Resilient Primal–Dual Optimization Algorithms for Distributed Resource Allocation

Berkay Turan, César A. Uribe, Hoi-To Wai, *Member, IEEE*, and Mahnoosh Alizadeh

***Abstract*—Distributed algorithms for multiagent resource allocation can provide privacy and scalability over centralized algorithms in many cyber-physical systems. However, the distributed nature of these algorithms can render these systems vulnerable to man-in-the-middle attacks that can lead to nonconvergence and infeasibility of resource allocation schemes. In this article, we propose attack-resilient distributed algorithms based on primal–dual optimization when Byzantine attackers are present in the system. In particular, we design attack-resilient primal–dual algorithms for static and dynamic impersonation attacks by means of robust statistics. For static impersonation attacks, we formulate a robustified optimization model and show that our algorithm guarantees convergence to a neighborhood of the optimal solution of the robustified problem. On the other hand, a robust optimization model is not required for the dynamic impersonation attack scenario and we are able to design an algorithm that is shown to converge to a near-optimal solution of the original problem. We analyze the performances of our algorithms through both theoretical and computational studies.**

***Index Terms*—Cyber-physical systems, distributed algorithms, gradient methods, multi-agent systems, optimization methods, robustness.**

I. Introduction

A NUMBER of multiagent optimization problems arise in a wide range of resource allocation systems that fall under the general umbrella of *network utility maximization* problems: in the pioneering example of congestion control in data networks [1], [2]; in determining the optimal price of electricity and enabling more efficient demand–supply balancing in smart power distribution systems [3], [4]; in managing user transmit powers and data rates in wireless cellular networks [5]; in determining optimal caching policies by content delivery networks [6]; in optimizing power consumption in wireless sensor networks with energy-restricted batteries [7], [8]; and in designing congestion control systems in urban traffic networks [9]. The shared goal among the above-mentioned problems is to minimize the sum of N user-specific cost functions, subject to a set of coupling constraints that depend on users' decisions.

In these resource allocation problems, the user-specific cost functions and the set of coupling constraints are considered private information to the users and to a central coordinator, respectively. Consequently, it is necessary to solve these problems in a distributed fashion allowing the agents to cooperate through communication with a central coordinator. Among others, primal–dual optimization methods [10] have been advocated as they naturally give rise to decomposable algorithms that favor distributed implementation [11]. In addition to their practical success, these methods are supported by strong theoretical guarantees where fast convergence to a near-optimal solution is well established [10].

However, the distributed nature of these methods also exposes the system to vulnerabilities not faced by their traditional centralized counterpart. Many of the existing algorithms assume the agents, and the communication channels between the central coordinator and the agents, to be *completely trustworthy*. In this article, we consider the setting where these communications are susceptible to adversarial attacks. An attacker can take over network subsystems, and deliberately edit the messages communicated to the central coordinator to any arbitrary value, i.e., a Byzantine attack. As we will demonstrate, this might result in an unstable system with possible damage to hardware and the overall system.

Our goal is to design attack-resilient primal–dual algorithms in order to solve multiagent resource allocation problems in the presence of Byzantine attackers. If a communication channel is attacked and becomes compromised, the attacker can modify messages and/or inject fresh messages into the network on the agents' behalf. We consider two scenarios with different attacker capabilities. A static impersonation attack scenario considers the set of agents communicating through compromised channels to be the same for the duration of the algorithm, whereas a dynamic impersonation attack scenario considers the case where all agents are susceptible to attacks and, hence, communicate through compromised channels for a *limited fraction* of the algorithm's runtime. Our main contributions are as follows.

Manuscript received April 29, 2020; revised August 5, 2020; accepted August 31, 2020. Date of publication September 17, 2020; date of current version February 26, 2021. This work was supported in part by UCOP under Grant LFR-18-548175, in part by the National Science Foundation under Grant #1847096, in part by the CUHK Direct Grant #4055113, and in part by the Yahoo! Research Faculty Engagement Program. Recommended by Associate Editor M. V. Salapaka. *(Berkay Turan and César A. Uribe contributed equally to this work.) (Corresponding author: Berkay Turan.)*

Berkay Turan and Mahnoosh Alizadeh are with the Department of Electrical and Computer Engineering, University of California–Santa Barbara, Santa Barbara, CA 93106 USA (e-mail: bturan@ucsb.edu; alizadeh@ucsb.edu).

César A. Uribe is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02138-1570 USA (e-mail: cauribe@mit.edu).

Hoi-To Wai is with the Department of SEEM, Chinese University of Hong Kong, Hong Kong (e-mail: htwai@se.cuhk.edu.hk).

Digital Object Identifier 10.1109/TCNS.2020.3024485

2325-5870 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 07,2021 at 18:46:03 UTC from IEEE Xplore. Restrictions apply.

1) We propose resilient distributed resource allocation algorithms under the two aforementioned attack scenarios that rely on robust mean estimation.
2) We provide convergence guarantees of the proposed algorithms. We show that our algorithm for the dynamic impersonation attack scenario converges to the optimal solution of the regularized problem, whereas our algorithm for the static impersonation attack scenario converges to an $\mathcal{O}(\alpha_1^2)$ neighborhood of the optimal solution of a robustified and regularized optimization model, where $\alpha_1 \in [0, \frac{1}{2})$ is a known upper bound on the fraction of attacked channels.
3) We provide empirical evidence that supports our theoretical results on convergence and preventing constraint violation. We do so via computational simulations on electric vehicle (EV) charging and power distribution applications.

Related work: Vulnerabilities of various types of distributed algorithms have been identified and addressed in a number of recent studies. Relevant examples can be found in [12]–[20], which study secure decentralized algorithms on a general network topology but consider consensus-based optimization models. There are two fundamental differences between distributed resource allocation and consensus problems that make these algorithms inapplicable in our case.

1) In resource allocation problems, each agent is solving its own optimal level of resource consumption, i.e., each agent is solving its own parameter, whereas consensus problems focus on all agents solving a shared (global) parameter.
2) Unlike resilient consensus algorithms, in resource allocation problems pertaining to access critical infrastructure systems, such as power or transportation networks, one cannot simply block a set of users' access to the network even if they are deemed likely to be attackers.

A recently popular line of works in [21]–[25] focuses on building resilient algorithms for distributed statistical learning. A crucial difference from this work is that they assume identical functions across the agents. In fact, we employ robust statistics [26], [27] to develop our resilient algorithms, and particularly, we develop novel results for robust mean estimation, a topic that has been recently rekindled in [28]–[30].

This article is a revised and extended version of the preliminary conference report [31]. This article expands [31] into multiple attack scenarios and includes numerical studies.

Paper Organization: The remainder of the article is organized as follows. In Section II, we provide an overview of the basic primal–dual algorithm for resource allocation. In Section III, we formally define two Byzantine attack models and demonstrate how Byzantine attacks can alter the primal–dual optimization procedure. In Section IV, we present two attack-resilient primal–dual algorithms corresponding to the different attack scenarios along with their convergence analysis. In Section V, we provide numerical results for our algorithms. Finally, Section VI concludes this article.

Notations: Unless otherwise specified, $\|\cdot\|$ denotes the standard Euclidean norm. For any $N \in \mathbb{N}$, $[N]$ denotes the finite set $\{1, \ldots, N\}$. Given $\boldsymbol{\theta}$, $\boldsymbol{\theta}_i$ indicates the ith block/entry of $\boldsymbol{\theta}$ that corresponds to the parameter of agent i. $\boldsymbol{\theta}_{i,j}$ denotes the jth element of vector $\boldsymbol{\theta}_i$.

Algorithm 1: PD-DRA Procedure.

```
for $k = 1, 2, \ldots$ do
    (Communication stage):
    1) Central coordinator receives $\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N$ from agents and computes $\overline{\boldsymbol{\theta}}^{(k)} := \frac{1}{N}\sum_{i=1}^N \boldsymbol{\theta}_i^{(k)}$, $\{\nabla_{\boldsymbol{\theta}} g_t(\overline{\boldsymbol{\theta}}^{(k)})\}_{t=1}^T$.
    2) Central coordinator broadcasts the vector $\overline{\boldsymbol{g}}^{(k)} := \sum_{t=1}^T \lambda_t^{(k)} \nabla_{\boldsymbol{\theta}} g_t(\overline{\boldsymbol{\theta}}^{(k)})$ to agents.
    (Computation stage):
    1) Agent $i$ computes the update for $\boldsymbol{\theta}_i^{(k+1)}$ according to (4a) using the received $\overline{\boldsymbol{g}}^{(k)}$.
    2) The central coordinator computes the update for $\boldsymbol{\lambda}^{(k+1)}$ according to (4b).
end for
```

II. Overview of Prima–Dual Algorithm for Resource Allocation

We consider the following multiagent optimization problem with an objective to minimize the average cost incurred by the agents, subject to a set of constraints that are functions of the average of the agents' parameters:

$$\begin{aligned} \min_{\boldsymbol{\theta}_i \in \mathbb{R}^d \forall i} \quad & f(\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^N f_i(\boldsymbol{\theta}_i) \\ \text{subject to} \quad & g_t\left(\tfrac{1}{N}\textstyle\sum_{i=1}^N \boldsymbol{\theta}_i\right) \leq 0, \qquad t = 1, \ldots, T \\ & \boldsymbol{\theta}_i \in \mathcal{C}_i, \qquad\qquad\qquad i = 1, \ldots, N \end{aligned} \tag{1}$$

where $f_i(\cdot) : \mathbb{R}^d \to \mathbb{R}$ is the continuously differentiable and convex cost function of agent i and $g_t(\cdot) : \mathbb{R}^d \to \mathbb{R}$ are a continuously differentiable and convex set of constraints. The parameter $\boldsymbol{\theta}_i$ of agent i is constrained to be in a compact convex set $\mathcal{C}_i \in \mathbb{R}^d$.

Running Example (Resource Allocation Problem) Throughout the article, we use the following toy example as a running example to clarify the concepts and the methods. We consider an EV charging example with 5 agents. The cost function $f_i(\cdot)$ is monotone decreasing and is the same for all agents. As an example, we set $f_i(\boldsymbol{\theta}_i) = (\boldsymbol{\theta}_i - 10)^2$ as the quadratic cost function that is monotonically decreasing for $0 \leq \boldsymbol{\theta}_i \leq 10$. There is a charging station with five EV charging points, three of which have a maximum charging rate of 7 kW, and two have a rate of 10 kW. The total rate at which the charging station is able to deliver electricity is determined by the grid, and let it be upper bounded by 25 kW (hence, the average rate is upper bounded by $25/5 = 5$ kW). Accordingly, the constraints of this system are stated as

$$\begin{aligned} & g\left((1/5)\textstyle\sum_{i=1}^5 \boldsymbol{\theta}_i\right) := (1/5)\textstyle\sum_{i=1}^5 \boldsymbol{\theta}_i - 5 \leq 0 \\ & 0 \leq \boldsymbol{\theta}_i \leq 7, \quad i = 1, 2, 3 \\ & 0 \leq \boldsymbol{\theta}_i \leq 10, \quad i = 4, 5. \end{aligned}$$

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 07,2021 at 18:46:03 UTC from IEEE Xplore. Restrictions apply.

