2022 American Control Conference (ACC)
Atlanta, USA, June 8-10, 2022

PrivOpt: an intrinsically private distributed optimization algorithm

Amir-Salar Esteki and Solmaz S. Kia, *Senior Member, IEEE*

***Abstract*— A critical factor for expanding the adoption of networked solutions is ensuring local data privacy of in-network agents implementing a distributed algorithm. In this paper, we consider privacy preservation in the distributed optimization problem in the sense that local cost parameters should not be revealed. Current approaches to privacy preservation normally propose methods that sacrifice exact convergence or increase communication overhead. We propose PrivOpt, an intrinsically private distributed optimization algorithm that converges exponentially fast without any convergence error or using extra communication channels. We show that when the number of the parameters of the local cost is greater than the dimension of the decision variable of the problem, no malicious agent, even if it has access to all transmitted-in and -out messages in the network, can obtain local cost parameters of other agents. As an application study, we show how our proposed PrivOpt algorithm can be used to solve an optimal resource allocation problem with the guarantees that the local cost parameters of all the agents stay private.**

I. Introduction

Distributed optimization algorithms are enabling means for optimal decision-making, distributed learning, and maximum likelihood estimation for distributed data in networked systems. In many in-network operations, it is highly desired that the parameters of the cost function of the agents stay private because these parameters are often a reflection of local privacy-sensitive data of the agents. For example, in the distributed economic dispatch problem in a smart grid, the cost function parameters of each individual generator are privacy-sensitive information that allows others to know the cost function of an agent [1]. The cost function is a critical business information that if revealed can promote other competitor to change their operational cost, e.g., to establish themselves as the least-cost utility provider in the market. Or, take the widely used machine learning problem of distributed linear regression, where each agent i has access to a part of the data (the data set and target points denoted by $\mathbf{X}^i \in \mathbb{R}^{n\times m}$ and $\mathbf{Y}^i \in \mathbb{R}^m$, respectively) and the goal is to minimize the total empirical risk of fitting a model to this data. In the distributed learning cost function formulation, the parameters of the local cost of each agent i includes $(\mathbf{X}^{i^\top}\mathbf{Y}^i, \mathbf{X}^{i^\top}\mathbf{Y}^i)$. If these parameters are revealed, in some scenarios where m is larger than n, i.e., dimension of the weight vector is larger than the number of data points, there is a concern regarding the privacy of local data $(\mathbf{X}^i, \mathbf{Y}^i)$ [2].

This paper considers the problem of private distributed unconstrained convex optimization problem in the sense that the parameters of the local cost, denoted by $\mathcal{C}^i$ should stay private for each agent i with respect to a malicious agent, which is defined in Definition 2. In an unconstrained convex optimization, a group of networked agents, each endowed with a local cost $f^i(\mathbf{x}, \mathcal{C}^i)$, use local interactions to obtain the minimizer of the global optimization problem $\sum_{i=1}^N f^i(\mathbf{x}, \mathcal{C}^i)$. This problem is one of the most studied ones in distributed optimization literature, for which there are proposed algorithms as a solution. The main solutions include EXTRA [3] and PI [4] algorithms; see [5] for a more extensive literature review of the topic. Even though these algorithms do not require the agents to communicate their local costs or their local gradients, since they communicate the decision variable that converges to the global minimizer, there is a breach of privacy when all the incoming and outgoing communication signals of an agent are known the malicious agent. In Appendix A, we show how a malicious agent can use a nonlinear least-squares observer [6] for obtaining cost parameters of the agents, when implementing the EXTRA algorithm.

Differential privacy and encryption are two popular methods to induce privacy in network operations. These methods, respectively, are used in [7] and [8] for the unconstrained distributed optimization problem. Differential privacy and perturbation methods [9] conceal the exact local parameters, however, in these stochastic privacy preservation mechanisms, the malicious agent can obtain an estimate on the true values with known quantifiable error covariance. In addition, in differential privacy, there is a trade-off between exact convergence and the level of privacy the method provides. On the other hand, encryption demands overhead communication and also a trusted third party to generate the public key. In the class of papers using obfuscation noise to disguise sensitive local data, [7] perturbs messages in order to preserve privacy and, therefore, results in having a steady-state error even when agents broadcast noiseless messages. To resolve this issue, [10] suggests perturbing local costs instead of messages, though exact convergence is still sacrificed for the sake of privacy. [11] introduces an algorithm that employs asynchronous updates and heterogeneous stepsizes among agents where the malicious agent cannot obtain the local cost. This method affects the rate of convergence since agents apply updates only at specific iterations. There are also other works where differential privacy is used to preserve privacy for constrained in-network optimization problems [12]–[14].

This paper proposes a novel algorithm, called PrivOpt, to solve the distributed optimization problem while preserving privacy of in-network agents against a malicious agent in the network which without perturbing/interrupting the execution of the algorithm, wants to obtain the private cost parameters of other agents. Unlike encryption and noise obfuscation methods like differential privacy, PrivOpt does not require any overhead communication channel, tagged

The authors are with the Department of Mechanical and Aerospace Engineering, University of California Irvine, Irvine, CA 92697, {aesteki,solmaz}@uci.edu. This work was supported by NSF award ECCS-1653838.

978-1-6654-5196-3/$31.00 ©2022 AACC

messages which are not practical in broadcast applications, or noisy messages which may sacrifice exact convergence for privacy. We also show that state-of-the-art algorithms such as EXTRA and PI cannot guarantee privacy preservation. Contrarily, PrivOpt can conceal the local cost parameters. To demonstrate our results, we show how PrivOpt can be used to solve an in-network economic dispatch problem (EDP) in a way that the parameters of local quadratic costs of all the agents is preserved, even if the malicious agent has access to all the transmitted in and out signals of the agent. We compare our solution to the distributed solver proposed in [15] and show not only PrivOpt preserves privacy of the agents but also it converges faster.

