Privacy-Preserving Power System Obfuscation: A Bilevel Optimization Approach

Terrence W.K. Mak, *Member, IEEE* Ferdinando Fioretto, Lyndon Shi, and Pascal Van Hentenryck *Member, IEEE*

***Abstract*—This paper considers the problem of releasing optimal power flow (OPF) test cases that preserve the privacy of customers (loads) using the notion of *Differential Privacy*. It is motivated by the observation that traditional differential privacy algorithms are not suitable for releasing privacy preserving OPF test cases: The added noise fundamentally changes the nature of the underlying optimization and often leads to test cases with no solutions. To remedy this limitation, the paper introduces the *OPF Load Indistinguishability* (OLI) problem, which guarantees load privacy while satisfying the OPF constraints and remaining close to the optimal dispatch cost. The paper introduces an exact mechanism, based on bilevel optimization, as well as three mechanisms that approximate the OLI problem accurately. These mechanisms enjoy desirable theoretical properties, and the computational experiments show that they produce orders of magnitude improvements over standard approaches on an extensive collection of test cases.**

I. Introduction

Releasing high-fidelity energy network test cases is crucial to support the design of efficient and effective algorithms aimed at improving energy network operations. However, the availability of realistic test cases is lagging and has been recognized as a severe impediment to continued scientific progress by federal agencies.[1] On the other hand, releasing such rich datasets raises fundamental privacy concerns: The releasing of the electrical load of a customer can cause significant economic damage, e.g., by revealing sensitive business activities and/or manufacturing processes. Indirectly, they may also reveal how transmission operators operate their networks, raising security issues [1].

To address these issues, several privacy-preserving frameworks have been proposed. In particular, Differential Privacy (DP) [2], [3] captures a desirable privacy property of computations over datasets. In particular, it allows to measure and bound the privacy risk associated with the participation of an individual to an analysis task. Differential privacy algorithms can be used to generate privacy-preserving datasets by introducing carefully calibrated noise to the entries of a dataset, which prevents the disclosure of sensitive information. However, when these privacy-preserving datasets are used as inputs to complex optimization algorithms, e.g., for solving *Optimal Power Flow* (OPF) problems, they may produce results that are fundamentally different from those obtained on the original data. This behavior is shown in Fig. 1, which illustrates the average error (measured as the L_1 distance) between the original and the privacy-preserving loads obtained by using Laplacian noise for a set of 29 networks.[2] As privacy guarantees increase (corresponding to increasing values of parameter α, to be introduced in Section III-B), the privacy-preserving loads become significantly higher than the original ones. Additionally, the number reported on each bar represents the percentage of *feasible* privacy-preserving instances for the AC-OPF problem: They reveal severe feasibility issues with the privacy-preserving procedures.

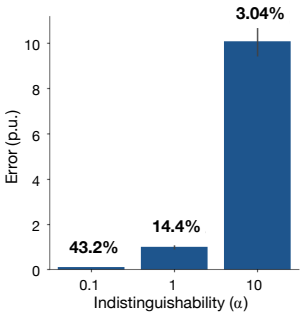


Fig. 1: Average L_1 error reported by the Laplace Mechanism. The percentages express the AC-OPF instances with feasible dispatches.

As a consequence, despite its strong theoretical foundations, industrial adoption of differential privacy has remained limited. Large-scale practical deployments of differential privacy were carried out by large data owners, such as Google [4] and Apple [5]. These applications, however, do not involve releasing data for solving complex optimization problems, but rather for evaluating a pre-defined set of *queries*, e.g., the count of individuals satisfying specific criteria for statistical analysis.

This paper is motivated by the desire of releasing OPF benchmarks that maintain the privacy of customer loads while overcoming the fidelity limitations of traditional differential privacy mechanisms. It formalizes the *OPF Load Indistinguishability* (OLI) problem and proposes a mechanism that applies complex Laplacian noise and leverages the post-processing immunity of differential privacy to redistribute the noise optimally using a bilevel optimization problem. To address the associated computational challenges, the paper proposes three mechanisms that approximate the bilevel optimization: A relaxation of the bilevel optimization, an implementation using the Fritz John conditions, and a Min-Max mechanism that exploits the structure of the bilevel optimization present in practice.

T.W.K. Mak, F. Fioretto, and P. Van Hentenryck are affiliated with the School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332. F. Fioretto is also affiliated with the Electrical Engineering and Computer Science Department, Syracuse University, Syracuse, NY 13244, USA. L. Shi is affiliated with University of Michigan, Ann Arbor, MI 48103. e-mail contacts: wmak@gatech.edu, ffiorett@syr.edu, lynshi@umich.edu, pvh@isye.gatech.edu.

[1]For instance, in 2015, ARPA-E initiated the Grid Data Program to produce high-quality datasets for security-constrained optimal power flows.

[2]The experimental settings are reported in all details in Section VII.

The main contributions of the paper can be summarized as follows: (1) It formalizes the OLI problem and its privacy and fidelity requirements; (2) It formalizes an ideal mechanism that satisfies these requirements using a post-processing step based on a bilevel optimization that redistributes the noise optimally; and (3) It presents a novel Min-Max mechanism that closely approximates the ideal mechanism, preserves its key theoretical properties, and is shown to produce high-fidelity privacy-preserving loads in reasonable time for large-scale test cases. The mechanism uses complex Laplacian noise to keep the relationship between active and reactive loads, guarantees that the OPF cost on the privacy-preserving loads is close to the original cost, and ensures that, in the worst case, the accuracy of the mechanism is at most a constant factor away to the optimal Laplacian mechanism. Moreover, in practice, the Min-Max mechanism provides orders of magnitude improvements compared to standard DP mechanisms.

II. Related Work

There is a rich literature on theoretical results of DP (see e.g., [3], [6]). The literature on DP applied to energy systems includes considerably fewer efforts. Ács and Castelluccia [7] exploit a direct application of the Laplace mechanism to hide user participation in smart meter datasets, achieving ϵ-DP. Zhao et al. [8] study a DP schema that exploits the ability of households to charge and discharge a battery to hide the real energy consumption of their appliances. Liao et al. [9] introduce Di-PriDA, a privacy-preserving mechanism for appliance-level peak-time load balancing control in the smart grid, aimed at masking the consumption of top-k appliances of a household. Finally, Zhou et al. [10] introduce the notion of monotonicity of the DC-OPF operator, which requires that monotonic changes in the network loads induce monotonic changes in the DC-OPF objective cost. This enables a characterization of the network sensitivity, which is useful to preserve the privacy of *monotonic networks*.

A different line of work, conducted by Karapetyan et al. [11] quantifies empirically the trade-off between privacy and utility in demand response systems. The authors analyze the effects of a simple Laplace mechanism on the objective value of the demand response optimization problem. Their experiments on a 4-bus micro-grid show drastic results: the optimality gap only converges to approximately 90% in some cases.

A DP schema that uses constrained post-processing was recently introduced by Fioretto et al. [12] and adopted to release private mobility data. In contrast, the proposal in this work releases the private data set through a mechanism that imposes constraints to ensure the problem solution cost is close to the solution cost of the original problem, and that the underlying optimal power flow constraints are satisfiable.

III. Preliminaries

A. Optimal Power Flow

Optimal Power Flow (OPF) is the problem of determining the best generator dispatch to meet the load demand in a power network. A power network is viewed as a graph (N, E) where the set of buses N represents the nodes and the set of lines E represents the edges. Note that E is a set of directed arcs and E^R is used to denote those arcs in E but in reverse direction. The AC power flow equations are based on complex quantities for current I, voltage V, admittance Y, and power S. These non-convex non-linear equations are a core building block in many power system applications. Practical applications typically include various operational constraints on the flow of power, which are captured in the AC OPF formulation presented in Model 1. The objective function $\mathcal{O}(\boldsymbol{S^g})$ captures the cost of the generator dispatch, with $\boldsymbol{S^g}$ denoting the vector of values $\langle S_i^g \mid i \in N \rangle$. Constraint (2) sets the reference angle to zero for the slack bus $i \in N$ to eliminate numerical symmetries. Constraints (3) and (4) capture the voltage and phase angle difference bounds. Constraints (5) and (6) enforce the generator output and line flow limits. Finally, Constraint (7) captures Kirchhoff's Current Law and Constraint (8) captures the Ohm's Law.

