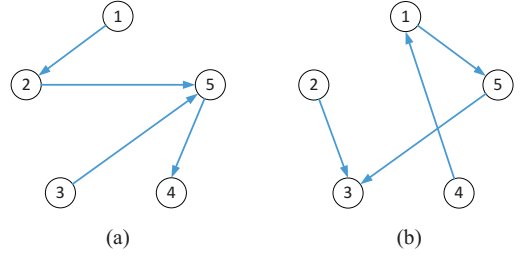


Fig. 1: A time-varying directed graph with 5 agents.

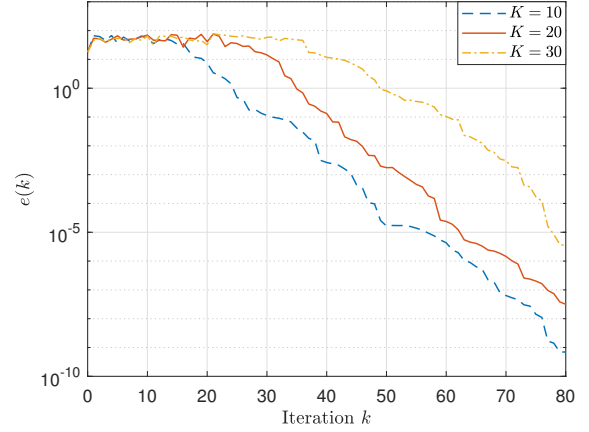


Fig. 2: The evolution of error  $e(k)$  under different  $K$ .

shows that as  $\varepsilon$  increases, the convergence rate  $\gamma$  decreases (i.e., the convergence speed increases), which confirms our analysis in Sec. III-B.

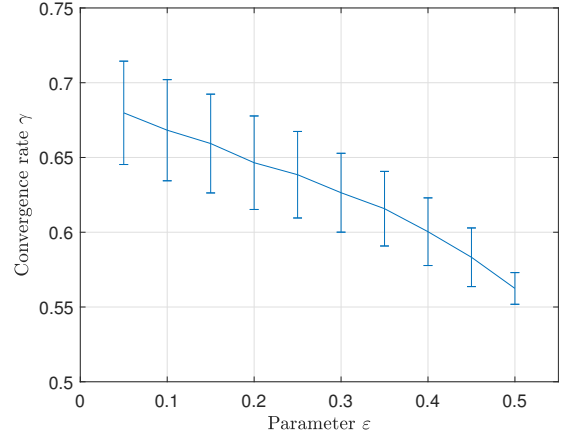


Fig. 3: The influence of  $\varepsilon$  on the convergence rate  $\gamma$ .

We then compared the proposed Algorithm 1 with existing data-obfuscation based approaches, more specifically, the differential-privacy based approach in [14], the decaying-noise approach in [27], and the finite-noise-sequence approach in [31]. Under the same setup as in the previous simulation, we chose the initial values as  $\{10, 15, 20, 25, 30\}$ , which led to an average value 20. We adopted the weight matrix  $\mathbf{W}$

from [14], i.e., the  $ij$ -th entry was  $w_{ij} = 1/(|\mathcal{N}_j^{out}| + 1)$  for  $i \in \mathcal{N}_j^{out} \cup \{j\}$  and  $w_{ij} = 0$  for  $i \notin \mathcal{N}_j^{out} \cup \{j\}$ . As the graph is directed and imbalanced, and does not meet the undirected or balanced assumption in [14], [27], [31], all three approaches failed to achieve average consensus, as shown in the numerical simulation results in Fig. 4, Fig. 5, and Fig. 6, respectively.