Notations: We follow [16] for graph theoretic terminologies. The interaction topology of N in-network agents is modeled by the undirected connected graph $\mathcal{G}(\mathcal{V},\mathcal{E},\mathbf{A})$ where $\mathcal{V}$ is the node set, $\mathcal{E}\subset\mathcal{V}\times\mathcal{V}$ is the edge set and $\mathbf{A}=[\mathsf{a}_{ij}]$ is the adjacency matrix defined such that $\mathsf{a}_{ij}>0$, if $(i,j)\in\mathcal{E}$, otherwise $\mathsf{a}_{ij}=0$. A graph is undirected if $\mathsf{a}_{ij}=\mathsf{a}_{ji}$ for all $i,j\in\mathcal{V}$. Moreover, a graph is connected if there is a directed path from every node to every other node. The degree of each node $i\in\mathcal{V}$ is $\mathsf{d}^i=\sum_{j=1}^{N}\mathsf{a}_{ij}$ and the Laplacian matrix of a graph $\mathcal{G}$ is $\mathbf{L}=\mathrm{Diag}(\mathsf{d}^1,\cdots,\mathsf{d}^N)-\mathbf{A}$. Furthermore, For a connected graph, we denote the eigenvalues of $\mathbf{L}$ by $\lambda_1,\cdots,\lambda_N$, where $\lambda_1=0$ and $\lambda_i\leq\lambda_j$, for $i<j$ and λ_2 and λ_N are, respectively, the smallest nonzero eigenvalue and maximum eigenvalue of $\mathbf{L}$. Finally, given an edge (i,j), i is called a neighbor of j, and vice versa. We let $\mathbf{1}_N$ denote the vector of N ones, and denote by $\mathbf{I}_N$ the $N\times N$ identity matrix. We also define $\mathfrak{r}=\frac{1}{\sqrt{N}}\mathbf{1}_N$ and $\mathfrak{R}\in\mathbb{R}^{N\times(N-1)}$, such that $\begin{bmatrix}\mathfrak{r} & \mathfrak{R}\end{bmatrix}\begin{bmatrix}\mathfrak{r}^\top\\ \mathfrak{R}^\top\end{bmatrix}=\begin{bmatrix}\mathfrak{r}^\top\\ \mathfrak{R}^\top\end{bmatrix}\begin{bmatrix}\mathfrak{r} & \mathfrak{R}\end{bmatrix}=\mathbf{I}_N$.

II. Problem setting

Let the interaction topology of the agents be an undirected and connected graph $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{A})$. Consider the distributed optimization problem where the set of N agents communicate over the graph $\mathcal{G}$ to obtain the global minimizer of the total cost $\sum_{i=1}^{N}f^i(\mathbf{x},\mathcal{C}^i)$, where $\mathbf{x}\in\mathbb{R}^n$ is the decision variable and $\mathcal{C}^i=\{q_1^i,q_2^i,\cdots,q_{n^i}^i\}$ denotes the set of local cost parameters of agent i with n^i representing the number of elements in $\mathcal{C}^i$. We assume that the local cost function of the agents is separable, i.e., $f^i(\mathbf{x},\mathcal{C}^i)=\sum_{l}^{n}f_l^i(\mathsf{x}_l,\mathcal{C}^i)$. In short, the collective goal is for each agent $i\in\mathcal{V}$ to obtain

$$\mathbf{x}^\star=\mathrm{argmin}_{\mathbf{x}\in\mathbb{R}^n}\sum\nolimits_{i=1}^{N}f^i(\mathbf{x},\mathcal{C}^i)$$

using local interactions by its neighbors, while having the guarantees that its local cost parameters $\mathcal{C}^i$ stay private. Note that each local cost $f^i(\mathbf{x},\mathcal{C}^i)$ is convex. To study privacy-preservation properties of an algorithm against a malicious agent let us present the following two definitions.

Definition 1 (Privacy Preservation)**.** Privacy of an agent i is preserved from a malicious agent if the malicious agent cannot obtain the exact value of the local cost parameters of agent i, i.e., $\mathcal{C}^i=\{q_1^i,q_2^i,\cdots,q_{n^i}^i\}$. □

Throughout this paper, we assume that $\mathcal{C}^i$ excludes the local cost's possible additive constant scalar term. Such an additive term never appears in gradient-based algorithms. Thus, there is no way for a malicious agent to obtain any information about it.

Definition 2 (Malicious Agent)**.** A malicious agent j is an agent in the network who wants to obtain the local parameters $\mathcal{C}^i=\{q_1^i,q_2^i,\cdots,q_{n^i}^i\}$ of another agent $i\in\mathcal{V}\backslash\{j\}$ without perturbing/interrupting the execution of the algorithm. The knowledge set of this malicious agent consists of (a) the network topology $\mathcal{G}(\mathcal{V},\mathcal{E},\mathbf{A})$, (b) its own local states and parameters $\mathcal{C}^j$, (c) all transmitted signals to and from an agent i, (d) Agent j also knows that the minimizer state $\mathbf{x}^i(k)$ of each agent $i\in\mathcal{V}$ converges asymptotically to $\mathbf{x}^\star$, (e) the malicious agent knows any special initialization condition of the algorithm if one exists. □