Note that $\boldsymbol{\theta}$ is a real number, hence dimension $d = 1$. The optimal solution in this example is to deliver electricity at a rate of 5 kW to all agents due to symmetry.

The optimization problem in (1) cannot be solved centrally, because the utility functions $f_i(\cdot)$ are private to the agents, and furthermore the coupling constraints on the resources are only known by a central coordinator. Accordingly, the goal of the primal–dual distributed resource allocation (PD-DRA) procedure in Algorithm 1 is to solve (1) in a distributed manner, where the agents observe a pricing signal received from the central coordinator and communicate their parameters to the central coordinator [10]. Consequent to this information exchange, the pricing signal and the agents' parameters are updated by the central coordinator and by the individual agents, respectively.

In order to derive the update rules used by Algorithm 1, we first consider the Lagrangian function of (1)

$$\mathcal{L}(\{\boldsymbol{\theta}_i\}_{i=1}^N; \boldsymbol{\lambda}) := \frac{1}{N}\sum_{i=1}^N f_i(\boldsymbol{\theta}_i) + \sum_{t=1}^T \lambda_t g_t\left(\frac{1}{N}\sum_{i=1}^N \boldsymbol{\theta}_i\right) \quad (2)$$

where $\lambda_t \geq 0$ is the dual variable associated with constraint $g_t(\cdot)$ and $\boldsymbol{\lambda} = [\lambda_1 \ldots \lambda_T]^\intercal \in \mathbb{R}_+^T$ is the vector of the dual variables. Under strong duality (e.g., when Slater's condition holds), solving problem (1) is equivalent to solving its dual problem

$$\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^T} \min_{\boldsymbol{\theta}_i\in\mathcal{C}_i \forall i} \mathcal{L}(\{\boldsymbol{\theta}_i\}_{i=1}^N; \boldsymbol{\lambda}). \quad (\mathrm{P})$$

As suggested in [10], we consider a regularized version of (P). Let us define

$$\begin{aligned}&\mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i\}_{i=1}^N; \boldsymbol{\lambda})\\ &:= \mathcal{L}(\{\boldsymbol{\theta}_i\}_{i=1}^N; \boldsymbol{\lambda}) + \tfrac{\upsilon}{2N}\textstyle\sum_{i=1}^N \|\boldsymbol{\theta}_i\|^2 - \tfrac{\upsilon}{2}\|\boldsymbol{\lambda}\|^2\end{aligned} \quad (3)$$

such that $\mathcal{L}_\upsilon(\cdot)$ is υ-strongly convex and υ-strongly concave in $\{\boldsymbol{\theta}_i\}_{i=1}^N$ and $\boldsymbol{\lambda}$, respectively.

Remark 1: Adding regularization terms is a typical technique used in optimization, called dual smoothing [32]. We add the regularization terms for the purposes of convergence analysis used in this article, which can be applied on strongly convex/concave functions. Indeed, adding the regularization terms might change the solution of the original optimization problem. However, as explained in [33, Prop. 5.2], by an appropriate selection of the regularization parameters, we can recover an optimality gap guarantee for the original problem based on the solution to the regularized problem.

We define the regularized problem as

$$\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^T} \min_{\boldsymbol{\theta}_i\in\mathcal{C}_i \forall i} \mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i\}_{i=1}^N; \boldsymbol{\lambda}). \quad (\mathrm{P}_\upsilon).$$

Let $\gamma > 0$ be the step size and $k \in \mathbb{Z}_+$ be the iteration index. The primal–dual recursion performs projected gradient descent/ascent on the primal/dual variables as follows:

$$\boldsymbol{\theta}_i^{(k+1)} = \mathcal{P}_{\mathcal{C}_i}\left(\boldsymbol{\theta}_i^{(k)} - \gamma\nabla_{\boldsymbol{\theta}_i}\mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N; \boldsymbol{\lambda}^{(k)})\right) \forall i \in [N] \quad (4a)$$

$$\boldsymbol{\lambda}^{(k+1)} = \left[\boldsymbol{\lambda}^{(k)} + \gamma\nabla_{\boldsymbol{\lambda}}\mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N; \boldsymbol{\lambda}^{(k)})\right]_+ \quad (4b)$$

where $\mathcal{P}_{\mathcal{C}_i}(\cdot)$ is the Euclidean projection operator to set $\mathcal{C}_i$ and $[\cdot]_+$ denotes the $\max\{\cdot, 0\}$ operator. According to (3), the gradients with respect to (w.r.t.) the primal and the dual variables are given, respectively, by

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_i}\mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N; \boldsymbol{\lambda}^{(k)}) = \frac{1}{N}\Big(&\nabla_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i^{(k)}) + \upsilon\boldsymbol{\theta}_i^{(k)}\\ &+ \textstyle\sum_{t=1}^T \lambda_t^{(k)}\nabla_{\boldsymbol{\theta}} g_t(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\frac{1}{N}\sum_{i=1}^N\boldsymbol{\theta}_i^{(k)}}\Big)\end{aligned} \quad (5a)$$

$$\left[\nabla_{\boldsymbol{\lambda}}\mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N; \boldsymbol{\lambda}^{(k)})\right]_t = g_t\left(\tfrac{1}{N}\textstyle\sum_{i=1}^N\boldsymbol{\theta}_i^{(k)}\right) - \upsilon\lambda_t^{(k)} \quad (5b)$$

for all i, t. It is worthwhile to highlight that both gradients depend on the average parameter $\overline{\boldsymbol{\theta}}^{(k)} := \frac{1}{N}\sum_{i=1}^N \boldsymbol{\theta}_i^{(k)}$. From (5a) and (5b), we can determine which variables should be communicated between the central coordinator and the agents so that the gradients can be computed locally (see Algorithm 1).

Since the regularized primal–dual problem is strongly convex/concave in primal/dual variables, Algorithm 1 converges linearly to the optimal solution of (P_υ) [10]. To study this, let us concatenate the primal and the dual variables and denote $\boldsymbol{z}^{(k)} := (\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N, \boldsymbol{\lambda}^{(k)})$ as the primal–dual variable at the kth iteration and define the mapping $\boldsymbol{\Phi}(\boldsymbol{z}^{(k)})$ as

$$\boldsymbol{\Phi}(\boldsymbol{z}^{(k)}) := \begin{pmatrix}\nabla_{\boldsymbol{\theta}}\mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N, \boldsymbol{\lambda}^{(k)})\\ -\nabla_{\boldsymbol{\lambda}}\mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N, \boldsymbol{\lambda}^{(k)})\end{pmatrix}. \quad (6)$$

Proposition 1: (see[10, Th. 3.5]) Assume that the map $\boldsymbol{\Phi}(\boldsymbol{z}^{(k)})$ is L_Φ Lipschitz continuous. For all $k \geq 1$, we have

$$\|\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^\star\|^2 \leq (1 - 2\gamma\upsilon + \gamma^2 L_\Phi^2)\|\boldsymbol{z}^{(k)} - \boldsymbol{z}^\star\|^2 \quad (7)$$

where $\boldsymbol{z}^\star$ is a saddle point to the (P_υ). Setting $\gamma = \upsilon/L_\Phi^2$ gives $\|\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^\star\|^2 \leq (1 - \upsilon^2/L_\Phi^2)\|\boldsymbol{z}^{(k)} - \boldsymbol{z}^\star\|^2 \; \forall\, k \geq 1$.

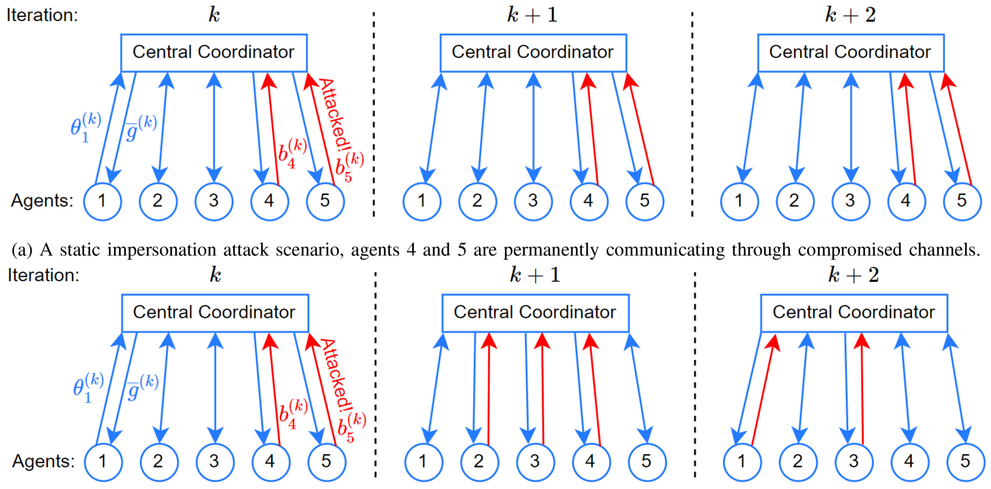
III. Problem Formulation

Even though the PD-DRA provides strong theoretical convergence guarantee, it relies on error-free communication between the central coordinator and the agents, and is not robust to attacks on the channels between the agents and the central coordinator, as described below.

We study a situation when the *uplink* communication channels between some of the agents and the central coordinator are compromised.[1] Let $\mathcal{A}^{(k)} \subset [N]$ be the set of agents communicating through *compromised uplink channels* at iteration k, whose identities are unknown to the central coordinator, and let $\mathcal{H}^{(k)} := [N] \setminus \mathcal{A}^{(k)}$ be the set of agents communicating through *trustworthy uplink channels* at iteration k. Instead of

[1] This article studies the case where only uplink channels are compromised. However, the case of downlink corruption can also be addressed. Since the downlink channel is a broadcast channel, a compromised downlink channel results in no agent receiving a trustworthy pricing signal. In that case, there is no optimization method based solution to that problem since there is no communication. If we assume however that all the downlink channels are point-to-point between the central coordinator and each agent, the methods developed in this article can be applied in a similar fashion.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 07,2021 at 18:46:03 UTC from IEEE Xplore. Restrictions apply.



(a) A static impersonation attack scenario, agents 4 and 5 are permanently communicating through compromised channels.

(b) A dynamic impersonation scenario, where the set of agents communicating through compromised channels are changing.

Fig. 1. Illustration of (a) static impersonation attack and (b) dynamic impersonation attack. Blue arrows represent trustworthy channels, whereas red arrows represent compromised channels. (a) Static impersonation attack scenario, agents 4 and 5 are permanently communicating through compromised channels. (b) Dynamic impersonation scenario, where the set of agents communicating through compromised channels is changing.

receiving $\boldsymbol{\theta}_i^{(k)}$ from each agent $i \in [N]$ at iteration k (Algorithm 1 Step 2.1), the central coordinator receives the following messages:

$$\boldsymbol{r}_i^{(k)} = \begin{cases} \boldsymbol{\theta}_i^{(k)}, & \text{if } i \in \mathcal{H}^{(k)} \\ \boldsymbol{b}_i^{(k)}, & \text{if } i \in \mathcal{A}^{(k)}. \end{cases} \tag{8}$$

We consider a Byzantine attack scenario, under which the messages sent through the compromised channels, $\boldsymbol{b}_i^{(k)}$, can be chosen arbitrarily by an adversary. This also encompasses faulty messages due to erroneous inputs or erroneous channels, since we set no restrictions on $\boldsymbol{b}_i^{(k)}$. The adversary's goal is to harm the system and cause suboptimalities. When the messages are erroneous or chosen adversarially, the central coordinator computes the gradients and therefore the pricing signal using these erroneous messages. The agents then update their parameters based on this erroneous pricing signal, which can lead to an overall suboptimal resource allocation. Moreover, the choice of the compromised channels $\mathcal{A}^{(k)}$ affects the impact of the attack and the precautions to be taken in order to defend against the attack. As such, we study two Byzantine attack scenarios that differ in the set of the compromised channels, as illustrated in Fig. 1.