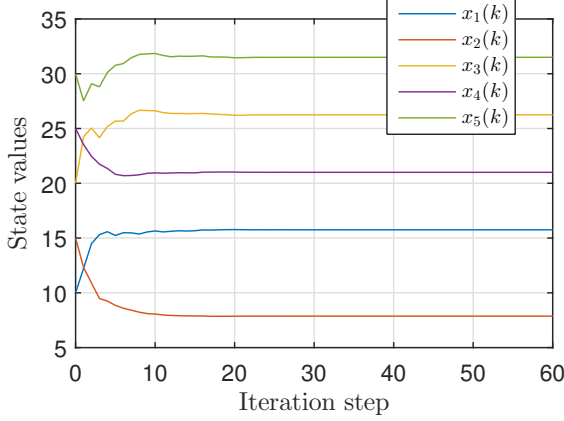


Fig. 4: The evolution of  $x_i(k)$  under the approach in [14].

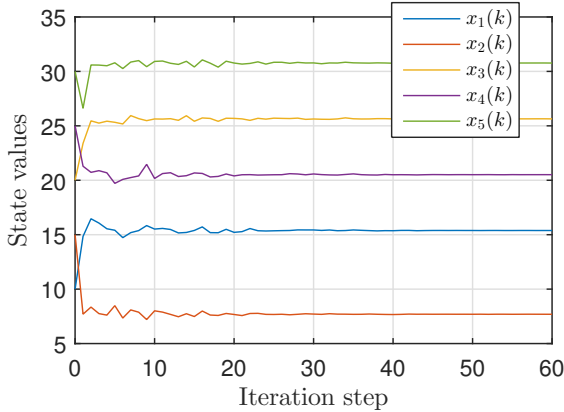


Fig. 5: The evolution of  $x_i(k)$  under the approach in [27].

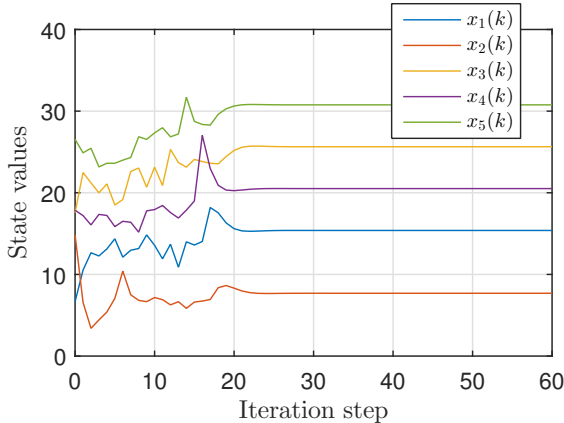


Fig. 6: The evolution of  $x_i(k)$  under the approach in [31].

Finally, we conducted numerical simulations to verify the scalability of our proposed Algorithm 1 using a network of  $N = 1,000$  agents. At every iteration  $k$ , each agent  $i$  was assumed to have three out-neighbors, i.e.,

$$\mathcal{N}_i^{out}(k) = \begin{cases} \{\bar{i} + 1, \bar{i} + 1 + 1, \bar{i} + 2 + 1\} & \text{if } k \text{ is even} \\ \{\bar{i} - 2 + 1, \bar{i} - 3 + 1, \bar{i} - 4 + 1\} & \text{if } k \text{ is odd} \end{cases} \quad (74)$$

where the superscript “ $\bar{\cdot}$ ” represents modulo operation on  $N$ , i.e.,  $\bar{i} \triangleq i \bmod N$ . The initial values  $x_i^0$  for  $i = 1, 2, \dots, N$  were randomly chosen from  $(-50, 50)$ .  $\varepsilon$  and  $K$  were set to 0.05 and 10, respectively. The evolution of estimation error  $e(k) = \|\pi(k) - \bar{x}^0 \mathbf{1}\|$  is shown in Fig. 7. It can be seen that  $e(k)$  converged to 0, implying that our proposed algorithm can guarantee the convergence of all agents to the actual average value even when the network size is large.