Model 1 P_{OPF}: AC Optimal Power Flow (AC-OPF)

$$\textbf{variables:}\quad S_i^g, V_i \;\; \forall i \in N, \quad S_{ij} \;\; \forall (i,j) \in E \cup E^R$$

$$\textbf{minimize:}\quad \mathcal{O}(\boldsymbol{S^g}) = \sum_{i \in N} c_{2i} (\Re(S_i^g))^2 + c_{1i} \Re(S_i^g) + c_{0i} \tag{1}$$

$$\textbf{subject to:}\quad \angle V_i = 0, \;\; i \in N \tag{2}$$

$$v_i^l \leqslant |V_i| \leqslant v_i^u \;\; \forall i \in N \tag{3}$$

$$-\theta_{ij}^{\Delta} \leqslant \angle (V_i V_j^*) \leqslant \theta_{ij}^{\Delta} \;\; \forall (i,j) \in E \tag{4}$$

$$S_i^{gl} \leqslant S_i^g \leqslant S_i^{gu} \;\; \forall i \in N \tag{5}$$

$$|S_{ij}| \leqslant s_{ij}^u \;\; \forall (i,j) \in E \cup E^R \tag{6}$$

$$S_i^g - S_i^d = \textstyle\sum_{(i,j) \in E \cup E^R} S_{ij} \;\; \forall i \in N \tag{7}$$

$$S_{ij} = Y_{ij}^* |V_i|^2 - Y_{ij}^* V_i V_j^* \;\; \forall (i,j) \in E \cup E^R \tag{8}$$

The paper uses $\boldsymbol{\mathcal{N}} = \langle N, E, \boldsymbol{S^d}, \boldsymbol{Y}, \boldsymbol{\theta^{\Delta}}, \boldsymbol{S}, \boldsymbol{s}, \boldsymbol{v} \rangle$ to succinctly describe a power network, where $\boldsymbol{S^d}$ represents the vector of power loads, $\boldsymbol{Y}$ the admittance matrix, and $\boldsymbol{\theta^{\Delta}}, \boldsymbol{S}, \boldsymbol{s}, \boldsymbol{v}$ represent, respectively, the phase angle difference bounds, the generator output limits, the line flow limits, and the voltage bounds. The paper further defines $n = |N|$ and $m = |E|$.

The OPF problem is specified in Model 1: It takes as input the power network $\boldsymbol{\mathcal{N}}$ and returns the optimal vector of generator dispatches $\boldsymbol{S^g}$ (with ties broken arbitrarily). Since this paper solely focuses on load obfuscation, $P_{\text{OPF}}(\boldsymbol{S^d})$ will be used to denote the OPF problem that takes as input a power network $\boldsymbol{\mathcal{N}}$ with loads $\boldsymbol{S^d}$ and returns an optimal generator dispatch. The solution set satisfying Constraints (2) to (8) for loads $\boldsymbol{S^d}$ is denoted by $\mathscr{C}_{PF}(\boldsymbol{S^d})$. Table I reviews the symbols and notation adopted in the paper.

B. Privacy Goals and Differential Privacy

Differential privacy [2] (DP) is a rigorous privacy notion used to protect the participation disclosure of an individual in a computation. The paper considers datasets $\boldsymbol{D} = \langle S_1^d, \ldots, S_n^d \rangle \in \mathbb{C}^n$ as n-dimensional complex-valued vectors describing the *active* and *reactive* load values S_i^d reported by load i in the network. In our application of interest, the data curator desires to release a *snapshot* of a power network, which includes information on the network topology, lines parameters, as well as the dispatch values of the load

TABLE I: Power Network Nomenclature.

$\boldsymbol{\mathcal{N}}$	A Power network	ϵ	The privacy loss
$\boldsymbol{S}^g$	The set of power generator dispatch	α	The indistinguishability value
$\boldsymbol{S}^d$	The set of load demands	β	The faithfulness parameter
P_{OPF}	The AC-OPF model	$\mathcal{O}^*$	The optimal costs of the original problem
$\mathscr{C}_{\text{PF}}$	The set of feasible AC power flow solutions	$\boldsymbol{x}$	A vector of variables or values
$\mathscr{C}_{\text{BL}}^{\text{FJ}}$	The Fritz John conditions of the OPF problems	x^l, x^u	The upper and lower bounds of quantity x
$\mathcal{M}_x$	A mechanism of type x	$\Re(\cdot), \Im(\cdot)$	The real and imaginary component of a complex number
P_x	The optimization problem for the accuracy phase of $\mathcal{M}_x$	Y^*, I^*	The conjugate of the admittance matrix Y and the complex current I

consumption at each bus. Loads are sensitive, and *the privacy goal is to obfuscate their value up to some quantity* $\alpha > 0$.

To *obfuscate* load values that are close to one another while retaining the distinction between those that are far apart from each other, the following *adjacency relation* is introduced:

$$\boldsymbol{D} \sim_\alpha \boldsymbol{D}' \Leftrightarrow \exists i \text{ s.t. } |S_i^d - S'^d_i| \leqslant \alpha \ \wedge \ S_j^d = S'^d_j, \forall j \neq i,$$

where $\boldsymbol{D}$ and $\boldsymbol{D}'$ are two datasets and $\alpha > 0$ is a positive real value. The above relates two load vectors that differ in at most one item by a value not greater than α. This relation ensures that a differentially private mechanism (introduced next) can protect *an individual load*, even if an attacker acquires information related to all other loads.

Informally speaking, a differentially private mechanism ensures that functions on two adjacent datasets (i.e., datasets differing on a single value by at most α) give similar results. The following definition formalizes this intuition.

Definition *1 (Differential Privacy [2]):* A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ is ϵ-differential private if, for any output response $O \subseteq \mathcal{R}$ and any two *adjacent* inputs $\boldsymbol{D} \sim_\alpha \boldsymbol{D}' \in \mathbb{C}^n$, for a fixed value $\alpha > 0$,

$$Pr[\mathcal{M}(\boldsymbol{D}) \in O] \leqslant e^\epsilon Pr[\mathcal{M}(\boldsymbol{D}') \in O]. \tag{9}$$

The level of *privacy* is controlled by the parameter $\epsilon \geqslant 0$, called the *privacy loss*, with small values denoting strong privacy. The level of *indistinguishability* is controlled by the parameter $\alpha > 0$.

The definition above was first introduced by Chatzikokolakis et. al [13]. It is a generalization of the *classical* differential privacy definition, that protects the participations of individuals into a dataset, to generic metric spaces. This work focuses on Euclidean spaces, since our datasets are in $\mathbb{C}^n$.

Differential privacy satisfies several important properties, including composition and immunity to post-processing. Parallel composition ensures that the privacy loss does not increases when several differentially private mechanisms are applied on different partitions of the dataset. For instance, in our target application, this would correspond to different subsets of loads. The paper uses $\boldsymbol{D} \cap \boldsymbol{D}'$ to denote the vector of loads whose locations are both in $\boldsymbol{D}$ and $\boldsymbol{D}'$. More formally, $\boldsymbol{D} \cap \boldsymbol{D}' = \langle S_i^d \,|\, i \in I_D \wedge i \in I_{D'} \rangle$, with I_D being the set of load locations of D.

Theorem *1 (Parallel Composition [14]):* Let $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ be disjoint subsets of $\boldsymbol{D}$ and $\mathcal{M}$ be an ϵ-differentially private mechanism. Computing $\mathcal{M}(\boldsymbol{D} \cap \boldsymbol{D}_1)$ and $\mathcal{M}(\boldsymbol{D} \cap \boldsymbol{D}_2)$ satisfies ϵ-differential privacy.

The immunity to post-processing ensures that applying arbitrary functions to the output of a differentially private-mechanism preserves its privacy guarantees.

Theorem *2 (Post-Processing Immunity [3]):* Let $\mathcal{M}$ be an ϵ-differential private mechanism and g be an arbitrary mapping from the set of possible output sequences to an arbitrary set. Then, $g \circ \mathcal{M}$ is ϵ-differential private.

Observing values of a dataset vector $\boldsymbol{D}$ is achieved through the means of *numeric queries*, i.e., functions mapping a dataset to a result set. A query Q can be made differentially private by injecting carefully calibrated noise to its output. The amount of noise to inject depends on the *sensitivity* of the query, denoted by Δ_Q and defined as

$$\Delta_Q = \max_{\boldsymbol{D} \sim_\alpha \boldsymbol{D}'} \left\| Q(\boldsymbol{D}) - Q(\boldsymbol{D}') \right\|_1 .$$

For instance, querying the values of a load from a dataset $\boldsymbol{D}$ is achieved through an identity query Q, whose sensitivity $\Delta_Q = \alpha$. The Laplace distribution with 0 mean and scale b, denoted by $\text{Lap}(\lambda)$, has a probability density function $\text{Lap}(x|\lambda) = \frac{1}{2\lambda} e^{-\frac{|x|}{\lambda}}$. It can be used to obtain an ϵ-differentially private algorithm to answer numeric queries [2]. In the following, $\text{Lap}(\lambda)^n$ denotes the i.i.d. Laplace distribution over n dimensions with parameter λ.

Theorem *3 (Laplace Mechanism [2]):* Let Q be a numeric query that maps datasets to $\mathbb{R}^n$. The Laplace mechanism that outputs $Q(\boldsymbol{D}) + z$, where $z \in \mathbb{R}^n$ is drawn from the Laplace distribution $\text{Lap}\left(\frac{\Delta_Q}{\epsilon}\right)^n$, achieves ϵ-differential privacy.

