Detailed Derivations of "A Privacy-Preserving Finite-Time Push-Sum based Gradient Method for Distributed Optimization over Digraphs"

Xiaomeng Chen¹, Wei Jiang², Themistoklis Charalambous³, Senior Member, IEEE, and Ling Shi¹, Fellow, IEEE

Abstract-This paper addresses the problem of distributed optimization, where a network of agents represented as a directed graph (digraph) aims to collaboratively minimize the sum of their individual cost functions. Existing approaches for distributed optimization over digraphs, such as Push-Pull, require agents to exchange explicit state values with their neighbors in order to reach an optimal solution. However, this can result in the disclosure of sensitive and private information. To overcome this issue, we propose a state-decomposition-based privacy-preserving finite-time push-sum (PrFTPS) algorithm without any global information, such as network size or graph diameter. Then, based on PrFTPS, we design a gradient descent algorithm (PrFTPS-GD) to solve the distributed optimization problem. It is proved that under PrFTPS-GD, the privacy of each agent is preserved and the linear convergence rate related to the optimization iteration number is achieved. Finally, numerical simulations are provided to illustrate the effectiveness of the proposed approach. Index Terms-Distributed optimization, privacy-preserving,

finite-time consensus, directed graph.

I. INTRODUCTION

In this paper, we consider an optimization problem in a multi-agent system of n agents. Each agent i has a private cost function f_i , which is known to itself only. All the agents aim to collaboratively solve the following optimization problem

$$\min_{x \in \mathbb{R}^p} F(x) := \sum_{i=1}^n f_i(x),$$
(1)

where x is the global decision variable. The agents are connected through a communication graph and can only transmit messages to their neighbors. By local computation and communication, each agent seeks a solution that minimizes the sum of all the local objective functions. Such a distributed paradigm facilitates breaking large-scale problems into sequences of smaller ones. That is why it has been widely adopted in several applications, such as power grids [1], sensor networks [2] and vehicular networks [3].

To solve problem (1), decentralized gradient descent (DGD) is the most commonly used algorithm, requiring diminishing stepsizes to ensure optimality [4]. To overcome the challenge of slow convergence caused by diminishing stepsizes, Xu et

²W. Jiang e-mail: wjiang.lab@gmail.com

al. [5] adopted the dynamic average consensus [6] to propose a gradient tracking (GT) method with a constant stepsize. Recently, Xin et al. [7] and Pu et al. [8] devised a modified GT algorithm called AB/Push-Pull algorithms for distributed optimization, which can be applied to a general digraph. A comprehensive survey on distributed optimization algorithms is provided by Yang et al. [9].

The aforementioned distributed algorithms share state values in each iteration, which can compromise the privacy of agents if they have private information. By hacking into communication links, an adversary could potentially access to transmitted messages among agents and potentially gather private information using an inferring algorithm. Mandal [10] presented theoretical analysis of privacy disclosure in distributed optimization, where the parameters of cost functions and generation power can be correctly inferred by an adversary. As the number of privacy leakage events is increasing, there is an urgent need to preserve privacy of each agent in distributed systems.

Recently, many results have been reported on the topic of privacy-preserving distributed optimization. One commonly used approach is differential privacy (DP) [11] due to its rigorous mathematical framework, proven privacy preservation properties and ease of implementation [12]. However, DP-based approaches face a fundamental trade-off between privacy and accuracy, which may result in suboptimal solutions [13]. To address this challenge, Lu et al. [14] combined distributed optimization methods with partially homomorphic encryption. Nonetheless, this approach has limitations due to high computation complexity and accurate results, Wang [15] proposed a privacy-preserving average consensus using a state decomposition mechanism that divides the state of a node into two sub-states.

It is worth noting that none of the aforementioned approaches is suitable for agents over digraphs. To preserve privacy of nodes interacting on a digraph, Charalambous et al. [16] proposed an offset-adding privacy-preserving push-sum, and Gao et al. [17] protected privacy by adding randomness on edge weights, both of which are only effective against honest-but-curious nodes (see Definition 3). To improve resilience to external eavesdroppers (see Definition 4), Chen et al. [18] extended the state decomposition mechanism to digraphs and introduced an uncertainty-based privacy notion. In terms of privacy-preserving distributed optimization over digraphs, Mao et al. [19] designed a privacy-preserving algorithm based on the push-gradient method with a decaying stepsize, which

¹X. Chen and L. Shi are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (email: xchendu@connect.ust.hk, eesling@ust.hk).

³T. Charalambous is with the Department of Electrical and Computer Engineering, School of Engineering, University of Cyprus, Nicosia, Cyprus. He is also a Visiting Professor at Aalto University, Espoo, Finland, and FinEst Centre for Smart Cities, Tallinn, Estonia. (e-mail: charalambous.themistoklis@ucy.ac.cy).

lacked a formal privacy notion. Wang and Nedić [20] designed a DP-oriented gradient tracking based algorithm (DPGT) that can ensure both differential privacy and optimality. However, it adopted a diminishing stepsize to ensure convergence, resulting in a slow convergence rate. To speed up the convergence, Chen et al. [21] proposed a state-decompositionbased push-pull (SD-Push-Pull) algorithm, which guarantees both linear convergence and differential privacy for digraphs. Nevertheless, SD-Push-Pull only converges to a suboptimal value.

Inspired by recent results that privacy can be enabled in consensus over digraphs by state decomposition [18] and that finite-time push-sum can be used in distributed optimization to deliver the optimal solution [22], this paper presents a novel PrFTPS algorithm that accurately computes the average value for digraphs in a finite time, as opposed to the asymptotic average consensus achieved in [18]. Then, combined with gradient decent, PrFTPS-GD is proposed to solve problem (1) allowing each node in a digraph to achieve optimal value linearly while preserving its privacy. The main contributions of this paper are summarized as follows:

- We propose PrFTPS (Algorithm 1) based GD algorithm (Algorithm 2) to solve problem (1) over digraphs. Moreover, we show that Algorithm 1 can compute the exact average value in finite time and Algorithm 2 guarantees the linear convergence to the optimal value of problem (1) (Theorem 1).
- 2) We analyze the privacy-preserving performance of PrFTPS-GD against honest-but-curious nodes and eavesdroppers (**Theorem 2**). Specifically, we adopt the uncertainty-based privacy notion [18] and show that the adversary has infinite uncertainty about agents' private information under certain topological conditions.
- 3) PrFTPS-GD performance is evaluated via simulations and compared with other state-of-the-art privacypreserving approaches (e.g., [20], [21]) over digraphs. It is shown that our approach apart from adopting an easily tuned constant stepsize (unlike the diminishing stepsize in [20]), it computes the optimal solution instead of the suboptimal one in [21].

