Dynamics based Privacy Preservation in Decentralized Optimization

Huan Gao a,1, Yongqiang Wang b,2, Angelia Nedić c

aSchool of Automation, Northwestern Polytechnical University, Xi’an 710129, China
bDepartment of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA
cSchool of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85281, USA

Abstract

With decentralized optimization having increased applications in various domains ranging from machine learning, control, sensor networks, to robotics, its privacy is also receiving increased attention. Existing privacy-preserving approaches for decentralized optimization achieve privacy preservation by patching decentralized optimization with information-technology privacy mechanisms such as differential privacy or homomorphic encryption, which either sacrifices optimization accuracy or incurs heavy computation/communication overhead. We propose an inherently privacy-preserving decentralized optimization algorithm by exploiting the robustness of decentralized optimization to uncertainties in optimization dynamics. More specifically, we present a general decentralized optimization framework, based on which we show that privacy can be enabled in decentralized optimization by adding randomness in optimization parameters. We further show that the added randomness has no influence on the accuracy of optimization, and prove that our inherently privacy-preserving algorithm has R-linear convergence when the global objective function is smooth and strongly convex. We also rigorously prove that the proposed algorithm can avoid the gradient of a node from being inferable by other nodes. Numerical simulation results confirm the theoretical predictions.

Key words: Privacy preservation; decentralized optimization.

1 Introduction

Decentralized optimization has received increased attention due to its vast applications in online learning [48], distributed sensing [2], formation control [34], source localization [51], and power system control [7]. In many of these applications, a network of nodes collectively solve the following problem

\[
\min_{x \in \mathbb{R}^d} F(x) \triangleq \sum_{i=1}^{n} f_i(x)
\]

where \(f_i : \mathbb{R}^d \to \mathbb{R}\) is a local objective function accessible only to node \(i\).

Over the past decade, a number of gradient-based first order algorithms have been developed to solve the problem. Early results include the decentralized subgradient (DGD) algorithm [26] which combines average consensus with (sub)gradient descent under diminishing stepsizes. Its convergence rate is \(O((\ln k)/\sqrt{k})\) for general convex functions and \(O((\ln k)/k)\) for strongly convex functions, where \(k\) is the number of iterations. [49] shows that the convergence rate of DGD can be improved under a fixed stepsizes but at the expense of optimization accuracy. To guarantee both fast convergence and exact solution under fixed stepsizes, many algorithms propose to replace the local gradients in DGD with an auxiliary variable which tracks the global gradient, with typical examples include Aug-DGM [46], DIGing [24, 33], ATC-DIGing [25], AsynDGM [47], AB [45], and Push-Pull [4, 32, 53]. These algorithms can achieve R-linear convergence. While most of existing algorithms assume a constant interaction graph, convergence of such gradient-tracking based algorithms on time-varying graphs have also been discussed in [24], [47], and [38]. Other

Suppose that a sequence \(\{x^k\}\) converges to \(x^*\) in some norm \(\| \cdot \|\). Then the convergence is R-linear with decay parameter \(\rho \in (0, 1)\) if there exists a positive constant \(c\) such that \(\|x^k - x^*\| \leq c\rho^k\) holds for any \(k\) [24].
However, none of the aforementioned algorithms consider the privacy of individual nodes, which is unacceptable in many applications. For example, in the rendezvous problem where a group of nodes use decentralized optimization to agree on the optimal assembly position, individual nodes may want to keep their initial positions private, which is important in hostile environments. As indicated in [15], without protection by an appropriate privacy mechanism, a node’s initial position can be easily inferred by an adversary in gradient-decent based rendezvous algorithms. Another example illustrating the importance of privacy in decentralized optimization is collaborative machine learning where gradients/model updates exchanged among participating machines may contain sensitive information such as personal medical record and salary [48].

To address the pressing need of privacy in decentralized optimization, recently results have emerged on privacy-preserving decentralized optimization. For example, differential-privacy based approaches are proposed in [15, 28, 42]. However, such approaches will unavoidably compromise the accuracy of optimization results. To enable privacy protection with guaranteed optimization accuracy, partially homomorphic encryption based approaches have been proposed in our own prior results [50, 52] as well as others’ [20]. However, such approaches will incur heavy computation and communication overhead. [48] and [19] showed that privacy can be obtained by incorporating a projection step or injecting constant uncertainties in stepsizes. However, both approaches have limitations in privacy protection: projection based defense requires individual agents to have a priori knowledge of the optimal solution, whereas constant uncertainties in stepsizes are unable to cover arbitrarily large variations on the gradients. Other approaches include [6, 18, 43]. However, they are only applicable to undirected graphs.

Recently, through using random coefficients and/or initial conditions, others as well as our group have proposed several private consensus algorithms [3, 8–10, 12, 16, 21, 22, 30, 36, 37]. These algorithms can protect the initial value of a node from being inferable by adversaries. Inspired by this line of research, in this paper, we propose to protect the gradient of participating nodes in decentralized optimization by leveraging the robustness of decentralized optimization dynamics. More specifically, by judiciously injecting uncertainties in optimization dynamics, we obfuscate exchanged information without affecting convergence to the exact optimal solution. We rigorously prove that the proposed algorithm can avoid the gradient of a node from being inferable by other nodes. Since protecting the gradient means protecting the values of the gradient function over the entire domain (or protecting both function types and function parameters), it is much more challenging than protecting a single initial value considered in the private consensus problem.

The main contributions of the paper are as follows: 1) We propose a dynamics based privacy protection approach for decentralized optimization that neither sacrifices optimization accuracy nor incurs heavy computation/communication overhead. This is in distinct difference from existing approaches based on differential privacy (which compromise optimization accuracy) and approaches based on homomorphic encryption (which incur heavy computation and communication overhead); 2) Our approach is also different from the multi-party secure computation approach in [13] which requires each node to communicate with two non-colluding external servers. By introducing randomness into interaction parameters, our approach is implementable in a fully decentralized manner without the assistance of any external servers; 3) We propose a new privacy definition based on the indistinguishability of gradient’s arbitrary variations to adversaries from the viewpoint of accessible information, which is stricter than the unobservability/unsolvability based privacy definitions; 4) To facilitate the dynamics based privacy design, we propose a general framework for gradient-tracking based decentralized optimization which includes as special cases many existing decentralized optimization algorithms, such as Aug-DGM [46], DIGing [24, 33], ATCDIGing [25], AsynDGM [47], AB [45], and Push-Pull [4, 32, 53]; 5) We analytically prove that despite the time-varying randomness injected into interaction parameters and the general framework, our proposed approach can still maintain $R$-linear convergence when the global objective function is strongly convex.

The rest of this paper is organized as follows. Sec. 2 formally introduces the problem and reviews some key concepts. A general framework for gradient-tracking based decentralized optimization is introduced in Sec. 3, followed by the inherently privacy-preserving algorithm in Sec. 4. In Sec. 4 we also analyze the convergence of the proposed algorithm and rigorously characterize its privacy-preserving performance. Examples and comparison with existing results are presented in Sec. 5. Finally we conclude this paper in Sec. 6.

**Notations:** $\mathbb{R}$ and $\mathbb{Z}_{\geq 0}$ denote real numbers and nonnegative integers, respectively. $\mathbb{R}^n$ denotes the Euclidean space of dimension $n$, and $\mathbb{R}^{m \times n}$ denotes the set of $m \times n$ matrices with real coefficients. $\mathbf{0}_n \in \mathbb{R}^n$ and $\mathbf{0}_{m \times n} \in \mathbb{R}^{m \times n}$ denote zero vector and $m \times n$ zero matrix, respectively. $\mathbf{1}_n \in \mathbb{R}^n$ denotes the $n \times 1$ all-ones vector, $\mathbf{1}_{m \times n} \in \mathbb{R}^{m \times n}$ denotes the $m \times n$ all-ones matrix, and $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ denotes the $n \times n$ identity matrix. For matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, its transpose is denoted as $\mathbf{A}^T \in \mathbb{R}^{n \times m}$. The notation $\| \cdot \|$ denotes the Euclidean norm of vectors and the spectral norm of matrices. The notation $\otimes$ represents the Kronecker product.

## 2 Problem Formulation

We characterize the interaction among nodes as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the index set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, whose elements are such that an ordered pair $(i, j)$ belongs to $\mathcal{E}$ if and only if there exists a directed link from node $j$ to node $i$, i.e., node...
Algorithm 1 A new decentralized optimization framework

Each node $i$ initializes $x_i^0$ randomly in $\mathbb{R}^d$ and sets $y_i^0 = \nabla f_i(x_i^0)$. At iteration $k$:

1. Node $i$ computes and sends $x_i^k$ as well as $\Lambda_i^k y_i^k$ to its out-neighbors $l \in \mathcal{N}_i^{out}$, where $\Lambda_i^k$ is a diagonal matrix denoting the stepsize.
2. After receiving $x_j^k$ and $\Lambda_j^k y_j^k$ from its in-neighbors $j \in \mathcal{N}_i^{in}$, node $i$ updates $x_i$ as:

$$x_i^{k+1} = \sum_{j \in \mathcal{N}_i^{in}\cup\{i\}} (R_{ij}^k x_j^k - \Lambda_{ij}^k y_j^k)$$

where $R_{ij}^k$ and $\Lambda_{ij}^k$ are coupling weight matrices.

3. After updating $x_i$, node $i$ computes and sends $C_{ii}^k y_i^k + B_{ii}^k (\nabla f_i(x_i^{k+1}) - \nabla f_i(x_i^k))$ to its out-neighbors $l \in \mathcal{N}_i^{out}$, where $C_{ii}^k$ and $B_{ii}^k$ are coupling weight matrices.

