Differentially Private ADMM for Regularized Consensus Optimization

Xuanyu Cao, *Senior Member, IEEE*, Junshan Zhang, *Fellow, IEEE*,
H. Vincent Poor, *Life Fellow, IEEE*, and Zhi Tian, *Fellow, IEEE*

***Abstract*—Due to its broad applicability in machine learning, resource allocation and control, the alternating direction method of multipliers (ADMM) has been extensively studied in the literature. The message exchange of ADMM in multi-agent optimization may reveal sensitive information of agents, which can be overheard by malicious attackers. This drawback hinders the application of ADMM to privacy-aware multi-agent systems. In this paper, we consider consensus optimization with regularization, in which the cost function of each agent contains private sensitive information, e.g., private data in machine learning and private usage patterns in resource allocation. We develop a variant of ADMM that can preserve agents' differential privacy by injecting noise to the public signals broadcast to the agents. We derive conditions on the magnitudes of the added noise under which the designated level of differential privacy can be achieved. Further, the convergence properties of the proposed differentially private ADMM are analyzed under the assumption that the cost functions are strongly convex with Lipschitz continuous gradients and the regularizer has smooth gradients or bounded subgradients. We find that to attain the best convergence performance given a certain privacy level, the magnitude of the injected noise should decrease as the algorithm progresses. Additionally, the choice of the number of iterations should balance the tradeoff between the convergence and the privacy leakage of ADMM, which is explicitly characterized by the derived upper bounds of convergence performance. Finally, numerical results are presented to corroborate the efficacy of the proposed algorithm. In particular, we apply the proposed algorithm to multi-agent linear-quadratic control with private information to showcase its merit in control applications.**

I. Introduction

With the development of big data analytics, vast amount of data are being collected and analyzed every day, many of which contain sensitive information of users, e.g., health data, social data, and location data. As such, privacy has become a pivotal concern for many users and lack of privacy preservation can severely compromise the reliability of the systems and services.

Since the seminal work [1] by Dwork *et al.*, the notion of differential privacy has become prevalent in privacy research due to its mathematical rigor and tractability. The Laplacian mechanism proposed in [1] has become the building block of many sophisticated private systems. Among others, multi-agent optimization theory is a field in need of privacy preservation. In the iterative optimization algorithms, the message exchanges between the agents and the central coordinator may contain some knowledge of the agents' objective/constraint functions, which are often related to sensitive private information, e.g., private data and usage patterns in distributed learning/control. Since the exchanged messages may be overheard by adversaries, agents' privacy is compromised and privacy-preserving optimization methods are imperative.

Because of its broad applicability and fast convergence, the alternating direction method of multipliers (ADMM) has pervasive usage [2] in many learning, control and resource allocation problems. In this paper, we develop and analyze a differentially private ADMM for consensus optimization with regularization under several assumptions such as strong convexity and Lipschitz continuity of gradient. The regularization term is incorporated to promote structures (e.g., sparsity) in machine learning or to stand for system cost in control and resource allocation. The cost functions of individual agents contain sensitive private information and their privacy needs to be preserved. To this end, we perturb the public signal (which may be overheard by adversaries) in each iteration of the consensus ADMM by injecting Laplacian noise. We derive conditions on the noise parameters under which the designated differential privacy can be ensured. Further, the convergence performance of the proposed differentially private ADMM is analyzed under the assumption that the cost functions are strongly convex with Lipschitz continuous gradients and the regularizer has smooth gradients or bounded subgradients. We find that to achieve the best convergence performance, the noise magnitudes should decrease as the algorithm progresses. In addition, there exists an optimal finite number of iterations that balances the tradeoff between the convergence of ADMM and the privacy leakage during the algorithm execution. Finally, numerical results are provided to verify the theoretical properties of the proposed algorithm. We apply the proposed algorithm to multi-agent linear-quadratic control with private information to demonstrate its merit in control applications. We note that the results in this paper are mostly theoretical. Real-world private optimization problems can be more complicated and other privacy-preserving techniques (e.g., encryptions) may need to be integrated with the techniques in this paper.

Related Works: The interfaces between differential privacy and other domains have been investigated extensively in the recent decade, e.g., private machine learning [3], [4] and private estimation [5], [6]. Besides differential privacy, there are various alternative privacy notions and approaches. Information-theoretic privacy includes mutual-information privacy [7] and conditional-entropy privacy [8]. Data privacy based on estimation accuracy was used in [9]. Further, partial homomorphic cryptography was leveraged for privacy-preserving consensus in [10], and a state decomposition approach was proposed in [11] for private average consensus. In this paper, we focus on differential privacy, which provides strong privacy guarantee: an adversary cannot infer the private information of an agent accurately even if he knows the private information of all other agents. A recent book [12] provided a comprehensive overview of various privacy-preserving methods (differential privacy, information-theoretic privacy and homomorphic encryption) from the perspective of systems and control, with applications to smart metering, traffic estimation and building management.

A line of research more closely related to this paper is differentially private optimization methods. In this regard, differentially private projected gradient descent was proposed in [13] with applications in electric vehicle charging. The objective function of the optimization problem in [13] only depended on the sum of all the agents' variables, and was very different from the consensus optimization problem in this paper. Additionally, the convergence bound of this paper (c.f. (22)) is better than that of [13] (c.f. (10) in [13]). Moreover, private

This research was supported by the Army Research Office under Grant W911NF-16-1-0448, and by NSF under Grants SaTC-1618768, CPS-1739344, 1704274, and 1741338.

X. Cao is with the Coordinated Science Lab, University of Illinois at Urbana-Champaign, Urbana, IL 61801. (email: xyc@illinois.edu)

J. Zhang is with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287. (email: junshan.zhang@asu.edu)

H. V. Poor is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544. (email: poor@princeton.edu)

Z. Tian is with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030. (email: ztian1@gmu.edu)

dual decomposition was presented in [14] to solve private convex programs. Specifically, the privacy guarantees in [14] were in terms of the (ϵ, δ)-differential privacy, which is a weaker notion of privacy than the ϵ-differential privacy used in this paper. Several additional uncommon assumptions beyond convexity were made in [14] to facilitate performance analysis, e.g., bounded gradient sensitivity of constraint functions, bounded width (i.e., bounded possible constraint violations), and a finite dual bound (i.e., violating the constraints by a can improve the objective function value by at most some constant times of a). In addition, differentially private distributed subgradient method was developed in [15], [16] for private online learning. One main concern of [15], [16] was that the ϵ-differential privacy was guaranteed for only one single iteration. If an adversary could keep eavesdropping for T iterations, then the algorithms in [15], [16] could only ensure ϵT-differential privacy, which was not useful for moderately large T. In contrast, in the current paper, the ϵ-differential privacy holds for the entire algorithm over all iterations.

In the literature, the convergence of ADMM has been analyzed extensively in both centralized [17] and decentralized settings [18]. ADMM has also been applied to various control problems. In [19], ADMM was used for model predictive control (MPC) and congestion control. ADMM was applied to ℓ_1 regularized MPC in [20], and was used to solve the dual problem of the distributed MPC in [21]. ADMM was also utilized for sensor/actuator selection in dynamical systems in [22] and compositional performance certification of interconnected systems in [23]. Private variants of ADMM were also studied in the recent works [24], [25] for empirical risk minimization (ERM), which is a special case of the generic consensus optimization problem considered in this paper.

II. Problem Formulation and Algorithm Development

In this section, we formulate the regularized consensus optimization problem and discuss the privacy concerns of consensus ADMM. Then, we introduce a framework of differential privacy for the optimization and propose a differentially private version of ADMM.

A. ADMM for regularized consensus optimization

We consider a multi-agent system with n agents. Each agent i has a cost function $f_i : \mathbb{R}^p \mapsto \mathbb{R}$. We study the following regularized consensus optimization problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^p}{\text{minimize}} \ \sum_{i=1}^{n} f_i(\boldsymbol{x}) + g(\boldsymbol{x}), \tag{1}$$

where $g : \mathbb{R}^p \mapsto \mathbb{R}$ is a regularization term. Problem (1) has pervasive applications in a multitude of engineering problems such as distributed machine learning and signal processing.