Notations: In this paper, \mathbb{R}^n and $\mathbb{R}^{n \times p}$ represent the set of n dimensional vectors and $n \times p$ dimensional matrices. \mathbb{Z}_{++} denotes the set of positive integers. $\mathbf{1}_n \in \mathbb{R}^n$, $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ and $\mathbf{0}_n \in \mathbb{R}^{n \times n}$ represent the vector of ones, the identity matrix and the zero matrix, respectively. For an arbitrary vector \boldsymbol{x} , we denote its *i*th element by \boldsymbol{x}_i . For an arbitrary matrix \boldsymbol{M} , we denote its element in the *i*th row and *j*th column by $[\boldsymbol{M}]_{ij}$. \otimes denotes the Kronecker product. The spectral radius of matrix \mathbf{A} is denoted by $\rho(\mathbf{A})$. Matrix \mathbf{A} is called row-stochastic if the sum of each row equals to 1 and the entries of \mathbf{A} are non-negative. Similarly, matrix \mathbf{A} is called column-stochastic if the sum of each column equals to 1 and the entries of \mathbf{A} are non-negative.

II. PRELIMINARIES AND PROBLEM STATEMENT

A. Network Model

We consider a digraph $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$ with n nodes, where the set of nodes and edges are $\mathcal{V} = \{1, \ldots, n\}$ and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$,

respectively. A communication link from node i to node j is denoted by $\varepsilon_{ji} = (j,i) \in \mathcal{E}$, indicating that node i can send messages to node j. The nodes who can send messages to node i are denoted as in-neighbours of node i and the set of these nodes is denoted as $N_i^- = \{j \in \mathcal{V} \mid \varepsilon_{ij} \in \mathcal{E}\}$. Similarly, the nodes who can receive messages from node i are denoted as out-neighbours of node i and the set of these nodes is denoted as $N_i^+ = \{j \in \mathcal{V} \mid \varepsilon_{ji} \in \mathcal{E}\}$. The cardinality of N_j^+ , is called the out-degree of node j and is denoted as $\mathcal{D}_j^+ = |N_j^+|$. A digraph is called strongly connected if there exists at least one directed path from any node i to any node j with $i \neq j$.

B. Push-Sum Algorithm

The push-sum algorithm, introduced originally in [23], aims at achieving average consensus for each node communicating over a digraph which satisfies the following assumption.

Assumption 1: The digraph G is assumed to be strongly connected.

Consider a network of n nodes, where each node has a private initial state, termed as $x_i(0)$. The push-sum algorithm introduces *two auxiliary varaibles*, $x_{i,1}(k)$ and $x_{i,2}(k)$, and assumes the out-degree is known for each node. The details are as follows: for each node i,

$$x_{i,l}(k+1) = \sum_{j \in N_i^- \cup \{i\}} p_{ij}(k) x_{j,l}(k), \qquad k \ge 0, l = 1, 2,$$

where $p_{ij}(k) = 1/(1 + \mathcal{D}_j^+), \forall i \in N_j^+ \cup \{j\}$ and $x_{i,1}(0) = x_i(0), x_{i,2}(0) = 1$ for $i \in \mathcal{V}$.

Proposition 1. [23] If a digraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with *n* nodes satisfies Assumption 1, then the ratio $r_i(k) := x_{i,1}(k+1)/x_{i,2}(k+1)$ asymptotically converges to the average of the initial values, *i.e.*, we have

$$\lim_{k \to \infty} r_i(k) = \frac{\sum_{i \in \mathcal{V}} x_i(0)}{n}, \forall i \in \mathcal{V}.$$

C. Information Set and Privacy Inferring Model

Before defining privacy, we first introduce the privacy inferring model. The adversary set \mathcal{A} is assumed to obtain some online data by eavesdropping on some edges $\mathcal{E}_r \subseteq \mathcal{E}$ and nodes $\mathcal{V}_a \subseteq \mathcal{V}$. The information set accessible to \mathcal{A} at time k is denoted as $\mathcal{I}_{\mathcal{A}}(k)$, which contains all transmitted information accessible to \mathcal{A} .

Then all the information accessible to \mathcal{A} at time iteration K is denoted as $\mathcal{I}_{\mathcal{A}}(0:K) \triangleq \{\mathcal{I}_{\mathcal{A}}(0), \mathcal{I}_{\mathcal{A}}(1), \dots, \mathcal{I}_{\mathcal{A}}(K)\}.$

With the above model, we adopt an uncertainty-based notion of privacy, which is proposed in [18]. Denote the private information of node *i* as $x_{p,i} \in \mathbb{R}^p$ and define a set $\Delta \mathcal{I}_{\mathcal{A}}(x_{p,i})$ as

 $\Delta \mathcal{I}_{\mathcal{A}}(x_{p,i}) = \{ \bar{x}_{p,i} \mid \text{the adversary's information set} = \mathcal{I}_{\mathcal{A}}(0:K) \},\$

which contains all possible states that can correspond to $x_{p,i}$ when the information set accessible to \mathcal{A} is $\mathcal{I}_{\mathcal{A}}(0:K)$.

The diameter of $\Delta \mathcal{I}_{\mathcal{A}}(x_{p,i})$ is defined as

$$\operatorname{Diam}\{\Delta \mathcal{I}_{\mathcal{A}}(x_{p,i})\} = \sup_{\bar{x}_{p,i}, \bar{x}'_{p,i} \in \Delta \mathcal{I}_{\mathcal{A}}(x_{p,i})} |\bar{x}_{p,i} - \bar{x}'_{p,i}|,$$

where $\bar{x}_{p,i}$ and $\bar{x}'_{p,i}$ are two different states that belong to set $\Delta \mathcal{I}_{\mathcal{A}}(x_{p,i})$.

Definition 1: The privacy of $x_{p,i}$ is preserved against \mathcal{A} if $\text{Diam}\{\Delta \mathcal{I}_{\mathcal{A}}(x_{p,i})\} = \infty$.

In this paper, we consider distributed optimization problems, where local objective gradients usually carry sensitive information. For example, in distributed-optimization-based localization and rendezvous, directly exchanging the gradient of an agent leads to disclosing its position [13]. Recent work shows that gradients are directly calculated from and embed sensitive information of training learning data [24]. Hence, the private information is the *gradient of each agent at all time iteration*. Then, we define the privacy preservation of each agent as follow.

Definition 2: For a network of n agents in distributed optimization, the privacy of agent j is preserved against \mathcal{A} if the privacy of its gradient value $\nabla f_j(x_j)$ evaluated at any point x_j is preserved.

We consider two types of adversaries, defined as follows.

Definition 3: An honest-but-curious adversary is a node or a group of nodes which knows the network topology and follows the system's protocol, attempting to infer the private information of other nodes.

Definition 4: An eavesdropper is an external adversary who has the knowledge of network topology, and is able to eavesdrop on a portion of coupling weights and transmitted data.

III. MAIN RESULTS

In this section, we first propose a privacy-preserving finitetime push-sum algorithm via a state decomposition mechanism. Then, we adopt the proposed privacy-preserving approach to address problem (1) based on a gradient descent approach.