4. After receiving $C_{ij}^k y_j^k + B_{ij}^k (\nabla f_j(x_j^{k+1}) - \nabla f_j(x_j^k))$ from its in-neighbors $j \in \mathcal{N}_i^{in}$, node $i$ updates $y_i$ as:

$$y_i^{k+1} = \sum_{j \in \mathcal{N}_i^{in}\cup\{i\}} (C_{ij}^k y_j^k + B_{ij}^k (\nabla f_j(x_j^{k+1}) - \nabla f_j(x_j^k)))$$

In (2)-(3) can be rewritten as

$$x_i^{k+1} = \sum_{j=1}^n (R_{ij}^k x_j^k - \Lambda_{ij}^k y_j^k)$$
$$y_i^{k+1} = \sum_{j=1}^n (C_{ij}^k y_j^k + B_{ij}^k (\nabla f_j^{k+1} - \nabla f_j^k))$$

where $\nabla f_j^k = \nabla f_j(x_j^k)$ is the gradient of $f_j$ evaluated at $x_j^k$, which is typically nonlinear in nature. (4) can be rewritten into a more compact matrix form

$$x_i^{k+1} = R^k x_i^k - \Lambda^k A^k y_i^k$$
$$y_i^{k+1} = C^k y_i^k + B^k (\nabla f_i^{k+1} - \nabla f_i^k)$$

where $x_i^k = [(x_1^k)^T \cdots (x_n^k)^T]^T$, $y_i^k = [(y_1^k)^T \cdots (y_n^k)^T]^T$, $\nabla f_i^k = [(\nabla f_1^k)^T \cdots (\nabla f_n^k)^T]^T$, and $R^k$, $A^k$, $C^k$, and $B^k$ are block matrices, with the $(ij)$-th block entry being $R_{ij}^k$, $A_{ij}^k$, $C_{ij}^k$, and $B_{ij}^k$, respectively. $\Lambda^k$ is a block diagonal matrix with the $i$-th diagonal block being $A_i^k$.

3.2 Relationship with Existing Decentralized Optimization Algorithms

The proposed decentralized optimization framework is very general and includes as special cases many popular decentralized optimization algorithms, such as Aug-DGM [46], DiGing [24, 33], ATC-DiGing [25], AsynDGM [47], ADE [45], and Push-Pull [4, 32, 53]. Table I summarizes the particular selections of parameters $R^k$, $A^k$, $B^k$, $C^k$, and $\Lambda^k$ in (5) to obtain some commonly used decentralized optimization algorithms.
Table I. Particular selections of parameters in the proposed framework to obtain some existing algorithms.

| Algorithm          | R^k | A^k | C^k | B^k | Λ^k |
|--------------------|-----|-----|-----|-----|-----|
| Aug-DGM [46]      | W   | W   | W   | W   | Λ   |
| DiGing [33]       | W   | I   | W   | I   | ΛI  |
| DiGing [24]       | W^k | I   | W^k | I   | ΛI  |
| ATC-DiGing [25]   | W   | W   | W   | W   | Λ   |
| AsynDGM [47]      | W^k | W^k | W^k | I   | A   |
| AΣ [45]           | R   | I   | C   | C   | ΛI  |
| Push-Pull [32]    | R   | R   | C   | C   | ΛI  |
| Push-Pull [4]     | R   | I   | C   | I   | Λ   |
| Push-Pull [53]    | R   | R   | C   | I   | ΛI  |

W and W^k are doubly stochastic, R is time-invariant and row-stochastic, C is time-invariant and column-stochastic, and Λ and Λ^k represent homogeneous and heterogeneous stepsize matrices, respectively. Note that Kronecker product with I^I_r is needed for these matrices when d > 1.

In fact, by setting R^k, A^k, C^k, B^k, and Λ^k to R, I, C, I, and Λ, respectively, our proposed algorithm in (5) can be rewritten as x^{k+1} = (R + C)x^k - CRx^{k-1} - A(∇f^k - ∇f^{k-1}) which becomes EXTRA [41].

As indicated in Table I, our proposed framework reduces to existing algorithms when the parameters are selected appropriately. Moreover, it can also give rise to new algorithms with special properties. In particular, in what follows we show that it results in new algorithms having inherent privacy-preserving capabilities.

4 Privacy-preserving Decentralized Optimization

Based on the general decentralized optimization framework proposed in Sec. 3, we can enable privacy-preservation in decentralized optimization without compromising optimization accuracy.

4.1 Privacy-preserving Design

To enable privacy, we propose to add randomness in step-size A^k and coupling weights R^k, A^k, C^k, and B^k for iterations k < K, where K is a positive integer. The detailed parameter design for each node i is given in Table II. More specifically, for iterations k < K, each node i selects the coupling weights C_{ji} = diag {c_{ji}(1), . . . , c_{ji}(d)} and B_{ji} = diag {b_{ji}(1), . . . , b_{ji}(d)} for j ∈ N^out following any chosen random distributions with support R such as Gaussian or Laplace distribution (so the coupling weights can be negative, positive, or zero). To guarantee the column-stochastic property of both C^k and B^k, each node i sets C_{ii} and B_{ii} as C_{ii} = I_d − ∑_{j∈N^out} C_{ji} and B_{ii} = I_d − ∑_{j∈N^out} B_{ji}, respectively. In this way, the column-stochastic property of both C^k and B^k for iterations k < K is guaranteed in a fully decentralized manner. Since we do not require R^k or A^k to be row-stochastic for iterations k < K, each node i can select R_{ji} and A_{ji} for j ∈ N^out ∪ {i} following any random distributions.

Furthermore, as indicated in Table II, the column stochastic property of C^k and B^k and the row stochastic property of R^k and A^k are required for iterations k ≥ K. Let us take C^k as an example to show how to ensure these properties in a fully decentralized manner. To meet the requirements of Table II, each node i selects a set of real values \{c_{ji} ∈ [η_i, 1] | j ∈ N^out ∪ {i}\} with their sum equal to one, then sets C_{ji} = c_{ji}I_d for j ∈ N^out ∪ {i}. Note that there are many ways to obtain such a set of real values with sum equal to one. For example, node i first randomly selects a set of real values \{p_{ji}^1 | j ∈ N^out ∪ {i}\} from [0, 1], then it determines c_{ji} by normalizing these values \{p_{ji}^1\} via c_{ji} = \frac{1}{1−(\sum_{l=1}^{n}p_{li}^1)\sum_{l=1}^{n}p_{li}^1 + \eta} for j ∈ N^out ∪ {i}. One can verify \sum_{j ∈ N^out ∪ {i}} c_{ji} = 1 and c_{ji} ∈ [η_i, 1] for j ∈ N^out ∪ {i}, and hence the column stochastic property of C^k is guaranteed in a fully decentralized manner.

Remark 1 We allow each agent i to randomly choose its associated coupling weights and stepizes for iterations k < K following any distributions. These distributions can be any continuous probability distribution with support R, e.g., Gaussian distribution, Laplace distribution, etc.

4.2 Convergence Analysis

In this subsection, inspired by the work in [38], we prove that our privacy-preserving algorithm has R-linear convergence. Since the parameters in iterations 0 ≤ k ≤ K − 1 vary randomly with time, we first analyze their influence on states x^K and y^K, based on which we can characterize the evolution of state variables after iteration K − 1. From the parameter design in Table II, we have \sum_{i=1}^{n} C_{i j} = I_d and \sum_{i=1}^{n} B_{i j} = I_d, which, in combination with (4) leads to

\[(\sum_{i=1}^{n} y_{i}^{k+1})^T = (\sum_{i=1}^{n} (y_{i}^{k} + \nabla f_{i}^{k+1} - \nabla f_{i}^{k}))^T \]  

for k ≤ K − 1, where we used the property (C_{i j})^T = C_{j i} and (B_{i j})^T = B_{j i} due to their diagonal structure. We can further rewrite (6) as \sum_{i=1}^{n} y_{i}^{k+1} = \sum_{i=1}^{n} \nabla f_{i}^{k+1} − \sum_{i=1}^{n} \nabla f_{i}^{k}. Given y_{i}^{0} = \nabla f_{i}^{0}, one can obtain

\[(\sum_{i=1}^{n} (y_{i}^{K} - \nabla f_{i}^{K})) = \cdots = (\sum_{i=1}^{n} (y_{i}^{0} - \nabla f_{i}^{0})) = 0_d \]  

The equality constraint in (7) reflects the influence of random time-varying parameters in iterations 0 ≤ k ≤ K − 1 on the
dynamics. Next, under this constraint, we study the system dynamics after iteration $k = 1$.

For our parameter design in Table II, we have $A^k = \lambda I_d$, $R^k_{ij} = r^k_{ij} I_d$, $A^k_{ij} = a^k_{ij} I_d$, $C^k_{ij} = c^k_{ij} I_d$, and $B^k_{ij} = b^k_{ij} I_d$ for $k \geq 1$. Constructing an $n \times n$ matrices $\bar{R}^k$, $\bar{A}^k$, $\bar{C}^k$, and $\bar{B}^k$ with the $(ij)$-th elements being $r^k_{ij}$, $a^k_{ij}$, $c^k_{ij}$, and $b^k_{ij}$, respectively, we have $R^k = \bar{R}^k \otimes I_d$, $A^k = \bar{A}^k \otimes I_d$, $C^k = \bar{C}^k \otimes I_d$, and $B^k = \bar{B}^k \otimes I_d$ for $k \geq 1$. Then for $k \geq 1$, we can rewrite the system dynamics (5) as

$$x^{k+1} = (\bar{R}^k \otimes I_d)x^k - \lambda(\bar{A}^k \otimes I_d)y^k$$

$$y^{k+1} = (\bar{C}^k \otimes I_d)y^k + (\bar{B}^k \otimes I_d)(\nabla f^{k+1} - \nabla f^k).$$

To facilitate convergence analysis, we introduce a state transformation, $s^k = ((\bar{V}^k)^{-1} \otimes I_d) x^k$, where $\bar{V}^k = \text{diag}(\bar{v}^k)$ with the evolution of $\bar{v}^k$ governed by

$$\bar{v}^{k+1} = \bar{C}^k \bar{v}^k$$

for $k \geq 1$. The value of $\bar{v}^k$ at iteration $K$ is set as $\frac{1}{n} 1_n$, i.e., $v^K = \frac{1}{n} 1_n$. Then (8) can be rewritten as

$$x^{k+1} = \bar{P}^k x^k - \lambda(\bar{A}^k \bar{V}^k \otimes I_d)s^k$$

$$s^{k+1} = P^k s^k + ((\bar{V}^{k+1})^{-1} \bar{B}^k \otimes I_d)(\nabla f^{k+1} - \nabla f^k)$$

for $k \geq K$ where $P^k = \bar{P}^k \otimes I_d$ and $\bar{P}^k = (\bar{V}^{k+1})^{-1} \bar{C}^k \bar{V}^k$. As proven in Lemma 4 in Appendix A, $\bar{P}^k$ is row-stochastic and the sequence $\{\bar{P}^k\}$ is ergodic for $k \geq K$. Furthermore, the sequence of $\{v^k\}$ for $k \geq K$ is an absolute probability sequence for $\{\bar{P}^k\}$ (see Appendix A for the definitions of ergodic sequence and absolute probability sequence). We define $\bar{x}_w$, $\bar{x}_w^k$, $r^k$, and $\tilde{s}_w^k$ as

$$\bar{x}_w = (\bar{v}^k)^T \otimes I_d) x^k$$

$$\bar{x}_w^k = x^k - 1_n \otimes \bar{x}_w$$

$$r^k = 1_n \otimes \bar{x}_w^k - 1_n \otimes x^k$$

$$\tilde{s}_w^k = s^k - (1_n(\bar{v}^k)^T \otimes I_d) s^k$$

where for $k \geq K$, $\{\phi^k\}$ is an absolute probability sequence for the ergodic sequence of row-stochastic matrices $\{\bar{R}^k\}$. Before presenting our main result, we introduce some auxiliary lemmas.