Running Example (Byzantine Attack) Let agent 1 be communicating through a compromised channel at all iterations, i.e., $\mathcal{A}^{(k)} = \{1\}\ \forall k$. The compromised message sent to the central coordinator is $\boldsymbol{b}_1^{(k)} = 1$ kW $\forall k$. This means that irrespective of $\boldsymbol{\theta}_1^{(k)}$, the central coordinator receives a message indicating agent 1 is willing to charge at rate of 1 kW.

A. Attack Scenarios

1) A *static impersonation attack*, where an adversary takes over a subset of uplink channels permanently and the set of agents communicating through compromised channels is fixed (i.e., $\mathcal{A}^{(k)} = \mathcal{A}\,\forall k$). Consequently, the central coordinator is never able to communicate reliably with agents $i \in \mathcal{A}$. In this case, it is not feasible to optimize the original problem (P) since the contribution from $f(\boldsymbol{\theta}_i) : i \in \mathcal{A}$ becomes unknown to the central coordinator. Yet, we assume that it is also not possible to deny access to resources for agents who are suspected of potentially being under attack. As a compromise, we formulate the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\theta}_i \in \mathcal{C}_i, i \in \mathcal{H}} \quad & f(\boldsymbol{\theta}) := \frac{1}{N} \sum_{i \in \mathcal{H}} f_i(\boldsymbol{\theta}_i) \\ \text{subject to} \quad & \max_{\boldsymbol{\theta}_j \in \mathcal{C}_j, j \in \mathcal{A}} g_t\left(\tfrac{1}{N} \textstyle\sum_{i=1}^{N} \boldsymbol{\theta}_i\right) \leq 0 \quad \forall t \in [T]. \end{aligned} \tag{9}$$

The objective of (9) is to minimize the cost of the agents with trustworthy channels subject to a *robust* set of constraints that consider the *worst case* scenario, in which the parameters of the agents with compromised channels are assumed to be maximizing the constraints (e.g., those agents are assumed to be consuming the maximum amount of resources). It is critical to mention that during a primal–dual algorithm scheme, the messages received through the compromised channels can be anything. The robust approach is to however ignore those messages and assume that the parameters of the agents communicating through those channels are maximizing the constraints so that the operation of the system is feasible under any circumstance. Our goal is to develop an attack-resilient PD-DRA to solve the robust optimization problem (9).

Running Example (Robust Optimization Model) Since agent 1 is sending a compromised message of 1 kW and their true parameter can be anything, the worst case approach is to assume that they are charging at the maximum

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 07,2021 at 18:46:03 UTC from IEEE Xplore. Restrictions apply.

rate, which is 7 kW for that agent. Hence, the robust constraint is

$$\max_{\boldsymbol{\theta}_j \in \mathcal{C}_j, j\in\mathcal{A}} g\left(\frac{1}{5}\sum_{i=1}^{5}\boldsymbol{\theta}_i\right) = \max_{\boldsymbol{\theta}_j \in \mathcal{C}_j, j\in\mathcal{A}} \frac{1}{5}\sum_{i=1}^{5}\boldsymbol{\theta}_i - 5$$

$$= \frac{1}{5}\sum_{i\in\mathcal{H}}\boldsymbol{\theta}_i + \max_{\boldsymbol{\theta}_j \in \mathcal{C}_j, j\in\mathcal{A}} \frac{1}{5}\sum_{j\in\mathcal{A}}\boldsymbol{\theta}_j - 5 = \frac{4}{5}\overline{\boldsymbol{\theta}}_{\mathcal{H}} - 3.6$$

where we used $|\mathcal{H}| = 4$ and the notation $\overline{\boldsymbol{\theta}}_{\mathcal{H}} = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\boldsymbol{\theta}_i$. The robust constraint states that

$$\frac{4}{5}\overline{\boldsymbol{\theta}}_{\mathcal{H}} - 3.6 \le 0 \Rightarrow \overline{\boldsymbol{\theta}}_{\mathcal{H}} \le 4.5.$$

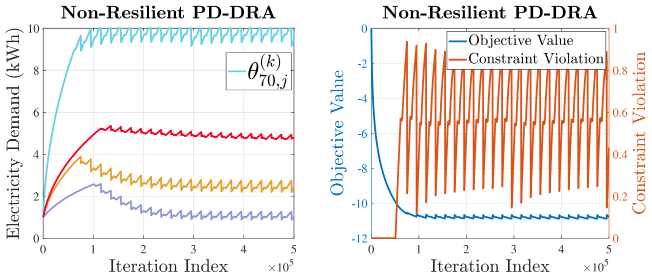
The optimal solution in this case is to deliver electricity at a rate of 4.5 kW to the trustworthy agents. Since the compromised agent has the same cost function, their true charging rate will also be 4.5 kW, even though the message sent is 1 kW and the central coordinator assumes their charging rate is 7 kW.

2) A *dynamic impersonation attack*, where all the agents might be affected by the adversarial attacks but only for a *limited* fraction of time and hence, the set of agents communicating through compromised channels $\mathcal{A}^{(k)}$ has to dynamically change with iteration k. As opposed to the static case, this scenario considers the case where the central coordinator is able to communicate reliably with all the agents at some iterations. Due to this distinction, it is necessary to mention that the static attack is not a special case of the dynamic attack and both scenarios are distinguishable from each other. The dynamic scenario could be applicable when agents do not have dedicated communication channels to the central coordinator and instead communicate over random access systems, which are more appropriate for distributed deployments. Hence, each user periodically accesses authenticated network devices/subsystems that are controlled by Byzantine adversaries and can alter the user's message. Our goal is to develop an attack-resilient PD-DRA algorithm that can still solve the original regularized problem (P_{υ}) in this environment.

B. Limitations of the Basic PD-DRA Algorithm

Applying the basic PD-DRA algorithm under a Byzantine attack scenario can lead to undesirable outcomes. Recall that the gradients in (5) depend on the average parameter $\overline{\boldsymbol{\theta}}^{(k)}$. Under a Byzantine attack scenario, if the central coordinator forms the naive average $\widetilde{\boldsymbol{\theta}}^{(k)} = (1/N)\sum_{i=1}^{N}\boldsymbol{r}_i^{(k)}$ and computes the gradients $\nabla g_t(\widetilde{\boldsymbol{\theta}}^{(k)})$ accordingly, this may result in large error since the deviation $\widetilde{\boldsymbol{\theta}}^{(k)} - (1/N)\sum_{i=1}^{N}\boldsymbol{\theta}_i^{(k)}$ can be large (proportional to the maximum diameter of $\mathcal{C}_i$s). This in turn can obstruct convergence and also overload the system by causing constraint violations.

Running Example (Basic PD-DRA Failure) If the central coordinator believes all the agents are sending trustworthy information, then the optimal solution will occur when one agent is demanding 1 kW and the others are demanding 6 kW (so that the average is 5 kW). But since the 1 kW message is compromised and all the agents have same cost function, the compromised agent's true electricity demand is also at a rate of 6 kW. Hence, the solution delivers electricity at an average rate of 6 kW, which is infeasible.



(a) (b)

Fig. 2. Illustration of basic PD-DRA algorithm failure under static impersonation attack. (a) Agents' parameters do not converge. (b) Objective function does not converge and moreover there is constraint violation. We only display one constraint for brevity.

We preview our numerical result of applying the basic PD-DRA method under a static impersonation attack scenario for an optimal EV charging application in Fig. 2. For constraint $g_t(\cdot)$, we define constraint violation as $\max\{0, g_t(\overline{\boldsymbol{\theta}}^{(k)})\}$. Observe that the PD-DRA method does not provide convergence and the first constraint is being violated. From resource allocation perspective, this means that the agents are asking to consume more resources than the available amount in the system, which is infeasible. For details regarding the experimental setup, please see Section V.

IV. Resilient PD-DRA Algorithms

Motivated by the failure of the basic PD-DRA procedure under Byzantine attack scenarios, resilient PD-DRA algorithms are necessary to optimize multiagent systems in a distributed manner when the system is susceptible to attacks. We hold the following assumption to be true throughout the rest of the article and propose two different attack resilient PD-DRA algorithms corresponding to the different attack scenarios outlined in Section III.

Assumption 1: For all $\boldsymbol{\theta} \in \mathbb{R}^d$ and for all t, the gradient of g_t is bounded with $\|\nabla g_t(\boldsymbol{\theta})\| \le B$ and is L-Lipschitz continuous. Moreover, since maximum resource that can be consumed by an agent is bounded due to limited amount of resources, we let $\mathbf{0} \in \mathcal{C}_i$ and upper bound the diameters of $\mathcal{C}_i$ by R

$$\max_{\boldsymbol{\theta},\boldsymbol{\theta}' \in \mathcal{C}_i} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\| \le R,\ i = 1, \ldots, N. \tag{10}$$

Running Example (Assumptions) The constraint in our running example satisfies $\nabla g(\boldsymbol{\theta}) = 1$, which is bounded by $B = 1$ and is $L = 0$-Lipschitz continuous. Since the maximum charging rate is upper bounded by 7 kW for three of the agents and by 10 kW for two of the agents, $R = 10$.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 07,2021 at 18:46:03 UTC from IEEE Xplore. Restrictions apply.

A. Static Impersonation Attack

Under this attack scenario, given the complete lack of any credible information on the resource consumption parameters of the agents that permanently communicate through compromised channels, the central coordinator can only hope to solve the robust optimization model defined in (9) instead. This formulation considers a worst case scenario on how much resources the compromised agents will consume, which ensures constraint satisfaction in all cases. However, the constraints in (9) require the knowledge of the set $\mathcal{A}$ and the sets $\mathcal{C}_j \forall j \in \mathcal{A}$, yet the central coordinator lacks this information.

Hence, in order to develop a robust optimization model that can handle the worst case scenario without the knowledge of $\mathcal{A}$, we let $\alpha_1 \geq |\mathcal{A}|/N$ as a known upper bound to the fraction of agents communicating through compromised channels and assume $\alpha_1 < 1/2$, where less than half of the agents are communicating through compromised channels.[2] Let $\overline{\boldsymbol{\theta}}_{\mathcal{H}} := \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \boldsymbol{\theta}_i$ be the mean of the agent's parameters that are sent through trustworthy channels. We then define the following set of constraints:

$$\overline{g}_t(\boldsymbol{\theta}) := g_t(\boldsymbol{\theta}) + \alpha_1\left(RB + \tfrac{1}{2}LR^2\right) \tag{11}$$

and formulate a conservative approximation of (9).

