Differentially Private Distributed Optimal Power Flow

Dvorkin, Vladimir; Van Hentenryck, Pascal; Kazempour, Jalal; Pinson, Pierre

Published in:
Proceedings of the 59th IEEE Conference on Decision and Control

Link to article, DOI:
10.1109/CDC42340.2020.9303768

Publication date:
2021

Document Version
Peer reviewed version

Differentially Private Distributed Optimal Power Flow

Vladimir Dvorkin[1], Pascal Van Hentenryck[2], Jalal Kazempour[1], Pierre Pinson[1]

***Abstract*— Distributed algorithms enable private Optimal Power Flow (OPF) computations by avoiding the need in sharing sensitive information localized in algorithms sub-problems. However, adversaries can still infer this information from the coordination signals exchanged across iterations. This paper seeks formal privacy guarantees for distributed OPF computations and provides differentially private algorithms for OPF computations based on the consensus Alternating Direction Method of Multipliers (ADMM). The proposed algorithms attain differential privacy by introducing static and dynamic random perturbations of OPF sub-problem solutions at each iteration. These perturbations are Laplacian and designed to prevent the inference of sensitive information, as well as to provide theoretical privacy guarantees for ADMM sub-problems. Using a standard IEEE 118-node test case, the paper explores the fundamental trade-offs among privacy, algorithmic convergence, and optimality losses.**

I. Introduction

Centralized OPF computations operate over large datasets of system parameters, such as electrical loads, and their unintended release poses privacy risks for data owners. Recognizing these risks, the literature suggests replacing the centralized computations with distributed algorithms [1], e.g., using the well-known Alternating Direction Method of Multipliers (ADMM). These algorithms distribute OPF computations among sub-problems that coordinate through primal and dual coordination signals without sharing the parameters used in their local computations and thus, preserving privacy. However, in the presence of side information, adversaries can reverse-engineer local parameters from observed coordination signals [2]. To overcome this limitation, this paper augments these ADMM-based OPF algorithms with *differential privacy*.

Differential privacy, first formalized by Dwork *et al.* [3], [4], is a theoretical framework *quantifying* the privacy risk associated with computing functions (queries) on datasets with sensitive information. It ensures that the same query applied to two adjacent datasets, i.e., differing by one item, return essentially similar results (i.e., up to specified parameters), thus preventing adversaries from learning any substantial information over individual items. Chatzikokolakis *et al.* [5] generalized this concept to a metric-based differential privacy for cases where publicly known participants have sensitive data to protect. For instance, in power systems, instead of hiding the presence of industrial customers, their electrical loads may need to be obfuscated (up to a certain threshold) to avoid revealing their commercial activities. Traditionally, differential privacy is achieved by adding Laplacian noise to query outputs and the noise can be calibrated using a small set of parameters (e.g., the privacy loss ε) to control the differences between the outputs on two adjacent datasets and obtain the guarantee known as $\varepsilon-$differential privacy [3].

Contributions: This paper applies differential privacy to distributed OPF computations using a consensus ADMM, where the OPF sub-problems coordinate voltage variables iteratively without disclosing their local load parameters. The paper first introduces an adversarial model and shows that, in the presence of additional information, adversaries can infer local load parameters from the sub-problem responses to the coordination signals. To remedy this privacy leak, the paper introduces two privacy-preserving ADMM algorithms for OPF computations using static (SP-ADMM) and dynamic (DP-ADMM) random perturbations of sub-problem solutions across iterations. The paper proves that the two algorithms ensure $\varepsilon-$differential privacy but differ in the amount of noise they introduce. The DP-ADMM algorithm ensures privacy at each iteration, but needs to scale the noise magnitude in order to minimize the privacy loss across multiple iterations. On the other hand, the SP-ADMM preserves differential privacy across all iterations uniformly but the worst-case sensitivity of its sub-problem solutions to load datasets must be defined ahead of the algorithm iterations. Numerical experiments highlight that, with a fine calibration of privacy parameters, the inference of loads from primal-dual coordination signals is equivalent to random guessing, *even if an adversary acquires all but one unknown sub-problem parameters*. The experiments also explore the convergence properties of the two algorithms and evaluate their fidelity with respect to the non-private ADMM. In particular, despite similar privacy properties, DP-ADMM results in smaller optimality losses, while SP-ADMM, exhibits faster convergence rates.

Related work: Differentially private distributed computation was first introduced by Zhang *et al.* [6] for the unconstrained empirical risk minimization (ERM) problem. The authors distribute the ERM problem using ADMM and obtain differential privacy for local training datasets by adding noise to either primal or dual variables. The privacy guarantees, however, are provided for a single ADMM iteration. Moreover, the results hold for the unconstrained ERM problem, and are not appropriate for heavily constrained OPF computations. Han *et al.* [7] build a private distributed projected gradient descent algorithm with gradient perturbations for electrical vehicle charging, preventing the inference of charging power from coordination signals. In the OPF context, Mak *et al.* [8] extend the centralized private release

[1]V. Dvorkin, J. Kazempour. and P. Pinson are with the Department of Electrical Engineering, Technical University of Denmark, Lyngby, Denmark. Email: {vladvo,seykaz,ppin}@elektro.dtu.dk

[2]Pascal Van Hentenryck is with the School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, USA. Email: pascal.vanhentenryck@isye.gatech.edu

of OPF test cases [9], [10] to a distributed ADMM-based algorithm. However, the work is meant for the private release of input datasets and does not provide the OPF solution itself.