ADMM can be utilized to solve Problem (1) in a distributed manner, in which each agent only handles its own cost function [2]. A central coordinator (e.g., a data aggregator or a system operator) communicates with agents by broadcasting public coordination signals iteratively so that Problem (1) is solved across all agents. To apply ADMM, we rewrite Problem (1) as follows:

$$\underset{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n,\boldsymbol{z}\in\mathbb{R}^p}{\text{minimize}} \ \sum_{i=1}^{n} f_i(\boldsymbol{x}_i) + g(\boldsymbol{z}) \tag{2a}$$

$$\text{subject to} \quad \boldsymbol{x}_i = \boldsymbol{z}, \ i = 1, ..., n. \tag{2b}$$

The augmented Lagrangian of Problem (2) is

$$\begin{aligned}&\mathfrak{L}_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda})\\ &= \sum_{i=1}^{n} f_i(\boldsymbol{x}_i) + g(\boldsymbol{z}) + \sum_{i=1}^{n} \boldsymbol{\lambda}_i^{\mathsf{T}}(\boldsymbol{x}_i - \boldsymbol{z}) + \frac{\rho}{2}\sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{z}\|^2,\end{aligned} \tag{3}$$

where $\rho > 0$ is an algorithm parameter and $\|\cdot\|$ is the Euclidean norm. Then, ADMM alternatingly minimizes the augmented Lagrangian $\mathfrak{L}_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda})$ with respect to $\boldsymbol{z}$ and $\boldsymbol{x}$ in conjunction with a dual ascent step. Specifically, the ADMM updates are

$$\boldsymbol{z}(k+1) = \underset{\boldsymbol{z}\in\mathbb{R}^p}{\arg\min} \left\{ g(\boldsymbol{z}) + \frac{\rho n}{2} \left\| \boldsymbol{z} - \bar{\boldsymbol{x}}(k) - \frac{1}{\rho}\bar{\boldsymbol{\lambda}}(k) \right\|^2 \right\}, \tag{4}$$

$$\boldsymbol{x}_i(k+1) = \underset{\boldsymbol{x}_i\in\mathbb{R}^p}{\arg\min} \left\{ f_i(\boldsymbol{x}_i) + \frac{\rho}{2} \left\| \boldsymbol{x}_i + \frac{1}{\rho}\boldsymbol{\lambda}_i(k) - \boldsymbol{z}(k+1) \right\|^2 \right\}, \quad i = 1, ..., n, \tag{5}$$

$$\boldsymbol{\lambda}_i(k+1) = \boldsymbol{\lambda}_i(k) + \rho(\boldsymbol{x}_i(k+1) - \boldsymbol{z}(k+1)), \ i = 1, ..., n, \tag{6}$$

where k is the iteration index, $\bar{\boldsymbol{x}}(k) := \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i(k)$, and $\bar{\boldsymbol{\lambda}}(k) := \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\lambda}_i(k)$. The coordinator updates $\boldsymbol{z}$ according to (4) and broadcasts the public signal $\boldsymbol{z}(k+1)$ to the agents. Each agent i updates $\boldsymbol{x}_i, \boldsymbol{\lambda}_i$ according to (5), (6), and sends the updated $\boldsymbol{x}_i(k+1), \boldsymbol{\lambda}_i(k+1)$ to the coordinator.

In many applications, the cost function f_i contains sensitive private information such as private data, private usage patterns of agent i. On the other hand, in ADMM, the public signal $\boldsymbol{z}(k)$ is very vulnerable to malicious attackers' eavesdropping since it is broadcast to all agents. With adequate side information (e.g., other agents' cost functions), an adversary may infer the private cost function of an agent from the public signal sequence $\{\boldsymbol{z}(k)\}$. Specifically, suppose an adversary knows all the individual cost functions $\{f_i\}_{i\neq j}$ except agent j's and aims to infer f_j. Such a strong adversary is in accordance with the connotations of differential privacy and can take place if all the agents $i \neq j$ collude with the adversary or their information is leaked to the adversary. In the following, we show how the adversary can infer f_j based on the overheard public signal sequence $\{\boldsymbol{z}(k)\}$ and the side information of $\{f_i\}_{i\neq j}$. For any $i \neq j$ and any k, since the adversary knows f_i and $\boldsymbol{z}(k+1)$, (5) can be written as $\boldsymbol{x}_i(k+1) = h_{i,k}(\boldsymbol{\lambda}_i(k))$, in which the function $h_{i,k} : \mathbb{R}^p \mapsto \mathbb{R}^p$ is known to the adversary. Thus, (6) becomes $\boldsymbol{\lambda}_i(k+1) = \boldsymbol{\lambda}_i(k) + \rho(h_{i,k}(\boldsymbol{\lambda}_i(k)) - \boldsymbol{z}(k+1))$, which is a recursive equation known to the adversary. As such, the adversary can infer the sequence $\{\boldsymbol{\lambda}_i(k)\}$ and further the sequence $\{\boldsymbol{x}_i(k)\}$ by using the function $h_{i,k}$. Moreover, since the adversary knows $\boldsymbol{z}(k+1)$ and the function g (public information), it can infer $\bar{\boldsymbol{x}}(k) + \frac{1}{\rho}\bar{\boldsymbol{\lambda}}(k)$ according to (4) in many cases. Based on $\bar{\boldsymbol{x}}(k) + \frac{1}{\rho}\bar{\boldsymbol{\lambda}}(k)$ and $\boldsymbol{x}_i(k), \boldsymbol{\lambda}_i(k)$ for all $i \neq j$, the adversary can further infer $\boldsymbol{\phi}_j(k) := \boldsymbol{x}_j(k) + \frac{1}{\rho}\boldsymbol{\lambda}_j(k)$ for all k. Combining this and (6) for agent j, we have $2\boldsymbol{\lambda}_j(k+1) = \boldsymbol{\lambda}_j(k) + \rho\boldsymbol{\phi}_j(k+1) - \rho\boldsymbol{z}(k+1)$, which is a recursive equation known to the adversary. Based on this recursive equation, the adversary can infer the sequence $\{\boldsymbol{\lambda}_j(k)\}$ and also the sequence $\{\boldsymbol{x}_j(k)\}$ by using the relation $\boldsymbol{x}_j(k) = \boldsymbol{\phi}_j(k) - \frac{1}{\rho}\boldsymbol{\lambda}_j(k)$. Finally, according to (5), for any k, we have

$$\boldsymbol{x}_j(k+1) = \underset{\boldsymbol{x}_j\in\mathbb{R}^p}{\arg\min} \left\{ f_j(\boldsymbol{x}_j) + \frac{\rho}{2} \left\| \boldsymbol{x}_j + \frac{1}{\rho}\boldsymbol{\lambda}_j(k) - \boldsymbol{z}(k+1) \right\|^2 \right\}. \tag{7}$$

Since the adversary knows $\boldsymbol{x}_j(k+1), \boldsymbol{\lambda}_j(k)$ and $\boldsymbol{z}(k+1)$, it can infer information about the function f_j from (7). In many cases, the function f_j is parameterized by some private parameters of agent j and (7) can be used to gain knowledge of these private parameters. With sufficient number of iterations $k = 1, ..., K$, the adversary may even infer all the private parameters of agent j and thus the private cost function f_j. This raises privacy concerns for the agents and motivates the privacy-preserving algorithm design in this paper based on noise injection. With the differential privacy guarantees of the algorithm in this paper, it is statistically hard for an adversary to infer f_j even when it knows all $f_i, i \neq j$ (when f_j changes, the distribution of the public signals $\{\widehat{\boldsymbol{z}}(k)\}$ does not change much). This protects the agents' private information from being inferred by very strong adversaries.

To solve consensus optimization problem with strongly convex functions, alternating minimization algorithm (AMA) [26] is another

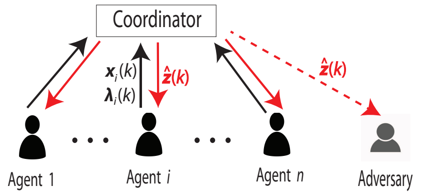

Coordinator
$\boldsymbol{x}_i(k)$
$\boldsymbol{\lambda}_i(k)$
$\hat{\boldsymbol{z}}(k)$
$\hat{\boldsymbol{z}}(k)$
Agent 1
Agent i
Agent n
Adversary

Fig. 1: Differentially private multi-agent optimization, where $\widehat{\boldsymbol{z}}(k)$ is the noisy coordination signal broadcast by the coordinator to all agents and is computed by (11) and (12) in Algorithm 1.

potential approach closely related to ADMM. Existing works show that AMA has global sublinear convergence rate for strongly convex optimization [26], [27]. In contrast, ADMM has global linear convergence rate for strongly convex problems [18]. To obtain faster convergence rate, we focus on ADMM, which is one of the most popular algorithms for solving consensus optimization. It is an interesting future direction to study the impact of privacy preservation on the performance of AMA.

B. Differentially private ADMM

Before presenting the private ADMM formally, we first formulate a differential privacy framework for the regularized consensus optimization problem (1). We define the database to be $\mathcal{D} = (f_1, ..., f_n)$, which contains the private cost functions of all agents. Our goal is to prevent this database from being inferred by an adversary. We denote the set of admissible cost functions as $\mathcal{F}$ so that $f_i \in \mathcal{F}$ for all i, i.e., $\mathcal{D} \in \mathcal{F}^n$. We further define the adjacency relation between databases.