Lemma 1: Under Assumption 1, the following problem yields a conservative approximation of (9), i.e., its feasible set is a subset of the feasible set of (9):

$$\begin{aligned} \min_{\boldsymbol{\theta}_i\in\mathcal{C}_i, i\in\mathcal{H}} \quad & \frac{1}{N}\sum_{i\in\mathcal{H}} f_i(\boldsymbol{\theta}_i) \\ \text{subject to} \quad & \overline{g}_t\left((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}\right) \leq 0 \, \forall t \in [T]. \end{aligned} \tag{12}$$

The proof can be found in Appendix A, which is provided in the online version [34]. All proofs and Appendices can be found in the online version [34].

Remark 2: The proof of Lemma 1 is done by upper bounding constraints of (9) using Assumption 1 and the fact that $\alpha_1 \geq |\mathcal{A}|/N$. The looser these upper bounds compared to the true values, the more conservative is (12). This approach potentially leaves less resources available to the agents communicating through trustworthy channels by assuming more than $|\mathcal{A}|$ number of agents having maximum possible impact on the constraints, irrespective of their set $\mathcal{C}_i$ or the true value/gradient of the constraints.

Running Example (Conservative Approximation) With $B = 1$, $L = 0$, and $R = 10$, the conservative approximation of the running example has the following constraint:

$$\begin{aligned} \overline{g}((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}) &= g((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}) + \alpha_1\left(RB + \frac{1}{2}LR^2\right) \\ &= (1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}} - 5 + 10\alpha_1. \end{aligned}$$

If $\alpha_1 = |\mathcal{A}|/N = 0.2$, then the upper bound is the fraction of compromised channels. In that case, the constraint is

$$0.8\overline{\boldsymbol{\theta}}_{\mathcal{H}} - 3 \leq 0 \Leftrightarrow \overline{\boldsymbol{\theta}}_{\mathcal{H}} \leq 3.75$$

which is more conservative compared to the constraint of the robust optimization model (which was $\overline{\boldsymbol{\theta}}_{\mathcal{H}} \leq 4.5$). The optimal solution in this case is to deliver electricity at a rate of 3.75 kW to the agents. The conservatism arises due to the difference between agent-specific maximum charging rate 7 kW and the absolute maximum charging rate 10 kW. Since the constraint is linear, the gradient is constant. Hence, the smoothness and Lipschitz bounds hold with equality without causing additional conservatism.

If however $\alpha_1 = 0.4$, then the central coordinator assumes two agents communicating through compromised channels. In this case, the conservative approximation has the constraint as

$$0.6\overline{\boldsymbol{\theta}}_{\mathcal{H}} - 1 \leq 0 \Leftrightarrow \overline{\boldsymbol{\theta}}_{\mathcal{H}} \leq \frac{5}{3}$$

which results in charging at an even slower rate since the central coordinator has to be robust against two agents charging at the maximum rate of 10 kW.

To develop an attack resilient PD-DRA algorithm, we again define the regularized Lagrangian function of (12)

$$\begin{aligned} &\overline{\mathcal{L}}_\upsilon(\{\boldsymbol{\theta}_i\}_{i\in\mathcal{H}}; \boldsymbol{\lambda}; \mathcal{H}) \\ &:= \tfrac{1}{N}\textstyle\sum_{i\in\mathcal{H}} f_i(\boldsymbol{\theta}_i) + \sum_{t=1}^T \lambda_t \overline{g}_t\left((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}\right) \\ &\quad + \tfrac{\upsilon}{2N}\textstyle\sum_{i\in\mathcal{H}} \|\boldsymbol{\theta}_i\|^2 - \tfrac{\upsilon}{2}\|\boldsymbol{\lambda}\|^2. \end{aligned} \tag{13}$$

The above function is $(1-\alpha_1)\upsilon$-strongly convex and concave in $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$, respectively (since $(1-\alpha_1) \leq \frac{|\mathcal{H}|}{N} \leq 1$). Our main task is to tackle the following modified problem of (P) under Byzantine attack on (some of) the uplinks:

$$\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^T} \min_{\boldsymbol{\theta}_i\in\mathcal{C}_i \forall i\in\mathcal{H}} \overline{\mathcal{L}}_\upsilon(\{\boldsymbol{\theta}_i\}_{i\in\mathcal{H}}; \boldsymbol{\lambda}; \mathcal{H}).(\mathrm{P}'_\upsilon)$$

Notice that (P'_υ) bears a similar form as (P) and thus one may apply the PD-DRA method to the former. The gradients w.r.t. primal/dual variables are given by

$$\begin{aligned} &\nabla_{\boldsymbol{\theta}_i}\overline{\mathcal{L}}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i\in\mathcal{H}}; \boldsymbol{\lambda}^{(k)}; \mathcal{H}) = \frac{1}{N}\Bigg(\nabla_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i^{(k)}) + \upsilon\boldsymbol{\theta}_i^{(k)}, \\ &+ \tfrac{(1-\alpha_1)N}{|\mathcal{H}|}\textstyle\sum_{t=1}^T \lambda_t^{(k)} \nabla_{\boldsymbol{\theta}}\overline{g}_t(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=(1-\alpha_1)\bar{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}}\Bigg) \quad \forall i \in \mathcal{H} \end{aligned} \tag{14a}$$

$$\left[\nabla_{\boldsymbol{\lambda}}\overline{\mathcal{L}}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i\in\mathcal{H}}; \boldsymbol{\lambda}^{(k)}; \mathcal{H})\right]_t = \overline{g}_t\left((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\right) - \upsilon\lambda_t^{(k)}. \tag{14b}$$

However, such an application requires the central coordinator to compute the sample average

$$\overline{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)} = \tfrac{1}{|\mathcal{H}|}\textstyle\sum_{i\in\mathcal{H}} \boldsymbol{\theta}_i^{(k)} \tag{15}$$

at each iteration. The above might not be computationally feasible under the attack model, since the central coordinator is oblivious to the identity of $\mathcal{H}$. As a solution, the central coordinator computes the robust mean $\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}$ of the received

[2] If more than half of the agents communicate through compromised channels, then the adversary controls the majority and therefore the median, which will be used to estimate the average parameter later in the article. In that case, there is no optimization-based solution the central coordinator can implement in order to securely run the system.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 07,2021 at 18:46:03 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2: Robust PD-DRA Algorithm.

```
Input: Each agent has initial state $\boldsymbol{\theta}_i^{(0)}$.
for $k = 1, 2, \ldots$ do
    (At the Central Coordinator):
    1) Receives $\{\boldsymbol{r}_i^{(k)}\}_{i=1}^N$, see (8), from agents.
    2) Computes robust mean $\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}$ using the estimator (16).
    3) Broadcasts the vector
       $\widehat{\boldsymbol{g}}_{\mathcal{H}}^{(k)} := \sum_{t=1}^T \lambda_t^{(k)} \nabla_{\boldsymbol{\theta}} \overline{g}_t((1-\alpha_1)\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)})$ to agents.
    4) Computes the update for $\boldsymbol{\lambda}^{(k+1)}$ with (18b).
    (At each agent $i \in \mathcal{H}$):
    1) Agent receives $\widehat{\boldsymbol{g}}_{\mathcal{H}}^{(k)}$.
    2) Agent computes update for $\boldsymbol{\theta}_i^{(k+1)}$ with (18a).
end for
```

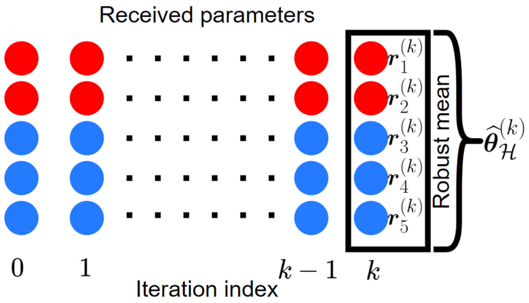


Fig. 3. Robust mean estimation under static impersonation attack. Red/blue circles correspond to parameters received through compromised/trustworthy channels, respectively. In this example, there are $N = 5$ agents and agents 1 and 2 are always communicating through compromised channels. At iteration k, the central coordinator computes the robust mean $\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}$ of the received parameters $\{\boldsymbol{r}_i^{(k)}\}_{i\in[N]}$.

parameters $\{\boldsymbol{r}_i^{(k)}\}_{i\in[N]}$ using a median-based mean estimator described next.

1) Overview of Median-Based Mean Estimation: Consider a set of N vectors $\{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^N$, among which at least $(1-\alpha_1)N$ are trustworthy ($\boldsymbol{x}_i \in \mathcal{H}$) and at most $\alpha_1 N$ are compromised ($\boldsymbol{x}_i \in \mathcal{A}$). We consider a simple median-based estimator applied to each coordinate $j = 1, \ldots, d$. First, define the coordinatewise median as

$$[\boldsymbol{x}_{med}]_j = med\left(\{[\boldsymbol{x}_i]_j\}_{i=1}^N\right)$$

where $med(\cdot)$ computes the median of the operand. Then, our estimator is computed as the mean of the nearest $(1-\alpha_1)N$ neighbors of $[\boldsymbol{x}_{med}]_j$. Our estimator is

$$[\widehat{\boldsymbol{x}}_{\mathcal{H}}]_j = \tfrac{1}{(1-\alpha_1)N} \textstyle\sum_{i\in\mathcal{N}_j} [\boldsymbol{x}_i]_j \tag{16}$$

where we have defined the set with $|\mathcal{N}_j| = (1-\alpha_1)N$ as

$$\mathcal{N}_j = \{i \in [N] : \left|[\boldsymbol{x}_i - \boldsymbol{x}_{med}]_j\right| \le r_j\}$$

such that r_j is chosen to satisfy $|\mathcal{N}_j| = (1-\alpha_1)N$.

The following bounds the performance of (16).

Proposition 2: Let $\overline{\boldsymbol{x}}_{\mathcal{H}}$ be the mean of the trustworthy vectors. Suppose that $\max_{i\in\mathcal{H}} \|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}\|_\infty \le r$, then for any $\alpha_1 \in (0, \frac{1}{2})$, it holds that

$$\|\widehat{\boldsymbol{x}}_{\mathcal{H}} - \overline{\boldsymbol{x}}_{\mathcal{H}}\| \le \frac{2\alpha_1}{1-\alpha_1}\left(1 + \sqrt{\frac{(1-\alpha_1)^2}{1-2\alpha_1}}\right) r\sqrt{d}. \tag{17}$$

The proof can be found in Appendix B of the online version [34]. We note that for sufficiently small α_1, the right-hand side on (17) can be approximated by $\mathcal{O}(\alpha_1 r\sqrt{d})$. Using this median-based mean estimator, we propose the robust PD-DRA algorithm as follows.

2) Robust PD-DRA Algorithm: We summarize the static impersonation attack resilient PD-DRA method in Algorithm 2. The algorithm behaves similarly as Algorithm 1 applied to (P'_υ), with the exception that the central coordinator is oblivious to $\mathcal{H}$, and it uses a robust mean estimator to find an approximate average for the signals sent through the trustworthy links, as illustrated in Fig. 3. This approximate value is used to compute the new price signals, and sent back to agents. In particular, the primal–dual updates are

$$\boldsymbol{\theta}_i^{(k+1)} = \mathcal{P}_{\mathcal{C}_i}\left(\boldsymbol{\theta}_i^{(k)} - \tfrac{\gamma}{N}\left(\widehat{\boldsymbol{g}}_{\mathcal{H}}^{(k)} + \nabla_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i^{(k)}) + \upsilon\boldsymbol{\theta}_i^{(k)}\right)\right) \tag{18a}$$

$$\lambda_t^{(k+1)} = \left[\lambda_t^{(k)} + \gamma\left(\overline{g}_t((1-\alpha_1)\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}) - \upsilon\lambda_t^{(k)}\right)\right]_+. \tag{18b}$$

We note that the update rule in (18a) is valid for agents in set $\mathcal{H}$, because the gradients of the Lagrangian are defined only for those agents in (14a). The agents in set $\mathcal{A}$ may or may not use the same update rule, however, this does not have any impact on the algorithm as they can never communicate their true parameters to the central coordinator.