II. Towards Distributed OPF Computations

Consider a power system as an undirected graph $\Gamma(\mathcal{B}, \Lambda)$, where $\mathcal{B}$ is the set of nodes and Λ is the set of transmission lines. Each transmission line has a susceptance $\beta \in \mathbb{R}_+^{|\Lambda|}$ and a capacity $\overline{f} \in \mathbb{R}_+^{|\Lambda|}$. The mapping functions $\mathrm{s} : \Lambda \mapsto \mathcal{B}$ and $\mathrm{r} : \Lambda \mapsto \mathcal{B}$ are used to return the sending and receiving ends of lines, respectively. The network topology is described by a weighted Laplacian matrix $B \in \mathbb{R}^{|\mathcal{B}|\times|\mathcal{B}|}$, where the weights on the lines are given by their susceptances. The network loads are given by vector $d \in \mathbb{R}_+^{|\mathcal{B}|}$. Each node generates an amount $p \in \mathbb{R}^{|\mathcal{B}|}$ of real power in the interval $[\underline{p}, \overline{p}]$ for a cost given by a quadratic function whose second- and first-order coefficients are $c_2 \in \mathbb{R}_+^{|\mathcal{B}|}$ and $c_1 \in \mathbb{R}_+^{|\mathcal{B}|}$, respectively. The OPF solution amounts to generator set-points $p \in \mathbb{R}^{|\mathcal{B}|}$ and voltage angles $\theta \in \mathbb{R}^{|\mathcal{B}|}$, obtained from the optimization

$$\min_{p,\theta} \quad c(p) = \sum_{i\in\mathcal{B}} c_{2i}p_i^2 + c_{1i}p_i \tag{1a}$$

$$\text{s.t.} \quad \underline{p}_i \leqslant p_i \leqslant \overline{p}_i, \ ,\forall i \in \mathcal{B}, \tag{1b}$$

$$-\overline{f}_l \leqslant \beta_\ell \left(\theta_{\mathrm{s}(\ell)} - \theta_{\mathrm{r}(\ell)}\right) \leqslant \overline{f}_l, \ \forall \ell \in \Lambda, \tag{1c}$$

$$\sum_{j\in\mathcal{B}} B_{ij}\theta_j = p_i - d_i, \forall i \in \mathcal{B}, \tag{1d}$$

which minimizes the total generation cost (1a). Inequality constraints (1b) and (1c) respectively ensure that generation and power flows are within their corresponding limits. Equations (1d) ensure the balance among the load, generation and power flow injection at every network node.

The centralized computation in (1) requires that all network parameters are submitted to a central entity. To avoid the need for sharing network parameters, one may consider instead, a distributed OPF computation, where the network is arbitrarily split into zones $z \in \mathcal{Z}$ [11]. The domestic nodes of each zone z are collected in a set $\mathcal{R}_z$, such that $\mathcal{R}_z \cap \mathcal{R}_{z'} = \emptyset, \forall z \neq z'$. The extended set $\mathcal{V}_z$ contains the domestic nodes of, and adjacent nodes to, zone z. The set of end nodes from the transmission lines adjacent to zone z is then defined as $\mathcal{M}_z = \mathcal{V}_z \cap \mathcal{V}_{z'}. \forall z \neq z'$. To enable the distributed computation, the voltage angles are duplicated per zone, i.e., they are redefined as $\theta \in \mathbb{R}^{|\mathcal{B}|\times|\mathcal{Z}|}$. Towards the purpose, the following consensus constraint is enforced:

$$\theta_{iz} = \overline{\theta}_i : \mu_{iz}, \ \forall z \in \mathcal{Z}, \ \forall i \in \mathcal{M}_z, \tag{2}$$

where θ_{iz} is a local copy of the voltage angle at node i, $\overline{\theta} \in \mathbb{R}^{|\mathcal{B}|}$ is the consensus variable, and $\mu \in \mathbb{R}^{|\mathcal{B}|\times|\mathcal{Z}|}$ is the dual variable of the consensus constraint. This decomposition separates the feasibility region (1b)-(1d) per zone, so we denote the constraint set of each zone by $\mathcal{F}_z$. Now, the computation boils down to the optimization of the following partial Lagrangian function:

$$\max_{\mu} \min_{p,\theta,\overline{\theta}} \mathcal{L}(\mu, p, \theta, \overline{\theta}) = c(p) + \mu_z^\top \left(\overline{\theta} - \theta_z\right)$$
$$\text{s.t.} \quad p, \theta \in \cap_{z\in\mathcal{Z}}\mathcal{F}_z,$$

where the objective function includes the dualized consensus constraint (2) with μ_z and θ_z being z^{th} columns of μ and θ, respectively. The distributed OPF computation is thus enabled by the following ADMM algorithm:

$$\theta_z^{k+1} \leftarrow \underset{(p,\theta_z)\in\mathcal{F}_z}{\operatorname{argmin}} \ \mathcal{L}(\mu^k, p, \theta_z, \overline{\theta}^k) + \frac{\rho}{2}\|\overline{\theta}^k - \theta_z\|_2^2, \ \forall z \in \mathcal{Z},$$

$$\overline{\theta}^{k+1} \leftarrow \underset{\overline{\theta}}{\operatorname{argmin}} \ \mathcal{L}(\mu^k, \theta^{k+1}, \overline{\theta}) + \frac{\rho}{2}\sum_{z\in\mathcal{Z}}\|\overline{\theta} - \theta_z^{k+1}\|_2^2,$$

$$\mu_z^{k+1} \leftarrow \mu_z^k + \rho\left(\overline{\theta}^{k+1} - \theta_z^{k+1}\right), \ \forall z \in \mathcal{Z},$$

where k is an iteration index and the squared norms denote the ADMM regularization terms augmented with a non-negative penalty factor ρ. The algorithm is indeed distributed, as it solely requires the exchange and update of primal and dual coordination signals between the neighboring zones.