**Lemma 1** Under Assumptions 1 and 2, and the parameter design in Table II, the following inequality holds for $k \geq K + N - 1$:

$$\|\bar{x}_w^k\| \leq \|r^k + \lambda Q n \sqrt{n} \| x^k - N + 1 \| + \lambda Q n \sqrt{n} \| x^k - N + 1 \|$$

$$+ \lambda Q n \sqrt{n} \| x^k - N + 1 \| + \lambda Q n \sqrt{n} \| x^k - N + 1 \|$$

$$+ \lambda Q n \sqrt{n} \| x^k - N + 1 \| + \lambda Q n \sqrt{n} \| x^k - N + 1 \|$$

where $\bar{v} = \text{max}\{\beta_1, \ldots, \beta_n\}$, $Q_R = 2n(1+\eta^{-n})/(1-\eta^{-n})$, $r^k = Q_R(1-\eta^{-n}) N^{-1}$, and $N = \text{max}\{N_R, N_P\}$ (see Lemmas 4 and 5 in Appendix A for more details on parameters $N_R$, $N_P$, $Q_R$, and $Q_R$).

**Proof:** The proof is given in Appendix B.

**Lemma 2** Under Assumptions 1 and 2, and the parameter design in Table II, the following inequality holds for $k \geq K$:

$$\|r^k\| \leq \lambda n \| s^k \| + (1 - \lambda n^{-1} \alpha_F) \| r^k \| + \lambda n \| s^k \|$$

where $\alpha_F$ is the stepsize $\lambda \leq 1/\beta_F$.

**Proof:** The proof is given in Appendix C.

**Lemma 3** Under Assumptions 1 and 2, and the parameter design in Table II, the following inequality holds for $k \geq K$:

$$\|r^k\| \leq \lambda n \| s^k \| + (1 - \lambda n^{-1} \alpha_F) \| r^k \| + \lambda n \| s^k \|$$

where $\alpha_F$ is the stepsize $\lambda \leq 1/\beta_F$. Before presenting our main result, we introduce some auxiliary lemmas.
Proof: The proof is given in Appendix D.

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

**Theorem 1** Under Assumptions 1 and 2, and the parameter design in Table II, our algorithm has R-linear convergence when the stepsize parameter \( \lambda \) is sufficiently small.

**Proof:** Denoting \( \mathbf{ξ}^k \) as \( \mathbf{ξ}^k = [\|\mathbf{s}^k\|, \|x^k\|, \|\tilde{x}^k\|]^T \) and invoking the results of Lemmas 1, 2, and 3, we have the following inequality for \( k \geq K + \bar{N} - 1 \)

\[
\begin{bmatrix}
\mathbf{ξ}^{k+1} \\
\vdots \\
\mathbf{ξ}^{k+\bar{N}+2}
\end{bmatrix} \leq \begin{bmatrix}
\mathbf{M}^1 + \lambda \mathbf{M}^2
\end{bmatrix}^{\bar{N}} \begin{bmatrix}
\mathbf{ξ}^k \\
\vdots \\
\mathbf{ξ}^{k+\bar{N}+1}
\end{bmatrix}
\]

where \( \mathbf{M}^1 \) and \( \mathbf{M}^2 \) are given by

\[
\mathbf{M}^1 = \begin{bmatrix}
\mathbf{M}^1_0 & \mathbf{M}^1_1 & \cdots & \mathbf{M}^1_{\bar{N}} & \mathbf{M}^1_{\bar{N}+1}
\end{bmatrix}, \\
\mathbf{M}^2 = \begin{bmatrix}
\mathbf{M}^2_0 & \mathbf{M}^2_1 & \cdots & \mathbf{M}^2_{\bar{N}} & \mathbf{M}^2_{\bar{N}+1}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\mathbf{I}_3 \\
\vdots \\
\mathbf{I}_3
\end{bmatrix}, \\
\begin{bmatrix}
0_{3 \times 3} \\
\vdots \\
0_{3 \times 3}
\end{bmatrix}
\]

respectively, with

\[
\mathbf{M}^1_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2t & 0 & r_P \end{bmatrix}, \\
\mathbf{M}^2_0 = \begin{bmatrix} n_{\beta}q & n_{\beta}q & q \\ 2t & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}
\]

To establish the R-linear convergence of our algorithm, it is sufficient to show that the spectral radius of \( \mathbf{M}(\lambda) \) is less than 1 when \( \lambda \) is small enough. To this end, we first prove that the spectral radius of \( \mathbf{M}^2 \) is 1, i.e., \( \rho(\mathbf{M}^2) = 1 \). Following Theorem 3.2 in [31], the determinant of \( \mathbf{I} - \mathbf{M}^1 \) is given by \( \det(z \mathbf{I} - \mathbf{M}^1) = (z^{\bar{N}} - r_R)(z^{\bar{N}} - r_P)(z - 1)z^{\bar{N} - 1} \). Since \( r_R, r_P \in (0, 1) \) holds, we have \( \rho(\mathbf{M}^1) = 1 \). Moreover, the eigenvalue 1 is simple, and its corresponding right and left eigenvectors are \( \mathbf{u} = 1_N \otimes [0 \ 1 \ 0]^T \) and \( \mathbf{w} = [0 \ 1 \ 0 \ \cdots \ 0]^T \), respectively. Denote the simple eigenvalue of \( \mathbf{M}(\lambda) \) as a function of \( \lambda \), i.e., \( p(\lambda) \). Given \( \mathbf{M}(\lambda) = \mathbf{M}^1 + \lambda \mathbf{M}^2 \), we have \( p(0) = 1 \). Using the matrix perturbation theory in Theorem 6.3.12 in [14], one can obtain

\[
\frac{dp(\lambda)}{d\lambda}\bigg|_{\lambda = 0} = \frac{w^T \mathbf{M}^2 \mathbf{u}}{w^T \mathbf{u}} = -n^{-1} \eta^{-1} \alpha_F < 0.
\]

Since eigenvalues are continuous functions of the elements of a matrix, the simple eigenvalue \( p(\lambda) \) is strictly less than 1 when \( \lambda \) is sufficiently small, implying that the spectral radius of \( \mathbf{M}(\lambda) \) is less than 1 when \( \lambda \) is sufficiently small. Noting that \( \mathbf{M}(\lambda) \) has nonnegative entries and \( \mathbf{M}(\lambda)^{\bar{N}+1} \) has all positive entries, from Theorem 8.5.1 and Theorem 8.5.2 in [14], we can obtain that each entry of \( \mathbf{M}(\lambda)^k \) will converge to zero at the rate of \( \mathcal{O}(\rho(\mathbf{M}(\lambda))^k) \). Therefore, when \( \lambda \) is sufficiently small, \( \|x^k - 1_N \otimes x^k\| \) converges to 0 at the rate of \( \mathcal{O}(\rho(\mathbf{M}(\lambda))^k) \), implying that our algorithm has R-linear convergence.

**Remark 2** Although adding randomness in optimization parameters in the first \( K \) iterations may delay the convergence of our algorithm (as shown in our numerical simulations in Fig. 4), it has no influence on the R-linear convergence rate. The added randomness in optimization parameters in the first \( K \) iterations is key to enable privacy protection, as elaborated in the following subsection.

4.3 Privacy Analysis

To protect \( f_i(\cdot) \), it suffices to protect the gradient function \( \nabla f_i(\cdot) \). In decentralized optimization, every participating node receives messages from its neighbors, and hence
could exploit them to infer other nodes’ private gradients. We denote all information accessible to node \( j \) as \( I_j \), which contains the coupling weights, step-sizes, and states associated with node \( j \), sent information from node \( j \) to its out-neighbors, and received information from node \( j \)’s in-neighbors to itself. A mathematical representation of \( I_j \) is given as follows:

\[
I_j = \{ \mathcal{T}_{j}^{\text{state}}(k) \cup \mathcal{T}_{j}^{\text{end}}(k) \cup \mathcal{T}_{j}^{\text{receive}}(k) \mid k = 0, 1, \ldots \}
\]

\[
\cup \{ A^k_{ji}, B^k_{ji}, C^k_{ji} \mid j < K, \forall l \in \mathcal{N}_i^{\text{in}} \} \\
\cup \{ A^k_{mj}, B^k_{mj} \mid k < K, \forall m \in \mathcal{N}_j^{\text{out}} \} \\
\cup \{ A^k_{lm}, R^k_{lm}, C^k_{lm}, B^k_{lm} \mid k \geq K, \forall l, m \in \mathcal{V} \}
\]

where

\[
\mathcal{T}_{j}^{\text{state}}(k) = \{ x^k_j, y^k_j, A^k_j, y^k_j x^k_j + B^k_j(\nabla f^k_j - \nabla f^k_j) \}
\]

\[
\mathcal{T}_{j}^{\text{end}}(k) = \{ x^k_j, A^k_j y^k_j, C^k_{mj} y^k_j + B^k_{mj}(\nabla f^k_j - \nabla f^k_j) \mid \forall m \in \mathcal{N}_j^{\text{out}} \}
\]

\[
\mathcal{T}_{j}^{\text{receive}}(k) = \{ x^k_j, A^k_j y^k_j, C^k_{jl} x^k_j + B^k_{jl}(\nabla f^k_j - \nabla f^k_j) \mid \forall l \in \mathcal{N}_i^{\text{in}} \}
\]

represent the respective state information, sent information, and received information of node \( j \) at iteration \( k \).