The Laplace mechanism can be extended to work on the complex plane by drawing the noise z from the Polar Laplacian distribution; Its probability density function, given a point (r, θ) in polar coordinates, is: $\text{PLap}(r, \theta|\lambda) = \frac{\lambda^2}{2\pi} r\, e^{-\lambda r}$. The (Polar) Laplace mechanism with parameter $\lambda = \alpha/\epsilon$ satisfies α-indistinguishability [13]. This approach was first introduced in [15]. Without loss of generality, the document refers to this approach as the Laplace mechanism.

While privacy is evaluated through the means of the indistinguishability value α, the *utility* of the privacy-preserving dataset is evaluated based on its fidelity with respect solving OPF problems and such that their solution cost is *close* to the solution cost computed using the original data.

C. A Bayesian Interpretation of DP

Differential privacy can also be interpreted as a bound on the ability of an attacker to learn anything substantially new about a dataset after observing a differentially private response about

the data. Using conditional probabilities, Definition 1 can be expressed and rewritten as:

$$\Pr[R \in O|\boldsymbol{D}] \leqslant e^{\epsilon} \Pr[R \in O|\boldsymbol{D}'], \tag{10}$$

where R is a random variable denoting the response of mechanism $\mathcal{M}$. An attacker tries to gain information on $\boldsymbol{D}$ by exploiting R, as captured by $\Pr[\boldsymbol{D}|R]$. Using Bayes' rule:

$$\Pr[\boldsymbol{D}|R] = \frac{\Pr[R|\boldsymbol{D}]\Pr[\boldsymbol{D}]}{\Pr[R]},$$

where $\Pr[\boldsymbol{D}]$ is the *prior* probability on dataset $\boldsymbol{D}$ (i.e., the attacker's belief of what the dataset may contain), and $\Pr[\boldsymbol{D}|R]$ is the attacker's posterior about $\boldsymbol{D}$ given the observation R. This Bayesian interpretation is interesting as it allow us to relate with the concept of *non-learnability*. This concept captures the change of the attacker's posterior with respect to her prior and requires these quantities to be mostly unchanged after observing the event R, i.e., $\Pr[\boldsymbol{D}|R] \sim \Pr[\boldsymbol{D}]$.

Differential privacy focuses on the difference of beliefs with respect to adjacent datasets:

$$\frac{\Pr[\boldsymbol{D}|R]}{\Pr[\boldsymbol{D}'|R]} \quad \text{when } \boldsymbol{D} \sim_{\alpha} \boldsymbol{D}'$$

that is, an attacker knows a-posteriori how much $\boldsymbol{D}$ is more likely than $\boldsymbol{D}'$ given the response R. Using Bayes'rule:

$$\frac{\Pr[\boldsymbol{D}|R]}{\Pr[\boldsymbol{D}'|R]} = \frac{\Pr[R|\boldsymbol{D}]\Pr[\boldsymbol{D}]}{\Pr[R|\boldsymbol{D}']\Pr[\boldsymbol{D}']} \leqslant e^{\exp} \frac{\Pr[\boldsymbol{D}]}{\Pr[\boldsymbol{D}']}. \tag{11}$$

The inequality follows from (10), and (11) tells us that differentially private mechanisms guarantee that ratios of posterior beliefs for adjacent datasets never increase too much when compared to their ratio of the belief a-priori.

The above also implies that even if an attacker has a hold on some *publicly available* information about the data (e.g., the OPF cost), its belief about a dataset (e.g., load values of customers) does not change too much after observing the output response of a differentially privacy mechanism. Of course, if the a-priori knowledge about the dataset is extensive, then it will remain as such even after observing the differentially private response. [3]

IV. OPF-Load Indistinguishability

The *OPF Load Indistinguishability* (OLI) problem aims at releasing the loads of a power system data in a privacy-preserving manner. It takes as input the loads $\boldsymbol{S}^d$ and returns obfuscated loads $\hat{\boldsymbol{S}}^d$ that satisfy two desiderata:

1) *Privacy*: $\boldsymbol{S}^d$ and $\hat{\boldsymbol{S}}^d$ are α-indistinguishable, i.e., they satisfy Definition 1 for a given value $\alpha > 0$.
2) *Fidelity*: The objective of the OPF problem with loads $\hat{\boldsymbol{S}}^d$ is close to the OPF objective with loads $\boldsymbol{S}^d$, i.e.,

$$\left|\mathcal{O}(P_{\text{OPF}}(\hat{\boldsymbol{S}}^d)) - \mathcal{O}^*\right| \leqslant \beta\mathcal{O}^* \tag{12}$$

for some parameter $\beta > 0$, where $\mathcal{O}^* = P_{\text{OPF}}(\boldsymbol{S}^d)$ is assumed to be a publicly available information.

The second condition makes sure that the privacy-preserving OPF is feasible and preserves the original OPF problem cost.

[3]This would be the case of small benchmark networks, e.g. with only one generator and one load where the cost may reveal the generation and load values.

V. Bilevel Optimization for Load Obfuscation

To address the OLI problem, this section introduces a novel mechanism, called $\mathcal{M}_{\text{OLI}}$, that consists of two phases:

1) **Privacy-Phase**: $\mathcal{M}_{\text{OLI}}$ first applies the Laplace mechanism to obtain an α-indistinguishable demand vector $\tilde{\boldsymbol{S}}^d$:

$$\mathcal{M}_{\text{PLap}}(\boldsymbol{S}^d, \alpha/\epsilon) = \tilde{\boldsymbol{S}}^d = \boldsymbol{S}^d + \text{PLap}(\alpha/\epsilon)^n, \tag{13}$$

where $\tilde{\boldsymbol{S}}^d$ is the resulting vector of noisy load demands.
2) **Fidelity-Phase**: $\mathcal{M}_{\text{OLI}}$ then post-processes the resulting noisy demands through the optimization problem P_{OLI} (introduced below) to obtain a new load vector $\hat{\boldsymbol{S}}^d$.

In the following, $\boldsymbol{\mathcal{N}}$ denotes the original power system data, $\tilde{\boldsymbol{\mathcal{N}}}$ the power system whose loads have been obfuscated using the Laplace mechanism (Equation (13)), as a result of the privacy phase, and $\hat{\boldsymbol{\mathcal{N}}}$ the post-processed power system obtained as a result of the fidelity phase.

After executing the two phases, $\mathcal{M}_{\text{OLI}}$ can then release the power system description $\hat{\boldsymbol{\mathcal{N}}}$ with the post-processed loads $\hat{\boldsymbol{S}}^d$. The heart of the OLI Mechanism is the optimization problem, P_{OLI}, executed during the fidelity phase to attain the OLI desiderata, and defined as:

$$P_{\text{OLI}} = \min_{(\hat{\boldsymbol{S}}^d, \boldsymbol{S}^g)} \|\hat{\boldsymbol{S}}^d - \tilde{\boldsymbol{S}}^d\|^2 \tag{O1}$$

$$\text{s.t.:} \quad \left|\mathcal{O}(\boldsymbol{S}^g) - \mathcal{O}^*\right| \leqslant \beta\mathcal{O}^* \tag{O2}$$

$$\boldsymbol{S}^g = P_{\text{OPF}}(\hat{\boldsymbol{S}}^d). \tag{O3}$$

P_{OLI} is a bilevel program that takes as input a power system $\tilde{\boldsymbol{\mathcal{N}}}$ with noisy load values $\tilde{\boldsymbol{S}}^d$, as well as two positive real numbers: α, the *indistinguishability level*, and β, which determines the required *fidelity* of the optimization problem over the privacy-preserving data. Additionally, the data owner provides the OPF model and the optimal objective value $\mathcal{O}^*$, which are considered public information.

P_{OLI} is a bilevel program, whose upper level objective (O1) minimizes the L2 distance between the noisy original loads $\tilde{\boldsymbol{S}}^d$ and the variables $\hat{\boldsymbol{S}}^d$ representing the new *post-processed* loads. Constraint (O3) captures the lower-level optimization: It computes an optimal generator dispatch $\boldsymbol{S}^g$ for the post-processed loads $\hat{\boldsymbol{S}}^d$ using Model 1. Constraint (O2) requires this generator dispatch to achieve β-fidelity with the respect to the original objective value.

The OLI post-processing can be thought as redistributing the noise of the Laplace mechanism to obtain new power load values that satisfy the OLI desiderata. It searches for a solution to the OPF that also satisfies the β-fidelity constraint. *A feasible solution always exists, since the original loads $\boldsymbol{S}^d$ trivially satisfy all constraints.* A similar approach has been successfully applied to other problems, e.g. load disaggregation [16].

Notice that the fidelity phase of $\mathcal{M}_{OLI}$ only uses the privacy-preserving loads $\tilde{\boldsymbol{S}}^d$ and public information.

Theorem *4:* $\mathcal{M}_{\text{OLI}}$ satisfies α-indistinguishability.