The rest of the paper makes the following assumption.

Assumption 1: The function $c(p)$ is convex and strictly monotone in p, the set $\mathcal{F}_z$ is compact and convex for all $z \in \mathcal{Z}$, and $\cap_{z\in\mathcal{Z}}\mathcal{F}_z$ has a non-empty interior. As a result, the distributed OPF algorithm converges to a unique optimal solution in a finite number of iterations [12].

III. Differential Privacy for OPF

The privacy goal in this paper is to ensure that individual loads cannot be inferred from the outputs of the ADMM sub-problems. Each sub-problem is thus considered as a *query* $\mathcal{Q}_z^k$ that maps the dataset of loads $\mathcal{D}_z = \{d_i\}_{i\in\mathcal{R}_z}$ to a vector of all voltage angles $\{\theta_{iz}^{k+1}\}_{i\in\mathcal{M}_z}$ to be released at iteration k. Under Assumption 1, each sub-problem has a unique response for a given load dataset, i.e., it computes a one-to-one mapping of the load dataset to the voltage solution. Hence, the release of the voltage solution leads to the leakage of the load dataset.

The dependencies between sub-problem solutions and their load datasets can be weakened by making queries $\mathcal{Q}_z^k$ differentially private, i.e., by adding a carefully calibrated noise to their outputs. More precisely, the added noise aims at making the outputs for two *adjacent* load datasets $\mathcal{D}_z$ and $\mathcal{D}_z'$ indistinguishable from each other.

Definition 1 (Adjacency [5]): $\mathcal{D}_z = \{d_i\}_{i\in\mathcal{R}_z}$ and $\mathcal{D}_z' = \{d_i'\}_{i\in\mathcal{R}_z}$ are $\alpha-$adjacent datasets, denoted by $\mathcal{D}_z \sim_\alpha \mathcal{D}_z'$, if they differ in one element by α, i.e.,

$$\exists i \text{ s.t. } \|d_i - d_i'\|_1 \leqslant \alpha \ \wedge \ d_j = d_j', \forall j \neq i.$$

Differential privacy relies on the concept of global sensitivity to calibrate the noise.

Definition 2 (Global Query Sensitivity): The global sensitivity Δ_z of query $\mathcal{Q}_z^k$ is defined by

$$\Delta_z^k \coloneqq \max_{\mathcal{D}_z\sim_\alpha\mathcal{D}_z'} \|\mathcal{Q}_z^k(\mathcal{D}_z) - \mathcal{Q}_z^k(\mathcal{D}_z')\|_1.$$

where $\mathcal{D}_z$ and $\mathcal{D}_z'$ belong to the universe $\mathcal{D}_z$ of all datasets of interest for zone z. Note that the datasets in $\mathcal{D}_z$ are projections of the globally feasible solutions of interest.

In practice, especially in off-peak hours, the global sensitivity is overly pessimistic. The notion of local query sensitivity can be used to obtain more precise upper bounds.

Definition 3 (Local Query Sensitivity): The local query sensitivity of query $\mathcal{Q}_z^k$ with respect to $\mathcal{D}_z$, denoted by

Algorithm 1 The SP-ADMM Algorithm

1: **Input:** Datasets $\mathcal{D}_z$, privacy parameters ε, α, algorithmic parameters $\gamma, \rho, K, \overline{\theta}^1, \mu^1$
2: Draw random samples $\xi_z \sim \text{Lap}(\frac{\Delta_z}{\varepsilon}), \forall z \in \mathcal{Z}$
3: **while** $k \neq K$ or $\sum_{z\in\mathcal{Z}} \|\tilde{\theta}_z^k - \overline{\theta}^{k+1}\|_2 \leqslant \gamma$ **do**
4: Update voltage angles $\theta_z^{k+1}, \forall z \in \mathcal{Z}$, by solving
$$\min_{(p,\theta_z)\in\mathcal{F}_z} \mathcal{L}_z(\mu^k, p, \theta_z, \overline{\theta}^k) + \frac{\rho}{2}\|\overline{\theta}^k - \theta_z\|_2^2$$
5: Perturb sub-problem solutions $\tilde{\theta}_z^{k+1} = \theta_z^{k+1} + \xi_z, \forall z \in \mathcal{Z}$,
6: Update consensus variables $\overline{\theta}_i^{k+1}, \forall i \in \mathcal{M}_z, z \in \mathcal{Z}$, as
$$\min_{\overline{\theta}} \mathcal{L}(\mu^k, \tilde{\theta}^{k+1}, \overline{\theta}) + \frac{\rho}{2}\sum_{z\in\mathcal{Z}} \|\overline{\theta} - \tilde{\theta}_z^{k+1}\|_2^2$$
7: Update dual variables $\mu_z^{k+1}, \forall z \in \mathcal{Z}$, by solving
$$\mu_z^{k+1} \leftarrow \mu_z^k + \rho\left(\overline{\theta}_z^{k+1} - \tilde{\theta}_z^{k+1}\right)$$
8: Iteration update $k \leftarrow k+1$
9: **Output:** Private OPF solution.

$\delta_z(\mathcal{D}_z)$, is defined as

$$\delta_z(\mathcal{D}_z) = \max_{\mathcal{D}'\sim_\alpha \mathcal{D}_z} \|\mathcal{Q}_z^k(\mathcal{D}_z) - \mathcal{Q}_z^k(\mathcal{D}'_z)\|_1. \quad (3)$$

For simplicity, when $\mathcal{D}_z$ is clear from the context, δ_z is used to denote $\delta_z(\mathcal{D}_z)$.