A. Privacy-Preserving Finite-Time Push-Sum Algorithm

The main idea of our privacy-preserving approach is a state decomposition mechanism.

Decomposition Mechanism: Let each node decompose its state $x_{i,l}(k)$ into two substates $x_{i,l}^{\alpha}(k)$ and $x_{i,l}^{\beta}(k)$, l = 1, 2. The initial values $x_{i,l}^{\alpha}(0)$ and $x_{i,l}^{\beta}(0)$ can be randomly chosen from the set of all real numbers under the following constraint

$$x_{i,1}^{\beta}(0) + x_{i,1}^{\alpha}(0) = 2x_i(0), x_{i,2}^{\alpha}(0) = 0, x_{i,2}^{\beta}(0) = 2, \forall i \in \mathcal{V},$$

where $x_i(0)$ denotes the private initial state of node *i*.

Under the state decomposition mechanism, the overall dynamics become

$$\begin{cases} x_{i,l}^{\alpha}(k+1) = \sum_{j \in N_i^- \cup \{i\}} p_{ij}(k) x_{j,l}^{\alpha}(k) + a_i^{\alpha,\beta}(k) x_{i,l}^{\beta}(k), \\ x_{i,l}^{\beta}(k+1) = a_i^{\beta,\alpha}(k) x_{i,l}^{\alpha}(k) + a_i^{\beta,\beta}(k) x_{i,l}^{\beta}(k), \end{cases}$$
(2)

with $i \in \mathcal{V}$, l = 1, 2. In this decomposition scheme, the substate $x_{i,l}^{\alpha}(k)$ is exchanged with other nodes while $x_{i,l}^{\beta}(k)$ is never shared with other nodes. The coupling weights between the two substates $x_{i,l}^{\alpha}(k)$ and $x_{i,l}^{\beta}(k)$ are asymmetric and denoted as $a_i^{\alpha,\beta}(k)$ and $a_i^{\beta,\alpha}(k)$. The update weights for substate $x_{i,l}^{\beta}(k)$ is denoted as $a_i^{\beta,\beta}(k)$. The outgoing link weight from agent *i* to agent *j* is denoted as $p_{ji}(k)$. These are design parameters and will be designed in the following weight mechanism (Section III-A.1).

We next introduce details of the weight mechanism to enable algorithm convergence and privacy preservation.

1) Weight mechanism: For $k = 0, \forall i \in \mathcal{V}$, we set $a_i^{\beta,\beta}(0) = 0, a_i^{\alpha,\beta}(0) = 1$ and $p_{ji}(0) = 0, \forall j \notin N_i^+$. Also, we allow $p_{ji}(0), \forall j \in N_i^+ \cup \{i\}$ and $a_i^{\beta,\alpha}(0)$ to be arbitrarily chosen from the set of all real numbers under the constraint

$$\sum_{j=1}^{n} p_{ji}(0) + a_i^{\beta,\alpha}(0) = 1.$$

For $k \ge 1$, we let $p_{ji}(k) = 1(2 + \mathcal{D}_i^+)$ for $j \in N_i^+ \cup \{i\}$ and $p_{ji}(k) = 0$, otherwise. Also,

$$a_i^{\beta,\alpha}(k) = \frac{1}{2 + \mathcal{D}_i^+}, \quad a_i^{\beta,\beta}(k) = a_i^{\alpha,\beta}(k) = \frac{1}{2}$$

Remark 1: Under the above weight mechanism, the statedecomposition-based push-sum (2) still preserves the property of conventional push-sum. Rigorous theoretical analysis will be provided in Section III-C.

To obtain the exact average value in finite time, we use the minimal polynomial associated with iteration (2), in conjunction with the final value theorem [25], [26]. Next, we provide definitions on minimal polynomials.

Definition 5: (Minimal Polynomial of a Matrix.) The minimal polynomial of matrix P, denoted by

$$Q(t) = t^{D+1} + \sum_{i=0}^{D} \alpha_i t^i,$$

is the monic polynomial of minimum degree D + 1 that satisfies $Q(P) = 0_n$ and α_i is the polynomial coefficient.

Definition 6: (Minimal Polynomial of a Matrix Pair.) The minimal polynomial associated with $[P, e_i^{\top}]$, denoted by

$$Q_j(t) = t^{D_j+1} + \sum_{i=0}^{D_j} \alpha_{j,i} t^i = 0, \alpha_{j,i} \in \mathbb{R},$$

is the monic polynomial of minimum degree $D_j + 1$ that satisfies $e_j^{\top}Q_j(P) = 0$.

In what follows, we will show how to use the coefficients of the minimal polynomial to obtain the final value in finite time. By using the iteration in (2), we have

$$\sum_{i=0}^{D_j+1} \alpha_{j,i} x_{j,1}^{\alpha}(k+i) = 0, \forall k \in \mathbb{Z}_{++},$$

where $\alpha_{j,D_j+1} = 1$. Thus, the minimal polynomial of a matrix is unique due to the monic property. We denote the \mathcal{Z} -transform of $x_{j,1}(k)$ as $X_{j,1} = \mathcal{Z}(x_{j,1}(k))$. By the time-shift property of the \mathcal{Z} -transform, it is easy to obtain that

$$X_{j,1}(z) = \frac{\sum_{i=1}^{D_j+1} \alpha_{j,i} \sum_{l=1}^{i} x_{j,1}^{\alpha}(l) z^{i-l}}{Q_j(z)}$$

2

Since the communication topology of the networked system is strongly connected, the minimal polynomial $Q_j(z)$ does not have any unstable poles apart from one. Hence, we can define polynomial

$$p_j(z) \triangleq \frac{Q_j(z)}{z-1} \triangleq \sum_{i=0}^{D_j} \beta_i^{(j)} z^i.$$

By the final value theorem [25] and [26], the final state values of (2) are computed as

$$\phi_{x_l}^{\alpha}(j) = \lim_{k \to \infty} x_{j,l}^{\alpha}(k) = \lim_{z \to 1} (z-1) X_{j,l}^{\alpha}(z) = \frac{(x_{l,D_j}^{\alpha})^\top \beta_j}{\mathbf{1}^\top \beta_j}$$

where

 $\Gamma \int (x^{\alpha})^{\top} \geq \Delta$

$$(x_{l,D_j}^{\alpha})^{\top} = (x_{j,l}^{\alpha}(1), x_{j,l}^{\alpha}(1), \dots, x_{j,l}^{\alpha}(D_j+1)), l = 1, 2,$$

and β_j is the coefficient vector of the polynomial $p_j(z)$.