Using the defined information set, we are in position to introduce the attacker model and our definition of privacy.

**Definition 4** A node \( j \) is called honest-but-curious if it follows all protocol steps correctly but is curious and tries to infer the gradient functions of other participating nodes using the information \( I_j \) accessible to itself.

**Definition 5** For a decentralized network of \( n \) nodes, the privacy of node \( i \) is preserved if honest-but-curious adversaries cannot distinguish the actual gradient \( \nabla f_i(\cdot) \) of node \( i \) from an arbitrarily large variation of the gradient \( \nabla f_i(\cdot) = \nabla f_i(\cdot) + \delta \) where \( \delta \) can be any vector in \( \mathbb{R}^d \). That is to say, there exist feasible coupling weights and step-sizes satisfying the requirements in Table II that make the information accessible to honest-but-curious adversaries exactly unchanged under arbitrary variations of node \( i \)’s gradient, which results in the undistinguishability of node \( i \)’s actual gradient.

Definition 5 means that an honest-but-curious adversary cannot even identify a range for a private function’s values and thus is more stringent than the privacy definition used in unobservability based approaches in [1, 29] which define privacy as the inability of an adversary to uniquely determine a protected value. Note that opacity has also been used to protect information in discrete-event systems [17, 39], which was recently extended to continuous-state dynamical systems [35]. In contrast to unobservability based approaches which consider adversaries having access to the entire output and control histories, opacity based approaches only consider adversaries having access to snapshots of the output and the set of controls. So opacity based privacy is weaker than unobservability based privacy [35], and hence is also weaker than the privacy defined here. Furthermore, in contrast to unobservability and opacity based approaches focusing on protecting some state values, we protect participating nodes’ gradient function values over the entire domain (or protecting both function types and function parameters), which is more challenging.

Before analyzing the privacy-preserving performance of our proposed algorithm, we first show that using deterministic parameters, our algorithm can protect the privacy of gradients. We consider two different scenarios: 1) a single adversary acting on its own (i.e., without collusion with other adversaries). Following a similar argument, one can also see that other commonly used algorithms have the same issue of breaching individual nodes’ privacy.

Next, we show that by adding randomness in interaction parameters, our algorithm can protect the privacy of gradients. We consider two different scenarios: 1) a single adversary acting on its own (i.e., without collusion with other adversaries) and 2) multiple adversaries colluding with each other.

### 4.3.1 Non-colluding case

In this case, we assume that there are \( 1 \leq q \leq n - 1 \) honest-but-curious nodes which try to infer node \( i \)’s gradient function without sharing information with each other.

**Theorem 2** In our decentralized optimization algorithm, the privacy of node \( i \) can be preserved if \( |\mathcal{N}_i^{\text{out}} \cup \mathcal{N}_i^{\text{in}}| \geq 2 \) holds.

**Proof** Our idea is based on the indistinguishability of \( \nabla f_i(\cdot) ’s \) arbitrary variations to any honest-but-curious node \( j \). According to Definitions 4 and 5, all the information available to node \( j \) to infer \( \nabla f_i(\cdot) \) is \( I_j \) (cf. (17)), and we
have to prove that when $\nabla f_i(\cdot)$ is altered to $\nabla  \tilde{f}_i(\cdot)$ (could have an arbitrarily large difference from $\nabla f_i(\cdot)$), the information accessible to node $j$, i.e., $\mathcal{I}_j$, could be exactly the same as $\mathcal{I}_j$ in (17). Therefore, we only need to prove that there exists such $\nabla \tilde{f}_i(\cdot)$ that makes $\mathcal{I}_j = \mathcal{I}_j$ hold. Given $|\mathcal{N}_i^{\text{out}} \cup \mathcal{N}_i^{\text{in}}| \geq 2$, there must exist a node $m \in \mathcal{N}_i^{\text{out}} \cup \mathcal{N}_i^{\text{in}}$ such that $m \neq j$ holds. So we only need to show that there exist feasible parameters (coupling weights and step sizes) satisfying the requirements in Table II making $\hat{\mathcal{I}}_j = \mathcal{I}_j$ hold under $\nabla \tilde{f}_i(\cdot) = \nabla f_i(\cdot) + \delta$ and $\nabla \tilde{f}_m(\cdot) = \nabla f_m(\cdot) - \delta$ for any $\delta = [\delta_1, \ldots, \delta_d]^T \in \mathbb{R}^d$. 

We consider $m \in \mathcal{N}_i^{\text{out}}$ and $m \in \mathcal{N}_i^{\text{in}}$, separately (note that if $m \in \mathcal{N}_i^{\text{out}} \cap \mathcal{N}_i^{\text{in}}$ holds, either of the considered cases can be used in the argument to draw a same conclusion):

Case I: If $m \in \mathcal{N}_i^{\text{out}}$ holds, it can be proven that $\hat{\mathcal{I}}_j = \mathcal{I}_j$ holds for any $\delta \in \mathbb{R}^d$ under $\nabla \tilde{f}_i(\cdot) = \nabla f_i(\cdot) + \delta$, $\nabla \tilde{f}_m(\cdot) = \nabla f_m(\cdot) - \delta$, and the following parameters

$$
\begin{align*}
\hat{A}_p^0 &= A_p^0 \quad \forall p \in \mathcal{V} \setminus \{i, m\} \\
\hat{\lambda}_i^0(l) &= \lambda_i^0(l) y_i^0[l] / (y_i^0[l] + \delta_l) \\
\hat{\lambda}_m^0(l) &= \lambda_m^0(l) y_m^0[l] / (y_m^0[l] - \delta_l) \\
\hat{R}_{pq}^0 &= R_{pq}^0 \quad \forall p, q \in \mathcal{V} \\
\hat{A}_{pq}^0 &= A_{pq}^0 \quad \forall p, q \in \mathcal{V} \\
\hat{C}_{pq}^0 &= C_{pq}^0 \quad \forall p, q \in \mathcal{V} \setminus \{i, m\} \\
\hat{c}_{pq}^0(l) &= c_{pq}^0(l) y_i^0[l] / (y_i^0[l] + \delta_l) \quad \forall p \in \mathcal{V} \setminus \{m\} \\
\hat{c}_{rm}^0(l) &= (c_{rm}^0(l) y_i^0[l] + \delta_l) / (y_i^0[l] + \delta_l) \\
\hat{c}_{pm}^0(l) &= (c_{pm}^0(l) y_m^0[l] - \delta_l) / (y_m^0[l] - \delta_l) \\
\hat{c}_{cm}^0(l) &= (c_{cm}^0(l) y_m^0[l] - \delta_l) / (y_m^0[l] - \delta_l) \\
\hat{C}_{pq}^0 &= C_{pq}^0 \quad \forall p, q \in \mathcal{V} \\
\hat{B}_{pq}^0 &= B_{pq}^0 \quad \forall p, q \in \mathcal{V} \\
\hat{A}_p^k &= A_p^k \quad \forall p \in \mathcal{V}, k = 1, 2, \ldots \\
\hat{R}_{pq}^k &= R_{pq}^k \quad \forall p, q \in \mathcal{V}, k = 1, 2, \ldots \\
\hat{A}_{pq}^k &= A_{pq}^k \quad \forall p, q \in \mathcal{V}, k = 1, 2, \ldots \\
\hat{C}_{pq}^k &= C_{pq}^k \quad \forall p, q \in \mathcal{V}, k = 1, 2, \ldots \\
\hat{B}_{pq}^k &= B_{pq}^k \quad \forall p, q \in \mathcal{V}, k = 1, 2, \ldots \\

\end{align*}
$$

where $l = 1, \ldots, d$ and “\setminus” represents set subtraction.

To see why the parameter setting in (20) can ensure $\hat{\mathcal{I}}_j = \mathcal{I}_j$ in the case $m \in \mathcal{N}_i^{\text{out}}$, we consider $k = 0$ and $k \geq 1$, separately. Under the feasible parameters for $k = 0$, it can be verified that the information accessible to node $j$ at the initial iteration $k = 0$ keeps unchanged and the states of each node $p \in \mathcal{V}$ satisfy $\hat{x}_i^0 = x_i^0$ and $\hat{y}_p^0 = y_p^0$. Then by setting feasible parameters for iterations $k \geq 1$ the same as the original ones without gradient variations, it is apparent that the information accessible to node $j$ keeps unchanged for iterations $k \geq 1$. Following a similar argument, one can verify that the parameter setting in (21) can ensure $\hat{\mathcal{I}}_j = \mathcal{I}_j$ in the case $m \in \mathcal{N}_i^{\text{in}}$.

In summary, we have $\hat{\mathcal{I}}_j = \mathcal{I}_j$ for $\nabla \tilde{f}_i(\cdot) = \nabla f_i(\cdot) + \delta$ under any $\delta \in \mathbb{R}^d$, which means that node $j$ cannot distinguish arbitrarily large variations on $\nabla f_i(\cdot)$ based on its accessible information. Therefore, our proposed algorithm can prevent the privacy of node $i$ if $|\mathcal{N}_i^{\text{out}} \cup \mathcal{N}_i^{\text{in}}| \geq 2$ holds.

Remark 3 Note that if we view the feasible parameters as solutions to guaranteeing $\hat{\mathcal{I}}_j = \mathcal{I}_j$, then there exist infinitely many solutions. The proof of Theorem 2 just provides one such solution.

Next we show that if the condition in Theorem 2 is not met, then the privacy of node $i$ can be breached.