Lemma 2: Algorithm 2 is a primal–dual algorithm [10] for (P'_υ) with perturbed gradients

$$\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(k)} = \nabla_{\boldsymbol{\theta}}\overline{\mathcal{L}}_\upsilon(\boldsymbol{\theta}^{(k)}; \boldsymbol{\lambda}^{(k)}; \mathcal{H}) + \boldsymbol{e}_{\boldsymbol{\theta}}^{(k)} \tag{19a}$$

$$\widehat{\boldsymbol{g}}_{\boldsymbol{\lambda}}^{(k)} = \nabla_{\boldsymbol{\lambda}}\overline{\mathcal{L}}_\upsilon(\boldsymbol{\theta}^{(k)}; \boldsymbol{\lambda}^{(k)}; \mathcal{H}) + \boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)} \tag{19b}$$

where we have used the concatenated variable as $\boldsymbol{\theta} = (\{\boldsymbol{\theta}_i\}_{i\in\mathcal{H}})$. Under Assumption 1 and assuming that $\lambda_t^{(k)} \le \overline{\lambda}$ for all k, we have

$$\begin{aligned}\|\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\| &\le (1-\alpha_1)\overline{\lambda}LT\|\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)} - \overline{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\| \\ &\quad + \frac{|\mathcal{H}| - (1-\alpha_1)N}{|\mathcal{H}|}\overline{\lambda}BT\end{aligned} \tag{20}$$

$$\|\boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}\| \le (1-\alpha_1)BT\|\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)} - \overline{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\|. \tag{21}$$

The proof can be found in Appendix C of the online version [34]. The assumption $\lambda_t^{(k)} \le \overline{\lambda}$ can be guaranteed since $\overline{g}_t((1-\alpha_1)\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)})$ is bounded, which is proven in Appendix H of the online version [34]. Furthermore, the performance analysis for the median-based estimator shows that

$$\|\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)} - \overline{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\| = \mathcal{O}(\alpha_1 R\sqrt{d}) \tag{22}$$

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 07,2021 at 18:46:03 UTC from IEEE Xplore. Restrictions apply.

when α_1 is small. Finally, based on Lemma 2, we can analyze the convergence of Algorithm 2. Let $\widehat{\boldsymbol{z}}^\star = (\widehat{\boldsymbol{\theta}}^\star, \widehat{\boldsymbol{\lambda}}^\star)$ be a saddle point of (P'_υ) and define

$$\overline{\boldsymbol{\Phi}}(\boldsymbol{z}^{(k)}) := \begin{pmatrix} \nabla_{\boldsymbol{\theta}} \overline{\mathcal{L}}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i\in\mathcal{H}}, \boldsymbol{\lambda}^{(k)}; \mathcal{H}) \\ -\nabla_{\boldsymbol{\lambda}} \overline{\mathcal{L}}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i\in\mathcal{H}}, \boldsymbol{\lambda}^{(k)}; \mathcal{H}) \end{pmatrix}. \tag{23}$$

We are ready to present our main result for static attacks.

Theorem 1: Assume the map $\overline{\boldsymbol{\Phi}}(\boldsymbol{z}^{(k)})$ is L_Φ-Lipschitz continuous. For Algorithm 2, for all $k \geq 0$ it holds

$$\|\boldsymbol{z}^{(k+1)} - \widehat{\boldsymbol{z}}^\star\|^2 \leq \left(1 - \gamma\upsilon' + 2\gamma^2 L_\Phi^2\right) \|\boldsymbol{z}^{(k)} - \widehat{\boldsymbol{z}}^\star\|^2 + \left(\frac{4\gamma}{\upsilon'} + 2\gamma^2\right) E_k \tag{24}$$

where $\upsilon' := (1-\alpha_1)\upsilon$ and $E_k := \|\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\|^2 + \|\boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}\|^2$ is the total perturbation at iteration k. Moreover, if we choose $\gamma < \upsilon'/2L_\Phi^2$ and E_k is upper bounded by $\overline{E}$ for all k, then

$$\limsup_{k\to\infty} \|\boldsymbol{z}^{(k)} - \widehat{\boldsymbol{z}}^\star\|^2 \leq \frac{\frac{4}{\upsilon'} + 2\gamma}{\upsilon' - 2\gamma L_\Phi^2} \overline{E}. \tag{25}$$

The proof can be found in Appendix D of the online version [34]. Combining with (22) shows that the resilient PD-DRA method converges to a $\mathcal{O}(\alpha_1^2 R^2 d)$ neighborhood of the saddle point of (P'_υ). Moreover, it shows that the convergence rate to the neighborhood is linear, which is similar to the classical analysis in [10].

B. Dynamic Impersonation Attack

Under this attack scenario, the set of agents communicating through compromised channels is dynamically changing with iterations. We make the following assumption on how frequently each agent's communications are compromised.

Assumption 2: Let m be a fixed window size and $\alpha_2 < 0.5$ be a known upper bound on how frequent an agent communicates through a compromised channel. Then, for all $k \geq m-1$ and for all agents $i \in [N]$, among the received parameters $\{\boldsymbol{r}_i^{(k-\ell)}\}_{\ell=0}^{m-1}$ at most $\alpha_2\, m$ are sent through compromised channels.

It is important to recall that the dynamic attack scenario does not generalize the static attack scenario and there is a significant distinction between the two. The static attack scenario assumes that a fixed set of agents' communications are *permanently* compromised. It may occur when the attacker compromises set of communication channels and those channels are assigned to the agents via a static channel allocation scheme.

On the contrary, for the dynamic attack scenario, each user's communications are vulnerable to attacks for *at most* a given α_2 fraction of iterations over a window of size m under Assumption 2, and hence each agent is able to communicate reliably with the central coordinator at some iterations. This scheme may occur when the attacker compromises a fixed set of communication channels (same as the static scenario), however, the channels are assigned to the agents via a dynamic channel allocation scheme (e.g., do a round-robin channel allocation. If there are m communication channels out of which $\alpha_2\, m$ are compromised, assigning channels dynamically in a cyclic way to the agents ensures that over a window of m, every agent has sent $\alpha_2\, m$ compromised messages). Although the attacker behaves the same way, we can simulate both scenarios by static/dynamic channel allocation. In cyber-physical systems, such dynamic allocation schemes are commonly used (e.g., dynamic IP assignment to be protected from hackers).

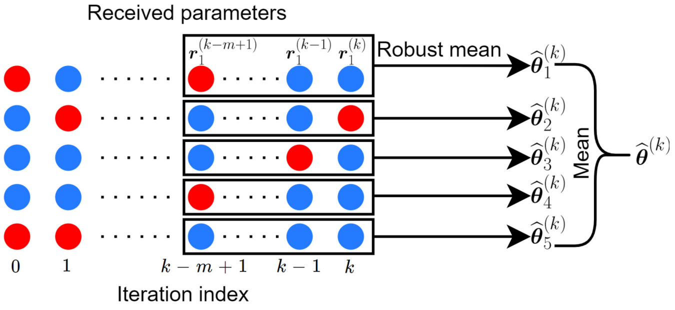

Received parameters
$\boldsymbol{r}_1^{(k-m+1)}$
$\boldsymbol{r}_1^{(k-1)}$
$\boldsymbol{r}_1^{(k)}$
Robust mean
$\widehat{\boldsymbol{\theta}}_1^{(k)}$
$\widehat{\boldsymbol{\theta}}_2^{(k)}$
$\widehat{\boldsymbol{\theta}}_3^{(k)}$
$\widehat{\boldsymbol{\theta}}_4^{(k)}$
$\widehat{\boldsymbol{\theta}}_5^{(k)}$
Mean
$\widehat{\boldsymbol{\theta}}^{(k)}$
0
1
$k-m+1$
$k-1$
k
Iteration index

Fig. 4. Robust mean estimation under dynamic impersonation attacks. Red/blue circles correspond to parameters received through compromised/trustworthy channels, respectively. In this example, there are $N=5$ agents and the set of agents communicating through compromised channels is changing at every iteration. At iteration k, the central coordinator computes the robust mean $\widehat{\boldsymbol{\theta}}_i^{(k)}$ of the received parameters $\{\boldsymbol{r}_i^{(k-\ell)}\}_{\ell=0}^{m-1}$ for all agents $i \in [N]$. Then, computes the naive average of $\{\widehat{\boldsymbol{\theta}}_i^{(k)}\}_{i=1}^N$ to get the average parameter $\widehat{\boldsymbol{\theta}}^{(k)}$.

Interestingly, it is possible to develop an algorithm that converges to the optimal solution of Problem (P_υ). The intuition behind is that the received parameters over a long period of time contain a fraction of trustworthy information that can be extracted by the algorithm to perform faithful computations.

Our algorithm is similar in nature to an *averaging gradient* scheme where the primal–dual updates utilize the averages of time-delayed gradients. Furthermore, the scheme is combined with the *robust mean* estimator developed in Section IV-A1 to approximate the averages of outdated gradients, as illustrated in Fig. 4. Specifically, the central coordinator chooses a window size of m. For any iteration $k \geq m-1$, instead of using $\boldsymbol{r}_i^{(k)}$ for computing the average parameter $\overline{\boldsymbol{\theta}}^{(k)}$ and the gradients, the central coordinator computes the robust mean $\widehat{\boldsymbol{\theta}}_i^{(k)}$ from the received parameters $\{\boldsymbol{r}_i^{(k-\ell)}\}_{\ell=0}^{m-1}$ using the median-based mean estimator (16) for all agents $i \in [N]$, applied on the sequence of historical received parameters. Note that we have replaced α_1 by α_2, N by m in this application. It then uses $\widehat{\boldsymbol{\theta}}^{(k)} := \frac{1}{N}\sum_{i=1}^N \widehat{\boldsymbol{\theta}}_i^{(k)}$ for computation of the primal–dual updates.

We summarize our robust averaging PD-DRA method in Algorithm 3. The primal–dual updates are described by

$$\theta_i^{(k+1)} = \mathcal{P}_{Cis}\left(\theta_i^{(k)} - \frac{\gamma}{N}\left(\widehat{\boldsymbol{g}}^{(k)} + \nabla_{\theta_i} f_i(\theta_i^{(k)}) + \upsilon\theta_i^{(k)}\right)\right) \tag{26a}$$

$$\lambda_t^{(k+1)} = \left[\lambda_t^{(k)} + \gamma(g_t(\widehat{\theta}^{(k)}) - \upsilon\lambda_t^{(k)})\right]_+. \tag{26b}$$

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 07,2021 at 18:46:03 UTC from IEEE Xplore. Restrictions apply.