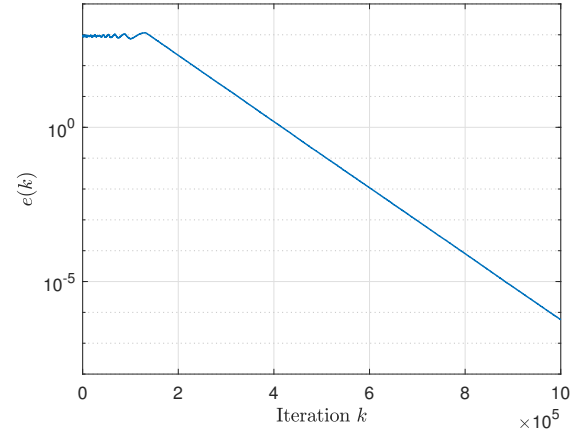


Fig. 7: The evolution of error  $e(k)$  in a network of  $N = 1,000$  agents.

## V. CONCLUSIONS

We proposed a confidential average consensus algorithm for time-varying directed graphs. In distinct difference from existing differential-privacy based approaches which enable confidentiality through compromising the accuracy of obtained consensus value, we leveraged the inherent robustness of average consensus to embed randomness in interaction dynamics, which guarantees confidentiality of participating agents without sacrificing the accuracy of average consensus. Finally, we provided numerical simulation results to confirm the effectiveness and efficiency of our proposed approach.

## REFERENCES

- [1] H. Gao, C. Zhang, M. Ahmad, and Y. Q. Wang, “Privacy-preserving average consensus on directed graphs using push-sum,” in *2018 IEEE Conference on Communications and Network Security (CNS)*, 2018.
- [2] J. E. Boillat, “Load balancing and poisson equation in a graph,” *Concurrency and Computation: Practice and Experience*, vol. 2, no. 4, pp. 289–313, 1990.
- [3] G. Cybenko, “Dynamic load balancing for distributed memory multiprocessors,” *J. Parallel Distrib. Comput.*, vol. 7, no. 2, pp. 279–301, 1989.
- [4] N. A. Lynch, *Distributed algorithms*. San Francisco, CA: Morgan Kaufmann Publishers, 1996.