Proof. Each load $\tilde{S}_i^d$ $(i \in N)$ obtained from the application of the Laplace mechanism (13) is α-indistinguishable by Theorem 3. Their combination produces a vector $\tilde{\boldsymbol{S}}^d$ that is α-indistinguishable by parallel composition (Theorem 1). The result follows from post-processing immunity (Theorem 2). □

Corollary *1:* $\mathcal{M}_{\text{OLI}}$ satisfies the privacy and fidelity desiderata of the OLI problem.

The above follows directly from Theorem 4 and the application of problem P_{OLI}.

In addition, $\mathcal{M}_{\text{OLI}}$ has a strong accuracy guarantee: It is no more than a constant factor away from optimality since the Laplace mechanism has been proved optimal for differentially private identity queries [17].

Theorem *5:* The optimal solution $\hat{\boldsymbol{S}}^{\boldsymbol{d}}$ to the optimization model P_{OLI} satisfies $\|\hat{\boldsymbol{S}}^d - \boldsymbol{S}^d\|_2 \leqslant 2\|\tilde{\boldsymbol{S}}^d - \boldsymbol{S}^d\|_2$.

Proof. We have

$$\|\hat{\boldsymbol{S}}^d - \boldsymbol{S}^d\|_2 \leqslant \|\hat{\boldsymbol{S}}^d - \tilde{\boldsymbol{S}}^d\|_2 + \|\tilde{\boldsymbol{S}}^d - \boldsymbol{S}^d\|_2 \tag{14}$$
$$\leqslant 2\|\tilde{\boldsymbol{S}}^d - \boldsymbol{S}^d\|_2. \tag{15}$$

where the first inequality follows from the triangle inequality on norms and the second inequality follows from

$$\|\hat{\boldsymbol{S}}^d - \tilde{\boldsymbol{S}}^d\|_2 \leqslant \|\tilde{\boldsymbol{S}}^d - \boldsymbol{S}^d\|_2$$

by optimality of $\hat{\boldsymbol{S}}^d$ and the fact that the original loads $\boldsymbol{S}^d$ are a feasible solution to the optimization program P_{OLI}. □

The theorem generalizes a prior result from [18]: It shows that the post-processing optimization restores feasibility and fidelity of loads and remains close to optimality.

VI. OPF-Load Indistinguishability Mechanisms

Although mechanism $\mathcal{M}_{\text{OLI}}$ meets the requirements of the OLI problem, bilevel programming is known to be challenging computationally. It is strongly NP-hard [19] and, in addition, even determining whether a solution is optimal is NP-hard [20]. To address the underlying computational challenge, this paper explores three approaches that only modify the fidelity phase of $\mathcal{M}_{\text{OLI}}$. Interestingly, Theorem 5 continues to hold for each of the proposed mechanisms.

A. The Relaxation Mechanism $\mathcal{M}_R$

The relaxation mechanism $\mathcal{M}_{\text{R}}$ aims to provide a relaxation P_{R} of the bilevel optimization that is efficient to compute.

$$P_{\text{R}} = \min_{(\hat{\boldsymbol{S}}^d, \boldsymbol{S}^g)} \quad \|\hat{\boldsymbol{S}}^d - \tilde{\boldsymbol{S}}^d\|^2 \tag{16}$$
$$\text{s.t.:} \quad \left|\mathcal{O}(\boldsymbol{S}^g) - \mathcal{O}^*\right| \leqslant \beta\mathcal{O}^* \tag{17}$$
$$\boldsymbol{S}^g \in \mathscr{C}_{PF}(\hat{\boldsymbol{S}}^d) \tag{18}$$

The relaxation problem P_{R} relaxes the requirement of optimality (O3) and only requires feasibility (18), hence reducing a bilevel program into single level. The mechanism essentially restores feasibility of the loads and ensures that there exists a dispatch $\boldsymbol{S}^g$ that is close to the original dispatch in cost. Since the relaxation relaxes optimality, the dispatch is not guaranteed to be optimal, yet it is simpler computationally. A version $\mathcal{M}_{\text{R}}$ protecting only active loads was proposed in [18].

B. The Fritz John Mechanism $\mathcal{M}_{FJ}$

A traditional approach to bilevel optimization is to replace the lower-level optimization by its *Fritz John* (FJ) conditions (that generalizes the KKT conditions). If $f^l(\boldsymbol{x}, \boldsymbol{y})$ is the lower-level objective and $g^l(\boldsymbol{x}, \boldsymbol{y})$ are the lower-level constraints (expressed as inequalities), the FJ conditions are given by

$$\nabla_y \lambda_0 f^l(\boldsymbol{x}, \boldsymbol{y}) - \textstyle\sum_{i=1}^{k^l} \lambda_i \nabla_y g_i^l(\boldsymbol{x}, \boldsymbol{y}) = 0 \tag{c1}$$
$$\lambda_i \geqslant 0 \quad \forall i \in [k^l] \tag{c2}$$
$$\lambda_i g_i^l = 0 \quad \forall i \in [k^l] \tag{c3}$$

where the λ_0, λ_i $(i \in [k^l])$ are *multipliers*. Mechanism $\mathcal{M}_{\text{FJ}}$ uses a post-processing optimization that augments the relaxation of the bilevel program with the FJ conditions, i.e.,

$$P_{\text{FJ}} = \min_{(\hat{\boldsymbol{S}}^d, \boldsymbol{S}^g, \boldsymbol{\lambda})} \quad \|\hat{\boldsymbol{S}}^d - \tilde{\boldsymbol{S}}^d\|^2 \tag{F1}$$
$$\text{s.t.:} \quad \left|\mathcal{O}(\boldsymbol{S}^g) - \mathcal{O}^*\right| \leqslant \beta\mathcal{O}^* \tag{F2}$$
$$\boldsymbol{S}^g \in \mathscr{C}_{PF}(\hat{\boldsymbol{S}}^d) \tag{F3}$$
$$(\boldsymbol{S}^g, \hat{\boldsymbol{S}}^d, \boldsymbol{\lambda}) \in \mathscr{C}_{\text{BL}}^{\text{FJ}}, \tag{F4}$$

where $\boldsymbol{\lambda}$ is the vector of FJ multipliers, $\mathscr{C}_{\text{BL}}^{\text{FJ}} = \{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) \in \mathbb{R}^{n+m+k^l} \mid \text{(c1), (c2), (c3) hold}\}$, and the FJ conditions (c1), (c2), and (c3) are for the OPF problem.

$\mathcal{M}_{\text{FJ}}$ guarantees feasibilty of the OPF for the post-processed loads. It does not guarantee global optimality of the resulting model since the lower-level program is not convex.

Even though the optimization problem P_{FJ} is single-level, it remains challenging to solve, as it is typically rewritten as a mixed integer non-linear program. Section VII analyzes its computational behavior for a large number of test cases.

C. The Min-Max Mechanism $\mathcal{M}_M$

The relaxation mechanism $\mathcal{M}_{\text{R}}$ produces the post-processed loads $\hat{\boldsymbol{S}}^d$ and their associated dispatch $\boldsymbol{S}^g$ that is close to $\mathcal{O}^*$. However, the optimal dispatch $\boldsymbol{S}_o^g = P_{\text{OPF}}(\hat{\boldsymbol{S}}^d)$ may not satisfy the fidelity constraint

$$\left|\mathcal{O}(\boldsymbol{S}_o^g) - \mathcal{O}^*\right| \leqslant \beta\mathcal{O}^*.$$

By optimality of the OPF problem, the following must hold:

$$\mathcal{O}(\boldsymbol{S}_o^g) \leqslant \mathcal{O}(\boldsymbol{S}^g) \leqslant \mathcal{O}^*(1+\beta).$$

Therefore,

$$\mathcal{O}(\boldsymbol{S}_o^g) < \mathcal{O}^*(1-\beta).$$

when the fidelity constraint is not satisfied. The key idea of the Min-Max mechanism $\mathcal{M}_{\text{M}}$ is to increase the post-processed loads in order to increase the corresponding OPF cost above $\mathcal{O}^*(1-\beta)$. Intuitively, increasing the amount of loads will increase the overall dispatch costs to maintain flow balance. This will also strengthen the chance to satisfy the fidelity requirement.

Algorithm 1 describes the Min-Max Mechanism, which operates in two phases. The first phase is responsible for finding a set of loads $\hat{\boldsymbol{S}}^{\boldsymbol{d}}$ resulting in an OPF dispatch that satisfies the β-fidelity requirement (Equation (12)). The second phase attempts at reducing the distance between the loads $\hat{\boldsymbol{S}}^{\boldsymbol{d}}$ and the Laplacian, private, loads $\tilde{\boldsymbol{S}}^d$.

In more details, the algorithm takes as input a parameter κ, used as a multiplicative factor for increasing the multiplier λ (introduced later) in the first phase, and a parameter λ^{tol}, expressing the tolerance to control the termination of phase 2.