Definition 1. *Given a positive number δ, two databases $\mathcal{D} = (f_1, ..., f_n) \in \mathcal{F}^n$ and $\widetilde{\mathcal{D}} = (\widetilde{f}_1, ..., \widetilde{f}_n) \in \mathcal{F}^n$ are said to be adjacent if there is some $j \in \{1, ..., n\}$ such that (i) $\|\nabla f_j(\boldsymbol{x}_j) - \nabla \widetilde{f}_j(\boldsymbol{x}_j)\| \leq \delta$ for any $\boldsymbol{x}_j \in \mathbb{R}^p$; (ii) $f_i(\boldsymbol{x}_i) = \widetilde{f}_i(\boldsymbol{x}_i)$ for any $\boldsymbol{x}_i \in \mathbb{R}^p$ and $i \neq j$.*

According to Definition 1, two databases $\mathcal{D}, \widetilde{\mathcal{D}}$ are adjacent if there is an agent j such that $\nabla f_j, \nabla \widetilde{f}_j$ are close and other agents' cost functions do not change across $\mathcal{D}, \widetilde{\mathcal{D}}$. Here, we require the gradients of the cost functions $f_j, \widetilde{f}_j$ (instead of themselves) to be close because constant shift of the cost function does not alter the optimal point of Problem (1) and it is the gradient of the cost function that determines the optimal point. Additionally, we define a mechanism $\boldsymbol{M}$ to be a mapping from a database $\mathcal{D}$ to a random vector $\boldsymbol{M}(\mathcal{D})$, which is the observation available to an adversary. Denote the set of possible outputs of $\boldsymbol{M}$ as range$(\boldsymbol{M})$. Then, the differential privacy of a mechanism is defined as follows.

Definition 2. *[1] A mechanism $\boldsymbol{M}$ is ϵ-differentially private if for any adjacent databases $\mathcal{D}, \widetilde{\mathcal{D}} \in \mathcal{F}^n$ and any set $\mathcal{S} \subset$ range$(\boldsymbol{M})$:*

$$\mathbb{P}(\boldsymbol{M}(\mathcal{D}) \in \mathcal{S}) \leq e^{\epsilon}\mathbb{P}\left(\boldsymbol{M}\left(\widetilde{\mathcal{D}}\right) \in \mathcal{S}\right), \tag{8}$$

where $\mathbb{P}$ stands for probability.

A well-known technique to ensure differential privacy is the Laplacian mechanism in the following.

Lemma 1. *[1] For a query function $\boldsymbol{q} : \mathcal{F}^n \mapsto \mathbb{R}^m$, suppose its sensitivity is bounded above by some constant Δ, i.e.,*

$$\sup_{\mathcal{D}, \widetilde{\mathcal{D}} \text{ are adjacent}} \left\|\boldsymbol{q}(\mathcal{D}) - \boldsymbol{q}\left(\widetilde{\mathcal{D}}\right)\right\| \leq \Delta. \tag{9}$$

Let $\boldsymbol{v}$ be a zero mean Laplacian noise with probability density function (PDF) proportional to $e^{-C\|\boldsymbol{v}\|}$, where $0 < C \leq \frac{\epsilon}{\Delta}$. Then, the mechanism $\boldsymbol{M}(\mathcal{D}) = \boldsymbol{q}(\mathcal{D}) + \boldsymbol{v}$ is ϵ-differentially private.

Clearly, in ADMM, the query function $\boldsymbol{q}$ consists of the public signals $(\boldsymbol{z}(1), ..., \boldsymbol{z}(K))$. Based on the Laplacian mechanism, we propose a differentially private version of ADMM for regularized consensus optimization as shown in Algorithm 1. Different from the non-private ADMM in (4)-(6), in differentially private ADMM, the coordinator broadcasts a noisy public signal $\widehat{\boldsymbol{z}}(k+1)$, which perturbs $\boldsymbol{z}(k+1)$ with additive Laplacian noise. Then, each agent i updates its $\boldsymbol{x}_i(k+1), \boldsymbol{\lambda}_i(k+1)$ based on the noisy signal $\widehat{\boldsymbol{z}}(k+1)$. The noise parameter $\alpha(k)$ determines the magnitude of the added noise $\boldsymbol{v}(k)$, $k \geq 2$: the smaller the value of $\alpha(k)$, the larger the magnitude of $\boldsymbol{v}(k)$. The noise parameters $\alpha(2), ..., \alpha(K)$ need to be chosen judiciously so that the differential privacy can be guaranteed while the algorithm can achieve the best convergence bound. Additionally, we note that the initial noise $\boldsymbol{v}(1)$ is set to be $\boldsymbol{0}$ because $\boldsymbol{z}(1)$ does not depend on the cost functions $\{f_i\}$ and contains no private information. The mechanism output of Algorithm 1 is the noisy public signals $(\widehat{\boldsymbol{z}}(1), ..., \widehat{\boldsymbol{z}}(K))$, which an adversary can access through overhearing. We note that the entire algorithm is only run for K iterations, where K is some finite number. After K iterations, the algorithm is terminated instead of being switched to conventional ADMM. Thus, after K iterations, no more signals are transmitted and no more private information is leaked.

The coordinator is assumed to be a trustworthy authority, which is a central entity serving many users or customers. For example, the coordinator can be a utility company in power grid or a technology company in data-intensive applications. The coordinator usually obeys the stipulations in data privacy and does not exploit agents' sensitive data maliciously. Instead, the privacy concern is that a potential adversary may overhear the signals broadcast by the coordinator, i.e., $\widehat{\boldsymbol{z}}(k)$ in Algorithm 1. The adversary may use the overheard signals $\{\widehat{\boldsymbol{z}}(k)\}$ to infer the private information of the agents. The model is illustrated in Fig. 1.

If the agents are networked, ADMM can also be used to obtain a fully decentralized algorithm without central coordinator, in which each agent communicates with its neighbors. In each iteration, the number of inter-agent communications is equal to the number of edges in the network and all of these communications can be potentially overheard by an adversary. This can lead to severe leakage of private information. To suppress the privacy leakage, we have to inject noise to all the inter-agent communications. This will inevitably degrade the optimization performance dramatically. As such, the fully decentralized algorithm is not very suitable for privacy-preserving optimization. Besides, direct communications/coordinations between neighboring agents may not always be possible, while a reliable central coordinator often exists in practice. Thus, we focus on the scenario with a trustworthy central coordinator in this paper.

Algorithm 1 terminates at a finite number of iterations K. We note that early termination alone without injecting noise cannot guarantee differential privacy, which is defined in terms of probability so that the mechanism has to be random by noise injection. When the public signals $(\widehat{\boldsymbol{z}}(1), ..., \widehat{\boldsymbol{z}}(K))$ are noiseless, the adversary can infer the private information of an agent based on the procedure of the deterministic algorithm and adequate side information, e.g., the private information of other agents. A possible inference method of the adversary has been elaborated in Subsection II-A.

The PDF of the injected noise $\boldsymbol{v}$ depends only on its length and is independent of its angle, i.e., the PDF of $\boldsymbol{v}$ is isotropic. We note that the PDF of $\boldsymbol{v}$ is proper since $\int \exp(-\alpha\|\boldsymbol{v}\|)d\boldsymbol{v} < \infty$ for any $\alpha > 0$. Generating such random vector $\boldsymbol{v} \in \mathbb{R}^p$ with PDF $p(\boldsymbol{v}) \propto e^{-\alpha\|\boldsymbol{v}\|}$ is not difficult. We note that the length of $\boldsymbol{v}$, denoted as $u = \|\boldsymbol{v}\|$, is a Gamma-distributed random variable with shape parameter p and scale parameter $\frac{1}{\alpha}$, i.e., $p(u) \propto u^{p-1}e^{-\alpha u}$ for $u > 0$. Moreover, since the PDF of $\boldsymbol{v}$ is isotropic, its angle is uniformly distributed over the unit sphere in $\mathbb{R}^p$. Multiplying its length and angle, we can generate the random vector $\boldsymbol{v}$.

Algorithm 1 Differentially Private ADMM

```
Parameters: number of iterations $K$ and positive numbers $\alpha(2), ..., \alpha(K)$.
Each agent $i$ initializes $\boldsymbol{x}_i(0), \boldsymbol{\lambda}_i(0)$ arbitrarily and sends them to the coordinator.
for $k = 0, 1, ..., K-1$ do
    The coodinator generates Laplacian noise $\boldsymbol{v}(k+1)$ independently. If $k = 0$, set $\boldsymbol{v}(1) = \mathbf{0}$. If $k \geq 1$, generate $\boldsymbol{v}(k+1)$ with PDF proportional to $e^{-\alpha(k+1)\|\boldsymbol{v}(k+1)\|}$.
    The coordinator computes:
        $\bar{\boldsymbol{x}}(k) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i(k), \ \bar{\boldsymbol{\lambda}}(k) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\lambda}_i(k), \quad (10)$
        $\boldsymbol{z}(k+1) = \underset{\boldsymbol{z} \in \mathbb{R}^p}{\arg\min} \left\{ g(\boldsymbol{z}) + \frac{\rho n}{2} \left\| \boldsymbol{z} - \bar{\boldsymbol{x}}(k) - \frac{1}{\rho}\bar{\boldsymbol{\lambda}}(k) \right\|^2 \right\}, \quad (11)$
        $\widehat{\boldsymbol{z}}(k+1) = \boldsymbol{z}(k+1) + \boldsymbol{v}(k+1). \quad (12)$
    The coordinator broadcasts $\widehat{\boldsymbol{z}}(k+1)$ to all agents.
    Each agent $i$ computes:
        $\boldsymbol{x}_i(k+1) = \underset{\boldsymbol{x}_i \in \mathbb{R}^p}{\arg\min} \left\{ f_i(\boldsymbol{x}_i) + \frac{\rho}{2} \left\| \boldsymbol{x}_i + \frac{1}{\rho}\boldsymbol{\lambda}_i(k) - \widehat{\boldsymbol{z}}(k+1) \right\|^2 \right\}, \quad (13)$
        $\boldsymbol{\lambda}_i(k+1) = \boldsymbol{\lambda}_i(k) + \rho(\boldsymbol{x}_i(k+1) - \widehat{\boldsymbol{z}}(k+1)). \quad (14)$
    Each agent $i$ sends $\boldsymbol{x}_i(k+1), \boldsymbol{\lambda}_i(k+1)$ to the coordinator.
end for
```