In most algorithms in the literature, e.g., the EXTRA [3]

$$\begin{aligned}\mathbf{x}^i(1)&=\sum\nolimits_{j=1}^{N}w_{ij}\mathbf{x}^j(0)-\gamma\nabla f^i(\mathbf{x}^i(0),\mathcal{C}^i),\\ \mathbf{x}^i(k+2)&=\mathbf{x}^i(k{+}1)+\sum_{j=1}^{N}w_{ij}\mathbf{x}^j(k{+}1)-\sum_{j=1}^{N}\tilde{w}_{ij}\mathbf{x}^j(k{+}1)\\ &\quad-\alpha(\nabla f^i(\mathbf{x}^i(k{+}1),\mathcal{C}^i)-\nabla f^i(\mathbf{x}^i(k),\mathcal{C}^i)),\\ &\quad i\in\mathcal{V},\quad k\in\mathbb{Z}_{>0},\end{aligned}$$

and PI [4] algorithms, agents communicate the local minimizer decision variable $\mathbf{x}^i$. The reader can easily confirm that in both EXTRA and PI algorithms, the value of $\nabla f^i(\mathbf{x}^i(k),\mathcal{C}^i)$ at each step $k\in\mathbb{Z}_{>0}$ can be trivially obtained by a malicious agent whose knowledge set is as in Definition 2 and trivially knows $\mathbf{x}^i(k)$. After a sufficient number of steps, the malicious agent can construct a system of consistent nonlinear equations where the number of equations (values of $\nabla f^i(\mathbf{x}^i(k),\mathcal{C}^i)$ at each step k) are equal to or greater than the number of variables ($\mathcal{C}^i$), i.e., there exists a finite $l\in\mathbb{Z}_{>0}$, such that $n\times k\geq n^i$. Obviously, l is the smallest positive integer that satisfies $l\geq\max\{1,n^i/n\}$. When the number of equations are equal to or greater than the number of unknowns, there are multiple methods to solve the nonlinear system of equations [6], [17], [18]. In the special case of quadratic cost functions, cost coefficients can be derived by the malicious agent by constructing a linear system of equations. An example observer that the malicious agent can use is the nonlinear least squares observer [6] which is presented in Algorithm 1 in Appendix A. A numerical example in which a malicious agent uses this least square observer to obtain the parameters of one of its neighbors is also given in Appendix A. The breach of privacy in the EXTRA and similarly in the PI algorithm then can be traced back to trivial availability of the agents' decision variable $\mathbf{x}^i(k)$ through the communication messages. In light of this observation, in what follows, we propose an alternative distributed optimization algorithm that employs a communication message that does not trivially reveal $\mathbf{x}^i(k)$.

III. PrivOpt algorithm

We propose the algorithm below, referred to as PrivOpt, as a solution to the private distributed unconstrained optimization problem,

$$\mathbf{v}^i(k+1)=\mathbf{v}^i(k)+\delta\sum\nolimits_{j=1}^{N}\mathbf{a}_{ij}(\mathbf{z}^i(k)-\mathbf{z}^j(k)), \tag{1a}$$

$$\begin{aligned}\mathbf{p}^i(k+1)=\mathbf{p}^i(k)+\delta\Big(&-\mathbf{p}^i(k)+\nabla f^i(\mathbf{x}^i(k),\mathcal{C}^i)\\ &-\sum\nolimits_{j=1}^{N}\mathbf{a}_{ij}(\mathbf{z}^i(k)-\mathbf{z}^j(k))-\mathbf{v}^i(k)\Big),\end{aligned} \tag{1b}$$

$$\mathbf{x}^i(k+1)=\mathbf{x}^i(k)-\delta\boldsymbol{\beta}^i(k)\mathbf{p}^i(k), \tag{1c}$$

where $\mathbf{z}^i=\mathbf{p}^i-\mathbf{x}^i$ and $\mathbf{v}^i(0)=\mathbf{0}$ and $\mathbf{x}^i(0),\mathbf{p}^i(0)\in\mathbb{R}^n$, for $i\in\mathcal{V}$. Moreover, $\delta>0$ and $\mathbf{0}<\boldsymbol{\beta}^i(k)\leq\mathbf{I}$ are the stepsize and a diagonal local weight matrix, respectively. In PrivOpt, instead of $\mathbf{x}^i$, agents communicate $\mathbf{z}^i$. This algorithm is inspired by consensus-based distributed optimal resource allocation algorithms [19], [20] where a dynamic average consensus, here represented by (1a) and (1b), enables the local state $\mathbf{p}^i$ of every agent $i\in\mathcal{V}$ to track $\sum_{i=1}^{N}\nabla f^i(\mathbf{x}^i,\mathcal{C}^i)$ scaled by $1/N$, while in (1c), $\mathbf{p}^i$ is used as descend direction to drive the local state $\mathbf{x}^i$ to the optimal solution $\mathbf{x}^\star$. $\mathbf{v}^i(k)$ is an integrator type feedback to correct the tracking error of $\mathbf{p}^i$. In PrivOpt, the local diagonal weight matrix $\boldsymbol{\beta}^i$ changes the local step size in (5b) and consequently, and, as will be shown later, helps disguise the exact value of the local states. This is because $\boldsymbol{\beta}^i$ is chosen locally by each agent and is not shared with other agents. The reader should note that use of a local step size in EXTRA and PI algorithm will change the final convergence point of the algorithm. Intuitively, one can see that in PrivOpt algorithm since (1a) and (1b) generate the descent direction, each agent is able to choose a local weighted step size. The next result establishes the exponential convergence of PrivOpt to $\mathbf{x}^\star$.