Theorem 3 In our decentralized optimization algorithm, the privacy of node $i$ cannot be preserved against node $j$ if

$$
\nabla f_m(\cdot) - \delta,
$$

and the following parameters

$$
\begin{align*}
\hat{A}_p^0 &= A_p^0 \quad \forall p \in \mathcal{V} \setminus \{i, m\} \\
\hat{\lambda}_i^0(l) &= \lambda_i^0(l) y_i^0[l] / (y_i^0[l] + \delta_l) \\
\hat{\lambda}_m^0(l) &= \lambda_m^0(l) y_m^0[l] / (y_m^0[l] - \delta_l) \\
\hat{R}_{pq}^0 &= R_{pq}^0 \quad \forall p, q \in \mathcal{V} \\
\hat{A}_{pq}^0 &= A_{pq}^0 \quad \forall p, q \in \mathcal{V} \\
\hat{C}_{pq}^0 &= C_{pq}^0 \quad \forall p, q \in \mathcal{V} \setminus \{i, m\} \\
\hat{c}_{pq}^0(l) &= c_{pq}^0(l) y_i^0[l] / (y_i^0[l] + \delta_l) \quad \forall p \in \mathcal{V} \setminus \{m\} \\
\hat{c}_{rm}^0(l) &= (c_{rm}^0(l) y_i^0[l] + \delta_l) / (y_i^0[l] + \delta_l) \\
\hat{c}_{pm}^0(l) &= (c_{pm}^0(l) y_m^0[l] - \delta_l) / (y_m^0[l] - \delta_l) \\
\hat{c}_{cm}^0(l) &= (c_{cm}^0(l) y_m^0[l] - \delta_l) / (y_m^0[l] - \delta_l) \\
\hat{C}_{pq}^0 &= C_{pq}^0 \quad \forall p, q \in \mathcal{V} \\
\hat{B}_{pq}^0 &= B_{pq}^0 \quad \forall p, q \in \mathcal{V} \\
\hat{A}_p^k &= A_p^k \quad \forall p \in \mathcal{V}, k = 1, 2, \ldots \\
\hat{R}_{pq}^k &= R_{pq}^k \quad \forall p, q \in \mathcal{V}, k = 1, 2, \ldots \\
\hat{A}_{pq}^k &= A_{pq}^k \quad \forall p, q \in \mathcal{V}, k = 1, 2, \ldots \\
\hat{C}_{pq}^k &= C_{pq}^k \quad \forall p, q \in \mathcal{V}, k = 1, 2, \ldots \\
\hat{B}_{pq}^k &= B_{pq}^k \quad \forall p, q \in \mathcal{V}, k = 1, 2, \ldots \\

\end{align*}
$$

where $l = 1, \ldots, d$. 

(21)
node $j$ is the only in-neighbor and out-neighbor of node $i$, i.e., $N^\text{in}_i = N^\text{out}_i = \{j\}$.

Proof: When $N^\text{out}_i = N^\text{in}_i = \{j\}$ holds, one can get the dynamics of $y_i^k$ from (3) as follows:

$$y_i^{k+1} = C_{ii}y_i^k + B_{ii}^k(\nabla f_i^{k+1} - \nabla f_i^k) + C_{ij}y_j^k + B_{ij}^k(\nabla f_j^{k+1} - \nabla f_j^k)$$

(22)

According to the parameter design in Table II, we have $y_i^k = C_{ii}y_i^k + C_{ij}y_j^k$ and $\nabla f_i^{k+1} - \nabla f_i^k = (B_{ii}^k + B_{ij}^k)(\nabla f_j^{k+1} - \nabla f_j^k)$ for $k \in \mathbb{Z}_{\geq 0}$ based on the facts $C_{ii} + C_{ij} = I_d$ and $B_{ii}^k + B_{ij}^k = I_d$. So we can rewrite (22) as

$$y_i^{k+1} = y_i^k - C_{ii}y_i^k + C_{ij}y_j^k - B_{ii}^k(\nabla f_i^{k+1} - \nabla f_i^k) + B_{ij}^k(\nabla f_j^{k+1} - \nabla f_j^k)$$

(23)

Denote $m_j^k$ as

$$m_j^k = -C_{ij}y_i^k + C_{jj}y_j^k - B_{ij}^k(\nabla f_i^{k+1} - \nabla f_i^k) + B_{jj}^k(\nabla f_j^{k+1} - \nabla f_j^k)$$

(24)

Note that $m_j^k$ is accessible to the honest-but-curious node $j$ because $C_{ij}y_j^k + B_{jj}^k(\nabla f_j^{k+1} - \nabla f_j^k)$ is the information node $i$ sends to node $j$, and $C_{ij}y_j^k + B_{jj}^k(\nabla f_j^{k+1} - \nabla f_j^k)$ is the information computed by node $j$. Plugging (24) into (23), node $j$ can obtain $y_i^{k+1} - y_i^k = \nabla f_i^{k+1} - \nabla f_i^k + m_j^k$ and further (note $y_i^0 = \nabla f_i^0$)

$$y_i^{k+1} = \nabla f_i^{k+1} + \sum_{l=0}^{k} m_j^l$$

(25)

Since $\lim_{k \to \infty} y_i^{k+1} = 0_d$ holds as $k$ goes to infinity, we have $\lim_{k \to \infty} \nabla f_i^{k+1} = \lim_{k \to \infty} \nabla f_i(x_i^{k+1}) = -\lim_{k \to \infty} \sum_{l=0}^{k} m_j^l$. Given $\lim_{k \to \infty} x_i^{k+1} = \lim_{k \to \infty} x_i^k = x^*$, node $j$ also knows $\lim_{k \to \infty} x_j^{k+1}$, meaning that an honest-but-curious node $j$ can infer the gradient of node $i$ at the global optimal solution $x^*$ using (25). Therefore, the privacy of node $i$ cannot be preserved against node $j$ when node $j$ is the only in-neighbor and out-neighbor of node $i$.

4.3.2 Colluding case

When a set of honest-but-curious nodes $\mathcal{A}$ collude, they can share information with each other to infer node $i$’s gradient function.

Theorem 4 In our decentralized optimization algorithm, the privacy of node $i$ can be preserved against a set of honest-but-curious nodes $\mathcal{A}$ if $(N^\text{out}_i \cup N^\text{in}_i) \not\subset A$ holds, i.e., there exists at least one node that belongs to $N^\text{out}_i \cup N^\text{in}_i$ but not $\mathcal{A}$.

Proof: To show that the privacy of node $i$ can be preserved, we have to show that no honest-but-curious node $j \in \mathcal{A}$ can distinguish arbitrary variations on $\nabla f_i(\cdot)$. In the colluding case, each node in $\mathcal{A}$ has access to the information accessible to any node in $\mathcal{A}$. So we represent the set of accessible information as

$$\mathcal{I}_{\mathcal{A}} = \{I_j \mid \forall j \in \mathcal{A}\}$$

(26)

where $\mathcal{I}_j$ is given by (17). Following the same line of reasoning in Theorem 2, to prove that node $j \in \mathcal{A}$ cannot distinguish arbitrary variations on $\nabla f_i(\cdot)$, it suffices to prove when $\nabla f_i(\cdot)$ is altered to $\nabla f_i(\cdot) + \delta$ and $\nabla f_m(\cdot) = \nabla f_m(\cdot) - \delta$, there exist respective feasible parameters in (20) (for the case $m \in N^\text{out}_i$) and (21) (for the case $m \in N^\text{in}_i$) that satisfy the requirements in Table II and make $\mathcal{I}_{A} = \mathcal{I}_{A}$ hold under any $\delta \in \mathbb{R}^d$. Therefore, our proposed algorithm can preserve the privacy of node $i$ against the set of honest-but-curious nodes $\mathcal{A}$ if $(N^\text{out}_i \cup N^\text{in}_i) \not\subset \mathcal{A}$ holds.

Next we show that if the condition in Theorem 4 is not met, then the privacy of node $i$ can be breached.

Theorem 5 In our decentralized optimization algorithm, the privacy of node $i$ cannot be preserved when all the in-neighbors and out-neighbors belong to $\mathcal{A}$, i.e., $(N^\text{out}_i \cup N^\text{in}_i) \subset \mathcal{A}$.

Proof: Following a similar line of reasoning for Theorem 3, Theorem 5 can be easily obtained and hence we omit the proof here.

Remark 4 It is worth noting that even using time-varying parameters, the AB algorithm cannot guarantee the privacy defined in Definition 5. This is because the AB algorithm does not allow negative coupling weights, leading to no feasible solutions for the parameters required in (20) and (21), which are necessary to make adversaries’ accessible information unchanged under arbitrary variations on the gradient.

Remark 5 From Theorem 1, one can see that the parameter $K$ does not affect optimization accuracy. Furthermore, from (20) and (21), we can see that only changing the coupling weights/stepsizes in the initial iteration $k = 0$ is enough to cover gradient variations. In other words, any $K \geq 1$ is sufficient to protect the defined privacy for gradients. However, a larger $K$ does provide additional privacy protection to more intermediate states, if the privacy of intermediate states is of concern. This is because each node $i$ sends $N^\text{in}_i y_i^k$ and $C_{ii} y_i^k + B_{ii}^k(\nabla f_i(x_i^{k+1})$ to its out-neighbors.
In the first $K$ iterations, as randomness is injected into the coupling weights ($C^{\|}_{i}$ and $B^{\|}_{i}$) and stepsizes ($\Lambda^{\|}$), the intermediate states $\tilde{y}^{k}_{i}$ and $\nabla f_{i}(x^{k+1}) - \nabla f_{i}(x^{k})$ are uninferrable by receiving nodes (note that in our algorithm $x_{i}^{0}$ is randomly initialized in $\mathbb{R}^{d}$ and hence does not carry sensitive information). Therefore, if intermediate states in the first $M$ iterations need to be protected, we can set $K$ to $M$ to achieve this goal without compromising accuracy (note that the gradient will always be protected for any $K \geq 1$).

It is also worth noting that although the randomness added in the first $K$ iterations does not affect the convergence rate (as proven in Theorem 1), it does delay the convergence since the algorithm only starts to converge after iteration $K$ (see Fig. 4). Of course, if the privacy of intermediate states is not of concern and only gradients need to be protected, it is preferable to set $K = 1$ to minimize the delay in the convergence process.