- [5] D. S. Scherber and H. C. Papadopoulos, "Locally constructed algorithms for distributed computations in ad-hoc networks," in *Proc. Information Processing Sensor Networks (IPSN)*, 2004, pp. 11–19.
- [6] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. Information Processing Sensor Networks (IPSN)*, 2005, pp. 63–70.
- [7] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [8] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, 2005.
- [9] O. L. Mangasarian, "Privacy-preserving linear programming," *Optimization Letters*, vol. 5, no. 1, pp. 165–172, 2011.
- [10] R. Hoenkamp, G. B. Huitema, and A. J. C. de Moor-van Vugt, "The neglected consumer: the case of the smart meter rollout in the netherlands," *Renew. Energy Law Policy Rev.*, pp. 269–282, 2011.
- [11] Y. Lou, L. Yu, S. Wang, and P. Yi, "Privacy preservation in distributed subgradient optimization algorithms," *IEEE Trans. Cybern.*, vol. 48, no. 7, pp. 2154–2165, 2017.
- [12] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, 1984.
- [13] Z. Zhang and M. Y. Chow, "Incremental cost consensus algorithm in a smart grid environment," in *2011 IEEE Power and Energy Society General Meeting*, 2011, pp. 1–6.
- [14] Z. Huang, S. Mitra, and G. Dullerud, "Differentially private iterative synchronous consensus," in *Proceedings of the 2012 ACM workshop on Privacy in the electronic society*, 2012, pp. 81–90.
- [15] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proc. Int. Conf. Distrib. Comput. Netw.*, 2015, pp. 4:1–4:10.
- [16] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private average consensus with optimal noise selection," *IFAC-PapersOnLine*, vol. 48, no. 22, pp. 203–208, 2015.
- [17] —, "Differentially private average consensus: obstructions, trade-offs, and optimal algorithm design," *Automatica*, vol. 81, pp. 221–231, 2017.
- [18] V. Katewa, F. Pasqualetti, and V. Gupta, "On privacy vs. cooperation in multi-agent systems," *Int. J. Control*, vol. 91, no. 7, pp. 1693–1707, 2018.
- [19] L. Gao, S. Deng, W. Ren, and C. Hu, "Differentially private consensus with quantized communication," *IEEE Trans. Cybern.*, 2019.
- [20] D. Ye, T. Zhu, W. Zhou, and S. Y. Philip, "Differentially private malicious agent avoidance in multiagent advising learning," *IEEE Trans. Cybern.*, 2019.
- [21] M. Kefayati, M. S. Talebi, B. H. Khalaj, and H. R. Rabiee, "Secure consensus averaging in sensor networks using random offsets," in *Proc. IEEE Int. Conf. Telecommun. Malaysia Int. Conf. Commun.*, 2007, pp. 556–560.
- [22] X. Wang, J. He, P. Cheng, and J. Chen, "Privacy preserving average consensus with different privacy guarantee," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 5189–5194.
- [23] Y. Wang, Z. Huang, S. Mitra, and G. E. Dullerud, "Differential privacy in linear distributed control systems: Entropy minimizing mechanisms and performance tradeoffs," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 1, pp. 118–130, 2017.
- [24] J. He, L. Cai, P. Cheng, J. Pan, and L. Shi, "Distributed privacy-preserving data aggregation against dishonest nodes in network systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1462–1470, 2018.
- [25] —, "Consensus-based data-privacy preserving data aggregation," *IEEE Transactions on Automatic Control*, vol. 64, no. 12, pp. 5222–5229, 2019.
- [26] J. He, L. Cai, C. Zhao, P. Cheng, and X. Guan, "Privacy-preserving average consensus: privacy analysis and algorithm design," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 1, pp. 127–138, 2018.
- [27] Y. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 753–765, 2017.
- [28] T. Charalambous, N. E. Manitaras, and C. N. Hadjicostis, "Privacy-preserving average consensus over digraphs in the presence of time delays," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2019, pp. 238–245.
- [29] N. Gupta, J. Kat, and N. Chopra, "Statistical privacy in distributed average consensus on bounded real inputs," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 1836–1841.
- [30] A. B. Pilet, D. Frey, and F. Taiani, "Robust privacy-preserving gossip averaging," in *International Symposium on Stabilizing, Safety, and Security of Distributed Systems*. Springer, 2019, pp. 38–52.
- [31] N. E. Manitaras and C. N. Hadjicostis, "Privacy-preserving asymptotic average consensus," in *2013 European Control Conference*, 2013, pp. 760–765.
- [32] X. Duan, J. He, P. Cheng, Y. Mo, and J. Chen, "Privacy preserving maximum consensus," in *2015 IEEE 54th Annual Conference on Decision and Control (CDC)*, 2015, pp. 4517–4522.
- [33] C. Altafini, "A dynamical approach to privacy preserving average consensus," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 4501–4506.
- [34] S. Pequito, S. Kar, S. Sundaram, and A. P. Aguiar, "Design of communication networks for distributed computation with privacy guarantees," in *2014 IEEE 53rd Annual Conference on Decision and Control (CDC)*, 2014, pp. 1370–1376.
- [35] I. D. Ridgley, R. A. Freeman, and K. M. Lynch, "Simple, private, and accurate distributed averaging," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2019, pp. 446–452.
- [36] A. Alaeddini, K. Morgansen, and M. Mesbahi, "Adaptive communication networks with privacy guarantees," in *Proceedings of 2017 American Control Conference*, 2017, pp. 4460–4465.
- [37] M. Kishida, "Encrypted average consensus with quantized control law," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 5850–5856.
- [38] C. N. Hadjicostis and A. D. Dominguez-Garcia, "Privacy-preserving distributed averaging via homomorphically encrypted ratio consensus," *IEEE Trans. Autom. Control*, 2020.
- [39] W. Fang, M. Zamani, and Z. Chen, "Secure and privacy preserving consensus for second-order systems based on paillier encryption," *arXiv preprint arXiv:1805.01065*, 2018.
- [40] M. Ruan, H. Gao, and Y. Wang, "Secure and privacy-preserving consensus," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4035–4049, 2019.
- [41] Y. Wang, "Privacy-preserving average consensus via state decomposition," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4711–4716, 2019.
- [42] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2014.
- [43] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.
- [44] J. Zhu, C. Xu, J. Guan, and D. O. Wu, "Differentially private distributed online algorithms over time-varying directed networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 4–17, 2018.
- [45] A. Scott, "Tactical data diodes in industrial automation and control systems," *SANS Institute InfoSec Reading Room*, 2015.
- [46] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proc. 44th IEEE Symp. Found. Comput. Sci.*, 2003, pp. 482–491.
- [47] F. Bénézit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli, "Weighted gossip: Distributed averaging using non-doubly stochastic matrices," in *Proc. IEEE Int. Symp. Inf. Theory*, 2010, pp. 1753–1757.
- [48] E. Seneta, *Non-negative Matrices and Markov Chains*. Springer, 1973.
- [49] J. A. Fill, "Eigenvalue bounds on convergence to stationarity for non-reversible markov chains, with an application to the exclusion process," *Ann. Appl. Probab.*, pp. 62–87, 1991.
- [50] C. N. Hadjicostis, A. D. Domínguez-García, and T. Charalambous, "Distributed averaging and balancing in network systems: with applications to coordination and control," *Foundations and Trends® in Systems and Control*, vol. 5, no. 2-3, pp. 99–292, 2018.
- [51] K. Liu, H. Kargupta, and J. Ryan, "Random projection-based multiplicative data perturbation for privacy preserving distributed data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp. 92–106, 2006.
- [52] S. Han, W. K. Ng, L. Wan, and V. C. S. Lee, "Privacy-preserving gradient-descent methods," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 6, pp. 884–899, 2010.
- [53] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 222–233, 2014.
- [54] A. Nedić, A. Olshevsky, and W. Shi, "A geometrically convergent method for distributed optimization over time-varying graphs," in *Proc. IEEE 55th Conf. Decis. Control*, 2016, pp. 1023–1029.
- [55] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