Proposition III.1. *Initialized at $\mathbf{x}^i(0),\mathbf{p}^i(0)\in\mathbb{R}^n$ and $\mathbf{v}^i(0)=\mathbf{0}_n$, $i\in\mathcal{V}$, let the agents implement the* PrivOpt *algorithm* (1) *over an undirected connected graph $\mathcal{G}$ using diagonal local weights $\mathbf{0}<\boldsymbol{\beta}^i(k)\leq\mathbf{I}$. Suppose that $f^i(\mathbf{x}^i,\mathcal{C}^i)$ is m^i-strongly convex and l^i-lipschitz. Then, the trajectories of* (1) *converge exponentially fast to*

$$\bar{\mathbf{v}}^i=\nabla f^i(\mathbf{x}^\star,\mathcal{C}^i),\quad \bar{\mathbf{p}}^i=\mathbf{0},\quad \bar{\mathbf{x}}^i=\mathbf{x}^\star,\quad i\in\mathcal{V}, \tag{2}$$

provided $0<\delta<\bar{\delta}<1$, and

$$\begin{aligned}\bar{\delta}=\max_{\phi_1,\phi_2}\min\Big\{&\frac{1}{4(1+4l+(1+l)\phi_2)},\frac{m\phi_2-\phi_1 l^2-\frac{1}{4}-3l^2}{4+\phi_1+2\phi_2},\\ &\frac{2+\frac{3}{4}\phi_1}{2+2(5+\phi_2)l+\frac{2+l}{2}\phi_1},\frac{\lambda_2\phi_2-1}{\lambda_N\phi_2(\lambda_N+1)}\Big\},\end{aligned} \tag{3}$$

where $0<\phi_1<\frac{m\phi_2-3l^2-1}{4l^2}$, $\phi_2>\frac{1}{\lambda_2}$, $m=\min\{m^i\}_{i\in\mathcal{V}}$ and $l=\max\{l^i\}_{i\in\mathcal{V}}$.

Due to limited space the proof of Proposition III.1 will appear elsewhere. In comparison to algorithms like EXTRA [3] and PI [4], in PrivOpt, $\mathbf{z}^i=\mathbf{p}^i-\mathbf{x}^i$ is communicated with neighbors instead of the local minimizer state $\mathbf{x}^i$. As we discussed in Section II knowing $\mathbf{x}^i$ by the malicious agent leads to breach of privacy in the EXTRA and PI algorithms. As we show below, because of this difference in the communication message, PrivOpt can guarantee privacy preservation for all agents in the network in the sense of Definition 1. In what follows without loss of generality, we assume that agent 1 is the malicious agent. In our analysis, according to Definition 2, the malicious agent knows the special initialization of the algorithm, i.e., $\mathbf{v}^i(0)=\mathbf{0}$.

Let

$$\mathbf{F}^i(k)=\sum\nolimits_{j=1}^{N}\mathbf{a}_{ij}(\mathbf{z}^i(k)-\mathbf{z}^j(k)). \tag{4}$$

Then, PrivOpt algorithm equivalently also reads as

$$\mathbf{z}^i(k+1)=\mathbf{z}^i(k)+\delta\Big(\nabla f^i(\mathbf{x}^i(k),\mathcal{C}^i)-\mathbf{F}^i(k)-\delta\sum_{l=0}^{k}\mathbf{F}^i(l)\Big), \tag{5a}$$

$$\begin{aligned}\mathbf{x}^i(k)=\ &\Pi_{l=0}^{k-1}\big(\mathbf{I}-\delta\boldsymbol{\beta}^i(l)\big)\,\mathbf{x}^i(0)\\ &-\delta\sum_{l=0}^{k-1}\Big(\Pi_{l'=l+1}^{k-1}\big(\mathbf{I}-\delta\boldsymbol{\beta}^i(l')\big)\boldsymbol{\beta}^i(l)\mathbf{z}^i(l)\Big),\end{aligned} \tag{5b}$$

for $i\in\mathcal{V}$. We show our privacy preservation by analyzing the extreme case that agent $i\in\mathcal{V}\backslash\{1\}$ and all the neighbors of agent i are neighbors of agent 1, i.e., agent 1 has access to all the messages transmitted in or out of agent i. At each step k the information available for agent 1 are δ and $\{\mathbf{z}^j(l)\}_{l=1}^{k}$ for all $j\in\mathcal{N}^i\cup\{i\}$. At each time step k, because of (5b), to know $\mathbf{x}^i(k)$ agent 1 needs $\mathbf{x}^i(0)$ and $\{\boldsymbol{\beta}^i(l)\}_{l=1}^{k}$. At each time step $k+1$, agent 1 can obtain the value of $\nabla f^i(\mathbf{x}^i(k),\mathcal{C}^i)$ by rearranging (5a) as

$$\nabla f^i(\mathbf{x}^i(k),\mathcal{C}^i)=\frac{1}{\delta}(\mathbf{z}^i(k+1)-\mathbf{z}^i(k))+\mathbf{F}^i(k)+\delta\sum_{l=1}^{k}\mathbf{F}^i(l). \tag{6}$$

Proposition III.2. *Let the agents of an undirected connected graph implement PrivOpt. Using the knowledge set in Definition 2, agent 1 almost always cannot obtain the exact value of $\mathbf{x}^i(k)$ and $\mathcal{C}^i$ of any agent in finite time.*

Proof: Given (6), at each finite time $k+1$ agent 1 knows the value of $\nabla f^i(\mathbf{x}^i(l),\mathcal{C}^i)$ for all $l=\{0,\cdots,k\}$. Then, at each finite time $k+1$, by substituting for $\mathbf{x}^i(k)$ from (5b) in (6), agent 1 can form $k+1$ set of n equations of the form

$$\mathbf{F}^i(k)=\mathbf{F}^i(\mathbf{x}^i(0),\{\boldsymbol{\beta}^i(l)\}_{l=1}^{k},\mathcal{C}^i)=\mathbf{0}. \tag{7}$$

Therefore, agent 1 has $(k+1)\times n$ equations but $n^i+n+(k+1)\times n$ unknowns, which are $\mathcal{C}^i$, $\mathbf{x}^i(0)$, and $\{\boldsymbol{\beta}^i(l)\}_{l=0}^{k}$. Therefore, the system of equations agent 1 has is underdetermined. As a result, almost always agent 1 cannot find a unique solution for its system of equations, unless the set of solutions of this system of nonlinear equations is a singleton. □

The privacy preservation study in infinite time should take into account that $\mathbf{x}^i(k)$ converges to $\mathbf{x}^\star$ as $k\to\infty$, and thus the only unknown in (6) is $\mathcal{C}^i$ after convergence.