**Remark 6** In the case where an adversary has side information such as a range of coefficients of the objective function, we cannot prove the adversary’s inability to distinguish $\nabla f_{i}(x_{i})$ from $\nabla \tilde{f}_{i}(x_{i}) = \nabla f_{i}(x_{i}) + \delta$ under an arbitrary $\delta$. Although it is difficult to quantify the influence of side information (e.g., a certain range of parameters) on enabled privacy due to possible complicated ways that the side information can affect the gradient, we can show that in specific cases, such as in the rendezvous problem, the adversary’s side information on parameter range could not be tightened after running the algorithm. For the sake of simple exposition, we consider the one-dimensional rendezvous problem. Each node $i$ has a local objective function $f_{i}(x_{i}) = \frac{1}{2}(x_{i} - p_{i})^{2}$. The gradient function of node $i$ is $\nabla f_{i}(x_{i}) = x_{i} - p_{i}$. Following our analysis in Theorem 2 and Theorem 4, the privacy of node $i$ can be preserved if it has an in-neighbor or out-neighbor not colluding with the adversarial node, i.e., the adversary cannot distinguish whether the gradient function of node $i$ is $\nabla f_{i}(x_{i}) = x_{i} - p_{i}$ or $\nabla \tilde{f}_{i}(x_{i}) = x_{i} - p_{i} + \delta = x_{i} - \tilde{p}_{i} (\tilde{p}_{i} \neq p_{i} - \delta)$ under any $\delta \in \mathbb{R}$. If the adversary has side information of the function type (i.e., quadratic) and parameter range $p_{i} \in [a, b]$, then the range of $\delta$ has to be reduced to $[p_{i} - b, p_{i} - a]$. This is because otherwise $\tilde{p}_{i}$ in $\nabla \tilde{f}_{i}(x_{i})$ will not be within the initial range $[a, b]$ known to the adversary. However, for any $\delta \in [p_{i} - b, p_{i} - a]$ (i.e., $\tilde{p}_{i} \in [a, b]$), following our analysis in Theorem 2 and Theorem 4, there always exist feasible parameters making all information known to the adversarial node under $\nabla f_{i}(x_{i})$ exactly the same as under $\nabla \tilde{f}_{i}(x_{i})$. Therefore, the adversarial node cannot distinguish $\nabla f_{i}(x_{i})$ from $\nabla \tilde{f}_{i}(x_{i})$ for any $\delta \in [p_{i} - b, p_{i} - a]$, meaning that the adversarial node cannot get a tighter range of $p_{i}$ than the initial knowledge (i.e., $[a, b]$) after running the algorithm.

**Remark 7** Our approach can be extended to enable privacy against adversaries wiretapping all communication links without compromising algorithmic accuracy by patching partially homomorphic encryption. For example, in our prior work [50] and [52], we can let each node generate and flood its public key before the optimization iteration starts.

**5 Numerical Simulations**

**5.1 Privacy Protection in the Rendezvous Problem**

First let’s consider the distributed rendezvous problem where a group of nodes want to agree on the nearest meeting point without revealing each other’s initial position [15]. Mathematically this can be modeled as the problem $\min_{x \in \mathbb{R}^{d}} F(x) = \sum_{i=1}^{n} f_{i}(x) = \sum_{i=1}^{n} \frac{1}{2} \|x - p_{i}\|^{2}$, where $p_{i}$ represents the initial position of node $i$.

For the simplicity of exposition, we consider the $d = 1$ case but similar results can be obtained when $d \neq 1$. We consider three nodes connected in a directed cycle as shown in Fig. 1 (a). Let node 3 be an honest-but-curious node which collects received data in an attempt to learn the gradient function of node 1. Node 2 does not collude with node 3. Parameter $K$ was set to 3. In the simulation, we first ran our algorithm and recorded $\tilde{I}_{3}$, the information accessible to node 3 (cf. (17)). Then we show that information accessible to node 3 can be exactly the same under a completely different gradient function $\nabla \tilde{f}_{1}(x_{1})$.

We represent the information accessible to node 3 in the new implementation as $\tilde{I}_{3}$. Fig. 2 shows $x^{3}_{1}, A^{\|}_{1} y^{1}_{1}$, and $C^{\|}_{11} y^{1}_{1} + B^{\|}_{11}(\nabla f^{k+1}_{1} - \nabla f^{k}_{1})$ in $\tilde{I}_{3}$ and $x^{3}_{2}, A^{\|}_{2} y^{1}_{2}$, and $C^{\|}_{32} y^{1}_{2} + B^{\|}_{32}(\nabla f^{k+1}_{1} - \nabla f^{k}_{1})$ in $\tilde{I}_{3}$, respectively. It can be seen that the trajectory of the observations of node 3 in both cases are identical. Fig. 3 shows the corresponding $\nabla f_{j}(x_{1})$ and $\nabla \tilde{f}_{j}(x_{1})$, which are clearly different. Since node 3 receives the same information under $\nabla \tilde{f}_{1}(x_{1}) \neq \nabla f_{1}(x_{1})$, it has no way to infer the real gradient function of node 1.

**5.2 Distributed Estimation Problem**

In this subsection we focus on the convergence performance of our algorithm and compare our algorithm with other decentralized optimization algorithms. We consider the canonical distributed estimation problem $\min_{x \in \mathbb{R}^{d}} F(x) = \sum_{i=1}^{n} (\|z_{i} - Q_{i} x\|^{2} + \sigma_{i} \|x\|^{2})$ in [47], where $n$ nodes cooperatively measure a certain unknown
We compared our algorithm with DIGing [25], ADD-OPT [44], and Subgradient-Push [27] to evaluate the influence of privacy design on optimization performance. The stepsize was set to 0.06 for all of the considered algorithms except Subgradient-Push whose stepsize was set to a diminishing sequence $\lambda^k = 1/k$. The simulation results on optimization error $\|x^k - 1_n \otimes x^*\|$ are depicted in Fig. 4.

From Fig. 4, it can be seen that all algorithms with a fixed stepsize had $R$-linear convergence, including our algorithm with guaranteed privacy. This corroborates our statement in Remark 2 that the added randomness has no influence on the $R$-linear convergence of our algorithm. Of course, it is worth noting that in our algorithm, the privacy-induced randomness delayed convergence to iteration step $k \geq 4$.

5.2.2 Comparison with [15]

We also compared our algorithm with the differential privacy based privacy-preserving approach for decentralized optimization proposed in [15]. We ran the algorithm in [15] under four different privacy levels, i.e., $\epsilon = 0.1, 1, 10, 100$, respectively. The domain of optimization in [15] was set to $\mathcal{X} = \{x \in \mathbb{R}^2 \mid \|x\| \leq 10\}$. Note that the optimal solution $x^* = \{0.6881, 0.5103\}^T$ resides in $\mathcal{X}$. For each privacy level $\epsilon$, we repeated the simulation for 1,000 times, and averaged the optimization error trajectories. Under a stepsize 0.06, we also measured the mean optimization error of our algorithm over 1,000 repetitions. The comparison results in Fig. 5 confirm the trade-off between privacy and accuracy for differential-privacy based approaches and demonstrate the advantage of our algorithm in ensuring optimization accuracy.

5.2.3 Comparison with [19]

Then we compared our algorithm with the privacy-preserving optimization algorithm in [19]. The closed convex projection set $\mathcal{X}$ was set to $\mathcal{X} = \{x \in \mathbb{R}^2 \mid \|x\| \leq 10\}$ for the algorithm in [19]. The other parameters for the algorithm in [19] were set as follows: each node $i$ randomly chose $c_i$ from $(0, 5)$, and updated its state with its subgradient once every $T_i$ iterations where $T_i$ was randomly chosen from $\{1, 2, 3, 4, 5\}$. The stepsize of our algorithm was set to 0.06. The results are shown in Fig. 6. Clearly our algorithm...
Finally, we conducted numerical simulations to verify the convergence properties of our algorithm and its privacy-preserving performance. In addition, to facilitate the privacy design, we also proposed a general framework of gradient-tracking based decentralization optimization, which includes many commonly used algorithms as special cases. Finally, we provided numerical simulation results to confirm the effectiveness and efficiency of our proposed algorithm.

5.2.4 Numerical Simulations on Large-scale Networks

Fig. 6. Comparison of optimization error $\|x^k - 1_n \otimes x^*\|$ between our algorithm and the algorithm in [19].

Fig. 7. The optimization error trajectory of our algorithm in a network of $n = 100$ nodes.

Fig. 5. Comparison of mean optimization error over 1,000 repetitions between our algorithm and the differential-privacy based approach in [15].

Appendix A: Absolute Probability Sequence

Definition 6 [38] Let $\{A^k\}$ be a sequence of row-stochastic matrices. A sequence of vectors $\{\pi^k\}$ is an absolute probability sequence for $\{A^k\}$ if $(\pi^k)^T = (\pi^{k+1})^T A^k$ holds for $k \geq 0$.

Definition 7 [40] A sequence of row-stochastic matrices $\{A^k\}$ is ergodic if the following limit exists for $k \geq 0$

$$\lim_{t\to\infty} A^tA^{t-1} \cdots A^{k+1} A^k = 1(\pi^k)^T$$

where $\{\pi^k\}$ is an absolute probability sequence for $\{A^k\}$.

For the absolute probability sequences $\{v^k\}$ and $\{q^k\}$ (associated with $\{\tilde{P}^k\}$ and $\{\tilde{R}^k\}$, respectively) in Sec. 4.2, we have the following lemmas.

Lemma 4 Under Assumptions 1 and 2, and the parameter design in Table II, the sequence of matrices $\{\tilde{P}^k\}$ for $k \geq K$ is row-stochastic and ergodic, and the sequence $\{v^k\}$ is the unique absolute probability sequence for $\{\tilde{P}^k\}$. Denote $Q_P$ as $Q_P = 2n(1 + (n\eta - n)^{n-1})/(1 - (n^{-1}\eta^n)^{n-1})$ and define a positive integer $N_P$ such that $r_P \triangleq Q_P(1 - (n^{-1}\eta^n)^{n-1})^{N_P-1} < 1$ holds. Then for any $k \geq K + N_P - 1$ and any vector $b \in \mathbb{R}^d$, vector $a \triangleq \tilde{P}^{k+1} \cdots \tilde{P}^{k-N_P+1} b$ always satisfies $\|\tilde{L}_i^n - 1_n(v^{k+1})^T \otimes I_d a\| \leq r_P \|\tilde{L}_i^n - 1_n(v^{k-N_P+1})^T \otimes I_d b\|$.  