Denote the following vectors of 2k + 1 successive discretetime values for the two iterations $x_{j,l}^{\alpha}(k), l = 1, 2$ at node j as

$$(x_{l,2k}^{\alpha})^{\top} = (x_{j,l}^{\alpha}(1), x_{j,l}^{\alpha}(1), \dots, x_{j,l}^{\alpha}(2k+1)), l = 1, 2.$$

Moreover, define the associated Hankel matrix and the difference vectors between successive values for l = 1, 2 as

$$\begin{bmatrix} x_{j,l}^{\alpha}(1) & x_{j,l}^{\alpha}(2) & \cdots & x_{j,l}^{\alpha}(k+1) \\ x_{j,l}^{\alpha}(2) & x_{j,l}^{\alpha}(3) & \cdots & x_{j,l}^{\alpha}(k+2) \\ \vdots & \vdots & \ddots & \vdots \\ x_{j,l}^{\alpha}(k+1) & x_{j,l}^{\alpha}(k+2) & \cdots & x_{j,lt}^{\alpha}(2k+1) \end{bmatrix}$$
$$(\bar{x}_{l,2k}^{\alpha})^{\top} \triangleq (x_{j,l}^{\alpha}(2) - x_{j,l}^{\alpha}(1), \dots, x_{j,l}^{\alpha}(2k+2) - x_{j,l}^{\alpha}(2k+1)).$$

It is shown in [26] that for arbitrary initial conditions $x_{j,1}^{\alpha}(1)$ and $x_{j,2}^{\alpha}(1)$, β_{j} can be computed as the kernel of the first defective Hankel matrices $\Gamma\{(\bar{x}_{1,2k}^{\alpha})^{\top}\}$ and $\Gamma\{(\bar{x}_{2,2k}^{\alpha})^{\top}\}$, except a set of initial conditions with Lebesgue measure zero.

From the above analysis, we know β_j and D_j can be different for node *j*. Thus, in existing works [22], [25], all nodes are assumed to know the upper bound of the network size. To relax this assumption, Charalambous and Hadjicostis [27] proposed a distributed termination mechanism, allowing all nodes to agree when to terminate their iterations, given they have all computed the average. The procedure is as follows:

- Once iterations (2) are initiated, each node j also initiates two counters c_j , $c_j(0) = 0$, and r_j , $r_j(0) = 0$. Counter c_j increments by one at every time step, i.e., $c_j(k+1) =$ $c_j(k) + 1$. The way counter r_j updates is described next.
- Alongside iterations (2) a max-consensus algorithm is initiated as well, given by

$$\theta_j(k+1) = \max_{v_i \in \mathcal{N}_j \cup \{v_j\}} \left\{ \max\{\theta_i(k), c_i(k)\} \right\}, \quad (3)$$

with $\theta_j(0) = 0$. Then, r_j is updated as follows:

$$r_j(k+1) = \begin{cases} 0, & \text{if } \theta_j(k+1) \neq \theta_j(k), \\ r_j(k) + 1, & \text{otherwise.} \end{cases}$$
(4)

• Once the Hankel matrices $\Gamma\{(\bar{x}_{1,D_j}^{\alpha})^{\top}\}$ and $\Gamma\{(\bar{x}_{2,D_j}^{\alpha})^{\top}\}$ lose rank, node j saves the count of the counter c_j at that time step, denoted by k_j^o , as c_j^o , i.e., $c_j^o \triangleq c_j[k_j^o]$, and it stops incrementing the counter, i.e., $\forall k' \ge k_j^o, c[k'] =$ $c_j[k_j^o] = c_j^o$. Note that $c_j^o = 2(D_j + 1) + 1$. • Node j can terminate iterations (2) when r_j reaches c_j^o .

Therefore, based on the distributed termination mechanism [27], [28], we design a privacy-preserving finite-time pushsum algorithm (PrFTPS) as presented in Algorithm 1, which guarantees the minimum number of iteration steps to obtain the exact average without any global information.

Algorithm 1 A Privacy-Preserving Finte-Time Push-Sum Algorithm (PrFTPS)

- 1: **Input:** Initial state $x_j(0)$, step t, graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$.
- 2: **if** t = 0 **then**
- 3: Run the privacy-preserving iteration (2) and the maxconsensus algorithm (3), store the vectors $(\bar{x}_{1,D_j}^{\alpha})^{\top}$, $(\bar{x}_{2,D_j}^{\alpha})^{\top}$, increment the value of the counter $c_j(k)$ and find the value of the counter $r_j(k)$ via (4).
- 4: Increase the dimension k of the Hankel matrices $\Gamma\{(\bar{x}_{1,D_j}^{\alpha})^{\top}\}$ and $\Gamma\{(\bar{x}_{2,D_j}^{\alpha})^{\top}\}$ until k_j^o at which they lose rank. Once this happens, store the kernel β_j of the first defective matrix and the value $c_j^o = 2(D_j + 1) + 1$.
- 5: Continue iteration (2) until iteration $k_{j,t}$ where $r_j(k_{j,t}) = c_j^o$ and store

$$D_{\max} = \frac{k_{j,t} - 2D_j - 2}{2} - 1$$

6: **else**

- 7: Run the privacy-preserving algorithm (2) for $k_{\text{max}} = D_{\text{max}} + 2$ steps with the same β_j .
- 8: **end if**
- 9: Compute the average value as $\hat{x}_j^{ave} = \frac{(x_{1,D_j}^{\alpha})^\top \beta_j}{(x_{2,D_j}^{\alpha})^\top \beta_j}.$
- 10: **Output:** Node $j \in \mathcal{V}$ outputs \hat{x}_j^{ave} .

Remark 2: Compared to existing state-decompositionbased privacy-preserving average consensus in [15] and [18], PrFTPS is applicable to general digraphs, while the method in [15] is limited to undirected graphs with doubly-stochastic methods. Moreover, our innovative weight mechanism (Section III-A.1) maintains constant weights in the privacypreserving iteration (2) for $k \ge 1$, as opposed to the timevarying weights in [15] and [18]. These constant weights play a crucial role in the final value theorem [26], allowing PrFTPS to compute an exact average consensus in finite time using the coefficients of the minimal polynomial associated with iteration (2). In contrast, the weight mechanisms in [15] and [18] only permit asymptotic average consensus, which limits their application to solving distributed optimization problems. Our proposed weight mechanism overcomes this limitation and facilitates the application of our PrFTPS algorithm to solve distributed optimization problems while preserving privacy, as shown in Algorithm 2.

B. Finite-Time Privacy-Preserving Push-Sum based Gradient Descent Algorithm

In this subsection, we design a PrFTPS based gradient method to address problem (1). We first assume the following conditions about Problem (1).

Assumption 2: Each objective function f_i is μ -strongly

convex with L-Lipschitz continuous gradients, i.e.,

$$\begin{split} \langle \nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle &\geq \mu ||\mathbf{x} - \mathbf{y}||^2, \\ ||\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})|| &\leq L ||\mathbf{x} - \mathbf{y}||, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p. \end{split}$$

Under Assumption 2, Problem (1) has a unique optimal solution $x^* \in \mathbb{R}^p$ [8].