Theorem III.3. *Let the agents of an undirected connected graph implement PrivOpt. Using the knowledge set in Definition 2, agent 1 almost always cannot obtain the exact value* *of $\mathcal{C}^i$ of any agent $i \in \mathcal{V}\backslash\{1\}$ provided $n^i > n$.*

Proof: Following through the proof of Proposition III.2, notice that at each time step $k+1$ agent 1 adds n equations to its $k \times n$ equations with $n^i + n + k \times n$ unknowns to make its equations $(k+1) \times n$. But it also adds n unknowns, which make its total unknowns $n^i + n + (k+1) \times n$. Only at infinite time when $\mathbf{x}^i$ converges to $\mathbf{x}^\star$ that agent 1 can add n equations without adding a new set of unknowns. But still the set of equations it has to solve to obtain all its unknowns is under-determined. Also if agent 1 only considers

$$\mathbf{F}^i(\infty) = \mathbf{F}^i(\mathbf{x}^\star, \mathcal{C}^i) = \mathbf{0},$$

if $n^i > n$, then the set of unknowns $\mathcal{C}^i$ are larger than the number of equations. As a result, almost always agent 1 cannot find a unique solution for its system of equations, unless the set of solutions of this system of equations is a singleton. □

In the EXTRA and PI algorithms, the malicious agent knows $\nabla f^i(\mathbf{x}^i(k), \mathcal{C}^i)$ and $\mathbf{x}^i(k)$ at any $k \in \mathbb{Z}_{\geq 0}$. Therefore, beyond the first step of the algorithm, at each iteration any equation the malicious agent adds to its set of equations, similar to (7), introduces no new unknowns. Thus, as mentioned earlier, after $l \geq \max\{1, n^i/n\}$ finite steps, the malicious agent can form an over-determined set of algebraic equations, and can use e.g., a least-square observer [6], [17], [18] to obtain local cost parameters in finite time. For example, in the problem of distributed linear regression, local data points may be revealed to the malicious agent in a finite number of steps [2]. This is in contrast to PrivOpt where as shown in Theorem III.3, there is always more unknowns than equations due to the use of time-varying local diagonal weight matrix $\boldsymbol{\beta}^i(k)$ at every $k \in \mathbb{Z}_{\geq 0}$. When implementing PrivOpt, the malicious agent needs to solve for $n^i + (k+2)n$ unknowns with $(k+2)n$ equations in each finite time $k+1$. Solving under-determined set of equations is a complicated problem in which even if the set of equations formed by knowing the value of $\nabla f^i(\mathbf{x}^i(k), \mathcal{C}^i)$ has a single real solution $\{q_1^i, q_2^i, \cdots, q_{n^i}^i\}$, the numerical solvers often have a difficult time in solving the problem and may require additional conditions on the under-determined set of nonlinear equations [21], [22].

IV. Privacy Preservation in Distributed Optimal Economic Dispatch via PrivOpt

Consider the quadratic optimal resource allocation problem

$$\boldsymbol{\mathfrak{p}}^\star = \text{argmin}_{\{\mathfrak{p}^i\}_{i=1}^N \subset \mathbb{R}} \sum\nolimits_{i=1}^N \frac{1}{2\mathfrak{b}^i}(\mathfrak{p}^i + \mathfrak{a}^i)^2, \quad \text{s.t.} \tag{8a}$$

$$\mathfrak{p}^1 + \mathfrak{p}^2 + \cdots + \mathfrak{p}^N = P_D, \quad i \in \mathcal{V}, \tag{8b}$$

where agents $\{1, \cdots, N\}$ interacting over a connected graph $\mathcal{G}$ want to find their corresponding component of $\boldsymbol{\mathfrak{p}}^\star = (\mathfrak{p}^{1\star}, \mathfrak{p}^{2\star}, \cdots, \mathfrak{p}^{N\star})$. We seek a distributed solution that ensures every agent $i \in \mathcal{V}$ can keep its corresponding local cost parameters $\mathfrak{a}^i$ and $\mathfrak{b}^i$ private. A practical example case is the distributed optimal economic dispatch problem where a group of N generator stations with quadratic costs, interacting over a connected graph, as for example shown in Fig. 1, seek to find their optimal dispatch power that collectively meets the demand P_D for all $i \in \mathcal{V}$ while resulting in minimum cost for the group [15]. As discussed in the opening paragraph of the introduction, in the economic dispatch problem, agents may want to keep the parameters of their local cost private.

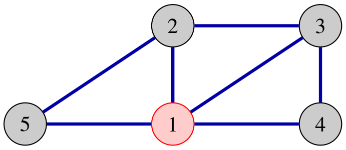

Fig. 1: An undirected connected graph with adjacency weights of $\mathbf{a}_{ij} = 1$, if $(i,j) \in \mathcal{E}$, otherwise $\mathbf{a}_{ij} = 0$.