The first phase is described in lines 3–11 of Algorithm 1. After initializing the multipliers $\lambda, \lambda^l, \lambda^u$ (line 4), it uses the Relaxation Mechanism introduced in Section VI-A to find a load vector $\hat{\boldsymbol{S}}^{\boldsymbol{d}}_{\boldsymbol{R}}$ (line 5). The latter is used to seed the iterative process that searches for a new load vector satisfying the β-fidelity requirement (lines 6–10). To do so, the algorithm solves the load maximization problem $P_{\mathrm{M}}(\lambda, \hat{\boldsymbol{S}}^d_R)$:

$$\begin{aligned} \text{argmax} \quad & \|\hat{\boldsymbol{S}}^d\| && \text{(M1)} \\ \text{s.t.:} \quad & \left|\mathcal{O}(\boldsymbol{S}^g) - \mathcal{O}^*\right| \leqslant \beta\mathcal{O}^* && \text{(M2)} \\ & \boldsymbol{S}^g \in \mathscr{C}_{PF}(\hat{\boldsymbol{S}}^d) && \text{(M3)} \\ & \|\hat{\boldsymbol{S}}^d - \tilde{\boldsymbol{S}}^d\| \leqslant \lambda \|\hat{\boldsymbol{S}}^d_R - \tilde{\boldsymbol{S}}^d\| && \text{(M4)} \end{aligned}$$

with increasing values of $\lambda \geqslant 1.0$ until it finds loads $\hat{\boldsymbol{S}}^d$ such that

$$\left|\mathcal{O}(P_{\mathrm{OPF}}(\hat{\boldsymbol{S}}^d)) - \mathcal{O}^*\right| \leqslant \beta\mathcal{O}^*.$$

Constraint (M4) ensures that the post-processed loads are not too far from the post-processed loads $\hat{\boldsymbol{S}}^d_R$ obtained by the relaxation mechanism $\mathcal{M}_{\mathrm{R}}$. Note that, when P_{M} is infeasible, increasing λ will have no effect if $\|\hat{\boldsymbol{S}}^d_R - \tilde{\boldsymbol{S}}^d\| = 0$ (i.e., the relaxation mechanism returns a zero objective). In this case, the algorithm replaces (M4) by: $\|\hat{\boldsymbol{S}}^d - \tilde{\boldsymbol{S}}^d\| \leqslant \lambda \cdot \gamma$ where γ is a small constant to allow feasibility by increasing λ (line 10).

The second phase of the mechanism is described in lines (12–20). In addition to the tolerance parameter λ^{tol}, it takes as input parameters λ^l and λ^u, constructed during phase 1, as well as the load vector $\hat{\boldsymbol{S}}^d_R$ resulting by solving the relaxation mechanism. Its goal is that of decreasing the value of λ obtained in the first phase to make it as tight as possible. It does so by using a binary search scheme, where at each iteration, the value λ is tighten within updated interval $[\lambda^l, \lambda^u]$ (lines 14–19). The process terminates when the range of the above interval is smaller than the given tolerance level λ^{tol} (line 17).

In general, phase 1 of mechanism $\mathcal{M}_{\mathrm{M}}$ is not guaranteed to terminate. However, it converges when the OPF problem is locally monotone around $\boldsymbol{S}^d$ for parameter β. The computational results also show that $\mathcal{M}_{\mathrm{M}}$ quickly converges on all the test cases, indicating that the OPF behaves monotonically around the solution of the bilevel optimization in these configurations. The next two definitions characterize local monotonicity.

Definition *2 (Load Neighborhood):* A load $\hat{\boldsymbol{S}}^d$ is a β-neighbor of $\boldsymbol{S}^d$ if there exists a dispatch $\boldsymbol{S}^g \in \mathscr{C}_{PF}(\hat{\boldsymbol{S}}^d)$ such that $\left|\mathcal{O}(\boldsymbol{S}^g) - \mathcal{O}^*\right| \leqslant \beta\mathcal{O}^*$ where $\mathcal{O}^* = \mathcal{O}(P_{\mathrm{OPF}}(\boldsymbol{S}^{\boldsymbol{d}}))$.

Definition *3 (Local Monotonicity):* The OPF is locally-monotone around $\boldsymbol{S}^d$ and β if

$$\|\hat{\boldsymbol{S}}^d_1\| \geqslant \|\hat{\boldsymbol{S}}^d_2\| \Rightarrow \mathcal{O}(P_{\mathrm{OPF}}(\hat{\boldsymbol{S}}^{\boldsymbol{d}}_{\boldsymbol{1}})) \geqslant \mathcal{O}(P_{\mathrm{OPF}}(\hat{\boldsymbol{S}}^{\boldsymbol{d}}_{\boldsymbol{2}}))$$

whenever $\hat{\boldsymbol{S}}^d_1$ and $\hat{\boldsymbol{S}}^d_2$ are β-neighbors of $\boldsymbol{S}^d$.

The definition of Local Monotonicity is similar, in spirit, to the monotonicity in [10]. However, local monotonicity only requires monotonicity around neighboring intervals of $\boldsymbol{S}^d$, instead of a more broad requirement on monotonicity for the feasible space.

Algorithm 1: The $\mathcal{M}_{\mathrm{M}}$ Fidelity Phase.

```
   Inputs : ⟨κ, λ^tol⟩
1  Ŝ^d_R, λ^l, λ^u ← Phase1(⟨κ⟩)
2  Ŝ^d ← Phase2(⟨λ^l, λ^u, Ŝ^d_R, λ^tol⟩)
   Output : Ŝ^d
3  Function Phase1(⟨κ⟩):
4      λ ← 1.0, λ^l ← 1.0, λ^u ← ∞
5      Ŝ^d_R ← P_R()
6      for i = 1, 2, ... do
7          Ŝ^d_(i) ← P_M(λ, Ŝ^d_R)
8          S^g_(i) ← P_OPF(Ŝ^d_(i))
9          if |O(S^g_(i)) − O*| ⩽ βO* then λ^u ← λ, terminate;
10         λ^l ← λ, λ ← λκ
11     return Ŝ^d_R, λ^l, λ^u
12 Function Phase2(⟨λ^l, λ^u, Ŝ^d_R, λ^tol⟩):
13     λ ← (λ^l + λ^u)/2
14     for i = 1, 2, ... do
15         Ŝ^d_(i) ← P_M(λ, Ŝ^d_R)
16         S^g_(i) ← P_OPF(Ŝ^d_(i))
17         if (λ^u − λ^l) < λ^tol then terminate;
18         if |O(S^g_(i)) − O*| ⩽ βO* then λ^u ← λ else
               λ^l ← λ ;
19         λ ← (λ^u + λ^l)/2
20     return P_M(λ^u, Ŝ^d_R)
```

Theorem *6:* $\mathcal{M}_{\mathrm{M}}$ converges when the OPF is locally-monotone around $\boldsymbol{S}^d$ and β.

Proof. Let $\hat{\boldsymbol{S}}^d_R$ be a β-neighbor of $\boldsymbol{S}^d$. First note that

$$\begin{aligned} \lambda \geqslant \lambda' \geqslant 1 &\Rightarrow \|P_M(\lambda, \hat{\boldsymbol{S}}^d_R)\| \geqslant \|P_M(\lambda', \hat{\boldsymbol{S}}^d_R)\| \\ &\Rightarrow \mathcal{O}(P_{\mathrm{OPF}}(P_M(\lambda, \hat{\boldsymbol{S}}^d_R))) \\ &\geqslant \mathcal{O}(P_{\mathrm{OPF}}(P_M(\lambda', \hat{\boldsymbol{S}}^d_R))) \end{aligned}$$

by definition of P_M and local monotonicity, which means that the OPF increases as λ grows. Moreover, there exists a value λ^* for which $\boldsymbol{S}^g_o = P_{\mathrm{OPF}}(\hat{\boldsymbol{S}}^d)$ is a feasible solution to $P_M(\lambda^*, \hat{\boldsymbol{S}}^d_R)$, i.e.,

$$\begin{aligned} &\left|\mathcal{O}(\boldsymbol{S}^g_o) - \mathcal{O}^*\right| \leqslant \beta\mathcal{O}^* \\ &\boldsymbol{S}^g_o \in \mathscr{C}_{PF}(\hat{\boldsymbol{S}}^d) \\ &\|\hat{\boldsymbol{S}}^d - \tilde{\boldsymbol{S}}^d\| \leqslant \lambda^* \|\hat{\boldsymbol{S}}^d_R - \tilde{\boldsymbol{S}}^d\|, \end{aligned}$$

where the following condition holds (due to optimality):

$$\|\boldsymbol{S}^d\| \leqslant \|P_M(\lambda^*, \hat{\boldsymbol{S}}^d_R)\|.$$

By local monotonicity,

$$\mathcal{O}(P_{\mathrm{OPF}}(\boldsymbol{S}^d)) \leqslant \mathcal{O}(P_{\mathrm{OPF}}(P_M(\lambda^*, \hat{\boldsymbol{S}}^d_R))).$$

Finally, by definition of P_M, there exists a feasible dispatch $\boldsymbol{S}^g_m$ for $P_M(\lambda^*, \hat{\boldsymbol{S}}^d_R)$ such that $\mathcal{O}(\boldsymbol{S}^g_m) \leqslant (1+\beta)\mathcal{O}^*$.

$$\begin{aligned} \mathcal{O}^* = \mathcal{O}(P_{\mathrm{OPF}}(\boldsymbol{S}^d)) &\leqslant \mathcal{O}(P_{\mathrm{OPF}}(P_M(\lambda^*, \hat{\boldsymbol{S}}^d_R))) \\ &\leqslant \mathcal{O}(\boldsymbol{S}^g_m) \\ &\leqslant (1+\beta)\mathcal{O}^* \end{aligned}$$

follows by definition of the OPF. □

Corollary *2:* When the OPF is locally-monotone around $\boldsymbol{S}^d$ for parameter β, mechanism $\mathcal{M}_{\mathrm{M}}$ solves the OLI problem.