To address problem (1) distributively, we propose the following PrFTPS based GD algorithm inspired by the distributed structure in [7], [8], [22]. Starting from the initial condition $x_i(0) \in \mathbb{R}^p$ and $y_i(0) = \nabla f_i(x_i(0))$, for all $t \ge 0$, we have

$$y_i(t) \leftarrow \text{Algorithm } 1(\nabla f_i(x_i(t)), t),$$
 (5a)

$$x_i(t+1) = \sum_{j \in N_i^- \cup \{i\}} \bar{a}_{ij} x_j(t) - \eta y_i(t),$$
 (5b)

where η is the stepsize and $\bar{\mathbf{A}} = [\bar{a}_{ji}] \in \mathbb{R}^n$ is row-stochastic. The details are summarized in Algorithm 2 in the following.

Algorithm 2 A Privacy-Preserving Finte-Time Push-Sum based GD Algorithm (PrFTPS-GD)

- 1: Initialization: Stepsize η , maximum optimization iteration number T, graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$.
- 2: Input: Node $i \in \mathcal{V}$ sets the initial value $x_i(0), y_i(0) =$ $\nabla f_i(x_i(0))$ and t = 0.
- 3: for $t \leq T$ do
- Put $\nabla f_i(x_i(t)), t$ as input to Algorithm 1 and get 4: \hat{x}^{ave} ; design $y_i(t) = \hat{x}^{ave}$. output
- Compute $x_i(t+1)$ using (5b) with $y_i(t)$. 5:
- $t \leftarrow t + 1$ 6:
- 7: end for
- 8: **Output:** Node $i \in \mathcal{V}$ obtains the solution x^* .

Algorithm 2 guarantees that the number of iterations needed at every optimization step $t \ge 1$ is the minimum. Fig. 1 shows the number of iterations needed at every optimization step.

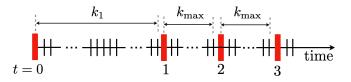


Fig. 1. The finite-time consensus algorithm is terminated after $k_1 =$ $4(D_{\text{max}} + 1)$ iterations in the first step of the optimization. After the first step, the consensus is terminated after $k_{max} = D_{max} + 1$ iterations.

Remark 3: Compared to conventional distributed optimization algorithms, such as AB [7] and Push-Pull [8], PrFTPS-GD requires additional communication rounds for each optimization step, as illustrated in Fig. 1. Although the separate timescales for optimization and consensus steps may slow down the convergence speed, they are crucial for ensuring privacy preservation and accuracy of PrFTPS-GD, as demonstrated by the rigorous theoretical analysis presented in Sections III-C and III-D.

C. Convergence Analysis

In this subsection, we provide the proof of the convergence and accuracy of Algorithm 1 and Algorithm 2.

From the weight mechanism, it can be seen that for k > 1, the coupling weights are constants. Hence, iteration (2) can be written by using matrix-vector notation as follows:

$$\boldsymbol{x_l}(k+1) = \boldsymbol{P}\boldsymbol{x_l}(k), \quad \forall k \ge 1, l = 1, 2, \tag{6}$$

where

$$\boldsymbol{x_l}(k) = [x_{1,l}^{\alpha}(k), \dots, x_{n,l}^{\alpha}(k), x_{1,l}^{\beta}(k), \dots, x_{n,l}^{\beta}(k)]^{\top},$$
$$\hat{\boldsymbol{P}} = \begin{bmatrix} \boldsymbol{P} & \frac{1}{2}\boldsymbol{I}_n\\ \boldsymbol{\Lambda} & \frac{1}{2}\boldsymbol{I}_n \end{bmatrix}$$

with $\Lambda = \operatorname{diag}(a_1^{\beta,\alpha}, \ldots, a_n^{\beta,\alpha})$ and $\boldsymbol{P} = [p_{ij}].$ Moreover, equation (5b) can be rewritten as

$$\boldsymbol{x}(t+1) = \boldsymbol{A}\boldsymbol{x}(t) - \eta \boldsymbol{y}(t), \tag{7}$$

where $\mathbf{A} = \bar{\mathbf{A}} \otimes \mathbf{I}_p, \boldsymbol{x}(t) = [x_1(t)^\top, \dots, x_n(t)^\top]^\top$ and $\boldsymbol{y}(t) =$ $[y_1(t)^{\top},\ldots,y_n(t)^{\top}]^{\top}.$

Before presenting Theorem 1, the following lemmas are needed.

Lemma 1: (Theorem 8.4.4 in [29]) Under Assumption 1, the matrix A has a unique nonnegative left eigenvector u^{+} (with respect to eigenvalue 1) with $u^{\dagger} \mathbf{1}_{np} = np$.

Lemma 2: (Adapted from Lemma 4 in [8]) Under Assumptions 1, there exists a matrix norm $|| \cdot ||_A$, defined as $||M||_A = AMA^{-1}$ for all $M \in \mathbb{R}^{np \times np}$, where $A \in \mathbb{R}^{np \times np}$ is invertible, such that $\sigma_A := ||\mathbf{A} - \frac{\mathbf{1}_{np}u^{\top}}{n}||_A < 1$, where \mathbf{A} is the update matrix defined in (7), and σ_A is arbitrarily close to the spectral radius $\rho(\mathbf{A} - \frac{\mathbf{1}_{np} u^{\top}}{n}) < 1.$

Now, we present Theorem 1 in the following.

Theorem 1: Under Assumptions 1 and 2, for each node $j \in$ V.

1) Algorithm 1 outputs the exact average of initial values

of all nodes, i.e., $\forall j \in \mathcal{V}, \hat{x}_{j}^{ave} = \frac{1}{n} \sum_{i \in \mathcal{V}} x_i(0)$. 2) When $0 < \eta < \frac{1}{\mu+L}$, where μ, L are defined in Assumption 2, Algorithm 2 converges linearly related to the optimization iteration number to the global optimal, i.e., $||\boldsymbol{x}(t) - \mathbf{1} \otimes x^{\star}||_2$ converges to 0 linearly.

Proof: See Appendix A.

D. Privacy-preserving Performance Analysis

In this subsection, we analyze the privacy-preserving performance of Algorithm 2 against honest-but-curious nodes and eavesdroppers. First, the following assumption is needed.

Assumption 3: Considering a digraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, each agent $i, \forall i \in \mathcal{V}$, does not know the structure of the whole network, i.e., the Laplacian of the network.

This assumption shows that agent i has no access to the whole consensus dynamics in (2), which is very natural in distributed systems since agent *i* is only aware of its outgoing link weights $p_{ii}(k), j \in N_i^- \cup \{i\}$. Without other agents' weights, matrix \hat{P} in (6) is inaccessible to agent *i*.

Note that only local gradient information is exchanged in Algorithm 1, and the outputs of Algorithm 1 are the same for each agent. Hence, if Algorithm 1 is able to preserve privacy of each agent in the network, we can deduce that Algorithm 2 can preserve privacy of each agent.

Next, we show the privacy preservation of Algorithm 1.