Using the KKT condition [23], the reader can confirm that the optimal solution for (8) is given by

$$\mathfrak{p}^{i\star} = \mathfrak{b}^i \mu^\star - \mathfrak{a}^i, \quad i \in \mathcal{V}, \tag{9}$$

where

$$\mu^\star = \frac{P_D + \sum_{i=1}^N \mathfrak{a}^i}{\sum_{i=1}^N \mathfrak{b}^i}, \tag{10}$$

is the Lagrange multiplier corresponding to equality constraint (8b). The distributed solution proposed in [15] (in the context of optimal power dispatch problem), which here we denote as EDP algorithm, is

$$\begin{aligned}\mu^i(k+1) = & \mu^i(k) - b(k) \sum\nolimits_{i=1}^N w_{ij}(\mu^i(k) - \mu^j(k)) \\ & - a(k)(\mathfrak{b}^i \mu^i(k) - \mathfrak{a}^i - \bar{P}_D),\end{aligned} \tag{11}$$

in which $\mu^i(k) \to \mu^\star$, $i \in \mathcal{V}$, $a(k), b(k) \to 0$ as $k \to \infty$ and $\bar{P}_D = \frac{P_D}{N}$. The reader can easily confirm that if an agent has access to all the incoming and outgoing signals of any agent $i \in \mathcal{V}$, in 3 number of steps, it can obtain $\mathfrak{a}^i$ and $\mathfrak{b}^i$. In Fig. 1 for example, agent 1 is the malicious agent that can derive $\mathfrak{a}^i$ and $\mathfrak{b}^i$ of any $i \in \mathcal{V}\backslash\{1\}$. An alternative distributed solution for (8), studied in [24], proposes to find $\mu^\star$ in a distributed manner using two PI average consensus algorithms that compute $\frac{\sum_{i=1}^N \mathfrak{a}^i}{N}$ and $\frac{\sum_{i=1}^N \mathfrak{b}^i}{N}$, so agents can use them to compute $\mu^\star$ and consequently $\mathfrak{p}^{i\star}$. The results in [24] show that a malicious agent j cannot find $(\mathfrak{a}^i, \mathfrak{b}^i)$ of any other agent $i \in \mathcal{V}/\{j\}$, if an only if agent i has at least one neighbor that is not a neighbor of the malicious agent. One can also enable agents to obtain $\frac{\sum_{i=1}^N \mathfrak{a}^i}{N}$ and $\frac{\sum_{i=1}^N \mathfrak{b}^i}{N}$ using the Laplacian average consensus algorithm augmented by an additive noise based privacy mechanism proposed in [25]. For this solution, also, privacy is only preserved for agents that have at least one incoming or outgoing signal that is not available to the malicious agent. For such private agents, however, the malicious agent can obtain an estimate on the cost parameters with a known covariance, see Fig. 2.

To obtain a distributed solution that preserves the privacy of all the agents, regardless of what communication messages are available to the malicious agent, we propose to solve (8) using the PrivOpt algorithm as shown next. Since PrivOpt solves the unconstrained optimization problem, we need to reformulate (8). We observe that $\mu^\star$ in (10) is the solution

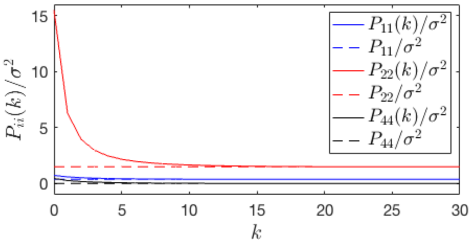

Fig. 2: Implementing the algorithm proposed in [25] for an average consensus problem in an undirected connected graph showed in [Fig 1] [25]. The malicious agent 5 in this scenario knows that in 99.7% of the times (3σ rule) the error rate to obtain local parameters $\mathfrak{a}^3$ and $\mathfrak{b}^3$ are respectively 0.071% and 0.879% according to the computed normalized covariances P_{ii} in the figure above. However, local parameters of agent 4, which all signals transmitted to and from it are accessible to agent 1, are fully revealed to the malicious agent, unlike in PrivOpt.

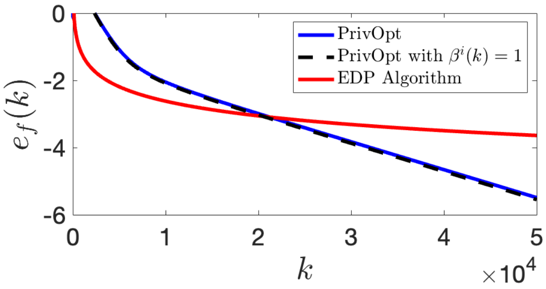

Fig. 3: Convergence error of PrivOpt compared to the EDP algorithm. PrivOpt converges exponentially fast, whereas EDP slows down due to its decreasing stepsizes. Observe that $\beta^i(k)$ does not affect convergence significantly.

of the unconstrained optimization problem

$$\mu^\star = \arg\min_{\mu\in\mathbb{R}} \sum\nolimits_{i=1}^{N} \underbrace{(\frac{1}{2}\mathfrak{b}^i\mu^2 - (\bar{P}_D + \mathfrak{a}^i)\mu)}_{f^i(\mu,\mathfrak{a}^i,\mathfrak{b}^i)},$$