The result above follows from Theorem 6.

Mechanism $\mathcal{M}_{\mathrm{M}}$ also provides strong accuracy guarantees.

Theorem *7:* Load vector $\hat{\boldsymbol{S}}^{\boldsymbol{d}}$ returned by mechanism $\mathcal{M}_{\mathrm{M}}$ satisfies $\|\hat{\boldsymbol{S}}^d - \boldsymbol{S}^d\|_2 \leqslant (\lambda^u + 1)\|\tilde{\boldsymbol{S}}^d - \boldsymbol{S}^d\|_2$.

Proof. At each iteration of Phases 1 and 2, Constraint (M4) implies

$$\|\hat{\boldsymbol{S}}^d - \tilde{\boldsymbol{S}}^d\|_2 \leqslant \lambda^u \|\hat{\boldsymbol{S}}^d_R - \tilde{\boldsymbol{S}}^d\|_2,$$

Since the solution $\hat{\boldsymbol{S}}^d_R$ of the relaxation mechanism P_{R} guarantees: $\|\hat{\boldsymbol{S}}^d_R - \tilde{\boldsymbol{S}}^d\|_2 \leqslant \|\tilde{\boldsymbol{S}}^d - \boldsymbol{S}^d\|_2$, it follows that

$$\begin{aligned}
&\|\hat{\boldsymbol{S}}^d - \tilde{\boldsymbol{S}}^d\|_2 \leqslant \lambda^u \|\tilde{\boldsymbol{S}}^d - \boldsymbol{S}^d\|_2 \\
\Longrightarrow\ &\|\hat{\boldsymbol{S}}^d - \tilde{\boldsymbol{S}}^d\|_2 + \|\tilde{\boldsymbol{S}}^d - \boldsymbol{S}^d\|_2 \leqslant (\lambda^u + 1)\|\tilde{\boldsymbol{S}}^d - \boldsymbol{S}^d\|_2 \\
\Longrightarrow\ &\|\hat{\boldsymbol{S}}^d - \boldsymbol{S}^d\|_2 \leqslant (\lambda^u + 1)\|\tilde{\boldsymbol{S}}^d - \boldsymbol{S}^d\|_2
\end{aligned}$$

where the last line uses the triangle inequality. □

VII. Experimental Results

This section presents an experimental analysis of the three proposed mechanisms. The results are concerned with both computational tractability and obfuscation quality. Since the relaxation mechanism typically does not return an optimal power flow, the experimental study also reports its distance compared to the optimal dispatch as returned by an AC-OPF solver. Finally, the results describe the behavior of the Min-Max mechanism, including the number of iterations to converge to an OPF solution.

The experiments are performed on a variety of NESTA benchmarks [21]. The parameter ϵ is fixed to 1.0, while the *indistinguishability level* α varies from 0.1 to 10 (in p.u.), and the *fidelity level* β varies from 10^{-3} to 10^{-1} (i.e. from 0.1% to 10% cost difference). The parameter λ in the Min-Max mechanism is initialized to 1.0 and initially grows by 5% per iteration until an optimal power flow satisfying the fidelity condition is found. The smallest λ (up to a convergence tolerance of 10^{-3}) is found subsequently via a binary search, as described in Algorithm 1. The Min-Max mechanism is limited to 3000 iterations and 48 hours of runtime. All the models are implemented using PowerModels.jl [22] in Julia with the nonlinear solver IPOPT [23].

TABLE II: Polar Laplace Mechanism $\mathcal{M}_{PLap}$: Percentage of feasible power flows for $\beta = 0.01$.

	indistinguishability (α)				indistinguishability (α)		
Benchmark	0.1	1.0	10.0	Benchmark	0.1	1.0	10.0
3_lmbd	96	42	0	39_epri	100	34	0
4_gs	78	42	0	57_ieee	0	0	0
5_pjm	100	92	0	73_ieee_rts	98	0	0
6_c	96	6	0	118_ieee	98	0	0
6_ww	50	0	0	189_edin	8	0	0
9_wscc	100	46	0	240_wecc	98	12	0
14_ieee	46	0	0	300_ieee	0	0	0
24_ieee_rts	100	0	0	1354_pegase	100	0	0
29_edin	100	100	16	1394sop_eir	0	0	0
30_as	0	0	0	1397sp_eir	0	0	0
30_fsr	6	0	0	1460wp_eir	0	0	0
30_ieee	2	0	0				

TABLE III: Convergence (%) with $\beta = 0.01$.

	Fritz John ($\mathcal{M}_{FJ}$)			Relaxation ($\mathcal{M}_R$)			Minmax ($\mathcal{M}_M$)		
Benchmark / α :	0.1	1.0	10.0	0.1	1.0	10.0	0.1	1.0	10.0
3_lmbd	74	70	32	100	100	98	100	100	100
4_gs	70	70	72	100	100	100	100	98^1	100
5_pjm	88	86	56	100	100	100	100	100	100
6_c	46	54	26	100	100	96	100	100	100
6_ww	92	68	54	100	100	96	100	100	100
9_wscc	94	64	48	100	100	100	100	98^1	100
14_ieee	50	30	34	100	96	100	100	100	100
24_ieee_rts	74	40	36	100	98	98	100	100	100
29_edin	68	58	54	100	100	100	100	100	100
30_as	44	28	48	98	100	98	100	100	100
30_fsr	66	36	60	100	100	98	100	100	100
30_ieee	46	28	44	100	98	100	100	100	100
39_epri	88	58	44	100	96	94	100	100	100
57_ieee	80	32	12	100	100	100	100	100	100
73_ieee_rts	56	30	34	100	94	96	100	100	98^0
118_ieee	84	46	20	100	98	100	100	100	100
189_edin	20	66	26	100	86	68	100	100	100
240_wecc	–	–	–	100	100	82	100	100	100
300_ieee	–	–	14	96	84	86	100	100	100
1354_pegase	–	–	–	100	68	74	100	98^0	100
1394sop_eir	–	–	–	100	82	74	100	100	100
1397sp_eir	–	–	–	96	76	88	100	100	100
1460wp_eir	–	–	–	100	44	62	100	100	100

A. Convergence Rate and Runtime

Table II shows the number (in percentage) among 50 instances that have a valid power flow solution after obfuscating loads using the Laplace mechanism $\mathcal{M}_{PLap}$. The results indicate that increasing the privacy level causes severe feasibility issues, with almost no benchmark returning any AC-feasible solution.

Recall that, even if the $\mathcal{M}_{PLap}$ obfuscation produces loads that satisfy the AC-OPF constraints, the associated OPF costs do not necessarily satisfy the fidelity requirement. In contrast, the three methods proposed in the paper, i.e., relaxation ($\mathcal{M}_R$), FJ ($\mathcal{M}_{FJ}$), and min-max ($\mathcal{M}_M$) address the fidelity requirement.

Table III compares $\mathcal{M}_R, \mathcal{M}_{FJ}$, and $\mathcal{M}_M$, showing the number of instances (in percentages) for which a feasible solution is found within the runtime limits.

The Fritz John mechanism $\mathcal{M}_{FJ}$ has significantly more convergence issues than the two other mechanisms. In particular, numerical stability issues arise on the complementary slackness conditions in many instances that fail to converge. In general, fewer instances converge when a large amount of Laplace noise, e.g., $\alpha = 10.0$, is applied in comparison to a small amount of noise, e.g., $\alpha = 0.1$. This suggests the presence of convergence issues when the starting point is far from a realistic solution.

Mechanism $\mathcal{M}_R$ exhibits significantly fewer numerical stability issues, e.g., numerical or convergence issues related to the non-linear solver. Since it is also used as the starting point of the Min-Max mechanism, constraint (M4) is replaced by $\|\hat{\boldsymbol{S}}^d - \tilde{\boldsymbol{S}}^d\| \leqslant \lambda \cdot \gamma$ where γ is a small constant and $\hat{\boldsymbol{S}}^d_R$ is replaced by $\tilde{\boldsymbol{S}}^d$ when the relaxation model did not converge.

Finally, the Min-Max mechanism $\mathcal{M}_M$ converges almost always. When it does not, the superscripts indicate the number of instances that converge to an AC-feasible solution, but reach the iteration or time limits before finding an OPF solution.