Remark 1: The maximal local sensitivity of $\mathcal{Q}_z^k$ depends not only on the magnitude of loads $\{d_i\}_{i\in\mathcal{R}_z}$, but also on their relative position with respect to the nodes in $\mathcal{M}_z$.

This work introduces noise drawn from a zero-mean Laplace distribution with scale b, denoted by $\text{Lap}(b)$ for short, with a probability density function $\text{Lap}(\xi|b) = \frac{1}{2b}\exp(-\frac{\|\xi\|_1}{b})$. It can be used to attain differential privacy for numerical queries as per the following result [3].

Theorem 1 (Laplace mechanism): Let $\mathcal{Q} : \mathcal{D} \mapsto \mathbb{R}$ be a query that maps dataset $\mathcal{D}$ to real numbers, and let Δ be a query sensitivity. The Laplace mechanism that outputs $\mathcal{Q}(\mathcal{D}) + \xi$ with $\xi \sim \text{Lap}(\Delta/\varepsilon)$ is $\varepsilon-$differential privacy, i.e.,

$$\mathbb{P}[\mathcal{Q}(\mathcal{D}) + \xi] \leqslant \mathbb{P}[\mathcal{Q}(\mathcal{D}') + \xi]\exp(\varepsilon),$$

where $\mathcal{D} \sim_\alpha \mathcal{D}'$ are any $\alpha-$adjacent datasets.

The parameter ε, called the *privacy loss*, bounds the multiplicative difference between distributions of query outputs on any two α-adjacent datasets. Stronger privacy requirements can be obtained by choosing smaller values for ε and larger values for α. The last building block is the *sequential composition* theorem [3], which characterizes the guarantees for sequential applications of differential privacy.

Theorem 2 (Sequential composition): Consider T runs of query function $\{\mathcal{Q}_z^t(\mathcal{D}_z)\}_{t=1}^T$ such that every run depends on the result of the previous runs, i.e.,

$$\mathcal{Q}_z^t(\mathcal{D}_z) = \mathcal{Q}_z^t\left(\mathcal{D}_z, \mathcal{Q}_z^1(\mathcal{D}_z), \mathcal{Q}_z^2(\mathcal{D}_z), \ldots, \mathcal{Q}_z^{t-1}(\mathcal{D}_z)\right).$$

Suppose that $\mathcal{Q}_z^t$ preserve ε_t-differential privacy for $\mathcal{Q}_z^{t'}$ for all $t' < t$. Then, the T-tuple mechanism $\mathcal{Q}_z = \left(\mathcal{Q}_z^1, \mathcal{Q}_z^2, \ldots, \mathcal{Q}_z^t\right)$ preserves $\sum_{t=1}^T \varepsilon_t-$differential privacy.

IV. Private Distributed OPF Algorithms

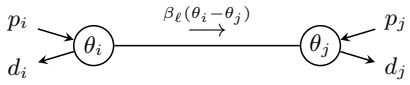
This section presents two differentially private ADMM algorithms for distributed OPF computations.

A. The SP-ADMM Algorithm

The SP-ADMM algorithm relies on a key insight: *the upper bound on the global query sensitivity is independent from the input coordination signals.*

Proposition 1: The global sensitivity of the ADMM sub-problem $\mathcal{Q}_z^k$ is upper-bounded by $\max_{\mathcal{D}_z\in\mathcal{D}_z} \max_i\{d_i\}_{i\in\mathcal{R}_z}$.

Proof: The result follows from the nodal balance constraint (1d). Consider that the voltage angle sensitivity to load changes reduces with the size of the network graph. Therefore, the worst-case sensitivity is observed in a minimal size two-node network, i.e.,



where node i is chosen as a reference node, i.e., $\theta_i = 0$. Then, the power balance at node j is $-\beta_\ell\theta_j = p_j - d_j$. The worst-case sensitivity is provided given that $p_j = 0$, so the change of load directly translates into $\beta_\ell\theta_j$. As $\beta_\ell \gg 1$ in power system networks, the change of θ_j is upper-bounded by the magnitude of d_j. In turn, d_j has to be chosen as the largest feasible load in a dataset universe $\mathcal{D}_z, \forall z \in \mathcal{Z}$. ■

As a result, the SP-ADMM algorithm, shown in Algorithm 1, generates the Laplacian noise $\xi_z \in \mathbb{R}^{|\mathcal{M}_z|}, \forall z \in \mathcal{Z}$, once, at the beginning of the algorithm, using an upper bound Δ_z on the global sensitivity. It takes the dataset, privacy and algorithm parameters as inputs, and runs ADMM iterations until reaching iteration limit K or the primal residual is below the tolerance γ. Unlike the conventional ADMM, once a sub-problem produces the optimal response to the dual and consensus variables, the algorithm perturbs the response with the initially generated noise. The perturbed solution $\tilde{\theta}_z^{k+1}$ then participates in the consensus and dual variable updates.

B. The DP-ADMM Algorithm

The DP-ADMM algorithm is fundamentally different: it uses the concept of local query sensitivity to perturb the phase angles differently at each iteration. Its key insight is the recognition that *the local sensitivity of the queries/sub-problems can be obtained by solving an optimization problem.* As a result, for each iteration, the DP-ADMM perturbs the phase angles using the local sensitivity. The DP-ADMM is outlined in Algorithm 2. The noise $\xi_z^k \in \mathbb{R}^{|\mathcal{M}_z|}, \forall z \in \mathcal{Z}$, is dynamically updated respecting the change of local sensitivity δ_z^k on $\alpha-$adjacent datasets. The sensitivity is obtained by identifying the individual load, whose $\alpha-$change brings the maximal change of the sub-problem solution. The rest of the algorithm is similar to SP-ADMM.

C. Properties

This section reviews the properties of the two algorithms.