Remark 1. *An alternative approach for privacy is to encrypt the public coordination signal $\boldsymbol{z}(k)$. Nevertheless, this encryption approach has at least two limitations. First, each agent needs to decrypt the public signal $\boldsymbol{z}(k)$, which may incur high computational overhead. When agents have low computational capability (e.g., cheap sensors), the encryption/decryption approach may be infeasible. Second, if an agent is compromised by the adversary, the decrypted public signal $\boldsymbol{z}(k)$ will be leaked to the adversary. This also exposes other agents' private information to the adversary since the public signal is related to all agents' private information. In contrast, when the proposed privacy-preserving ADMM algorithm is used, an agent's private information is protected even if all the other agents are compromised. This is a salient feature of differential privacy and can be achieved via adding noise to the public signal $\boldsymbol{z}(k)$ (i.e., $\widehat{\boldsymbol{z}}(k)$) as we demonstrate in Algorithm 1.*

III. Analysis of Privacy and Convergence

In this section, we derive conditions on the noise parameters $\{\alpha(k)\}$, under which ϵ-differential privacy can be achieved. We further study the convergence performance of Algorithm 1 and derive the optimal values of $\{\alpha(k)\}$ so that the convergence upper bound is minimized. Before proceeding to the formal analysis, we first make the following assumptions on the consensus optimization problem.

Assumption 1. *For any $f \in \mathcal{F}$, the following conditions hold.*

1) *f is differentiable.*
2) *f is τ-strongly convex ($\tau > 0$), i.e., $(\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}))^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{y}) \geq \tau\|\boldsymbol{x} - \boldsymbol{y}\|^2$, for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$.*
3) *∇f is L-Lipschitz continuous, i.e., $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$, for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$.*

Assumption 2. *g is a convex function. Further, for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$, $\boldsymbol{h} \in \partial g(\boldsymbol{x})$, $\widehat{\boldsymbol{h}} \in \partial g(\boldsymbol{y})$, we have $\left\| \widehat{\boldsymbol{h}} - \boldsymbol{h} \right\| \leq G + M\|\boldsymbol{x} - \boldsymbol{y}\|$, where G, M are two non-negative constants, and ∂g denotes the subdifferential of g.*

Assumption 3. $\rho > \max\left\{2L, \frac{M}{n}\right\}$.

In Assumption 1, strong convexity and Lipschitz continuous gradient are typical assumptions used to establish the linear convergence of ADMM [18]. We note that ADMM can still converge sublinearly for general convex (not necessarily strongly convex) functions. In the classical results in the monograph [2] by Boyd *et al.*, under convexity assumption, convergence to the optima is shown while no result on the convergence rate is established. To achieve faster linear convergence rate, existing works on non-private ADMM usually assume strong convexity [18], which is also adopted in this paper for better convergence rate. Assumption 1 can hold for many practical problems. For instance, in LASSO (c.f. the numerical example in this paper), f_i's are quadratic and Assumption 1 holds.

To ensure the convergence (not necessarily linear convergence) of conventional non-private ADMM, the typical assumption on g is that it is convex, proper and closed (or lower semicontinuous) [2]. In this paper, we make the assumption that the (sub)-gradient of g is either bounded or Lipschitz continuous (c.f. Assumption 2). The main purpose of this assumption is to facilitate privacy guarantee, which is not considered in conventional ADMM. In particular, to obtain a finite sensitivity upper bound H (c.f. (16) and Lemma 3), we need finite G and M, i.e., Assumption 2. On the other hand, the assumptions of Lipschitz continuous gradient or bounded (sub)-gradient are not uncommon in existing works, e.g., [13], [15], [16], [28]. Many frequently used regularizers have either Lipschitz continuous gradient or bounded subgradient so that Assumption 2 is satisfied. For example, the quadratic regularizer in ridge regression has Lipschitz continuous gradient, while the ℓ_1 norm for sparsity regularization has bounded subgradient.

The main purpose of Assumption 3 is to facilitate privacy analysis (c.f. the definition of the sensitivity bound H in (16) and the proof of Lemma 3), which is not taken into consideration by conventional non-private ADMM. We note that ρ is a user-defined algorithm parameter, which can be chosen to satisfy Assumption 3.

Assumptions 1, 2 and 3 can hold in typical control setups. For instance, in Section IV, we show that these assumptions hold for a multi-agent finite-horizon linear-quadratic control problem (23).

A. Condition for Differential Privacy

In this subsection, we derive the condition for ϵ-differential privacy of Algorithm 1 in terms of the noise parameters $\alpha(2), ..., \alpha(K)$. We will make use of the following sequential composition theorem for differential privacy.

Lemma 2. *[29] Consider a sequence of mechanisms $(\boldsymbol{M}_1, ..., \boldsymbol{M}_K)$, where the output of $\boldsymbol{M}_k$ depends on the outputs of $\boldsymbol{M}_1, ..., \boldsymbol{M}_{k-1}$ as follows:*

$$\boldsymbol{M}_k(\mathcal{D}) = \boldsymbol{M}_k(\mathcal{D}, \boldsymbol{M}_1(\mathcal{D}), ..., \boldsymbol{M}_{k-1}(\mathcal{D})), \ k = 1, ..., K. \quad (15)$$

If $\boldsymbol{M}_k(\cdot, \boldsymbol{a}_1, ..., \boldsymbol{a}_{k-1})$ is ϵ_k-differentially private for any given $\boldsymbol{a}_1 \in$ range($\boldsymbol{M}_1$), ..., $\boldsymbol{a}_{k-1} \in$ range($\boldsymbol{M}_{k-1}$), $\forall k = 1, ..., K$, then the joint mechanism $\boldsymbol{M} = (\boldsymbol{M}_1, ..., \boldsymbol{M}_K)$ is $\sum_{k=1}^{K} \epsilon_k$-differentially private.

Clearly, the mechanism of Algorithm 1, $(\widehat{\boldsymbol{z}}(1), ..., \widehat{\boldsymbol{z}}(K))$, is comprised of K sequential mechanisms. To show the differential privacy of Algorithm 1, we need to establish the differential privacy of $\widehat{\boldsymbol{z}}(t+1)$ given $\widehat{\boldsymbol{z}}(1), ..., \widehat{\boldsymbol{z}}(t)$. To this end, we derive the sensitivity of $\boldsymbol{z}(t+1)$ given $\widehat{\boldsymbol{z}}(1), ..., \widehat{\boldsymbol{z}}(t)$ in the following lemma. Define a positive constant H as (the positivity follows from Assumption 3)

$$H := \frac{G}{\rho n - M} + \frac{3\delta\rho}{(\rho - 2L)(\rho n - M)}. \quad (16)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.3022856, IEEE Transactions on Automatic Control

Lemma 3. *Suppose Assumptions 1, 2, 3 hold. When the values of $\widehat{\boldsymbol{z}}(1), ..., \widehat{\boldsymbol{z}}(t)$ are given, the sensitivity of $\boldsymbol{z}(t+1)$ is upper bounded by H. In other words,*

$$\sup_{\mathcal{D},\widetilde{\mathcal{D}} \text{ are adjacent}} \left\{ \left\| \boldsymbol{z}(t+1;\mathcal{D}) - \boldsymbol{z}(t+1;\widetilde{\mathcal{D}}) \right\| \middle| \text{given } \widehat{\boldsymbol{z}}(1), ..., \widehat{\boldsymbol{z}}(t) \right\} \\ \leq H,\ \forall t = 1, ..., K-1, \tag{17}$$

where $\boldsymbol{z}(t+1;\mathcal{D})$ is the value of $\boldsymbol{z}(t+1)$ for database $\mathcal{D}$.