Under Algorithm 1, the information set accessible to the set of honest-but-curious nodes N at time k can be defined as

$$\mathcal{I}_{\mathcal{N}}(k) = \{ x_{a,l}^{\alpha}(k), x_{a,l}^{\rho}(k), p_{ja}(k), p_{ap}(k), x_{p,l}^{\alpha}(k), \\ \mid p \in N_a^-, a \in \mathcal{N}, j \in \mathcal{V}, l = 1, 2 \}.$$

Similar, an eavesdropper \mathcal{R} is assumed to eavesdrop some edges $\varepsilon_{ij} \in \mathcal{E}_{\mathcal{R}}$ and its information set is denoted by

$$\mathcal{I}_{\mathcal{R}}(k) \triangleq \{ x_{j,l}^{\alpha}(k), p_{ij}(k) \mid \forall \varepsilon_{ij} \in \mathcal{E}_{\mathcal{R}}, j \in \mathcal{V}, l = 1, 2 \}.$$

Theorem 2: Under Assumptions 1 and 3, for each node $j \in \mathcal{V}$, under Algorithm 1, the privacy of agent j can be preserved:

1) Against a set of honest-but-curious nodes \mathcal{N} if at least one neighbor of node j does not belongs to \mathcal{N} , i.e., $N_j^+ \cup N_j^- \notin \mathcal{N}$.

2) Against eavesdropper \mathcal{R} if there exists one edge ε_{mj} or ε_{jm} that eavesdropper \mathcal{R} cannot eavesdrop, where $m \in N_j^+$ or $m \in N_j^-$.

IV. SIMULATIONS

Consider a strongly connected digraph containing n = 5 agents. The following distributed least squares problem is considered:

$$\min_{x \in \mathbb{R}^p} F(x) = \frac{1}{n} \sum_{i=1}^n f_i(x) = \frac{1}{5} \sum_{i=1}^5 ||A_i x - b_i||^2, \quad (8)$$

where $A_i \in \mathbb{R}^{q \times p}$ is only known to node $i, b_i \in \mathbb{R}^q$ is the measured data and $x \in \mathbb{R}^p$ is the common decision variable. In this simulation, we set q = p = 3 and $\eta = 0.1$. All elements of A_i and b_i are set from independent and identically distributed samples of standard normal distribution $\mathcal{N}(0, 1)$. The finite-time consensus (Algorithm 1) stage consists of $k_1 = 64$ (t = 0) and $k_{\max} = 17$ $(t \ge 1)$ communication steps inside each PrFTPS-GD optimization iteration, i.e., the optimization variable $\boldsymbol{x}(0)$ takes 64 steps to become $\boldsymbol{x}(1)$ and then variable $\boldsymbol{x}(t)$ is updated every 17 iterations for $t \ge 1$.

The normalized residual $\sum_{i}^{5}(||x_i(t) - x^*||/||x_i(0) - x^*||)$ is illustrated in Fig. 2 to compare PrFTPS-GD with DPGT [20] and SD-Push-Pull [21]. Notice that as we have multiple consensus steps in Algorithm 1 inside our PrFTPS-GD while there is only one step in DPGT and SD-Push-Pull, we have scaled each PrFTPS-GD optimization iteration number to include the consensus number (i.e., k_1 and k_{max}) directly. It is shown that the proposed PrFTPS-GD converges linearly related to optimization iteration number. The stepsizes of all algorithms are manually tuned to obtain the corresponding best convergence performance. For SD-Push-Pull, the stepsize is set to be $\eta = 0.1$ and it can be seen that SD-Push-Pull only converge to suboptimality while PrFTPS-GD can converge to the optimal point. In terms of DPGT, we choose the stepsize with the diminishing sequence as

$$\lambda^k = \frac{0.02}{1+0.1k}, \quad \gamma_1^k = \frac{1}{1+0.1k^{0.9}}, \quad \gamma_2^k = \frac{1}{1+0.1k^{0.8}}$$

Fig. 2 demonstrates clearly that PrFTPS-GD converges faster to the optimal solution than DPGT.

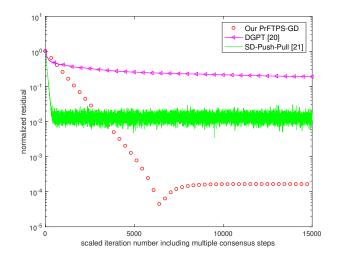


Fig. 2. Performance comparison among our proposed PrFTPS-GD, DPGT [20] and SD-Push-Pull [21].

V. CONCLUSION AND FUTURE WORK

In this paper, a privacy-preserving finite-time push-sum based gradient descent algorithm is proposed to solve the distributed optimization problem over a directed graph. Compared to existing privacy-preserving algorithms in the literature, the proposed one can converge linearly to the global optimum. Moreover, privacy of each agent is preserved via a state decomposition mechanism.

Future work includes considering constrained optimization problems in large-scale. Moreover, privacy-preserving distributed optimiation algorithm with quantization communication is a potential research direction.

APPENDIX

A. Proof of Theorem 1

1) By the weight mechanism and the iteration in (2) at iteration k = 0, we have $\mathbf{1}^{\top} \boldsymbol{x}_{1}(1) = \mathbf{1}^{\top} \boldsymbol{x}_{1}(0) = 2 \sum_{i=1}^{n} x_{i}(0)$ and $\mathbf{1}^{\top} \boldsymbol{x}_{2}(1) = \mathbf{1}^{\top} \boldsymbol{x}_{2}(0) = 2n$.

Since \hat{P} is irreducible, column-stochastic with positive diagonals, from Perron-Frobenius theorem, we have $\rho(\hat{P}) = 1$. Denote $\boldsymbol{v} = [v_i]$ as the right eigenvector corresponding to the eigenvalue of 1, we have $\lim_{k\to\infty} \hat{P}^k = \boldsymbol{v} \mathbf{1}^\top$. Then,

$$\frac{x_{j,1}^{\alpha}(\infty)}{x_{j,2}^{\alpha}(\infty)} = \frac{[\hat{P}^{\infty} x_{1}(1)]_{j}}{[\hat{P}^{\infty} x_{2}(1)]_{j}} = \frac{[v\mathbf{1}^{\top} x_{1}(1)]_{j}}{[v\mathbf{1}^{\top} x_{2}(1)]_{j}} = \frac{\sum_{i=1}^{n} x_{i}(0)}{n}.$$

Since the coefficient β_j is independent of the initial node state,

$$\hat{x}_{j}^{ave} = \frac{(x_{1,D_{j}}^{\alpha})^{\top} \boldsymbol{\beta}_{j}}{(x_{2,D_{j}}^{\alpha})^{\top} \boldsymbol{\beta}_{j}} = \frac{x_{j,1}^{\alpha}(\infty)}{x_{j,2}^{\alpha}(\infty)} = \frac{\sum_{i=1}^{n} x_{i}(0)}{n}.$$
 (9)