Proof: Given $\tilde{P}^k = (\tilde{V}^{k+1} - 1_n \tilde{C} \tilde{V}^k)^T$, it can be verified that $\tilde{P}^k$ is row-stochastic since $\sum_{j=1}^n P_{ij}^k = 1$ holds for...
all $i = 1, \ldots, n$. Using Lemma 4 in [26], we can see that the sequence of matrices $\{\bar{P}^k\}$ for $k \geq K$ is ergodic since the limit in (27) exists under Assumptions 1 and 2, and the parameter design in Table II. Combining $\bar{V}^k = \text{diag}(v^k)$ and $\bar{P}^k = (\bar{V}^{k+1})^{-1} \bar{C}^k \bar{V}^k$ leads to $v^k$ is the absolute probability sequence for $\{\bar{P}^k\}$. Further using Lemma 1 in [23], we have that $\{v^k\}$ is the unique absolute probability sequence for $\{\bar{P}^k\}$ due to the ergodicity of $\{\bar{P}^k\}$.

Now we show that the positive entries of $\bar{P}^k$ has a lower bound $\eta^n/n$, i.e., $p_{ij}^k \geq \eta^n/n$ for any $j \in N_i^{\text{out}} \cup \{i\}$ where $p_{ij}^k$ represents the $ij$-th entry of $\bar{P}^k$. To this end, we first prove that all entries of $v^k$ have a lower bound $\eta^n/n$, i.e., $v_i^k \geq \eta^n/n$ for $i = 1, \ldots, n$ and $k \geq K$. Denoting $\bar{C}_{k,K} = \bar{C}^k \cdots \bar{C}^{K}$, from (9) we have

$$v^k = \bar{C}_{k-1,K} v^k = \bar{C}_{k-1,K} 1/n \quad (28)$$

for $k \geq K$. Then we divide the proof into two parts: $K \leq k \leq K + n - 1$ and $k \geq K + n$, respectively.

**Part 1:** $v_i^k \geq \eta^n/n$ for $K \leq k \leq K + n - 1$. Given $\bar{C}_{k,1} = \bar{C}_k$, one can verify that the inequality $v_i^k \geq \eta^n/n$ holds for $i = 1, \ldots, n$ and $k \leq K + n - 1$.

**Part 2:** $v_i^k \geq \eta^n/n$ for $k \geq K + n$. Under Assumptions 1 and 2, and the parameter design in Table II, following the arguments in Lemma 2 in [26], we can obtain $\bar{C}_{k-1,k+1}$ is column-stochastic, $\bar{C}_k$ is also column-stochastic. Further using the fact $\bar{C}_{k-1,K} \subseteq \bar{C}_{k-1,k+1} \bar{C}_{k-1,n}$, and yields $\bar{C}_{k-1,K}$ is column-stochastic. Since $\bar{C}_k$ is column-stochastic, $\bar{C}_{k-1,n}$ is column-stochastic. Therefore, $\bar{C}_{k-1,k}$ is column-stochastic, it follows naturally from (28) that $1/n \bar{v}_i^k = \sum_{j=1}^n \bar{C}_{j-1,k} v_j^k \geq \eta^n/n$ for $i = 1, \ldots, n$ and $k \geq K$. In summary, we have $v_i^k \geq \eta^n/n$ for $i = 1, \ldots, n$ and $k \geq K$. Since $\bar{C}_{k-1,K}$ is column-stochastic, it follows naturally from (28) that $1/n \bar{v}_i^k = \sum_{j=1}^n \bar{C}_{j-1,k} v_j^k \geq 1$ holds for $k \geq K$, which implies $v_i^k \in [\eta^n/n, 1]$ for $i = 1, \ldots, n$ and $k \geq K$.

Given $\tilde{P}^k = (\bar{V}^{k+1})^{-1} \bar{C}^k \bar{V}^k$, we have $\theta_{ij}^k = c_{ij}^k \phi_i^k (\phi_j^k)^+ / v_i^{k+1}$. Further using $\eta^n/n$ for $j \in N_i^{\text{out}} \cup \{i\}$ and $p_{ij}^k = 0$ for $j \notin N_i^{\text{out}} \cup \{i\}$.

Denote $P_{t,k}$ as $P_{t,k} = P^t \bar{P}^{t-1} \cdots \bar{P}^k$, one can obtain

$$[(I_n - 1_n(v^k)^+) \otimes I_d] P_{t,k} = ((I_n - 1_n(v^k)^+) (I_n - 1_n(v^{k-N}^+) \otimes I_d)$$

for any $k \geq K + n - 1$ and any vector $b \in \mathbb{R}^d$.

**Lemma 5** Under Assumptions 1 and 2, and the parameter design in Table II, the sequence of row-stochastic matrices $\{\bar{R}^k\}$ for $k \geq K$ is ergodic, and $\{\phi^k\}$ is the unique absolute probability sequence for $\{\bar{R}^k\}$. Denote $Q_R = 2n(1 + \eta^{(n-1)}) / (1 - \eta^{n-1})$. Then for any $k \geq K + n - 1$ and any vector $b \in \mathbb{R}^d$, vector $\theta \triangleq \bar{R}^k \bar{R}^{k-1} \cdots \bar{R}^{k-n+1}$ always satisfies $\|((I_n - 1_n(\phi^{k+n}) \otimes I_d) b \| \leq (KR_{\text{max}})^n \|I_n - 1_n(\phi^{k+n}) \otimes I_d\| \|b\|$. 

**Proof:** Following a similar line of reasoning for Lemma 4, Lemma 5 can be easily obtained and hence we omit the proof here.

**Appendix B: Proof of Lemma 1**

**Proof:** From (8), we have $x^{k+1} = R_{k-k-N+1}x^{k-N+1} - \lambda(A^k x^k + \sum_{l=1}^{N-1} R_{k-k-l} A^{k-l} y^{k-l})$ for $k \geq K + N - 1$ where $R_{k-k-N+1} = R^{k} R^{k-1} \cdots R^{k-N+1}$. Then, it follows

$$\|\bar{x}^{k+N-1}\| \leq \|((I_n - 1_n(\phi^{k+n}) \otimes I_d) R_{k-k-N+1} x^{k-N+1}\|$$

$$+ \lambda \cdot \|((I_n - 1_n(\phi^{k+n}) \otimes I_d) A^{k+N} y^{k-N-1}\|$$

$$+ \lambda \cdot \sum_{l=1}^{N-1} \|((I_n - 1_n(\phi^{k+n}) \otimes I_d) R_{k-k-l} A^{k-l} y^{k-l}\|$$

$$\leq \lambda R_k \|\bar{x}^{k+N-1}\|$$

where in the derivation we used Lemma 5 and the facts $\|I_n - 1_n(\phi^{k+n}) \| \leq 2\sqrt{n}$, $\|A^k\| \leq \sqrt{n}$, and $2\sqrt{n} \leq Q_R$. Following Lemma 1 in [38], we can obtain

$$\|y^{k}\| \leq \eta K \|\bar{x}^{k}\| + n \beta \|x^k\| + \|\bar{x}^{k+N-1}\|$$

for $k \geq N$. Further combing (29) and (30) leads to the result in (11) for $k \geq K + N - 1$.

**Appendix C: Proof of Lemma 2**

**Proof:** Given $v^k = 1_n \otimes \bar{x}_w^k - 1_n \otimes x^* = 1_n(\phi^k)^T \otimes \phi^k \otimes 1_n(z^w)^T - 1_n \otimes x^*$
where \( \delta^k \) is the \( (\phi^k+1)^T \vec{A}^k v^k \), and in the derivation we used the facts |
\( \vec{I}_n \) and \( \lambda \) are parallel, so we have that the inequality in (12) holds for \( \phi^k \) such that
\( \frac{\lambda}{2} \lambda^k \leq \sqrt{n/\eta} \).

Now we show that \( \delta^k \in [\eta^{-1}n, 1] \). From the proof of Lemma 4 in Appendix A, we have \( \lambda \phi_{i,j} \leq \eta^{-1}n \) for \( k \geq \overline{K} \) where \( v_i \in [\eta^{-1}n, 1] \) for \( k \geq \overline{K} \) and the norm of \( v_i \) denotes the \( i \)-th entry of vector \( v^i \). Given \( \sum_{j=1}^{n} a_{i,j} \leq 1 \) for \( k \geq \overline{K} \), we can obtain \( \eta^{-1}n \leq \sum_{j=1}^{n} \alpha_j v_{i,j} \leq 1 \) for each \( i = 1, \ldots, n \). Therefore, we have
\( \delta^k = \sum_{j=1}^{n} \phi_{i,j} \sum_{j=1}^{n} a_{i,j} v_{i,j} \in [\eta^{-1}n, 1] \) based on the fact \( \sum_{j=1}^{n} \phi_{i,j} \leq 1 \).

Next we focus on the second last term on the right hand side of (31), which can be expressed as follows
\[
\left\| \left[ \lambda (\phi^k)^T \vec{I}_n \right] \vec{x}^k - 1_n \otimes \vec{x}^* - \lambda \delta^k (1_n \otimes \vec{x}_w) \right\| \leq \left( \sum_{i=1}^{n} \beta_i^2 \left\| (\vec{x}^k)^T - \vec{x}^* \right\|^2 \right)^{1/2} \leq \beta \left\| \vec{x}^k - \vec{x}^* \right\| 
\]

As the global objective function \( F \) is \( \alpha_F \)-strongly convex and \( \beta_F \)-smooth with \( \alpha_F \leq \beta_F \), from Lemma 10 in [33], we obtain
\[
\left\| \vec{x}_w - \vec{x}^* - \lambda \delta \nabla F(\vec{x}_w) \right\| \leq \frac{1}{\sqrt{n}} \left( 1 - \lambda n^{-1} \eta^{-1} \alpha_F \right) \left\| \vec{r}^k \right\| 
\]

if \( \lambda \leq 1/\beta_F \). Therefore, combining (31), (32), and (33), we have that the inequality in (12) holds for \( k \geq \overline{K} \) if \( \lambda \) satisfies \( \lambda \leq 1/\beta_F \).