Theorem 3: Let $\tilde{\mathcal{Q}}_z^k(\mathcal{D}_z)$ be a randomized sub-problem of zone z acting on optimization dataset $\mathcal{D}_z$ i.e.,

$$\tilde{\mathcal{Q}}_z^k(\mathcal{D}_z) = \theta_z^{k+1} + \xi_z^k.$$

Let δ_z^k be a sensitivity of $\mathcal{Q}_z^k(\mathcal{D}_z)$ for all $\alpha-$adjacent dataset D'_z. Then, if the random perturbation ξ_z^k is sampled from the

Algorithm 2 The DP-ADMM Algorithm

Input: Datasets $\mathcal{D}_z$, privacy parameters ε, α, algorithmic parameters $\gamma, \rho, K, \overline{\theta}^1, \mu^1$

while $k \neq K$ or $\sum_{z\in\mathcal{Z}} \|\tilde{\theta}_z^k - \overline{\theta}^{k+1}\|_2 \leqslant \gamma$ **do**

Update voltage angles $\theta_z^{k+1}, \forall z \in \mathcal{Z}$, by solving

$$\min_{(p,\theta_z)\in\mathcal{F}_z} \mathcal{L}_z(\mu^k, p, \theta, \overline{\theta}^k) + \frac{\rho}{2}\|\overline{\theta}^k - \theta_z\|_2^2$$

For μ_z^k and $\overline{\theta}^k$, compute sensitivity $\delta_z^k, \forall z \in \mathcal{Z}$, by solving

$$\delta_z^k = \max_{\mathcal{D}'\in\mathcal{D}} \|\mathcal{Q}_z^k(\mathcal{D}_z) - \mathcal{Q}_z^k(\mathcal{D}'_z)\|_1,$$
$$\text{s.t. } \|\mathcal{D}_z - \mathcal{D}'_z\|_1 \leqslant \alpha$$

Perturb sub-problem solutions by $\xi_z^k \sim \text{Lap}(\frac{\delta_z^k}{\varepsilon}), \forall z \in \mathcal{Z}$,

$$\tilde{\theta}_z^{k+1} = \theta_z^{k+1} + \xi_z^k$$

Update consensus variables $\overline{\theta}_i^{k+1}, \forall i \in \mathcal{M}_z, z \in \mathcal{Z}$, as

$$\min_{\overline{\theta}} \mathcal{L}(\mu^k, \tilde{\theta}^{k+1}, \overline{\theta}) + \frac{\rho}{2}\sum_{z\in\mathcal{Z}} \|\overline{\theta}_z - \tilde{\theta}_z^{k+1}\|_2^2$$

Update dual variables $\mu_z^{k+1}, \forall z \in \mathcal{Z}$, by solving

$$\mu_z^{k+1} \leftarrow \mu_z^k + \rho\left(\overline{\theta}_z^{k+1} - \tilde{\theta}_z^{k+1}\right)$$

Iteration update $k \leftarrow k+1$

Output: Private OPF solution.

probability distribution with density function $\text{Lap}(\frac{\delta_z^k}{\varepsilon})$, then $\tilde{\mathcal{Q}}_z^k(\mathcal{D})$ provides $\varepsilon-$differential privacy at iteration k, i.e.,

$$\mathbb{P}[\tilde{\mathcal{Q}}_z^k(\mathcal{D}_z)] \leqslant \mathbb{P}[\tilde{\mathcal{Q}}_z^k(\mathcal{D}'_z)]\text{exp}(\varepsilon),\ \forall \mathcal{D}'_z \in \mathcal{D}_z.$$

Proof: The proof follows a similar line of arguments as for numerical queries. We consider the ratio of probabilities that the query $\tilde{\mathcal{O}}_z^k$ returns the same solution $\hat{\theta}_z^{k+1}$ on two $\alpha-$adjacent datasets $\mathcal{D}_z \sim_\alpha \mathcal{D}'_z$ at ADMM iteration k:

$$\frac{\mathbb{P}[\tilde{\mathcal{Q}}_z^k(\mathcal{D}_z) \in \hat{\theta}_z^{k+1}]}{\mathbb{P}[\tilde{\mathcal{Q}}_z^k(\mathcal{D}'_z) \in \hat{\theta}_z^{k+1}]} = \frac{\mathbb{P}[\mathcal{Q}_z^k(\mathcal{D}_z) + \text{Lap}(\xi_z|\frac{\delta_z^k}{\varepsilon}) \in \hat{\theta}_z^{k+1}]}{\mathbb{P}[\mathcal{Q}_z^k(\mathcal{D}'_z) + \text{Lap}(\xi_z|\frac{\delta_z^k}{\varepsilon}) \in \hat{\theta}_z^{k+1}]}$$

$$= \prod_{i\in\mathcal{M}_z} \frac{\frac{\varepsilon}{2\delta_z^k}\text{exp}\left(-\frac{\|\xi_{iz} - \mathcal{Q}_{iz}^k(\mathcal{D}_z)\|_1}{\delta_z^k}\right)}{\frac{\varepsilon}{2\delta_z^k}\text{exp}\left(-\frac{\|\xi_{iz} - \mathcal{Q}_{iz}^k(\mathcal{D}'_z)\|_1}{\delta_z^k}\right)}$$

$$= \prod_{i\in\mathcal{M}_z} \text{exp}\left(\frac{\varepsilon\left(\|\xi_{iz} - \mathcal{Q}_{iz}^k(\mathcal{D}'_z)\|_1 - \|\xi_{iz} - \mathcal{Q}_{iz}^k(\mathcal{D}_z)\|_1\right)}{\delta_z^k}\right)$$

$$\leqslant \prod_{i\in\mathcal{M}_z} \text{exp}\left(\frac{\varepsilon\|\mathcal{Q}_{iz}^k(\mathcal{D}_z) - \mathcal{Q}_{iz}^k(\mathcal{D}'_z)\|_1}{\delta_z^k}\right)$$

$$= \text{exp}\left(\frac{\varepsilon\|\mathcal{Q}_z^k(\mathcal{D}_z) - \mathcal{Q}_z^k(\mathcal{D}'_z)\|_1}{\delta_z^k}\right), \tag{4}$$

where the second equality follows from the definition of the probability density function of the Laplace distribution, and the inequality follows from the inequality of norms. Recall Definition 3 of the local sensitivity. Hence, by substituting (3) in (4), we obtain the desired result. ∎