where $f^i(\mu, \mathfrak{a}^i, \mathfrak{b}^i)$ is the local cost of agent $i \in \mathcal{V}$ with parameters $(\mathfrak{a}^i, \mathfrak{b}^i)$. Thus, we can implement PrivOpt to obtain μ^* by choosing $f^i(\mu, \mathfrak{a}^i, \mathfrak{b}^i)$ as the local costs of the agents. Subsequently, every agent $i \in \mathcal{V}$ obtains $\mathfrak{p}^{i\star}$ using its local parameters $\mathfrak{a}^i$ and $\mathfrak{b}^i$ from (9). Implementing PrivOpt as the solution for (8) with local weights $\beta^i(k) = \frac{1}{2}(1 + \sin(ik))$, by virtue of Theorem III.3 since $n^i = 2 > n = 1$ and the parameters appear linearly in the gradient, we have the guarantees that the privacy of all the agents in the network is preserved. Figure 3 compares the convergence performance of our proposed PrivOpt-based solution of (8) with that of the EDP algorithm (11), where $e_f(k) = \log\sqrt{\sum_{i=1}^{N} \|\frac{\mathfrak{p}^i(k)-\mathfrak{p}^{i\star}}{\mathfrak{p}^{i\star}}\|^2}$. As we can see, our solution not only preserves privacy of the agents, but also converges faster than the EDP algorithm. This is because even though EDP converges faster at the beginning, eventually it slows down due to the vanishing nature of its stepsizes $a(k)$ and $b(k)$. On the other hand, PrivOpt-based solution converges exponentially fast. Next, we conduct a series of simulations to investigate whether the use of local weights β^i enables agent 5 to preserve privacy against the malicious agent. Since the malicious agent cannot obtain an accurate estimate of $\mu^5(k)$ when k is relatively small due to (5b), we let agent 1 estimate $\mu^5(k)$ when $k \to \infty$. If the malicious agent forms an under-determined system of equations with unknowns $\mu^5(0)$ and $\{\beta^5(l)\}_{l=0}^{k}$, it is computationally hard to solve for $\mathcal{C}^5$ due to the huge number of unknowns if multiple number of steps are used. Therefore, agent 1 uses a practical value of $\mu^1(\infty)$ in place of $\mu^5(\infty) = \mu^\star$. Next, let is consider practical stopping condition at various values of $e_f < \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. For each of these scenarios, agent 1 uses the last 1000 steps in each of the four scenarios to obtain $\mathfrak{a}^5$ and $\mathfrak{b}^5$ using the linear least-squares method [6]. As we can see in the table below, where each row represents the results of each case, the estimated values have significant errors compared to the actual values of $\mathfrak{a}^5 = 2567.2$ and $\mathfrak{b}^5 = 208.2$. In the best scenario, the error rate to obtain $\mathfrak{a}^5$ and $\mathfrak{b}^5$ are respectively 8.3% and 6.7%. We can also observe that by using all the steps in the execution, less accurate results is achieved due to the effect of $\mu^5(0)$ in (5b). Note that in this special case, since in earlier steps of the execution $\mu^1(k)$ and $\mu^5(k)$ are significantly different, we let agent 1 use arbitrary values for $\mu^5(0)$ and $0 < \beta^5(k) \leq 1$ to estimate the value of $\mu^5(k)$ with $\hat{\mu}^5(k)$ at each step. Eventually, as $k \to \infty$, $\hat{\mu}^5(k) \to \mu^5(k)$.

Case #	Stopping Criteria	$\hat{\mathfrak{a}}^5$	$\hat{\mathfrak{b}}^5$
1	$e_f < 10^{-2}$	4922	339
2	$e_f < 10^{-3}$	2205	150
3	$e_f < 10^{-4}$	2291	187
4	$e_f < 10^{-5}$	2780	222
5	$k = \{1, \cdots, 5\times 10^5\}$	532	10

V. Conclusion

This paper considered the problem of privacy preservation in an in-network unconstrained convex optimization, in the sense of keeping local cost parameters of the agents private. We showed that unlike the existing algorithms, e.g., EXTRA and PI, our proposed distributed solution intrinsically preserves the privacy of all the agents in the network, when the number of private parameters is larger than the number of decision variables of the optimization problem. Our proposed algorithm, PrivOpt, provided this strong privacy-preservation guarantee without requiring extra communication or using additive perturbation noises that may perturb the algorithm's convergence. As an application study, we considered an in-network optimal resource allocation problem. We showed how our proposed PrivOpt could be used as a privacy-preserving solution for this problem, while some known distributed solutions fail to provide such guarantees.

References

[1] A. Mandal, "Privacy preserving consensus-based economic dispatch in smart grid systems," in *International Conference on Future Network Systems and Security*, pp. 98–110, Springer, 2016.

[2] S. Han, W. K. Ng, L. Wan, and V. C. Lee, "Privacy-preserving gradient-descent methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 6, pp. 884–899, 2009.

[3] W. Shi, Q. Ling, G. Wu, and W. Yin, “Extra: An exact first-order algorithm for decentralized consensus optimization,” *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[4] T. Yang, Y. Wan, H. Wang, and Z. Lin, “Global optimal consensus for discrete-time multi-agent systems with bounded controls,” *Automatica*, vol. 97, pp. 182–185, 2018.

[5] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, “A survey of distributed optimization,” *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[6] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*, vol. 2. Chapman & Hall/CRC Boca Raton, FL, 2004.

[7] Z. Huang, S. Mitra, and N. Vaidya, “Differentially private distributed optimization,” in *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, pp. 1–10, 2015.

[8] Y. Lu and M. Zhu, “Privacy preserving distributed optimization using homomorphic encryption,” *Automatica*, vol. 96, pp. 314–325, 2018.

[9] D. Han, K. Liu, H. Sandberg, S. Chai, and Y. Xia, “Privacy-preserving dual averaging with arbitrary initial conditions for distributed optimization,” *IEEE Transactions on Automatic Control*, 2021.

[10] E. Nozari, P. Tallapragada, and J. Cortés, “Differentially private distributed convex optimization via functional perturbation,” *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 395–408, 2016.

[11] Y. Lou, L. Yu, S. Wang, and P. Yi, “Privacy preservation in distributed subgradient optimization algorithms,” *IEEE transactions on cybernetics*, vol. 48, no. 7, pp. 2154–2165, 2017.

[12] S. Mao, Y. Tang, Z. wei Dong, K. Meng, Z. Y. Dong, and F. Qian, “A privacy preserving distributed optimization algorithm for economic dispatch over time-varying directed networks,” *IEEE Transactions on Industrial Informatics*, 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, 2016.

[14] M. T. Hale and M. Egerstedt, “Differentially private cloud-based multi-agent optimization with constraints,” in *American Control Conference*, pp. 1235–1240, 2015.

[15] S. Kar and G. Hug, “Distributed robust economic dispatch in power systems: A consensus+ innovations approach,” in *2012 IEEE Power and Energy Society General Meeting*, pp. 1–8, IEEE, 2012.