Algorithm 3: Averaging PD-DRA Algorithm.

```
Input: Each agent has initial state $\boldsymbol{\theta}_i^{(0)}$.
for $k = 0, 1, \ldots, m-2$ do
    Apply basic PD-DRA (Run Algorithm 1).
end for
for $k = m-1, m, \ldots$ do
    (At the Central Coordinator):
    1) Receives $\{\boldsymbol{r}_i^{(k)}\}_{i=1}^N$, see (8), from agents.
    2) For all agents $i = 1, \ldots, N$, computes robust
       mean $\widehat{\boldsymbol{\theta}}_i^{(k)}$ of $\{\boldsymbol{r}_i^{(k-\ell)}\}_{\ell=0}^{m-1}$ using the estimator (16)
       with parameters $\alpha_1 \to \alpha_2$, $N \to m$.
    3) Computes $\widehat{\boldsymbol{\theta}}^{(k)} := \frac{1}{N}\sum_{i=1}^N \widehat{\boldsymbol{\theta}}_i^{(k)}$.
    4) Broadcasts the vector
       $\widehat{\boldsymbol{g}}^{(k)} := \sum_{t=1}^T \lambda_t^{(k)} \nabla_{\boldsymbol{\theta}} g_t(\widehat{\boldsymbol{\theta}}^{(k)})$ to agents.
    5) Computes the update for $\boldsymbol{\lambda}^{(k+1)}$ with (26b).
    (At each agent $i$):
    1) Agent receives $\widehat{\boldsymbol{g}}^{(k)}$.
    2) Agent computes update for $\boldsymbol{\theta}_i^{(k+1)}$ with (26a).
end for
```

Lemma 3: Algorithm 3 is a primal–dual algorithm for (P_υ) with perturbed gradients

$$\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(k)} = \nabla_{\boldsymbol{\theta}} \mathcal{L}_\upsilon(\boldsymbol{\theta}^{(k)}; \boldsymbol{\lambda}^{(k)}) + \boldsymbol{e}_{\boldsymbol{\theta}}^{(k)} \tag{27a}$$

$$\widehat{\boldsymbol{g}}_{\boldsymbol{\lambda}}^{(k)} = \nabla_{\boldsymbol{\lambda}} \mathcal{L}_\upsilon(\boldsymbol{\theta}^{(k)}; \boldsymbol{\lambda}^{(k)}) + \boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)} \tag{27b}$$

where we have used concatenated variable as $\boldsymbol{\theta} = (\{\boldsymbol{\theta}_i\}_{i\in N})$. Under Assumption 1 and assuming that $\lambda_t^{(k)} \leq \overline{\lambda}$ for all k, we have

$$\|\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\| \leq \frac{\overline{\lambda} LT}{N} \sum_{i=1}^N \|\boldsymbol{\theta}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)}\| \tag{28a}$$

$$\|\boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}\| \leq \frac{BT}{N} \sum_{i=1}^N \|\boldsymbol{\theta}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)}\|. \tag{28b}$$

The proof can be found in Appendix E of the online version [34]. The assumption $\lambda_t^{(k)} \leq \overline{\lambda}$ can be guaranteed since $g_t(\widehat{\boldsymbol{\theta}}^{(k)})$ is bounded, which is proven in Appendix H of the online version [34]. Let $\boldsymbol{z}^{(k)} := (\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N, \boldsymbol{\lambda}^{(k)})$ be the primal–dual variable at the kth iteration and define the mapping $\boldsymbol{\Phi}(\boldsymbol{z}^{(k)})$ as in (6). We observe that the algorithm's behavior is similar to the incremental aggregated gradient method in [35]–[37]. The following lemma, which is inspired by [35]–[37], upper bounds the perturbation in the gradients in (28) by the maximum optimality gap in a finite window of size $2m-1$.

Lemma 4: Assume the map $\boldsymbol{\Phi}(\boldsymbol{z}^{(k)})$ is L_Φ-Lipschitz continuous. Let $E_k := \|\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\|^2 + \|\boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}\|^2$. Then, for all $k \geq 2(m-1)$, we have

$$E_k \leq \gamma^2 \overline{C} \max_{0\leq \ell \leq 2(m-1)} \|\boldsymbol{z}^{(k-\ell)} - \boldsymbol{z}^\star\|^2 \tag{29}$$

where

$$\overline{C} = \left(\frac{T^2(\overline{\lambda}^2\, L^2 + B^2)}{N}\right) \times \left(\frac{1}{L_\Phi} + (1+\sqrt{d})\overline{\lambda} LT\right)^2 \times \left(\frac{1+C_\alpha}{1-\alpha_2} + C_\alpha\right)^2 \times (m-1)^2 \tag{30}$$

and

$$C_\alpha = \frac{2\alpha_2}{1-\alpha_2}\left(1 + \sqrt{\frac{(1-\alpha_2)^2}{1-2\alpha_2}}\right)\sqrt{d}.$$

The proof can be found in Appendix F of the online version [34]. Using on Lemmas 3 and 4, we can analyze the converge of Algorithm 3.

Theorem 2: Assume the map $\boldsymbol{\Phi}(\boldsymbol{z}^{(k)})$ is L_Φ-Lipschitz continuous. For Algorithm 3, for all $k \geq 2(m-1)$ it holds that

$$\|\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^\star\|^2 \leq (1 - \gamma\upsilon + 2\gamma^2 L_\Phi^2)\|\boldsymbol{z}^{(k)} - \boldsymbol{z}^\star\|^2 + \left(\frac{4\gamma}{\upsilon} + 2\gamma^2\right)\gamma^2 \overline{C} \max_{0\leq \ell\leq 2(m-1)} \|\boldsymbol{z}^{(k-\ell)} - \boldsymbol{z}^\star\|^2. \tag{31}$$

Moreover, if we choose γ sufficiently small such that it satisfies

$$\upsilon - 2\gamma L_\Phi^2 - \frac{4\overline{C}\gamma^2}{\upsilon} - 2\overline{C}\gamma^3 > 0$$

then

$$\|\boldsymbol{z}^{(k)} - \boldsymbol{z}^\star\|^2 \leq \rho^{k-2(m-1)} \|\boldsymbol{z}^{(2(m-1))} - \boldsymbol{z}^\star\|^2 \tag{32}$$

and

$$\lim_{k\to\infty} \|\boldsymbol{z}^{(k)} - \boldsymbol{z}^\star\|^2 = 0 \tag{33}$$

where $\rho = (1 - \gamma\upsilon + 2\gamma^2 L_\Phi^2 + \frac{4\overline{C}\gamma^3}{\upsilon} + 2\overline{C}\gamma^4)^{\frac{1}{1+2(m-1)}}$.

The proof can be found in Appendix G of the online version [34]. Theorem 2 shows that the robust averaging PD-DRA method converges *geometrically* to the optimal solution of (P_υ) under said assumptions.

C. Remarks

A few remarks highlighting design criteria to be explored in practical implementations are in the following order.

1) Theorem 1 illustrates a tradeoff in the choice of the step size γ between convergence speed and accuracy. In particular, (24) shows that the rate of convergence factor $1 - \gamma\upsilon + 2\gamma^2 L_\Phi^2$ can be minimized by setting $\gamma = \upsilon/(4L_\Phi^2)$. Meanwhile, the asymptotic upper bound in (25) is increasing with γ and it can be minimized by setting $\gamma \to 0$.
2) Theorem 2 illustrates a tradeoff between the window size m and the convergence rate. Observe that increasing the window size m decreases the rate of convergence by increasing ρ [(30) and (32)]. On the other hand, the likelihood that Assumption 2 holds true in a stochastic setting (e.g., channels being compromised with some probability) increases with a larger window size m.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 07,2021 at 18:46:03 UTC from IEEE Xplore. Restrictions apply.

3) Under the dynamic impersonation attack scenario, the choice of α_2 does not affect convergence accuracy to the saddle point of (P_v), but only changes the convergence rate. As such, choosing the largest α_2 such that $\alpha_2 m = \lfloor \frac{m-1}{2} \rfloor$ (i.e., assuming maximum possible number of iterates received through compromised channels) makes the algorithm robustly applicable to all dynamic impersonation attack scenarios regardless of the frequency of the attack.
4) In case the central coordinator cannot identify the attack scenario as static or dynamic impersonation (or the attack can be a mixture of both), a mixture of both Algorithms 2 and 3 can be applied. In particular, this can be done by adding Step 6.2 of Algorithm 3 before Step 2 of Algorithm 2, and applying the rest of Algorithm 2 as it is. The central coordinator first computes robust parameters $\widehat{\boldsymbol{\theta}}_i^{(k)}$ by computing the robust mean of $\{\boldsymbol{r}_i^{(k-\ell)}\}_{\ell=0}^{m-1}$ for all agents, and then computes the robust mean of $\{\widehat{\boldsymbol{\theta}}_i^{(k)}\}_{i=1}^{N}$. This effectively makes Algorithm 2 robust to possible dynamic impersonation attacks on uplink channels that are thought to be trustworthy for all iterations.

V. Numerical Study

In this section, we demonstrate the performance of our methods and verify our theoretical claims by applying our algorithms for the following:

1) an EV charging coordinator under static impersonation attack;
2) an EV charging coordinator under dynamic impersonation attack;
3) a power distribution network with flexible demand under dynamic impersonation attack.

The EV charging coordinator problem resembles classic network utility maximization problems such as those studied in communication networks, whereas the power distribution network problem has more nuisances that we will discuss next. To solve the convex optimization problems in order to get the optimal solutions, we used CVX, a package for specifying and solving convex programs [38].

A. EV Charging Facility

In this study, the aim is to optimize EV charging demand over time. We consider multiple EVs receiving charge under the same local feeder/transformer. Each agent (or EV owner) has a different utility of charging at different times. Hence, at a given time period, it is desired to charge those EVs who have a higher utility (or less cost) for that time period. This problem falls into the broad category of network utility maximization problems, which can be formulated as

$$\min_{\boldsymbol{\theta}_i \in \mathbb{R}_+^d \forall i} \quad f(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} f_i(\boldsymbol{\theta}_i) \tag{34a}$$

$$\text{subject to} \quad \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\theta}_i \preceq \overline{\boldsymbol{e}} \tag{34b}$$

$$\boldsymbol{\theta}_i^{\min} \preceq \boldsymbol{\theta}_i \preceq \boldsymbol{\theta}_i^{\max} \quad \forall i \tag{34c}$$

$$\Theta_i^{\min} \leq \mathbf{1}^T \boldsymbol{\theta}_i \leq \Theta_i^{\max} \quad \forall i \tag{34d}$$

where $N \times \overline{\boldsymbol{e}} \in \mathbb{R}^d$ is the vector of maximum available transformer capacity in all time periods and $\preceq$ denotes componentwise inequality between the vectors. The available capacity changes with time of day as exogenous load on the transformer varies with time as well. The elements $\{\boldsymbol{\theta}_{i,j}\}_{j=1}^{d}$ of the vector $\boldsymbol{\theta}_i$ correspond to the electricity demand of the EV i at time slots $j = 1 \ldots d$. Constraint (34c) restricts the amount an EV can charge at each time slot, whereas constraint (34d) bounds the total amount an EV can charge. For this study, we set the cost function to be

$$f_i(\boldsymbol{\theta}) = -\sum_{j=1}^{d} \beta_{i,j} \log \boldsymbol{\theta}_{i,j} \tag{35}$$

where $\beta_{i,j}$ are generated randomly from a uniform distribution in [0, 1]. We study this problem under both attack scenarios for $N = 100$ EVs.