Remark 2: Theorem 3 holds not only for the DP-ADMM, but also for the SP-ADMM algorithm when used with an upper bound on the global sensitivity.

Remark 3: Theorem 3 makes use of the local sensitivity δ_z^k, thus attaining local $\varepsilon-$differential privacy. By substituting δ_z^k with the global query sensitivity Δ_z, the algorithms provide global $\varepsilon-$differential privacy. The robustness of the two approaches to privacy attacks is analyzed in Section VI.

Observe that every new iteration of the DP-ADMM algorithm reveals more information to an adversary, thus diminishing the privacy guarantee. Assume that the algorithm implementation can limit the adversary to observing T iterations, e.g., by using secure switching of communication channels. Then, the following result applies.

Theorem 4: Let $\tilde{\mathcal{Q}}_z^k(\mathcal{D}_z) = \theta_z^{k+1} + \xi_z^k$ be a randomized query as specified in Theorem 3 with the difference that the noise ξ_z^k is drawn from $\text{Lap}(T\frac{\delta_z^k}{\varepsilon})$. Then, DP-ADMM preserves $\varepsilon-$differential privacy across T iterations.

Proof: It follows from combining Theorems 2 and 3.

Finally, observe that the feasibility of the OPF solution is not affected by either dynamic or static perturbations, as the two algorithms add noise only to the unconstrained consensus and dual variable updates.

V. Adversarial Problem

The strength of differentially private algorithms is their robustness to *side* information. The framework guarantees that, even if an adversary obtains information on *all but one* items in a dataset, the privacy of the remaining one item is ensured. This section presents an adversarial problem for this worst-case scenario of privacy attack on OPF sub-problems.

Consider a set $\mathcal{T} = \{k - T, \ldots, k\}$ of ADMM iterations observed by an adversary. Let $\hat{\theta}_z^{t+1}$ be a response of each sub-problem $z \in \mathcal{Z}$ to dual and consensus variables μ_z^t and $\overline{\theta}_z^t$ at iteration $t \in \mathcal{T}$. For sub-problem z, the adversarial inference problem can be formulated as the following empirical risk minimization problem across T iterations:

$$\min_{\hat{p}^t, \hat{\theta}_z^t, \hat{d}_i} \sum_{t\in\mathcal{T}} c_z\left(\hat{p}^t\right) - [\mu_z^t]^\top \hat{\theta}_z^t + \sum_{t\in\mathcal{T}} \frac{\rho}{2}\|\overline{\theta}^t - \hat{\theta}_z^t\|_2^2 + \Upsilon\sum_{t\in\mathcal{T}} \|\hat{\theta}_z^t - \tilde{\theta}_z^{t+1}\|_2 \tag{5a}$$

$$\text{s.t. Equations (1b)} - \text{(1c)},\ \forall t \in \mathcal{T}, \tag{5b}$$

$$\sum_{m\in\mathcal{V}_z} B_{nm}\hat{\theta}_{mz}^t = \hat{p}_n^t - d_n, \forall n \in \mathcal{R}_z\backslash i, t \in \mathcal{T}, \tag{5c}$$

$$\sum_{m\in\mathcal{V}_z} B_{im}\hat{\theta}_{mz}^t = \hat{p}_i^t - \hat{d}_i, \forall t \in \mathcal{T}, \tag{5d}$$

where decision variables are indicated with a $(\hat{\cdot})$ notation, and the rest are the parameters available to an adversary. The unknown load magnitude $\hat{d}_i$ at node i of interest is modelled as a decision variable. An adversary seeks the value of $\hat{d}_i$ that minimizes the Euclidean distance between the voltage variables $\hat{\theta}_z^{t+1}$ modeled in the adversarial problem and the voltage solution $\tilde{\theta}_z^{t+1}$ released by the sub-problem at all iterations $t \in \mathcal{T}$. By penalizing the distance with a sufficiently large coefficient Υ, an adversary identifies the load magnitude that produces the same voltage solution as that released by the sub-problem, thus identifying the unknown load magnitude.

VI. Numerical Experiments

This section examines the proposed Algorithms 2 and 1 using a standard IEEE 118-node test case with a 3-zone

lay-out taken from [13, case 118-3]. The algorithms are compared in terms of their robustness to privacy attacks, convergence properties, and fidelity with respect to the non-private ADMM algorithm. By default, we set ADMM penalty factor $\rho = 100$, iteration limit $K = 300$, algorithm tolerance $\gamma = 0.5$, and coefficient $\Upsilon = 10^6$. The privacy requirements are selected such that the privacy loss is fixed $\varepsilon = 1$ whereas the adjacency coefficient varies in the range $\alpha = \{1, 2.5, 5, 7, 10\}\%$. For the given algorithmic parameters, the standard non-private ADMM converges to the optimal OPF solution in 59 iterations.