2) Denote $\bar{x}(t) = u^{\top} \boldsymbol{x}(t)/n, \bar{y}(t) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(x_i(t))$ and $g(t) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\bar{x}(t))$. From the above analysis, we know that at time iteration t, each agent can obtain the average gradient at time t via Algorithm 1, i.e, $y_i(t) = \bar{y}(t), \boldsymbol{y}(t) = 1\bar{y}(t)$. Hence, from iteration (7), we have

$$\bar{x}(t+1) - x^* = \bar{x}(t) - \eta \bar{y}(t) - x^* = \bar{x}(t) - \eta g(t) - x^* - \eta (\bar{y}(t) - g(t)),$$

$$\boldsymbol{x}(t+1) - \mathbf{1}\bar{\boldsymbol{x}}(t+1) = \mathbf{A}\boldsymbol{x}(t) - \eta\boldsymbol{y}(t) - \mathbf{1}\bar{\boldsymbol{x}}(t) + \eta\mathbf{1}\bar{\boldsymbol{y}}(t)$$
$$= (\mathbf{A} - \mathbf{1}_n \boldsymbol{u}^\top / n)(\boldsymbol{x}(t) - \mathbf{1}\bar{\boldsymbol{x}}(t)).$$

Based on [30, Lemma 8(c) and 10], if $\eta < 1/(\mu + L)$, we have

$$\begin{aligned} ||\bar{x}(t+1) - x^{\star}||_{2} &\leq (1 - \eta\mu) ||\bar{x}(t) - x^{\star}||_{2} + \eta ||\bar{y}(t) - g(t)|| \\ &\leq (1 - \eta\mu) ||\bar{x}(t) - x^{\star}||_{2} + \frac{\eta L}{\sqrt{n}} ||\boldsymbol{x}(t) - \mathbf{1}\bar{x}(t)||_{2}. \end{aligned}$$
(10)

Moreover, from the result of Lemma 2, we can obtain that

$$||\bar{x}(t+1) - x^{\star}||_{2} \le (1 - \eta\mu)||\bar{x}(t) - x^{\star}||_{2} + \frac{\eta qL}{\sqrt{n}}||\boldsymbol{x}(t) - \mathbf{1}\bar{x}(t)||_{A}$$

Denote $V(t) = [||\bar{x}(t+1) - x^{\star}||_2, ||\boldsymbol{x}(t) - \mathbf{1}\bar{x}(t)||_A]^{\top}$, we have

$$V(t+1) \le \boldsymbol{M}V(t), \tag{11}$$

where the transition matrix $M = \begin{bmatrix} 1 - \eta \mu & \eta q L / \sqrt{n} \\ 0 & \sigma_A \end{bmatrix}$.

Since $0 < 1 - \eta \mu < 1$ and $0 < \sigma_A < 1$, we can obtain that the spectral radius of M is strictly less than 1 and therefore $||\boldsymbol{x}(t) - \boldsymbol{1} \otimes x^*||_2$ converges to zero linearly.

B. Proof of Theorem 2

Since under Algorithm 1, the maximum communication round equals to k_1 , the information set $\mathcal{I}_{\mathcal{N}}(0:k_1)$ and $\mathcal{I}_{\mathcal{R}}(0:k_1)$ denote all the information accessible to the adversary. From Algorithm 2, it can be seen that the private information $\nabla f_j(x_j(t)), \forall t \ge 0$ is regarded as the input of Algorithm 1. Hence, it suffices to prove that the privacy of initial value of agent $j, x_j(0)$ is preserved under Algorithm 1.

1) Denote $\mathcal{L} = \mathcal{V} \setminus \mathcal{A}$ as the set of legitimate nodes. Since $N_j^+ \cup N_j^- \notin \mathcal{A}$, there exists at least one node m that belongs to $N_j^+ \cup N_j^-$ but not \mathcal{A} . Fix any feasible information set $\mathcal{I}_{\mathcal{N}}(0:k_1)$. We denote $\{x_{n,1}^{\alpha}(0)', x_{n,1}^{\beta}(0)', p_{in}(0)' \mid n \in \mathcal{L}, i \in \mathcal{V}\}$ as an arbitrary set of initial substate values and weights that satisfies $\mathcal{I}_{\mathcal{N}}(0:k_1)$. Hence, we have $x_j(0)' = (x_{j,1}^{\alpha}(0)' + x_{j,1}^{\beta}(0)')/2$. We then denote $x_j(0)''$ as $x_j(0)'' = x_i(0)' + e$, where e is an arbitrary real number. Next we show that there exists a set of values $\{x_{n,1}^{\alpha}(0)'', x_{n,1}^{\beta}(0)'', p_{in}(0)'', n \in \mathcal{L}, i \in \mathcal{V}\}$ which makes $x_j(0)'' \in \Delta_j(\mathcal{I}_{\mathcal{N}}(0:k_1))$. The initial substate values $x_{n,1}^{\alpha}(0)'', x_{n,1}^{\beta}(0)''$ are denoted as follows.

$$\begin{aligned} x_{q,1}^{\alpha}(0)'' &= x_{q,1}^{\alpha}(0)', x_{q,1}^{\beta}(0)'' = x_{q,1}^{\beta}(0)', \forall q \in \mathcal{L} \setminus \{j, m\}, \\ x_{m,1}^{\alpha}(0)'' &= x_{m,1}^{\alpha}(0)', x_{m,1}^{\beta}(0)'' = x_{m,1}^{\beta}(0)' - 2e, \\ x_{j,1}^{\alpha}(0)'' &= x_{j,1}^{\alpha}(0)', x_{j,1}^{\beta}(0)'' = x_{j,1}^{\beta}(0)' + 2e. \end{aligned}$$

$$(12)$$

Then we consider the following two situations.

Situation I: Consider $m \in N_j^-$, then the information set sequence accessible to set \mathcal{N} equals to $\mathcal{I}_{\mathcal{N}}(0:k_1)$ under the initial substate values in (12) and the following weights:

$$a_{n}^{(\beta,\alpha)}(0)'' = a_{n}^{(\beta,\alpha)}(0)', \forall n \in \mathcal{L},$$

$$p_{mm}(0)'' = (p_{mm}(0)'x_{m,1}^{\alpha}(0)' + 2e)/x_{m,1}^{\alpha}(0)',$$

$$p_{jm}(0)'' = (p_{jm}(0)'x_{m,1}^{\alpha}(0)' - 2e)/x_{m,1}^{\alpha}(0)',$$

$$p_{qm}(0)'' = p_{qm}(0)', \forall q \in \mathcal{L} \setminus \{j, m\},$$

$$p_{np}(0)'' = p_{np}(0)', \forall n \in \mathcal{L}, \forall p \in \mathcal{L} \setminus \{m\}.$$
(13)