Table IV presents the runtimes (averaged over converged runs) of the three mechanisms. The Fritz John mechanism is

TABLE IV: Runtime (sec) results averaged over converged runs and $\beta = 0.01$.

	Fritz John ($\mathcal{M}_{FJ}$)			Relaxation ($\mathcal{M}_R$)			Minmax ($\mathcal{M}_M$)		
Benchmark / α :	0.1	1.0	10.0	0.1	1.0	10.0	0.1	1.0	10.0
3_lmbd	0.43	0.38	0.61	0.01	0.02	0.02	0.23	0.30	0.12
4_gs	0.35	0.61	0.70	0.01	0.01	0.01	0.79	0.73	0.23
5_pjm	0.92	0.71	1.10	0.02	0.02	0.02	0.38	1.48	0.36
6_c	1.16	1.10	1.44	0.02	0.01	0.02	0.23	0.11	0.24
6_ww	1.37	1.99	2.52	0.02	0.02	0.02	0.23	0.11	0.15
9_wscc	0.72	0.76	0.91	0.02	0.02	0.02	0.52	0.48	0.22
14_ieee	2.38	7.20	6.02	0.03	0.03	0.03	0.69	0.56	0.89
24_ieee_rts	30.69	30.35	45.59	0.06	0.20	0.99	1.38	1.52	2.33
29_edin	130.06	123.76	135.70	0.20	0.27	0.34	0.53	43.07	27.78
30_as	10.00	10.10	14.18	0.07	0.09	0.08	0.87	0.98	0.89
30_fsr	8.97	8.96	16.05	. 0.09	0.12	0.07	1.23	1.46	0.80
30_ieee	8.26	11.12	11.71	0.05	0.07	0.07	0.78	1.29	1.26
39_epri	8.46	11.76	26.60	0.08	0.12	0.10	1.09	7.40	1.16
57_ieee	58.42	128.42	146.31	0.11	0.14	0.17	1.35	4.70	9.25
73_ieee_rts	172.47	361.65	288.31	0.21	0.29	3.55	5.80	2.66	19.23
118_ieee	112.89	209.39	175.56	0.33	0.37	0.51	8.28	9.82	8.83
189_edin	535.72	440.02	295.17	0.34	0.47	0.53	2.41	55.04	51.81
240_wecc	–	–	–	1.93	2.19	2.26	9.39	111.75	138.48
300_ieee	–	–	1313.91	1.02	1.37	1.63	455.98	433.03	205.18
1354_pegase	–	–	–	4.76	17.28	14.98	13.53	1613.98	2300.29
1394sop_eir	–	–	–	6.23	6.91	7.86	70.06	811.67	804.94
1397sp_eir	–	–	–	7.10	8.73	7.31	139.26	923.28	561.29
1460wp_eir	–	–	–	7.69	8.96	8.10	194.91	1408.14	732.98

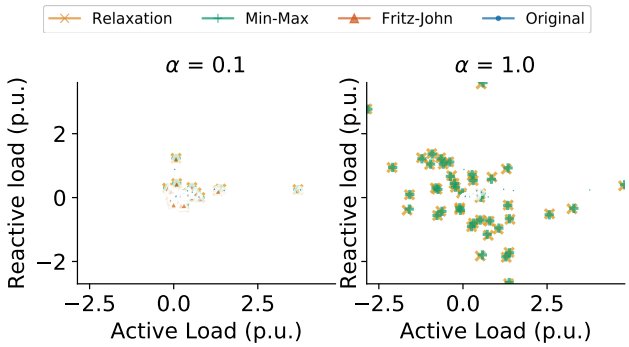


Fig. 2: Loads from the original dataset and the relaxation ($\mathcal{M}_R$), Fritz John ($\mathcal{M}_{FJ}$), and Min-Max ($\mathcal{M}_M$) mechanisms on the NESTA-57 bus system. $\alpha = 0.1$ (left) and $\alpha = 1.0$ (right), $\beta = 0.1$; x-axis/y-axis report the active/reactive components.

often several orders of magnitude slower and does not scale to large test cases. In contrast, the relaxation mechanism is extremely efficient even for large instances. Even though the runtimes for the Min-Max mechanism can be two orders of magnitude slower than the relaxation mechanism, they remain well beyond the time limits even for the largest test cases.

B. Accuracy of the Mechanisms

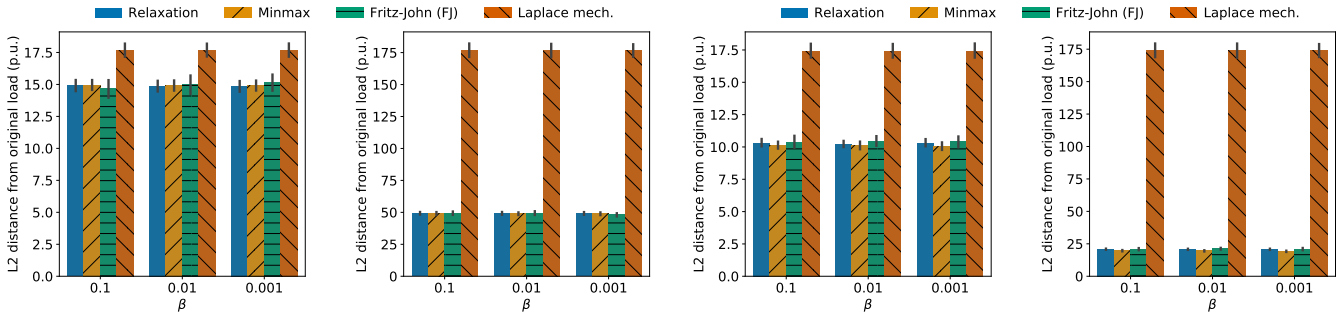
Figure 2 depicts the values of all the loads before and after the obfuscation process on the NESTA-57 bus system. After post-processing, the loads are indeed different from the original ones and the three post-processing mechanisms agree closely with each other. To compare the differences between the three mechanisms, Figures 3a and 3b provide a more detailed view and present the load distances (in L2 norm) of the Fritz John, relaxation, Min-Max, and Laplace mechanisms with respect to the original loads, averaged over 50 runs, for the NESTA-73 and the NESTA-189 test cases. *The key outcome is that the three proposed mechanisms have smaller distances to the original loads than those reported by the Laplace mechanism.* Not surprisingly, a large α (more Laplace noise) may lead to infeasible problems (as shown in Table II), and therefore, the solutions computed by these mechanisms typically have more considerable distances.

C. Quality of the Relaxation Mechanism

Figures 4a and 4b illustrate the loss in fidelity of the relaxation and the Min-Max mechanisms. The box-plots report the percentage difference between the OPF cost on the original and obfuscated loads produced by $\mathcal{M}_{\mathrm{R}}$ and $\mathcal{M}_{\mathrm{M}}$, i.e., $100 \times \frac{\mathcal{O}(P_{\mathrm{OPF}}(\hat{\boldsymbol{S}}^d)) - \mathcal{O}^*}{\mathcal{O}^*}$, on the NESTA-57 and NESTA-189 test cases. Note that the upper and lower quantiles of the boxplots in Figure 4a(left) coincide with the range of the data. The results show that the differences can reach up to 10% when β is very small. *This highlights the benefits of the Min-Max mechanism in achieving greater fidelity.* The gray bar indicates the median sample (over 50 instances) and it mostly overlaps with the outer-bar showing the minimum or maximum values. This is due to that most of the instances converge with solutions with active β-fidelity constraint (O2).

D. Analysis of the Min-Max Mechanism

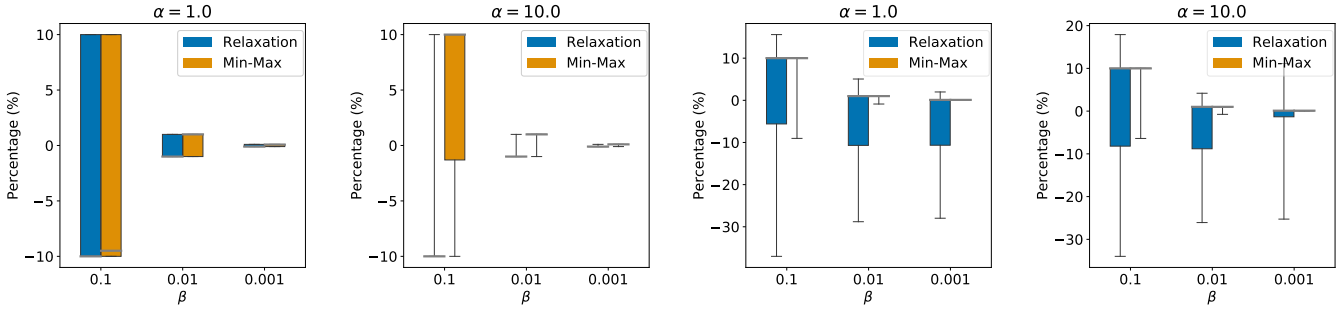
Figures 5 and 6 present two case studies, which use, respectively, the 118-bus and the 240-bus systems, and illustrate the behavior of the Min-Max mechanism. The figures describe the generation dispatch costs found by the load maximization (lines 7 and 15) in Algorithm (1) and by the OPF (lines 8 and 16) (top-left), the evolution of the λ parameter (top-right), the total active load (MW) (bottom-left), and the load distances with the respect to the $\mathcal{M}_{PLap}$ obfuscated loads and the original loads throughout the iterative process (bottom-right). The



(a) NESTA-73 bus system: $\alpha = 1.0$ (left) and $\alpha = 10.0$ (right).