Appendix D: Proof of Lemma 3

**Proof:** Denoting \( z^k \) as \( z^k = \nabla f^{k+1} - \nabla f^k \), from (10) we have \( z^k = P_{k,k-N+1} z^{k-N+1} + ((\vec{v}^k)^T \vec{I}_d) \vec{B}_N z^k + \sum_{l=1}^{N-1} P_{k,k-l+1} ((\vec{v}^k)^T \vec{I}_d) \vec{B}_l z^l \) for \( k \geq \overline{K} + N - 1 \). Given \( \vec{s}_w = \phi_{i,j} \) and \( \vec{s}_w = (1_n \otimes \vec{x}_w)^T \vec{I}_d \vec{s}_w \), one can obtain the following result for \( k \geq \overline{K} + N - 1 \)
\[
\left\| z^{k+1} \right\| \leq r_P \left\| (\vec{I}_n - 1_n (\vec{v}^{k+1-N+1} \vec{B}_N) S^{k-N+1} \right\| 
\]

where we used the facts \( \left\| \vec{I}_n - 1_n (\vec{v}^{k+1-N+1}) \vec{B}_N \right\| \leq 2 \sqrt{n} \leq Q_P \), \( \left\| (\vec{v}^{k-N}) \vec{B}_N \right\| \leq n/\eta \), \( \left\| \vec{B}_N \right\| \leq \sqrt{n} \), and Lemma 4 in Appendix A.

Using the relationship \( z^k = \nabla f^{k+1} - \nabla f^k \), we can obtain the following relationship for \( k \geq \overline{K} \)
\[
\left\| z^k \right\| \leq \left( \sum_{i=1}^{n} \beta_i^2 \left\| (\vec{x}^{k+1-N+1} - \vec{x}_w)^T \vec{I}_d \right\|^2 \right)^{1/2} \leq \beta \left\| (\vec{I}_n - 1_n) \vec{x}^k \right\| + \lambda \beta \left\| \vec{A}^k \right\| 
\]

\[
\leq \beta \left\| (\vec{I}_n - 1_n) \vec{x}^k \right\| + \lambda \beta \left\| \vec{A}^k \right\| 
\]

\[
\leq (2 \sqrt{n} \beta + \lambda n \sqrt{n} \beta^2) \left\| \vec{x}_w \right\| + \lambda \sqrt{n} \beta^2 \left\| \vec{r}^k \right\| + \lambda \sqrt{n} \beta \left\| \vec{s}_w \right\| 
\]

where we used the facts \( \left\| \vec{I}_n - 1_n (\vec{v}^{k+1-N+1}) \vec{B}_N \right\| \leq 2 \sqrt{n} \leq Q_P \), \( \left\| (\vec{v}^{k-N}) \vec{B}_N \right\| \leq n/\eta \), \( \left\| \vec{B}_N \right\| \leq \sqrt{n} \), and Lemma 4 in Appendix A.

Therefore, from (34) and (35), we can obtain the inequality in (13) for \( k \geq \overline{K} + N - 1 \).
[14] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

[15] Z. Huang, S. Mitra, and N. Vaidya. Differentially private distributed optimization. In *Proc. Int. Conf. Distrib. Comput. Netw.*, pages 4:1–4:10, 2015.

[16] S. S. Kia, J. Cortés, and S. Martinez. Dynamic average consensus under limited control authority and privacy requirements. *Int. J. Robust Nonlinear Control*, 25(13):1941–1966, 2015.

[17] D. Lefebvre and C. N. Hadjicostis. Privacy and safety analysis of timed stochastic discrete event systems using markovian trajectory-observers. *Discret. Event Dyn. Syst.*, pages 1–28, 2020.

[18] Q. Li, R. Heusdens, and M. G. Christensen. Privacy-preserving distributed optimization via subspace perturbation: a general framework. *IEEE Trans. Signal Process.*, 68:5983–5996, 2020.

[19] Y. Lou, L. Yu, S. Wang, and P. Yi. Privacy preservation in distributed subgradient optimization algorithms. *IEEE Trans. Cybern.*, 48(7):2154–2165, 2018.

[20] Y. Lu and M. Zhu. Privacy preserving distributed optimization using homomorphic encryption. *Automatica*, 96:314–325, 2018.

[21] N. E. Manitara and C. N. Hadjicostis. Privacy-preserving asymptotic average consensus. In *2013 Eur. Control Conf.*, pages 760–765, 2013.

[22] Y. Mo and R. M. Murray. Privacy preserving average consensus. *IEEE Trans. Autom. Control*, 62(2):753–765, 2017.

[23] A. Nedić and J. Liu. On convergence rate of weighted-averaging dynamics for consensus problems. *IEEE Trans. Autom. Control*, 62(2):766–781, 2016.

[24] A. Nedić, A. Olshevsky, and W. Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM J. Optim.*, 27(4):2597–2633, 2017.

[25] A. Nedić, A. Olshevsky, W. Shi, and C. A. Uribe. Geometrically convergent distributed optimization with uncoordinated step-sizes. In *Proc. 2017 American Contr. Conf.*, pages 3950–3955, 2017.

[26] A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control*, 54(1):48–61, 2009.

[27] A. Nedić and A. Ozdaglar. Distributed optimization over time-varying directed graphs. *IEEE Trans. Autom. Control*, 60(3):601–615, 2015.

[28] E. Nozari, P. Tallapragada, and J. Cortés. Differentially private distributed convex optimization via functional perturbation. *IEEE Trans. Control Netw. Syst.*, 5(1):395–408, 2016.

[29] S. Pequito, S. Kar, S. Sundaram, and A. P. Aguiar. Design of communication networks for distributed computation with privacy guarantees. In *Proc. IEEE 53rd Conf. Decis. Control*, pages 1370–1376, 2014.

[30] A. B. Pilet, D. Frey, and F. Taiani. Robust privacy-preserving gossip averaging. In *Proc. 21st Int. Symp. Stabilization, Saf., Secure. Distrib. Syst.*, pages 38–52. Springer, 2019.

[31] P. D. Powell. Calculating determinants of block matrices. *arXiv preprint arXiv:1112.4379*, 2011.

[32] S. Pu, W. Shi, J. Xu, and A. Nedić. A push-pull gradient method for distributed optimization in networks. In *Proc. IEEE 57th Conf. Decis. Control*, pages 3385–3390, 2018.

[33] G. Qu and N. Li. Harnessing smoothness to accelerate distributed optimization. *IEEE Trans. Control Netw. Syst.*, 5(3):1245–1260, 2017.

[34] R. L. Raffard, C. J. Tomlin, and S. P. Boyd. Distributed optimization for cooperative agents: Application to formation flight. In *Proc. IEEE 43rd Conf. Decis. Control*, volume 3, pages 2453–2459, 2004.

[35] B. Ramasubramanian, W. R. Cleaveland, and S. Marcus. Notions of centralized and decentralized opacity in linear systems. *IEEE Trans. Autom. Control*, 65(4):1442–1455, 2019.

[36] I. D. Ridgley, R. A. Freeman, and K. M. Lynch. Simple, private, and accurate distributed averaging. In *Proc. 57th Annua. Allerton Conf. Commun. Control Comput.*, pages 446–452, 2019.

[37] M. Ruan, H. Gao, and Y. Wang. Secure and privacy-preserving consensus. *IEEE Trans. Autom. Control*, 64(10):4035–4049, 2019.

[38] F. Saadatniaki, R. Xin, and U. A. Khan. Decentralized optimization over time-varying directed graphs with row and column-stochastic matrices. *IEEE Trans. Autom. Control*, 65(11):4769–4780, 2020.

[39] A. Saboori and C. N. Hadjicostis. Verification of initial-state opacity in security applications of discrete event systems. *Inform. Sciences*, 246:115–132, 2013.

[40] E. Seneta. *Non-negative matrices and Markov chains*. Springer Science & Business Media, 2006.

[41] W. Shi, Q. Ling, G. Wu, and W. Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM J. Optim.*, 25(2):944–966, 2015.

[42] Y. Wang and A. Nedic. Tailoring gradient methods for differentially-private distributed optimization. *arXiv preprint arXiv:2202.01113*, 2022.

[43] Y. Wang and V. Poor. Decentralized stochastic optimization with inherent privacy protection. *IEEE Trans. Autom. Control*, 2022.

[44] C. Xi, R. Xin, and U. A. Khan. Add-opt: Accelerated distributed directed optimization. *IEEE Trans. Autom. Control*, 63(5):1329–1339, 2018.

[45] R. Xin and U. A. Khan. A linear algorithm for optimization over directed graphs with geometric convergence. *IEEE Control Syst. Lett.*, 2(3):315–320, 2018.

[46] J. Xu, S. Zhu, Y. C. Soh, and L. Xie. Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In *Proc. IEEE 54th Conf. Decis. Control.*, pages 2055–2060, 2015.

[47] J. Xu, S. Zhu, Y. C. Soh, and L. Xie. Convergence of asynchronous distributed gradient methods over stochastic networks. *IEEE Trans. Autom. Control*, 63(2):434–448, 2017.

[48] F. Yan, S. Sundaram, S. V. N. Vishwanathan, and Y. Qi. Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties. *IEEE Trans. Knowl. Data Eng.*, 29(11):2483–2493, 2012.

[49] K. Yuan, Q. Ling, and W. Yin. On the convergence of decentralized gradient descent. *SIAM J. Optim.*, 26(3):1835–1854, 2016.

[50] C. Zhang, M. Ahmad, and Y. Wang. Admm based privacy-preserving decentralized optimization. *IEEE Trans. Inf. Forensic Secur.*, 14(3):565–580, 2018.

[51] C. Zhang and Y. Wang. Distributed event localization via alternating direction method of multipliers. *IEEE Trans. Mob. Comput.*, 17(2):348–361, 2018.

[52] C. Zhang and Y. Wang. Enabling privacy-preservation in decentralized optimization. *IEEE Trans. Control Netw. Syst.*, 62:679–689, 2018.

[53] S. Zhang, X. Yi, J. George, and T. Yang. Computational convergence analysis of distributed optimization algorithms for directed graphs. In *Proc. IEEE 15th Int. Conf. Control Autom.*, pages 1096–1101, 2019.