The proof of Lemma 3 is presented in the supplementary file. A sketch of the proof is given as follows. We consider two adjacent databases $\mathcal{D}$ and $\widetilde{\mathcal{D}}$. From (13) and (14), we obtain an upper bound for $\|\boldsymbol{x}_j(k+1;\widetilde{\mathcal{D}})-\boldsymbol{x}_j(k+1;\mathcal{D})\|$ in terms of $\|\boldsymbol{x}_j(k;\widetilde{\mathcal{D}})-\boldsymbol{x}_j(k;\mathcal{D})\|$. Using this bound repeatedly, we can get an upper bound for $\|\boldsymbol{x}_j(k;\widetilde{\mathcal{D}}) - \boldsymbol{x}_j(k;\mathcal{D})\|$. Then, we can get an upper bound for $\|\boldsymbol{\lambda}_j(k;\widetilde{\mathcal{D}})-\boldsymbol{\lambda}_j(k;\mathcal{D})\|$ accordingly. Thus, $\|\bar{\boldsymbol{x}}(k;\mathcal{D})-\bar{\boldsymbol{x}}(k;\widetilde{\mathcal{D}})\|$ and $\|\bar{\boldsymbol{\lambda}}(k;\mathcal{D}) - \bar{\boldsymbol{\lambda}}(k,\widetilde{\mathcal{D}})\|$ can also be upper bounded. Finally, by (11), we get the upper bound for $\|\boldsymbol{z}(k+1;\mathcal{D}) - \boldsymbol{z}(k+1;\widetilde{\mathcal{D}})\|$. Based on Lemma 3 and the Laplacian mechanism, we can readily derive the condition for ϵ-differential privacy of Algorithm 1 by invoking the sequential composition property in Lemma 2.

Theorem 1. *Suppose Assumptions 1, 2, 3 hold. If $\sum_{k=2}^{K} \alpha(k) \leq \frac{\epsilon}{H}$, then Algorithm 1 is ϵ-differentially private.*

B. Convergence Analysis

In this subsection, we analyze the convergence performance of Algorithm 1. We first make a few definitions. We denote $(\boldsymbol{x}_1^*, ..., \boldsymbol{x}_n^*, \boldsymbol{z}^*)$ as the optimal point of Problem (2). The optimal point of the original consensus problem (1) is denoted as $\widehat{\boldsymbol{x}}$. It is obvious that $\boldsymbol{x}_1^* = \cdots = \boldsymbol{x}_n^* = \boldsymbol{z}^* = \widehat{\boldsymbol{x}}$. We further denote the unique dual optimal point of Problem (2) as $(\boldsymbol{\lambda}_1^*, ..., \boldsymbol{\lambda}_n^*)$[1]. We denote $\bar{\boldsymbol{\lambda}}^* = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\lambda}_i^*$. We define $\boldsymbol{x}^* \in \mathbb{R}^{np}$ and $\boldsymbol{\lambda}^* \in \mathbb{R}^{np}$ to be the concatenations of all $\boldsymbol{x}_i^*$'s and all $\boldsymbol{\lambda}_i^*$'s, respectively. We further define $\boldsymbol{x}(k) \in \mathbb{R}^{np}$ and $\boldsymbol{\lambda}(k) \in \mathbb{R}^{np}$ to be the concatenation of all $\boldsymbol{x}_i(k)$'s and all $\boldsymbol{\lambda}_i(k)$'s across agents, respectively. In addition, we define

$$\pi(k) := \mathbb{E}\left[\frac{1}{2\rho}\|\boldsymbol{\lambda}(k) - \boldsymbol{\lambda}^*\|^2 + \frac{\rho}{2}\|\boldsymbol{x}(k) - \boldsymbol{x}^*\|^2\right], \tag{18}$$

which measures the distance between the algorithm iterates and the optima. The expectation in (18) is taken with respect to the probability distribution from which the noise vectors $\{\boldsymbol{v}(k)\}$ are generated. Let θ be arbitrary number within the interval $(0, 1)$. Define $\beta = \min\left\{\frac{2\theta\tau}{\rho}, \frac{2\rho(1-\theta)}{L}\right\} > 0$. Then, we have the following main theorem regarding the convergence performance of Algorithm 1. Its proof is relegated to the supplementary file.

Theorem 2. *Under Assumptions 1, 2, 3, Algorithm 1 satisfies*

$$\sqrt{\pi(K)} \leq \frac{\sqrt{\pi(0)}}{\left(\sqrt{1+\beta}\right)^K} + \frac{4\sqrt{n\rho p(p+1)}}{\left(\sqrt{1+\beta}\right)^{K+2}} \sum_{l=2}^{K} \frac{\left(\sqrt{1+\beta}\right)^l}{\alpha(l)}. \tag{19}$$

Moreover, we have

$$\sum_{i=1}^{n} \mathbb{E}\left[\|\boldsymbol{x}_i(K) - \widehat{\boldsymbol{x}}\|^2\right] \leq \frac{2}{\rho}\pi(K). \tag{20}$$

A sketch of the proof is given as follows. Exploiting the KKT conditions of Problem (2) and the update equations in Algorithm 1, we can show that $\pi(k) \geq (1+\beta)\pi(k+1) - \mathbb{E}[\boldsymbol{v}(k+1)^{\mathsf{T}}\boldsymbol{\xi}(k)]$. Here, $\boldsymbol{\xi}(k)$ is a random vector whose expected length can be upper bounded in terms of $\pi(k+1)$. Then, we get an upper bound for $\sqrt{\pi(k+1)}$ in terms of $\sqrt{\pi(k)}$ and $\mathbb{E}[\|\boldsymbol{v}(k+1)\|^2]$. Using the bound repeatedly and noting that $\mathbb{E}\left[\|\boldsymbol{v}(l)\|^2\right] = \frac{p(p+1)}{\alpha(l)^2}$, we obtain (19).

From (19), we can clearly see the impact of noise parameters $\{\alpha(l)\}$ on the convergence performance of Algorithm 1. When $\alpha(l)$ decreases, the noise magnitude increases, and the privacy preservation is improved at the cost of convergence performance as shown in (19). Next, we design the optimal noise parameters $\alpha(2), ..., \alpha(K)$ so that the convergence bound in (19) is minimized while the ϵ-differential privacy can be ensured by Theorem 1.

Theorem 3. *Under Assumptions 1, 2, 3, the optimal noise parameters $\alpha^*(2), ..., \alpha^*(K)$ that minimize the convergence bound in* (19) *while satisfying the ϵ-differential privacy are*

$$\alpha^*(l) = \frac{\epsilon(1+\beta)^{\frac{l-2}{4}}\left[(1+\beta)^{\frac{1}{4}} - 1\right]}{H\left[(1+\beta)^{\frac{K-1}{4}} - 1\right]},\ l = 2, ..., K. \tag{21}$$

Moreover, with the optimal $\{\alpha^*(l)\}$, we have

$$\sqrt{\pi(K)} \leq \frac{\sqrt{\pi(0)}}{(1+\beta)^{\frac{K}{2}}} \\ + \frac{4H\sqrt{n\rho p(p+1)}}{\epsilon\left[(1+\beta)^{\frac{3}{4}} - (1+\beta)^{\frac{1}{2}}\right]}\left[1 - \left(\frac{1}{1+\beta}\right)^{\frac{K-1}{4}}\right]^2. \tag{22}$$

The proof of Theorem 3 is relegated to the supplementary file. The proof follows from straightforward application of Cauchy's inequality. From (21), we see that the optimal noise parameter $\alpha^*(l)$ increases with l. In other words, as Algorithm 1 progresses, the injected amount of noise in each iteration should decrease. This has the following intuitive explanation. On one hand, the differential privacy of Algorithm 1 does not distinguish between the added noise in different iterations (Theorem 1 concerns the plain sum of all $\alpha(k)$), because of the uniform constant bound for sensitivity in every iteration (Lemma 3) and the sequential composition theorem for differential privacy (Lemma 2 concerns the plain sum of all ϵ_k). On the other hand, the convergence performance of Algorithm 1 relies more on the quality of later iterations than that of the earlier iterations (the bound in (19) puts more weight on $\frac{1}{\alpha(l)}$ with larger iteration index l). Thus, to guarantee differential privacy and minimize its impact on the convergence performance, we should inject more (resp. less) noise in the earlier (resp. later) iterations.

Additionally, (22) manifests a clear effect of the ϵ-differential privacy on the convergence performance of Algorithm 1. For smaller ϵ, the level of privacy is improved at the expense of convergence performance degradation. Furthermore, the two terms in the bound of (22) correspond to the bound of traditional non-private ADMM and the performance loss incurred by privacy preservation, respectively. As the number of iterations K goes to infinity, the first term of the bound decreases to zero, while the second term increases to a positive constant. An optimal finite number of iterations K exists to minimize the bound in (22). This contrasts to most traditional non-private optimization algorithms, in which the convergence performance always gets better as more iterations are carried out. The reason is that, as more iterations are conducted, the amount of revealed information increases. To combat this information revelation and preserve a given privacy level, we have to inject more noise in each iteration, which in turn harms the convergence performance. We note that analogous phenomenon has been observed in [13] for projected gradient descent.