Situation II: Consider $m \in N_j^+$, then the information set sequence accessible to set \mathcal{N} equals to $\mathcal{I}_{\mathcal{N}}(0:k_1)$ under the initial substate values in (12) and the following weights:

$$a_{n}^{(\beta,\alpha)}(0)'' = a_{n}^{(\beta,\alpha)}(0)', \forall n \in \mathcal{L}, p_{jj}(0)'' = (p_{jj}(0)'x_{j,1}^{\alpha}(0)' - 2e)/x_{j,1}^{\alpha}(0)', p_{mj}(0)'' = (p_{mj}(0)'x_{j,1}^{\alpha}(0)' + 2e)/x_{j,1}^{\alpha}(0)',$$
(14)
$$p_{qj}(0)'' = p_{qj}(0)', \forall q \in \mathcal{L} \setminus \{j, m\}, p_{np}(0)'' = p_{np}(0)', \forall n \in \mathcal{L}, \forall p \in \mathcal{L} \setminus \{j\}.$$

Summarizing Situations I and II, we have that $x_j(0)'' = x_j(0)' + e \in \Delta_j(\mathcal{I}_N(0:k_1))$, then

$$\operatorname{Diam}(\Delta_j(\mathcal{I}_{\mathcal{N}}(0:k_1))) \ge \sup_{e \in \mathbb{R}} |x_j(0)' - (x_j(0)' + e)| = \infty.$$

From Definition 1, the first statement is proved.

2) For the second statement, due to the topology constraints, the eavesdropper \mathcal{R} can not eavesdrop on ε_{mj} or ε_{jm} , where $m \in N_j^+$ or $m \in N_j^-$, i.e., $p_{mj}(0)$ or $p_{jm}(0)$ is inaccessible to \mathcal{R} . Moreover, since the self-weights $p_{jj}(0), p_{mm}(0)$ are not transmitted over the communication network, the eavesdropper \mathcal{R} can not obtain any information of $p_{jm}(0), p_{mm}(0)$ or $p_{mj}(0), p_{jj}(0)$). Then, similar to the proof in the first statement, we have $\text{Diam}(\Delta_j(\mathcal{I}_{\mathcal{R}}(0 : k_1))) = \infty$, i.e., the privacy of $x_j(0)$ is preserved.

REFERENCES

- P. Braun, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann, "A distributed optimization algorithm for the predictive control of smart grids," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3898–3911, 2016.
- [2] S. Dougherty and M. Guay, "An extremum-seeking controller for distributed optimization over sensor networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 928–933, 2016.
 [3] R. Mohebifard and A. Hajbabaie, "Distributed optimization and coordi-
- [3] R. Mohebifard and A. Hajbabaie, "Distributed optimization and coordination algorithms for dynamic traffic metering in urban street networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1930–1941, 2018.
- [4] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.
- [5] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes," in *Proceedings of the 54th IEEE Conference on Decision* and Control (CDC), 2015, pp. 2055–2060.
- [6] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms," *IEEE Control System Magazine*, vol. 39, no. 3, pp. 40–72, 2019.
- [7] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control System Letters*, vol. 2, no. 3, pp. 315–320, 2018.
- [8] S. Pu, W. Shi, J. Xu, and A. Nedić, "Push-pull gradient methods for distributed optimization in networks," *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 1–16, 2020.
- [9] T. Yang, X. Yi, J. Xu, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, W. Hong, Z. Lin and K. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.
- [10] A. Mandal, "Privacy preserving consensus-based economic dispatch in smart grid systems," in *Future Network Systems and Security: Second International Conference (FNSS)*, 2016, pp. 98–110.
- [11] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography: Third Theory of Cryptography Conference*, pp. 265–284, 2006.
- [12] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private average consensus: Obstructions, trade-offs, and optimal algorithm design," *Automatica*, vol. 81, pp. 221–231, 2017.

- [13] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proceedings of the 16th International Conference on Distributed Computing and Networking*, 2015, pp. 1–10.
- [14] Y. Lu and M. Zhu, "Privacy preserving distributed optimization using homomorphic encryption," *Automatica*, vol. 96, pp. 314–325, 2018.
- [15] Y. Wang, "Privacy-preserving average consensus via state decomposition," *IEEE Transactions on Automatic Control*, vol. 64, no. 11, pp. 4711–4716, 2019.
- [16] T. Charalambous, N. E. Manitara, and C. N. Hadjicostis, "Privacypreserving average consensus over digraphs in the presence of time delays," in *Proceedings of the 57th IEEE Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2019, pp. 238–245.
- [17] H. Gao, C. Zhang, M. Ahmad, and Y. Wang, "Privacy-preserving average consensus on directed graphs using push-sum," in *IEEE Conference on Communications and Network Security (CNS)*, 2018, pp. 1-9.
- [18] X. Chen, L. Huang, K. Ding, S. Dey, and L. Shi, "Privacy-preserving push-sum average consensus via state decomposition," *IEEE Transactions on Automatic Control*, 2023, doi: 10.1109/TAC.2023.3256479.
- [19] S. Mao, Y. Tang, Z. wei Dong, K. Meng, Z. Y. Dong, and F. Qian, "A privacy preserving distributed optimization algorithm for economic dispatch over time-varying directed networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1689–1701, 2021.
- [20] Y. Wang and A. Nedić, "Tailoring gradient methods for differentiallyprivate distributed optimization," *IEEE Transactions on Automatic Control*, 2023, doi: 10.1109/TAC.2023.3272968.
- [21] X. Chen, L. Huang, L. He, S. Dey, and L. Shi, "A differential private method for distributed optimization in directed networks via state decomposition," *IEEE Transactions on Control of Network Systems*, 2023, doi: 10.1109/TCNS.2023.3264932.

- [22] W. Jiang and T. Charalambous, "A fast finite-time consensus based gradient method for distributed optimization over digraphs," in *Proceedings* of the 61st IEEE Conference on Decision and Control (CDC), 2022, pp. 6848–6854.
- [23] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2003 pp. 482–491.
- [24] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," Advances in neural information processing systems, vol. 32, 2019.
- [25] T. Charalambous, Y. Yuan, T. Yang, W. Pan, C. N. Hadjicostis, and M. Johansson, "Distributed finite-time average consensus in digraphs in the presence of time delays," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 4, pp. 370–381, 2015.
- [26] Y. Yuan, G.-B. Stan, L. Shi, and J. Gonçalves, "Decentralised final value theorem for discrete-time LTI systems with application to minimal-time distributed consensus," in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC)*, 2009, pp. 2664–2669.
- [27] T. Charalambous and C. N. Hadjicostis, "When to stop iterating in digraphs of unknown size? An application to finite-time average consensus," in *IEEE European Control Conference (ECC)*, 2018, pp. 1–7.
- [28] W. Jiang and T. Charalambous, "Fully distributed alternating direction method of multipliers in digraphs via finite-time termination mechanisms," arXiv:2107.02019, 2021.
- [29] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, 2012.
- [30] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1245–1260, 2017.