1) Static Impersonation Attack: We simulated various static impersonation attack scenarios and ran Algorithm 2. The results are displayed in Fig. 5.

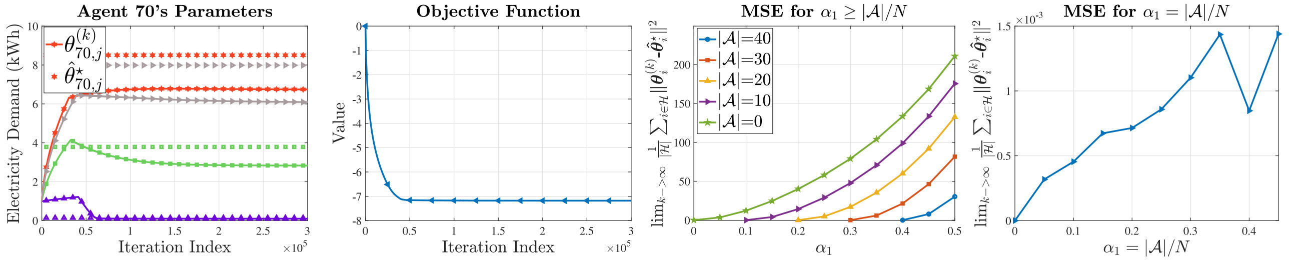
In Fig. 5(a), we plot agent 70's electricity demand for some time periods, with $|\mathcal{A}|/N = 0.2$ and $\alpha_1 = 0.3$. Each different color corresponds to a different dimension of the parameter vector (i.e., electricity demand for different time periods). A colored solid line corresponds to a dimension of the parameter vector iterates generated by the algorithm. A dashed line with the same marker and color as a solid line is the optimal value corresponding to that dimension of the parameter vector, which is the solution of the regularized robust optimization problem [formulated as (P'_v)] of (34). Observe that Algorithm 2 successfully provides convergence to a close neighborhood of the optimal solution of the regularized robust optimization problem. Furthermore, in Fig. 5(b), we show that the objective function value converges, as opposed to a nonresilient PD-DRA method that is shown to oscillate and violate the constraint in Fig. 2. Our robust optimization model on the other hand ensures there is no constraint violation.

In Fig. 5(c), we plot the mean squared error (MSE) in primal variables $\boldsymbol{\theta}_i$ for different number of compromised channels $|\mathcal{A}|$ and different choices of α_1, which is the upper bound on fraction of compromised links known by the central coordinator. The MSE is calculated by

$$\mathrm{MSE} = \lim_{k\to\infty} \frac{1}{|\mathcal{H}|} \sum_{i\in\mathcal{H}} \|\boldsymbol{\theta}_i^{(k)} - \hat{\boldsymbol{\theta}}_i^{\star}\|^2 \tag{36}$$

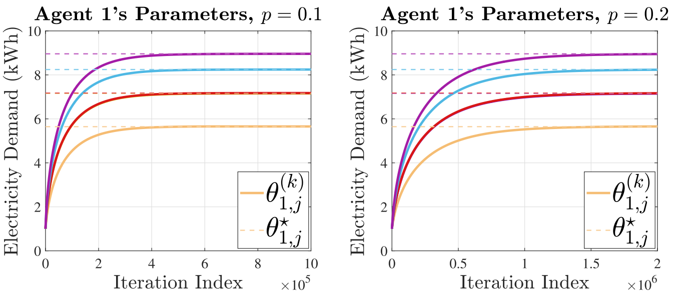
where $\hat{\boldsymbol{\theta}}_i^{\star}$ is the solution to (P'_v) with $\alpha_1 = |\mathcal{A}|/N$, i.e., the solution to the regularized and robustified problem with the knowledge of the compromised channels. Naturally, the looser the upper bound, the larger the error, since it increases the amount of conservatism. Hence, having an accurate upper bound on fraction of compromised channels significantly improves the performance.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 07,2021 at 18:46:03 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c) (d)

Fig. 5. Numerical study results for optimal EV charging under static impersonation attack. (a) Optimal parameter of agent 70 converges to a neighborhood of the optimal solution of the robust optimization problem for $|\mathcal{A}|/N = 0.2$ and $\alpha_1 = 0.3$. (b) Algorithm provides convergence of the objective function value. (c) MSE for different number of compromised channels and different choices of upper bound α_1. (d) MSE when $\alpha_1 = |\mathcal{A}|/N$.



(a) $p = 0.1$, $m = 20$, $\alpha_2 = 0.45$ (b) $p = 0.2$, $m = 100$, $\alpha_2 = 0.49$

Fig. 6. Numerical study results demonstrating convergence of Algorithm 3 for optimal EV charging under two dynamic impersonation attack scenarios: (a) $p = 0.1$, $m = 20$, $\alpha_2 = 0.45$; (b) $p = 0.2$, $m = 100$, $\alpha_2 = 0.49$. Observe that the number of iterations it takes to converge for (b) is much larger than for (a).

Finally, in Fig. 5(d), we exhibit the efficacy of our approach with median-based mean estimation. We plot the mse in primal variables, when the upper bound on α_1 is tight, i.e., $\alpha_1 = |\mathcal{A}|/N$. The error tends to increase with $|\mathcal{A}|/N$, however, considering the magnitude, the error is negligible and we can conclude that the median-based mean estimator performs well.

2) Dynamic Impersonation Attack: We simulated a dynamic impersonation attack scenario and ran Algorithm 3. To simulate a dynamic impersonation attack, we assigned a probability p for an uplink to be compromised at each iteration.[3] For $p = 0.1$, we picked a window size $m = 20$ and $\alpha_2 = 0.45$, whereas for $p = 0.2$, we picked a window size $m = 100$ and $\alpha_2 = 0.49$. The results are displayed in Fig. 6. Each different color corresponds to a different dimension of the parameter vector. A colored solid line corresponds to a dimension of the parameter vector iterates generated by the algorithm. A dashed line with the same color as a solid line is the optimal value corresponding to that dimension of the parameter vector, which is the solution of the regularized optimization problem [formulated as (P_v)] of (34).

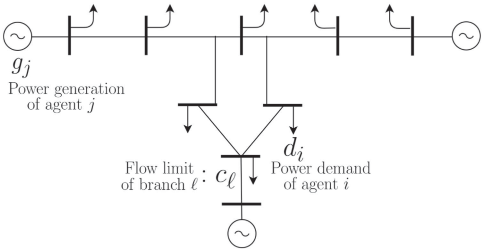


Fig. 7. IEEE 9 bus system with three generators (supplies) represented by sources and eight loads (demands) represented by arrows.

In both scenarios, Algorithm 3 successfully provides convergence to the optimal solution of the regularized problem. Observe that for $p = 0.2$, we chose a larger window size and a larger α_2 in order to meet Assumption 2. However, this restricts us to choose a smaller step size γ as dictated by Theorem 2 and in turn slower convergence. This highlights an important tradeoff between robustness and convergence rate, where a larger window size m and larger α_2 makes the algorithm more robust while decreasing the convergence rate.

B. Power Distribution Network

We consider the IEEE $N = 9$ bus system with $N_g = 3$ generators and $N_\ell = 8$ loads, as shown in Fig. 7. The power network cost minimization problem can be stated as

$$\min_{d_i, g_i \in \mathbb{R}^+} \quad f(\boldsymbol{d}, \boldsymbol{g}) = -\sum_{i=1}^{N_\ell} U_i(d_i) + \sum_{i=1}^{N_g} C_i(g_i) \tag{37a}$$

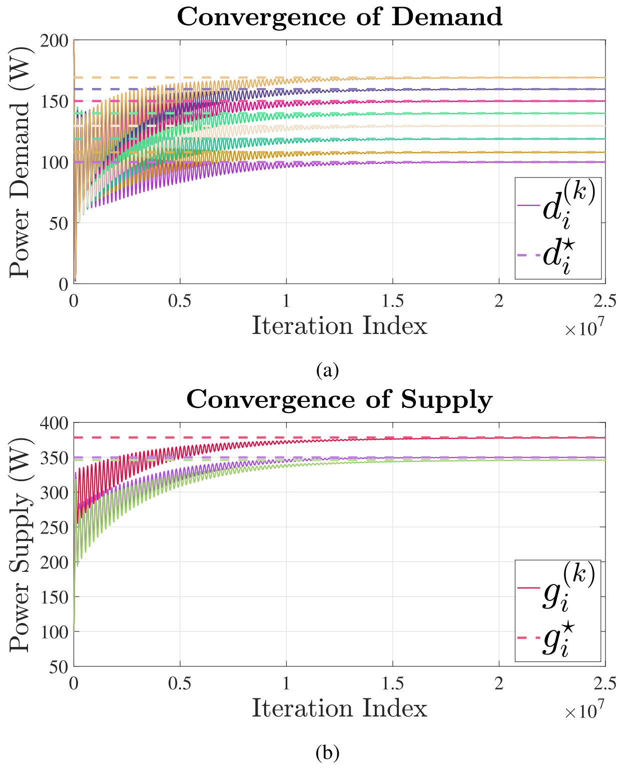
$$\text{subject to} \quad \mathbf{1}^T(\boldsymbol{d} - \boldsymbol{g}) = 0 \tag{37b}$$

$$\mathbf{H}(\boldsymbol{d} - \boldsymbol{g}) \leq \boldsymbol{c} \tag{37c}$$

where $\boldsymbol{d} = [d_1 \dots d_N]^T$ and $\boldsymbol{g} = [g_1 \dots g_N]^T$ are the vectors of load and generation at each node, respectively ($d_i = 0$ for nodes without load and $g_j = 0$ for nodes without generators). The first constraint (37b) ensures the power supply is equal to

[3] Although a probabilistic scenario does not guarantee that Assumption 2 holds, with sufficiently large window size m and α_2, it holds with high probability at each iteration. Even though we do not study this scenario theoretically, our algorithm still performs well.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 07,2021 at 18:46:03 UTC from IEEE Xplore. Restrictions apply.



(a)

(b)

Fig. 8. Numerical study results for power network under dynamic impersonation attack: (a)/(b) displaying convergence of the demand/supply, respectively.

the demand, and the second constraint (37c) is the power flow constraint limiting the power flow on each branch.

Observe that the formulation in (37) does not directly match with our general formulation in (1) mainly due to the presence of equality constraint (37b), which prevents the application of the robustified formulation in (9) and hence the robust PD-DRA algorithm for static impersonation attacks. Nevertheless, our algorithm for dynamic impersonation attacks can still be applied since it does not require any robustified constraints (which cannot be done for equality constraints).

We have chosen the utility function for load i to be $U_i(d_i) = \beta_i \log d_i$ and randomly generated β_i from a uniform distribution in [500, 1000]. For generators, we set the cost function $C_i(g_i) = e^{c_i g_i}$, where $c_1 = 0.01$, $c_2 = 0.011$, $c_3 = 0.012$. We obtained the power transfer distribution factor matrix $\mathbf{H}$ and the vector of flow limits $\boldsymbol{c}$ from MATPOWER [39]. To simulate a dynamic impersonation attack scenario, we assigned a probability p for an uplink to be compromised at each iteration. We ran Algorithm 3 for $p = 0.15$, $m = 75$, and $\alpha_2 = 0.49$.