1) Robustness to the Privacy Attacks: The robustness of the algorithms to the load inference is assessed by using the adversarial model in (5). The adversarial model identifies *all* network loads if the standard ADMM is used. The random perturbations of sub-problem solutions, however, prevent the adversary from inferring the actual loads. The results focus on the load at bus 20, which has a median load in the first zone. As per Theorem 3, by specifying the adjacency coefficient α, the algorithms guarantee that, at a given iteration k, an adversary cannot distinguish the magnitude of unknown load d_{20} from any other magnitude in the range $d_{20} \pm \alpha$.

The load inference results for the DP-ADMM algorithm are shown in Fig. 1. The plots show the inferred load at every iteration of the algorithm assuming that only a single iteration is available to an adversary. The inferred load is given as a probability density with each observation corresponding to a single iteration. By increasing α, the inferred load deviates more substantially from the true value of 9 MW, hence, the probability of recovering the true load magnitude reduces. The load obfuscation with SP-ADMM is depicted in Fig. 2. Since the noise is fixed across iterations, the results display 1000 ADMM runs. Similarly to the DP-ADMM algorithm, increasing values of α result in wider distributions of inferred loads. It is important to note that the attacker observes only one sample from these distributions. Observe that the variance of load distributions is notably larger than that of DP-ADMM for a given adjacency value. Moreover, the support of the distributions in Fig. 2 extends drastically with increasing values of α, making the load inference essentially equivalent to a random guess.

Fig. 2 further shows that the use of upper bound on the global sensitivity Δ_z in SP-ADMM, which is set to the largest installed load in the system, results in much stronger privacy protection than the use of local sensitivity, which is set to be at least as much as the maximum local sensitivity observed across DP-ADMM iterations, i.e., $\overline{\delta}_z = \max_k\{\delta_z^k\}_{k=1}^K$. Although both methods enable privacy protection, the formal privacy guarantees provided by SP-ADMM are only achieved with the use of global sensitivity.

Finally, observe that every new iteration of DP-ADMM reveals more information to an adversary, as shown on the left plot in Fig. 3. If the attack budget, i.e., the number of compromised iterations, increases up to T, an adversary recovers the load more precisely. To overcome this limitation, Theorem 4 can be applied to preserve $\varepsilon-$differential privacy across T iterations. This requires to scale the noise parameters by T. The corresponding results in the right plot in Fig. 3 show that the magnitude of the noise increases substantially, thus reducing the quality of load inference even with more information available to an adversary.

TABLE I
OPTIMALITY LOSS INDUCED BY DP-ADMM AND SP-ADMM (%)

α,%	1	2.5	5	7	10
DP-ADMM	0.48	0.92	1.23	1.51	3.83
SP-ADMM	0.28	4.33	11.0	11.35	20.41

2) Convergence analysis: The convergence statistics of the two algorithms obtained with 100 simulation runs are summarized in Fig. 4. The figure shows the evolution of the aggregated primal residual across iterations highlighting important differences between dynamic and static perturbations of sub-problem solutions. With dynamic perturbations, the DP-ADMM algorithm perturbs the sub-problem solutions at every iteration, and the magnitude of the noise increases with α. While the non-private ADMM converges in 59 iterations on this test case, the DP-ADMM requires up to 300 iterations in average, depending on the choice of α. In contrast, the SP-ADMM exhibits a similar computational complexity as the non-private ADMM algorithm. Moreover, in average, the convergence of SP-ADMM is not affected by the choice of adjacency coefficient.

3) Fidelity analysis: It remains to quantify the loss in efficiency of differentially-private OPF solutions. The average optimality loss induced by DP-ADMM and SP-ADMM algorithms for 100 runs is provided in Table 4. The results for attack budget $T = 1$ show that with increasing privacy requirements, both algorithms converge in sub-optimal solutions as compared to the non-private ADMM solution. However, due to dynamically updated zero-mean perturbations, the DP-ADMM has notability better fidelity than the SP-ADMM, which fixes the noise across iterations and constantly steers the OPF dispatch from the optimal solution. This unfolds the following trade-offs among the algorithms: despite better convergence properties of SP-ADMM, it yields a larger optimality gap compared to the DP-ADMM, which demonstrates weaker convergence statistics.

VII. CONCLUSIONS

Although the distributed algorithms have been long trusted to preserve the privacy of network parameters in OPF computations, this paper shows that the standard distributed algorithms do not ensure the information integrity as the sensitive parameters, e.g., electrical loads, are leaked through the exchange of coordination signals. To overcome this limitation, this paper introduces two privacy-preserving OPF ADMM algorithms that satisfy the definition of ε-differential privacy. The algorithms provide privacy be means of either static or dynamic perturbations of the sub-problem solutions at each iteration. The paper shows theoretically and through numerical results that the two algorithms are able to negate the adversarial inference of sensitive information from coordination signals. Despite their complementary privacy properties, the numerical performance of the two algorithms is

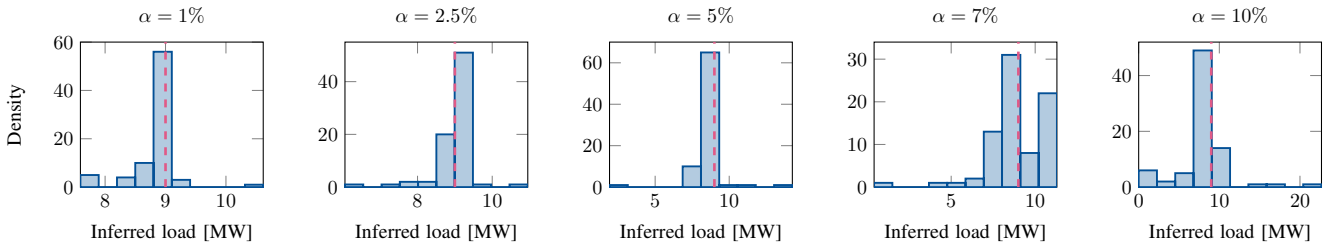


Fig. 1. DP-ADMM: Results of privacy attack on the load sited at node 20. The plots show the densities of inferred load by an adversary across iterations for different adjacency coefficients for a single simulation run.