In (22), if we let $\epsilon \to \infty$ (i.e., no privacy guarantee), the linear convergence of conventional ADMM is recovered. We note that ADMM may still converge sublinearly without the strong convexity and Lipschitz smoothness assumptions in this paper. In such a case,

[1] By the KKT conditions of Problem (2), we know that $\nabla f_i(\boldsymbol{x}_i^*) + \boldsymbol{\lambda}_i^* = \boldsymbol{0}$ for $i = 1, ..., n$. Further, we know that $\boldsymbol{x}_1^* = \cdots = \boldsymbol{x}_n^* = \widehat{\boldsymbol{x}}$ ($\widehat{\boldsymbol{x}}$ is the optimal solution to Problem (1)), since Problems (1) and (2) are equivalent. Because of the strong convexity of the objective function of Problem (1), its optimal point $\widehat{\boldsymbol{x}}$ is unique. As such, all the $\boldsymbol{x}_i^*$'s are unique. By the KKT condition $\boldsymbol{\lambda}_i^* = -\nabla f_i(\boldsymbol{x}_i^*)$, the dual optimal solution $(\boldsymbol{\lambda}_1^*, ..., \boldsymbol{\lambda}_n^*)$ is thus unique.

the convergence bound in (22) is expected to become looser (the first part of (22) will become a sublinear bound, e.g., $\mathcal{O}(1/K)$, and the second part of (22) will change to a larger form as β becomes zero when $\tau = 0$ and $L = \infty$, i.e., no strong convexity and Lipschitz continuous gradient assumptions). With the more stringent strong convexity and Lipschitz smoothness assumptions, we are able to derive tighter convergence bound in this paper.

The optimal integer K that minimizes the bound (22) is either the floor or the ceiling of $1+4\log_{1+\beta}\left(1+\frac{\sqrt{\pi(0)}\left((1+\beta)^{\frac{1}{4}}-1\right)\epsilon}{4H\sqrt{n\rho p(p+1)}}\right)$. The derivation of this optimal K is relegated to the supplementary file. In practice, to compute this theoretically optimal K, one needs an estimate of $\pi(0) = \frac{1}{2\rho}\|\boldsymbol{\lambda}(0)-\boldsymbol{\lambda}^*\|^2+\frac{\rho}{2}\|\boldsymbol{x}(0)-\boldsymbol{x}^*\|^2$, i.e., estimates of the ranges of the primal/dual optima $\boldsymbol{x}^*$ and $\boldsymbol{\lambda}^*$. If such estimates are not available, one has to resort to numerical trial-and-error in order to find a satisfactory value of K empirically.

According to Theorem 1, a smaller H makes privacy preservation easier. Thus, by the definition of H in (16), a larger ρ is preferred for privacy preservation. Nevertheless, too large ρ will lead to slow convergence of ADMM. To see this, we recall the definition of β as $\beta = \min\left\{\frac{2\theta\tau}{\rho}, \frac{2\rho(1-\theta)}{L}\right\}$, where $\theta \in (0,1)$ is arbitrary. Setting $\theta = \frac{\rho^2}{\rho^2+\tau L}$, we obtain the optimal (largest) β as $\beta = \frac{2\tau\rho}{\rho^2+\tau L}$. If ρ is too large, then β is very small and the convergence of the ADMM can be very slow, c.f. the first term $\frac{\sqrt{\pi(0)}}{(1+\beta)^{\frac{K}{2}}}$ on the R.H.S. of (22) in Theorem 3. Hence, an appropriate ρ should be neither too large nor too small in order to balance the privacy preservation and the convergence of ADMM. In practice, ρ is chosen empirically based on trial-and-error while keeping Assumption 3 satisfied.

When the individual cost functions are nonconvex and Assumption 1 no longer holds, the convergence to global optima cannot be ensured even in the absence of privacy considerations. Nevertheless, the convergence of conventional ADMM to stationary points can be established for nonconvex problems without privacy preservation [30]. As such, when noise is injected to preserve privacy, we expect that the nonconvex ADMM would converge to some neighborhood of a stationary point. The size of the neighborhood is related to the bias caused by the added noise. The smaller ϵ is, the larger this bias becomes.

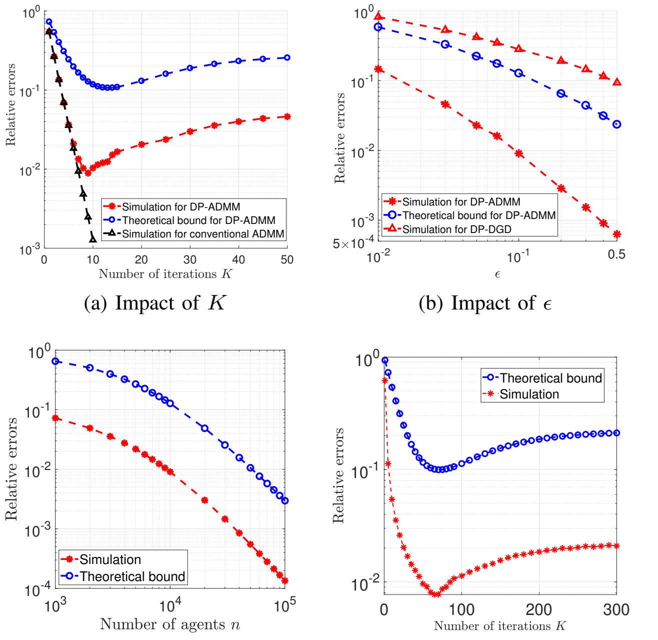
IV. Numerical Experiments

In this section, simulation results are presented to confirm the theoretical properties of Algorithm 1. In the experiments, we let the set of admissible cost functions be $\mathcal{F}_{\tau,L} = \{f : \mathbb{R}^p \mapsto \mathbb{R} | \exists\ \tau\boldsymbol{I}_p \preceq \boldsymbol{B} \preceq L\boldsymbol{I}_p,\ \boldsymbol{c} \in \mathbb{R}^p$ such that $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\mathsf{T}\boldsymbol{B}\boldsymbol{x} + \boldsymbol{c}^\mathsf{T}\boldsymbol{x}, \forall \boldsymbol{x} \in \mathbb{R}^p\}$, where $L > \tau > 0$ are constants. The cost function $f_i \in \mathcal{F}_{\tau,L}$ of agent i is denoted as $f_i(\boldsymbol{x}_i) = \frac{1}{2}\boldsymbol{x}_i^\mathsf{T}\boldsymbol{B}_i\boldsymbol{x}_i + \boldsymbol{c}_i^\mathsf{T}\boldsymbol{x}_i$. Further, we set the regularizer to be $g(\boldsymbol{x}) = \gamma\|\boldsymbol{x}\|_1$, where $\gamma > 0$ is a constant. In such a case, the regularized consensus optimization (1) becomes multi-agent LASSO, which has broad applications in statistics. Assumption 1 is clearly satisfied. Assumption 2 is also satisfied by setting $G = 2\gamma\sqrt{p}$ and $M = 0$. Assumption 3 is satisfied by choosing $\rho > 2L$. Hence, all the theoretical results in Section III can be applied. The parameters of the simulations are chosen as follows: $n = 10000$, $p = 5$, $\tau = 1$, $L = 2$, $\rho = 5$, $\delta = 1$, $\gamma = 100$. We note that 10000 agents can be realistic in large-scale systems. For instance, in communication networks, the resource (e.g., bandwidth) allocation problems can involve a huge number of mobile users, who may have private information such as locations. As another example, in the power scheduling problems for electric vehicle (EV) charging, the power grid may involve a huge number of EV users, who may also have private information such as the start/ending time of the charging (which corresponds to the users' activity patterns). The values of ϵ and K are specified in each experiment. $\boldsymbol{x}(0)$ and $\boldsymbol{\lambda}(0)$ are initialized to be $\mathbf{0}$. The noise parameters $\alpha(2), ..., \alpha(K)$ are set optimally pursuant to (21).

We first fix $\epsilon = 0.1$ and study the impact of the number of iterations K on the convergence performance of Algorithm 1. After K iterations, the expected total squared error of all agents is $\sum_{i=1}^n \mathbb{E}[\|\boldsymbol{x}_i(K) - \widehat{\boldsymbol{x}}\|^2]$, where $\widehat{\boldsymbol{x}}$ is the optimal solution to (1). The initial total squared error of all agents is $n\|\widehat{\boldsymbol{x}}\|^2$ (note that $\boldsymbol{x}_i(0) = \mathbf{0}$ for all i initially). The ratio between these two numbers (called relative error hereafter), i.e., $\frac{\sum_{i=1}^n \mathbb{E}[\|\boldsymbol{x}_i(K)-\widehat{\boldsymbol{x}}\|^2]}{n\|\widehat{\boldsymbol{x}}\|^2}$, measures the convergence progress of the algorithm and is used as the performance criterion in the simulations. We also compute the theoretical upper bound of this relative error by using (20) in conjunction with (22). The result is shown in Fig. 2-(a). As K increases, in accordance with the theoretical bound, the simulated relative error first decreases and then increases. There is a finite optimal value of K (9 for the simulation and 13 for the theoretical bound) for the convergence performance. The reason for this phenomenon has been discussed after Theorem 3. Specifically, to achieve better convergence of ADMM, more iterations should be conducted. In such a case, more private information is revealed and hence more noise injection is needed for privacy preservation, which in turn deteriorates the convergence performance. Therefore, a judicious choice of the number of iterations K is necessary to balance the tradeoff between the convergence of ADMM and the privacy leakage during the algorithm execution. We further note that, though the theoretical bound is quantitatively loose compared to the simulation result, their qualitative trends are similar. As such, the theoretical bound can still provide useful information in many cases, e.g., predetermining the number of iterations K. Further, we illustrate the performance of conventional ADMM (with no noise injection and no privacy guarantee) in Fig. 2-(a). The conventional ADMM exhibits linear convergence, as one would expect by setting $\epsilon \to \infty$ in (22), i.e., no privacy guarantee. In Fig. 2-(a), we only plot the first 10 iterations of conventional ADMM because later iterations go beyond the main scope of this figure (after 30 iterations, the relative error of conventional ADMM is 2×10^{-9}). Comparing the simulation results of DP-ADMM and conventional ADMM, we can see that they have very similar accuracy when the number of iterations K is within 6. With more iterations, the DP-ADMM has to add more noise in each iteration in order to suppress the total privacy leakage and the impact of noise on accuracy starts to emerge when $K \geq 7$. When $K = 9$, the DP-ADMM achieves the minimum relative error of 8.9×10^{-3}, which is reasonably close to that of the conventional ADMM (4.8×10^{-3}). Using $K \geq 10$ iterations is meaningless for DP-ADMM since this will deteriorate its accuracy. In contrast, the error of conventional ADMM will converge linearly to zero with more iterations.