[16] F. Bullo, J. Cortes, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*, vol. 27. Princeton University Press, 2009.

[17] C. Grosan and A. Abraham, “Solving nonlinear equation systems using evolutionary algorithms,” in *Proceedings of genetic & evolutionary computation conference, Seattle, USA, Proceedings on CD*, 1996.

[18] C. Grosan and A. Abraham, “A new approach for solving nonlinear equations systems,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 38, no. 3, pp. 698–714, 2008.

[19] S. S. Kia, “Distributed optimal in-network resource allocation algorithm design via a control theoretic approach,” *Systems & Control Letters*, vol. 107, pp. 49–57, 2017.

[20] A. Cherukuri and J. Cortes, “Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment,” *Automatica*, vol. 74, pp. 183–193, 2016.

[21] X. Chen and T. Yamamoto, “Newton-like methods for solving underdetermined nonlinear equations with nondifferentiable terms,” *Journal of Computational and Applied Mathematics*, vol. 55, no. 3, pp. 311–324, 1994.

[22] B. Polyak and A. Tremba, “Solving underdetermined nonlinear equations by newton-like method,” *arXiv preprint arXiv:1703.07810*, 2017.

[23] H. W. Kuhn and A. W. Tucker, “Nonlinear programming,” in *Traces and emergence of nonlinear programming*, pp. 247–258, Springer, 2014.

[24] A. S. Esteki and S. S. Kia, “Deterministic privacy preservation in static average consensus problem,” *IEEE Control Systems Letters*, vol. 5, no. 6, pp. 2036–2041, 2020.

[25] Y. Mo and R. M. Murray, “Privacy preserving average consensus,” *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 753–765, 2016.

Appendix A

An example observer that the malicious agent can use to obtain the private parameters of its neighbors in EXTRA algorithm is the non-linear least squares observer [6] given in Algorithm 1. In Algorithm 1, the notations are as follows: $\hat{\mathbf{q}} = \{\hat{q}^i_1, \hat{q}^i_2, \cdots \hat{q}^i_{n^i}\}^\top$ is the estimate of local parameters $\mathcal{C}^i = \{q^i_1, q^i_2, \cdots q^i_{n^i}\}^\top$, $\tilde{\mathbf{y}} = \{\nabla f^i(x^i(1), \mathcal{C}^i), \nabla f^i(x^i(2), \mathcal{C}^i), \cdots, \nabla f^i(x^i(k+1), \mathcal{C}^i)\}^\top$ is the observations from $k+1$ iterations, ϵ is a prescribed small value, $\mathbf{W} \in \mathbb{R}^{m\times m}$ is the weighting matrix used to weight the relative importance of each measurement, and $\delta J = \frac{|J_t - J_{t-1}|}{J_t}$ is a stopping condition.

Algorithm 1: Nonlinear least-squares observer

```
initialize with $\hat{\mathbf{q}} \in \mathbb{R}^{n^i}$ ;
while $t \leq T$ do
    $\Delta\mathbf{y}_c = \tilde{\mathbf{y}} - \mathbf{f}(\hat{\mathbf{q}})$, $J_t = \mathbf{y}_c^\top \mathbf{W}\mathbf{y}_c$, $\mathbf{H} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}|_{\hat{\mathbf{q}}}$;
    $\Delta\hat{\mathbf{q}} = (\mathbf{H}^\top \mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\top \mathbf{W}\Delta\mathbf{y}_c$ ;
    if $\delta J < \frac{\epsilon}{||\mathbf{W}||}$ then
        STOP
    else
        $\hat{\mathbf{q}} = \hat{\mathbf{q}} + \Delta\hat{\mathbf{q}}$ ;
        $t = t + 1$ ;
    end
end
```

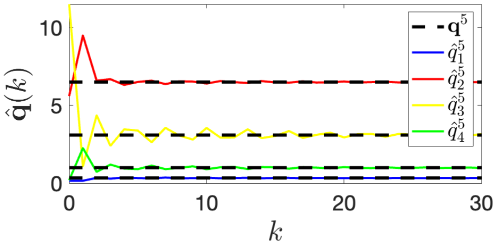

Fig. 4: The least-squares observer of agent 1 results in $\hat{q}^5_1$, $\hat{q}^5_2$, $\hat{q}^5_3$ and $\hat{q}^5_4$ convergence to the true values of $\mathbf{q}^5 = (0.34, 6.5, 3.1, 1.0)$.

Example: Let the agents of the network in Fig. 1 implement the EXTRA algorithm to solve the distributed optimization problem where the local costs are given by $f^i(x^i, \mathcal{C}^i) = q^i_1 e^{q^i_2 x^i} + q^i_3 e^{-q^i_4 x^i}$, $i \in \{1, 2, 3, 4, 5\}$. Let the malicious agent be agent 1. Because agent 1 is the neighbor of all the agents in the network, it can use the nonlinear least-square observer in Algorithm 1 to reconstruct the private parameters of all the other agents in the network. For example, agent 1's observer's performance when it wants to obtain the parameters of agent 5 is depicted in Fig. 4 for a scenario with the following numerical values: $q^5_1 = 0.34, q^5_2 = 6.5, q^5_3 = 3.1, q^5_4 = 1.0$ and initial condition $x^5(0) = 0$. Using the knowledge about the graph topology and the communication messages $x^5(0) = 0, x^5(1) = 0.02, x^5(2) = 0.07, x^5(3) = 0.09$, agent 1 obtains $\nabla f^5(x^5(0)) = -0.9, \nabla f^5(x^5(1)) = -0.53, \nabla f^5(x^5(2)) = 0.57, \nabla f^5(x^5(3)) = 1.11$. We set $\epsilon = 0.01$. As seen in Fig. 4, agent 1 can reconstruct all the parameters of agent 5 using a least-square observer.