The results are shown in Fig. 8. In Fig. 8(a) and (b), each different color corresponds to a different agent. A colored solid line corresponds to an agent's parameter iterates generated by the algorithm. A dashed line with the same color as a solid line is the optimal value of that agent's parameter, which is the solution of the regularized optimization problem [formulated as (P_v)] of (37). Our algorithm successfully generates sequences that convergence to the optimal solution of the regularized problem for both power supplying and power demanding agents.

VI. Conclusion

In this article, we studied two strategies for establishing primal–dual algorithms for resource allocation in the presence of Byzantine attackers. Specifically, we considered static and dynamic impersonation attack scenarios and proposed an attack-resilient primal–dual algorithm for each scenario based on robust mean estimation techniques. We derived bounds for the performance (in terms of distance to optimality) of the proposed algorithms and show that our algorithm for static impersonation attack converges to a neighborhood of the optimal solution of the regularized and robustified resource allocation problem, whereas our algorithm for dynamic impersonation attack converges to the optimal solution of the original regularized problem. We verify our theoretical results via computational simulations for network utility maximization problems involving optimal distributed resource allocation, such as power distribution networks.

References

[1] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, “Rate control for communication networks: Shadow prices, proportional fairness and stability,” *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, Mar. 1998.

[2] S. H. Low and D. E. Lapsley, “Optimization flow control. I. Basic algorithm and convergence,” *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 861–874, Dec. 1999.

[3] P. Samadi, A. Mohsenian-Rad, R. Schober, V. W. S. Wong, and J. Jatskevich, “Optimal real-time pricing algorithm based on utility maximization for smart grid,” in *Proc. IEEE Int. Conf. Smart Grid Commun.*, Oct. 2010, pp. 415–420.

[4] N. Li, L. Chen, and S. H. Low, “Optimal demand response based on utility maximization in power networks,” in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2011, pp. 1–8.

[5] M. Chiang and J. Bell, “Balancing supply and demand of bandwidth in wireless cellular networks: Utility maximization over powers and rates,” in *Proc. IEEE Conf. Comput. Commun.*, Mar. 2004, vol. 4, pp. 2800–2811.

[6] M. Dehghan, L. Massoulie, D. Towsley, D. Menasche, and Y. C. Tay, “A utility optimization approach to network cache design,” in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[7] M. Zhao, J. Li, and Y. Yang, “Joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks,” in *Proc. 23rd Int. Teletraffic Congr.*, 2011, pp. 238–245.

[8] R. Deng, Y. Zhang, S. He, J. Chen, and X. Shen, “Maximizing network utility of rechargeable sensor networks with spatiotemporally coupled constraints,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1307–1319, May 2016.

[9] N. Mehr, J. Lioris, R. Horowitz, and R. Pedarsani, “Joint perimeter and signal control of urban traffic via network utility maximization,” in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Oct. 2017, pp. 1–6.

[10] J. Koshal, A. Nedić, and U. V. Shanbhag, “Multiuser optimization: Distributed algorithms and error analysis,” *SIAM J. Optim.*, vol. 21, no. 3, pp. 1046–1081, 2011.

[11] D. P. Palomar and M. Chiang, “A tutorial on decomposition methods for network utility maximization,” *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[12] S. Sundaram and C. N. Hadjicostis, “Distributed function calculation via linear iterative strategies in the presence of malicious agents,” *IEEE Trans. Autom. Control*, vol. 56, no. 7, pp. 1495–1508, Jul. 2011.

[13] F. Pasqualetti, A. Bicchi, and F. Bullo, “Consensus computation in unreliable networks: A system theoretic approach,” *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 90–104, Jan. 2012.

[14] R. Gentz, S. X. Wu, H.-T. Wai, A. Scaglione, and A. Leshem, “Data injection attacks in randomized gossiping,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 4, pp. 523–538, Dec. 2016.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 07,2021 at 18:46:03 UTC from IEEE Xplore. Restrictions apply.

[15] S. Sundaram and B. Gharesifard, “Distributed optimization under adversarial nodes,” *IEEE Trans. Autom. Control*, vol. 64, no. 3, pp. 1063–1076, Mar. 2019.

[16] Y. Chen, S. Kar, and J. Moura, “Resilient distributed estimation: Sensor attacks,” *IEEE Trans. Autom. Control*, vol. 64, no. 9, pp. 3772–3779, Sep. 2019.

[17] W. Ben-Ameur, P. Bianchi, and J. Jakubowicz, “Robust distributed consensus using total variation,” *IEEE Trans. Autom. Control*, vol. 61, no. 6, pp. 1550–1564, Jun. 2016.

[18] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, “Resilient asymptotic consensus in robust networks,” *IEEE J. Sel. Areas Commun.*, vol. 31, no. 4, pp. 766–781, Apr. 2013.

[19] J. S. Baras and X. Liu, “Trust is the cure to distributed consensus with adversaries,” in *Proc. 27th Mediterranean Conf. Control Autom.*, Jul. 2019, pp. 195–202.

[20] N. Ravi, A. Scaglione, and A. Nedić, “A case of distributed optimization in adversarial environment,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2019, pp. 5252–5256.

[21] J. Feng, H. Xu, and S. Mannor, “Distributed robust learning,” 2014, *arXiv:1409.5937*.

[22] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, “Byzantine-robust distributed learning: Towards optimal statistical rates,” in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 5650–5659.

[23] D. Alistarh, Z. Allen-Zhu, and J. Li, “Byzantine stochastic gradient descent,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4618–4628.

[24] Y. Chen, L. Su, and J. Xu, “Distributed statistical machine learning in adversarial settings: Byzantine gradient descent,” *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 2, pp. 44:1–44:25, 2017.

[25] D. Data, L. Song, and S. Diggavi, “Data encoding methods for Byzantine-resilient distributed optimization,” in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2019, pp. 2719–2723.

[26] D. L. Donoho and P. J. Huber, “The notion of breakdown point,” in *A Festschrift for Erich L. Lehmann*. Belmont, CA, USA: Wadsworth, 1983, pp. 157–184.

[27] P. J. Huber, *Robust Statistics*. New York, NY, USA: Springer, 2011.

[28] S. Minsker *et al.*, “Geometric median and robust estimation in Banach spaces,” *Bernoulli*, vol. 21, no. 4, pp. 2308–2335, 2015.

[29] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart, “Robust estimators in high dimensions without the computational intractability,” in *Proc. IEEE Annu. Symp. Found. Comput. Sci.*, 2016, pp. 655–664.

[30] J. Steinhardt, M. Charikar, and G. Valiant, “Resilience: A criterion for learning in the presence of arbitrary outliers,” 2017, *arXiv:1703.04940*.

[31] C. A. Uribe, H. Wai, and M. Alizadeh, “Resilient distributed optimization algorithms for resource allocation,” in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 8341–8346.

[32] Y. Nesterov, “Smooth minimization of non-smooth functions,” *Math. Program.*, vol. 103, no. 1, pp. 127–152, 2005.

[33] C. A. Uribe, S. Lee, A. Gasnikov, and A. Nedić, “A dual approach for optimal algorithms in distributed optimization over networks,” *Optim. Methods Softw.*, pp. 1–40, 2020.

[34] B. Turan, C. A. Uribe, H.-T. Wai, and M. Alizadeh, “Resilient primal-dual optimization algorithms for distributed resource allocation,” 2020, *arXiv:2001.00612*.

[35] D. Blatt, A. O. Hero, and H. Gauchman, “A convergent incremental gradient method with a constant step size,” *SIAM J. Optim.*, vol. 18, no. 1, pp. 29–51, 2007.

[36] M. Gürbüzbalaban, A. Ozdaglar, and P. A. Parrilo, “On the convergence rate of incremental aggregated gradient algorithms,” *SIAM J. Optim.*, vol. 27, no. 2, pp. 1035–1048, 2017.

[37] P. Tseng and S. Yun, “Incrementally updated gradient methods for constrained and regularized optimization,” *J. Optim. Theory Appl.*, vol. 160, no. 3, pp. 832–853, Mar. 2014.

[38] M. Grant and S. Boyd, “CVX: MATLAB software for disciplined convex programming, version 2.1,” Mar. 2014. [Online]. Available: http://cvxr.com/cvx

[39] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, “MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education,” *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

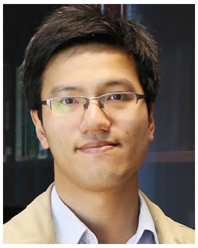
[40] H. R. Feyzmahdavian, A. Aytekin, and M. Johansson, “A delayed proximal gradient method with linear convergence rate,” in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, Sep. 2014, pp. 1–6.

Berkay Turan received the B.Sc. degree in electrical and electronics engineering as well as in physics from Boğaziçi University, Istanbul, Turkey, in 2018. He is currently working toward the Ph.D. degree in electrical and computer engineering with the University of California–Santa Barbara, Santa Barbara, CA, USA.

His research interests include optimization and reinforcement learning for the design, control, and analysis of smart infrastructure systems such as the power grid and transportation systems.

César A. Uribe received the M.Sc. degrees in systems and control from the Delft University of Technology, Delft, The Netherlands, and in applied mathematics from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2013 and 2016, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, in 2018.

He is currently a Postdoctoral Associate with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA. His research interests include distributed learning and optimization, decentralized control, algorithm analysis, and computational optimal transport.



Hoi-To Wai (Member, IEEE) received the B.Eng. (with First-Class Honors) and the M.Phil. degrees in electronic engineering from The Chinese University of Hong Kong (CUHK), Hong Kong, in 2010 and 2012, respectively, and the Ph.D. degree in electrical engineering from Arizona State University (ASU), Tempe, AZ, USA, in 2017.

He is currently an Assistant Professor with the Department of Systems Engineering and Engineering Management, CUHK. He has held research positions with ASU, UC Davis, Telecom ParisTech, Ecole Polytechnique, and LIDS, MIT. His research interests include signal processing, machine learning, and distributed optimization, with a focus on their applications to network science.

Dr. Wai’s dissertation has received the 2017’s Dean’s Dissertation Award from the Ira A. Fulton Schools of Engineering of ASU. He was the recipient of the Best Student Paper Award at the IEEE International Conference on Acoustics, Speech, and Signal Processing 2018.

Mahnoosh Alizadeh received the B.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2009, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of California–Davis, Davis, CA, USA, in 2013 and 2014, respectively.

She is currently an Assistant Professor of Electrical and Computer Engineering with the University of California–Santa Barbara, Santa Barbara, CA, USA. From 2014 to 2016, she was a Postdoctoral Scholar with Stanford University. Her research interests include designing network control, optimization, and learning frameworks to promote efficiency and resiliency in societal-scale cyber-physical systems.

Dr. Alizadeh was the recipient of the NSF CAREER Award and the Best Paper Award from HICSS-53 power systems track. She was also the recipient of the Richard C. Dorf Award for outstanding research accomplishment at the University of California–Davis.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 07,2021 at 18:46:03 UTC from IEEE Xplore. Restrictions apply.