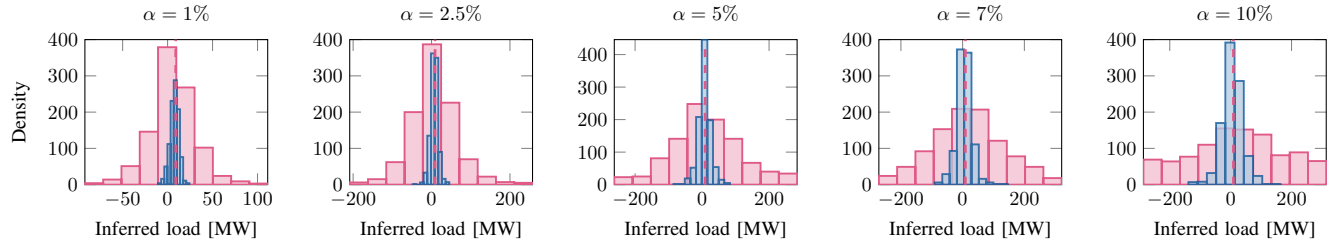


Fig. 2. SP-ADMM: Results of privacy attack on the load sited at node 20. The plots depict the distribution of inferred load by an adversary across 1000 simulation runs for different adjacency coefficient. The red and blue distributions are given for the global and local sensitivities Δ_z and $\overline{\delta}_z$, respectively.

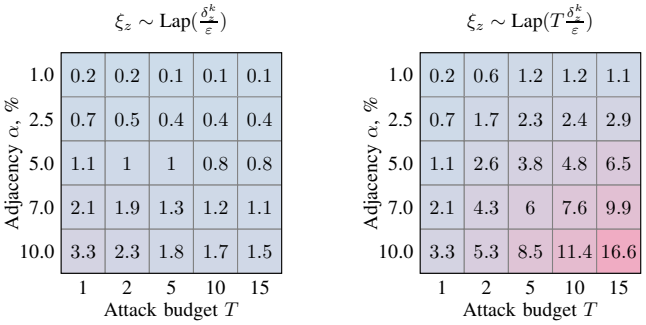


Fig. 3. DP-ADMM: Mean absolute inference error, i.e., mismatch between the actual and inferred loads in MWh, across last T iterations with (right) and without (left) application of Theorem 4 for 100 simulation runs.

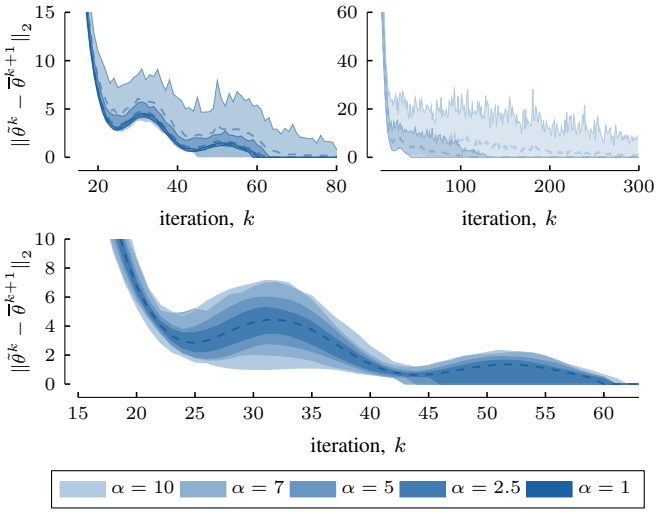


Fig. 4. Convergence of DP-ADMM (top) and SP-ADMM (bottom) algorithms on the 3-zone IEEE 118-node system for different adjacency coefficient α in %. The dashed lines indicate the average residual across 100 runs, whereas the colored areas indicate the spread between the minimum and maximum values of the residual at iteration k. Best view in colors.

mutually exclusive: if static perturbations demonstrate a more robust convergence, their fidelity with respect to the non-private solution is lower than that of dynamic perturbations with weaker convergence statistics.

REFERENCES

[1] D. K. Molzahn *et al.*, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[2] R. Shokri *et al.*, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 3–18.

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

[4] C. Dwork *et al.*, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[5] K. Chatzikokolakis *et al.*, "Broadening the scope of differential privacy using metrics," in *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 2013, pp. 82–102.

[6] T. Zhang and Q. Zhu, "Dynamic differential privacy for ADMM-based distributed classification learning," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 172–187, 2016.

[7] 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.

[8] T. W. Mak, F. Fioretto, and P. Van Hentenryck, "Privacy-preserving obfuscation for distributed power systems," *arXiv preprint arXiv:1910.04250*, 2019.

[9] F. Fioretto and P. Van Hentenryck, "Constrained-based differential privacy: Releasing optimal power flow benchmarks privately," in *CPAIOR*. Springer, 2018, pp. 215–231.

[10] F. Fioretto, T. W. Mak, and P. Van Hentenryck, "Differential privacy for power grid obfuscation," *arXiv preprint arXiv:1901.06949*, 2019.

[11] T. Erseghe, "A distributed approach to the OPF problem," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 1, pp. 1–13, 2015.

[12] S. Boyd *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[13] J. Guo, G. Hug, and O. K. Tonguz, "Intelligent partitioning in distributed optimization of electric power systems," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1249–1258, 2015.