- [56] A. Nedić, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, 2010.
- [57] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*. Springer, 1999, pp. 223–238.
- [58] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
- [59] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469–472, 1985.



**Huan Gao** was born in Shandong, China. He received the B.S. degree in automation and the M.Sc. degree in control theory and control engineering from Northwestern Polytechnical University, Xi'an, Shaanxi, China, in 2011 and 2015, respectively, and the Ph.D. degree in electrical engineering from Clemson University, Clemson, SC, USA, in 2020. He is currently an Associate Professor with the School of Automation, Northwestern Polytechnical University. His research interests include decentralized optimization, cooperative control, and privacy

preservation in distributed systems.



**Yongqiang Wang** was born in Shandong, China. He received the B.S. degree in electrical engineering and automation, the B.S. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, Shaanxi, China, in 2004, and the M.Sc. and Ph.D. degrees in control science and engineering from Tsinghua University, Beijing, China, in 2009. From 2007 to 2008, he was with the University of Duisburg-Essen, Germany, as a Visiting Student. He was a Project Scientist with the University of California at Santa Barbara, Santa Barbara. He is

currently an Associate Professor with the Department of Electrical and Computer Engineering, Clemson University. His current research interests include distributed control, optimization, and learning, with emphasis on privacy protection. He currently serves as an associate editor for *IEEE Transactions on Control of Network systems* and *IEEE Transactions on Automatic Control*.