Next, we vary ϵ from 0.01 to 0.5 and examine its impact on the convergence performance of Algorithm 1. For each value of ϵ, the number of iterations K is chosen optimally. The simulated relative errors and their theoretical bounds are shown in Fig. 2-(b). As ϵ increases, the privacy requirement becomes less stringent, and the relative error of Algorithm 1 decreases, in accordance with the intuition and the theoretical bound. This suggests that we can always increase ϵ to improve the accuracy of DP-ADMM at the cost of loosening the privacy requirement. The error of the DP-ADMM can be made arbitrarily small with sufficiently large ϵ, in which case the DP-ADMM tends to become conventional ADMM with no privacy guarantee. In light of this, the proposed DP-ADMM highlights a flexible tradeoff between accuracy and privacy. In different applications, one can use different values of ϵ to achieve satisfactory balance between accuracy and privacy. In particular, when $\epsilon \to \infty$, the DP-ADMM degenerates to the conventional ADMM. Further, from Fig. 2-(b), though the theoretical bound is quantitatively loose

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.3022856, IEEE Transactions on Automatic Control



(a) Impact of K

(b) Impact of ϵ

(c) Impact of n

(d) Higher problem dimension and larger condition number

Fig. 2: Impact of various factors on the convergence performance

in comparison with the simulated relative error, the slopes of the theoretical and simulated curves are close to each other. This indicates that the theoretical bound can characterize the decaying rate of the errors with respect to the privacy level ϵ accurately. Additionally, the optimal number of iterations K are 4, 9, 15, when the values of ϵ are 0.01, 0.1, 0.5, respectively. This trend coincides with the theoretical bound in (22): the optimal K minimizing the bound in (22) takes the form of $1 + 4\log_{1+\beta}(1 + \mathcal{O}(\epsilon))$, which increases with ϵ. In fact, when ϵ grows, Algorithm 1 tends to be non-private ADMM, which certainly benefits from more iterations. Additionally, we adapt the differentially private distributed gradient descent (DP-DGD) algorithm in [31] for the setting in this paper (the original formulation in [31] is over a network with no central coordinator) and illustrate its performance in Fig.2-(b) for comparison. It can be observed that the accuracy of Algorithm 1 is significantly better than that of the DP-DGD. This can be attributed to the better convergence properties of ADMM over gradient descent in general. Besides, to preserve the privacy of the individual functions, the DP-DGD in [31] uses geometrically decaying stepsizes and it takes hundreds of iterations for the DP-DGD to converge (in fact, the theoretical convergence results in [31] require that the number of iterations goes to infinity). This further corroborates the computational advantage of Algorithm 1, which needs much fewer iterations to converge to a satisfactory solution.

Additionally, we show the impact of the number of agents n on the relative errors (both theoretical upper bounds and the simulated ones) of the proposed algorithm in Fig. 2-(c). The privacy level ϵ is fixed to be 0.1 and the number of iterations K are chosen to be optimal. We let n vary between 10^3 and 10^5. We observe that the relative error of the algorithm decreases with n. This confirms the intuition that a large number of agents can facilitate privacy preservation since each agent's impact on the public signal is small. This phenomenon has been explained by the discussion after Theorem 1.

Further, we consider more realistic problem data with higher problem dimension and larger condition number. Specifically, we set $L = 10$ (so that the condition number L/τ becomes larger), $p = 20$, $\rho = 30$ and $\epsilon = 1$. The other parameters remain unchanged. In such a case, the relative errors of Algorithm 1 for different numbers of iterations K are shown in Fig. 2-(d). We observe that there is still a finite optimal value for K (65 for the simulation and 70 for the theoretical upper bound). Compared with the results in Fig. 2-(a), the optimal value of K has increased as the problem becomes harder to solve (higher dimension and worse condition number) and the privacy requirement becomes less stringent, both of which favor more iterations. This larger number of iterations is more realistic in most practical scenarios.

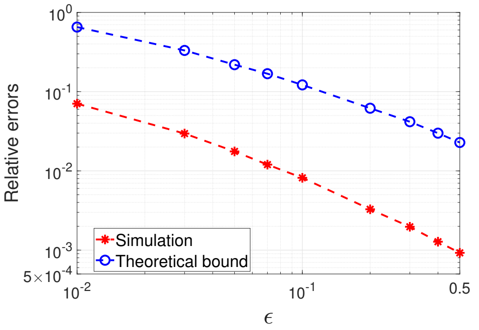


Fig. 3: Impact of the privacy level ϵ on the performance of a multi-agent finite-horizon linear-quadratic control problem, in which the cost coefficients of each agent are private information.

Finally, we apply the proposed algorithm to a multi-agent finite-horizon linear-quadratic control problem as follows:

$$\min_{\boldsymbol{x},\boldsymbol{u}} \sum_{i=1}^{n}\sum_{t=1}^{m}\left(\boldsymbol{x}_t^\top \boldsymbol{Q}_i \boldsymbol{x}_t + \boldsymbol{a}_i^\top \boldsymbol{x}_t\right) + \sum_{t=0}^{m-1}\left(\boldsymbol{u}_t^\top \boldsymbol{P}\boldsymbol{u}_t + \boldsymbol{b}^\top \boldsymbol{u}_t\right) \quad (23a)$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t, \ t = 0, ..., m-1, \quad (23b)$$

where n is the number of agents and m is the time horizon. $\boldsymbol{x}_t \in \mathbb{R}^w$ is the system state and $\boldsymbol{u}_t \in \mathbb{R}^r$ is the control input. When the state is $\boldsymbol{x}_t$, the cost of agent i is given by the first summand in (23a), where the coefficients $\boldsymbol{Q}_i$ and $\boldsymbol{a}_i$ are private information of agent i. To implement the control input $\boldsymbol{u}_t$, the controller incurs a cost given by the second summand in (23a). The cost coefficients $(\boldsymbol{P}, \boldsymbol{b})$ of the controller and the linear system coefficients $(\boldsymbol{A}, \boldsymbol{B})$ are public information. $\boldsymbol{x}_0$ is the given initial state and the optimization variables in (23) are $\boldsymbol{x} = [\boldsymbol{x}_1^\top, ..., \boldsymbol{x}_m^\top]^\top$ and $\boldsymbol{u} = [\boldsymbol{u}_0^\top, ..., \boldsymbol{u}_{m-1}^\top]^\top$.

Since the private data are distributed in each agent and the public coordination signal is noisy for privacy-preservation, it is challenging to derive a simple closed-loop control policy as in the classical non-private linear-quadratic regulator. Alternatively, we view (23) as a *planning* problem with private information and solves for all the control inputs $\{\boldsymbol{u}_t\}_{t=0,...,m-1}$ in advance by transforming (23) into a consensus optimization problem amenable to Algorithm 1.

Let $\boldsymbol{C} \in \mathbb{R}^{mw\times mr}$ be a matrix such that the (t,l)-th $w \times r$ block of $\boldsymbol{C}$ is $\boldsymbol{A}^{t-l}\boldsymbol{B}$ if $t \geq l$ and $\boldsymbol{0}$ otherwise. Let $\boldsymbol{D} = [\boldsymbol{A}^\top, (\boldsymbol{A}^2)^\top, ..., (\boldsymbol{A}^m)^\top]^\top$. Then, from (23b), we have $\boldsymbol{x} = \boldsymbol{C}\boldsymbol{u} + \boldsymbol{D}\boldsymbol{x}_0$. Using this to eliminate $\boldsymbol{x}$ in the objective function (23a), we can rewrite (23) equivalently as

$$\min_{\boldsymbol{u}} \sum_{i=1}^{n} f_i(\boldsymbol{u}) + g(\boldsymbol{u}), \quad (24)$$

where $f_i(\boldsymbol{u}) = (\boldsymbol{C}\boldsymbol{u} + \boldsymbol{D}\boldsymbol{x}_0)^\top \widetilde{\boldsymbol{Q}}_i(\boldsymbol{C}\boldsymbol{u} + \boldsymbol{D}\boldsymbol{x}_0) + \boldsymbol{a}_i^\top \sum_{t=1}^{m}(\boldsymbol{A}^t\boldsymbol{x}_0 + \sum_{s=0}^{t-1}\boldsymbol{A}^{t-1-s}\boldsymbol{B}\boldsymbol{u}_s)$, $g(\boldsymbol{u}) = \sum_{t=0}^{m-1}(\boldsymbol{u}_t^\top \boldsymbol{P}\boldsymbol{u}_t + \boldsymbol{b}^\top \boldsymbol{u}_t)$, and $\widetilde{\boldsymbol{Q}}_i = \text{diag}(\boldsymbol{Q}_i, ..., \boldsymbol{Q}_i)$. Problem (24) is clearly in the form of (1) so that we can apply the proposed algorithm. In the following, we verify

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.3022856, IEEE Transactions on Automatic Control

Assumptions 1, 2, 3. Suppose there exist positive constants $\widehat{\tau}$ and $\widehat{L}$ such that $\widehat{\tau}\boldsymbol{I} \preceq \boldsymbol{Q}_i \preceq \widehat{L}\boldsymbol{I}$. Assume that $\boldsymbol{B}$ has full column rank, i.e., rank$(\boldsymbol{B}) = r$. Then, it can be shown that $\boldsymbol{C}$ also has full column rank, i.e., rank$(\boldsymbol{C}) = mr$. Thus, $\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}$ is positive definite and $\lambda_{\min}(\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}) > 0$, where $\lambda_{\min}(\cdot)$ means the smallest eigenvalue. Therefore, $\nabla^2 f_i(\boldsymbol{u}) = 2\boldsymbol{C}^{\mathsf{T}}\widetilde{\boldsymbol{Q}}_i\boldsymbol{C} \succeq 2\widehat{\tau}\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C} \succeq \tau\boldsymbol{I}$ so that f_i is τ-strongly convex, where $\tau = 2\widehat{\tau}\lambda_{\min}(\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}) > 0$. Further, $\boldsymbol{C}^{\mathsf{T}}\widetilde{\boldsymbol{Q}}_i\boldsymbol{C} \preceq \widehat{L}\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C} \preceq \widehat{L}\|\boldsymbol{C}\|^2\boldsymbol{I}$. Hence, ∇f_i is L-Lipschitz continuous, where $L = 2\widehat{L}\|\boldsymbol{C}\|^2$. Thus, Assumption 1 holds. Similarly, we can show that ∇g is $2\sqrt{m}\|\boldsymbol{P}\|$-Lipschitz continuous, i.e., Assumption 2 holds with $M = 2\sqrt{m}\|\boldsymbol{P}\|$ and $G = 0$. Assumption 3 can always be satisfied by choosing ρ large enough. We apply Algorithm 1 to this optimal control problem with parameters $n = 10000$, $m = 20$, $r = 2$, $w = 3$. The positive definite matrices $\{\boldsymbol{Q}_i\}$ and $\boldsymbol{P}$ are generated randomly so that their eigenvalues are within $[0.3, 1]$ (we first generate symmetric matrices with random entries and then scale/shift (by scalar multiples of identity matrix) them to move their eigenvalues into this interval). Each entry of the vectors $\{\boldsymbol{a}_i\}$, $\boldsymbol{b}$ and matrices $\boldsymbol{A}, \boldsymbol{B}$ is selected randomly from $[-1, 1]$. The relative errors and their theoretical upper bounds for different values of ϵ are shown in Fig. 3. We observe that the relative errors still decrease with ϵ, which confirms the tradeoff between privacy and optimization performance in the optimal control problem. With the proposed algorithm, a multi-agent control system can protect the privacy of the agents, at the expense of (slight) degradation in control performance. For instance, in Fig. 3, when the required privacy level is $\epsilon = 0.5$, the relative error of the privacy-preserving controls is smaller than 10^{-3}. This can be very useful in practice. For example, in robotic systems or sensor networks deployed in adversarial environments such as battlegrounds, the locations of the robots or sensors can be sensitive private information, which should not be inferred by potential eavesdroppers accurately. The proposed algorithm can be used to control the robots and sensors in a nearly optimal way while protecting their location information.

V. Conclusion

In this paper, we have developed a differentially private ADMM for regularized consensus optimization by perturbing the public signals with additive Laplacian noise. We have derived the conditions on the noise parameters under which the proposed algorithm preserves ϵ-differential privacy. Further, the convergence performance of the algorithm has been analyzed. We have found that to achieve the best convergence with given privacy requirement, the magnitude of the added noise should decrease as the algorithm progresses. In addition, there exists a finite optimal number of iterations, in the sense of balancing the tradeoff between the convergence of ADMM and the privacy leakage during the algorithm execution. Numerical results have been presented to confirm the analytical properties of the proposed algorithm and an application in multi-agent linear-quadratic control with private information is provided. We note that the results in this paper are mostly theoretical and constitute a preliminary step of making ADMM private. Real-world private optimization problems can be more complicated than the simplified model here and other privacy-preserving techniques may need to be integrated with the techniques in this paper.

References

[1] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC*, pp. 265–284, 2006.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, 2011.

[3] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: the SuLQ framework," in *ACM PODS*, pp. 128–138, 2005.

[4] A. Beimel, S. P. Kasiviswanathan, and K. Nissim, "Bounds on the sample complexity for private learning and private data release," in *Theory of Cryptography Conference*, pp. 437–454, Springer, 2010.

[5] J. Le Ny and G. J. Pappas, "Differentially private filtering," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 341–354, 2014.

[6] M. Ye and A. Barg, "Optimal schemes for discrete distribution estimation under locally differential privacy," *IEEE Transactions on Information Theory*, vol. 64, no. 8, pp. 5662–5676, 2018.

[7] S. Li, A. Khisti, and A. Mahajan, "Information-theoretic privacy for smart metering systems with a rechargeable battery," *IEEE Transactions on Information Theory*, vol. 64, no. 5, pp. 3679–3695, 2018.

[8] P. Venkitasubramaniam, "Privacy in stochastic control: A markov decision process perspective," in *Allerton*, pp. 381–388, 2013.

[9] J. He, L. Cai, and X. Guan, "Preserving data-privacy with added noises: Optimal estimation and privacy analysis," *IEEE Transactions on Information Theory*, vol. 64, no. 8, pp. 5677–5690, 2018.

[10] M. Ruan, H. Gao, and Y. Wang, "Secure and privacy-preserving consensus," *IEEE Trans. Autom. Control*, vol. 64, no. 10, 2019.

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

[12] F. Farokhi, *Privacy in Dynamical Systems*. Springer, 2020.

[13] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed constrained optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 50–64, 2017.

[14] J. Hsu, Z. Huang, A. Roth, and Z. S. Wu, "Jointly private convex programming," in *SODA*, pp. 580–599, 2016.

[15] C. Li, P. Zhou, L. Xiong, Q. Wang, and T. Wang, "Differentially private distributed online learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 8, pp. 1440–1453, 2018.

[16] J. Zhu, C. Xu, J. Guan, and D. O. Wu, "Differentially private distributed online algorithms over time-varying directed networks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 1, pp. 4–17, 2018.

[17] T. Lin, S. Ma, and S. Zhang, "On the global linear convergence of the ADMM with multiblock variables," *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1478–1497, 2015.

[18] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization.," *IEEE Trans. Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[19] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Püschel, "Distributed ADMM for model predictive control and congestion control," in *IEEE CDC*, pp. 5110–5115, 2012.

[20] M. Annergren, A. Hansson, and B. Wahlberg, "An ADMM algorithm for solving ℓ_1 regularized MPC," in *IEEE CDC*, pp. 4486–4491, 2012.

[21] R. Rostami, G. Costantini, and D. Görges, "ADMM-based distributed model predictive control: Primal and dual approaches," in *IEEE CDC*, pp. 6598–6603, 2017.

[22] N. K. Dhingra, M. R. Jovanović, and Z.-Q. Luo, "An ADMM algorithm for optimal sensor and actuator selection," in *CDC*, pp. 4039–4044, 2014.

[23] C. Meissen, L. Lessard, M. Arcak, and A. K. Packard, "Compositional performance certification of interconnected systems using ADMM," *Automatica*, vol. 61, pp. 55–63, 2015.

[24] T. Zhang and Q. Zhu, "Dynamic differential privacy for ADMM-based distributed classification learning," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 172–187, 2017.

[25] X. Zhang, M. M. Khalili, and M. Liu, "Improving the privacy and accuracy of ADMM-based distributed algorithms," in *ICML*, 2018.

[26] P. Tseng, "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM Journal on Control and Optimization*, vol. 29, no. 1, pp. 119–138, 1991.

[27] A. Beck and M. Teboulle, "A fast dual proximal gradient algorithm for convex minimization and applications," *Operations Research Letters*, vol. 42, no. 1, pp. 1–6, 2014.

[28] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[29] C. Dwork, A. Roth, *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[30] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM Journal on Optimization*, vol. 26, no. 1, 2016.

[31] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *ACM ICDCN*, 2015.