(b) NESTA-189 bus system: $\alpha = 1.0$ (left) and $\alpha = 10.0$ (right).

Fig. 3: Load distance (L2 norm) between the obfuscation mechanism, $\mathcal{M}_R, \mathcal{M}_M, \mathcal{M}_{FJ}, \mathcal{M}_{PLap}$ and original dataset.



(a) NESTA-57 bus system: $\alpha = 1.0$ (left) and $\alpha = 10.0$ (right).

(b) NESTA-189 bus system: $\alpha = 1.0$ (left) and $\alpha = 10.0$ (right).

Fig. 4: Percentage difference on the AC-OPF optimal dispatch costs after obfuscation by $\mathcal{M}_R$ and $\mathcal{M}_M$.

shaded region denotes the cost range for the β-neighborhood. In case study 1, a 5% increase on λ is sufficient to find post-processed loads satisfying the mechanism conditions (one iteration of phase 1). The graphs thus show the binary search to tighten λ (phase 2): λ is decreased until the OPF on the post-processed loads is not within the bounds specified by β, before being increased slightly. Case study 2 is the exact opposite: It requires substantial increases of λ, and the mechanism spends almost all its time in phase 1, finding post-processed loads whose OPF value is accurate enough. Note that the distance with respect to the original loads does not vary much during these optimizations, highlighting that the Min-Max mechanism preserves the accuracy of the relaxation mechanism, although it enforces much stricter constraints. Observe also the monotonicity of the loads: as λ increases (resp. decreases), the OPF cost increases (resp. decreases).

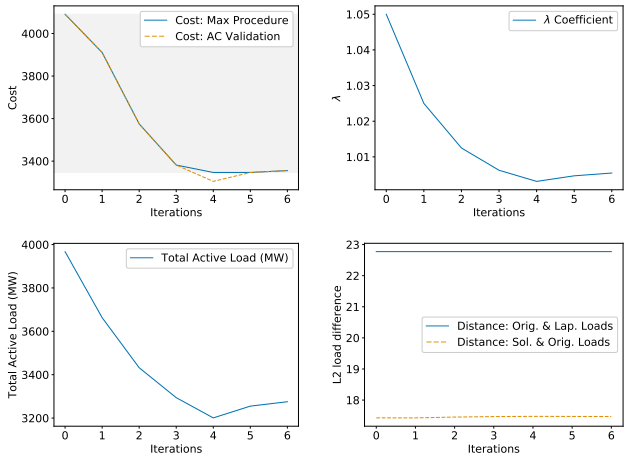


Fig. 5: (Case study 1) Dispatch costs, lambda parameter, total active load (MW), and load distance (L2) on NESTA-118. $\alpha = 1.0, \beta = 0.1$.

E. Extension: OPF Cost Obfuscation

This section illustrates the extensibility of the proposed post-processing based framework. Suppose that the optimal OPF objective costs $\mathcal{O}^*$ cannot be publicly released. To guarantee differential privacy, we are thus required to obfuscate the value $\mathcal{O}^*$. The sensitivity of $\mathcal{O}^*$ to load indistinguishability α is given by the following expression:

$$\Delta_{\mathcal{O}^*} = \max_{\boldsymbol{S}^d \sim_\alpha \boldsymbol{S}^{d'}} \left\| P_{\text{OPF}}(\boldsymbol{S}^d) - P_{\text{OPF}}(\boldsymbol{S}^{d'}) \right\|_1 .$$

When P_{OPF} is modeled by a convex program, this sensitivity can be computed exactly and, by Theorem 3, $\tilde{\mathcal{O}}^* = \mathcal{O}^* + z$ where z is drawn from Laplace distribution Lap $\left(\frac{\Delta_{\mathcal{O}^*}}{2\epsilon}\right)$. Here, half of the privacy budget ϵ is allocated to protect the value $\mathcal{O}^*$ and the other half to protect the load values.

Figures 7a and 7b illustrate the loss in fidelity of the relaxation mechanism with the obfuscated market costs $\tilde{\mathcal{O}}^*$ for 50 instances using a DC power flow model.

The two box-plots report the percentage difference between the obfuscated market costs and the OPF costs with the post-processed loads produced by $\mathcal{M}_R$ and $\mathcal{M}_M$, i.e.,

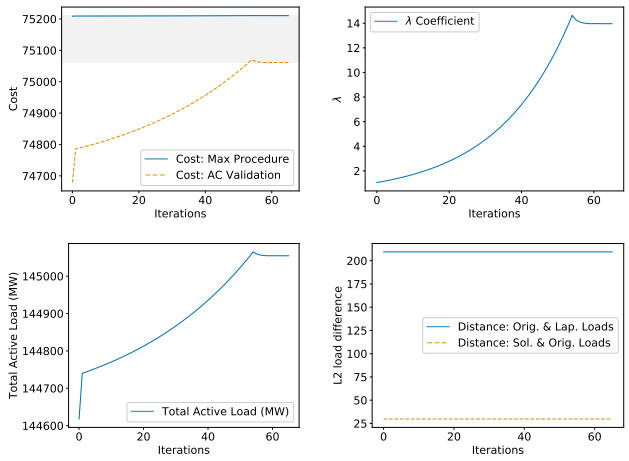


Fig. 6: (Case study 2) Dispatch costs, lambda parameter, total active load (MW), and load distance (L2) on NESTA-240. $\alpha = 1.0, \beta = 0.001$.

$100 \times \frac{\mathcal{O}(P_{\text{OPF}}(\hat{\boldsymbol{s}}^d)) - \tilde{\mathcal{O}}^*}{\tilde{\mathcal{O}}^*}$, on the NESTA-57 and NESTA-189 test cases. The results further indicate relaxation mechanism can be difficult to maintain fidelity and the differences can reach more than 60% for the extreme case. In contrast, the Min-Max mechanism $\mathcal{M}_M$ preserves fidelity even when the original costs are obfuscated.

VIII. Conclusions

This paper introduced the Optimal power flow Load Indistinguishability (OLI) problem to release optimal power flow (OPF) test cases that preserve load privacy. To solve the OLI problem, the paper proposed an ideal mechanism that leverages the post-processing immunity of DP to cast the production of a private load as a bilevel optimization problem that redistributes the noise introduced by a randomized mechanism to ensure OPF fidelity and accuracy. To meet the computational challenges of the bilevel optimization, the paper then introduced three mechanisms that respectively exploit the Fritz John conditions, a relaxation of the bilevel formulation, and a Min-Max procedure that alternates between the bilevel relaxation and a load maximization, until privacy-preserving and high-fidelity loads are found. The proposed mechanisms enjoy desirable theoretical properties. They achieve ϵ-differential privacy, ensure that the released dataset can produce feasible solutions for the OPF problem, and are a constant factor away from optimality. The mechanisms have been evaluated on the largest collection of OPF test cases available. Computational results show that the mechanisms provide orders of magnitude improvements in accuracy compared to traditional approaches (e.g., the Laplace mechanism) and preserve the salient computational features of the test cases. These results indicate that the proposed mechanisms have the potential to become an important tool to release sensitive data sets for competitions and benchmarking.

Future work will be devoted to study theoretical properties on local monotonicity of OPF problems, and extend the proposed methods to time-series data.

Acknowledgment

The authors would like to thank Kory Hedman for extensive discussions about obfuscation techniques and effective attack strategies. They are also thankful to Enpeng Yuan for pointing out several typos in the paper. Finally, the authors are grateful to the anonymous reviewers for their valuable comments. This research is partly funded by the ARPA-E Grid Data Program under Grant 1357-1530.

References

[1] F. Fioretto, T. W. K. Mak, and P. Van Hentenryck, "Privacy-preserving obfuscation of critical infrastructure networks," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, pp. 1086–1092.

[2] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC*, vol. 3876. Springer, 2006, pp. 265–284.

[3] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2013.

[4] G. Fanti, V. Pihur, and Ú. Erlingsson, "Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 3, pp. 41–61, 2016.

[5] A. Greenberg, "Apple's "differential privacy" is about collecting your data—but not your data," *Wired, June*, 2016.

[6] S. Vadhan, "The complexity of differential privacy," in *Tutorials on the Foundations of Cryptography*. Springer, 2017, pp. 347–450.

[7] G. Ács and C. Castelluccia, "I have a dream!(differentially private smart metering)." in *Information hiding*, vol. 6958. Springer, 2011, pp. 118–132.

[8] J. Zhao, T. Jung, Y. Wang, and X. Li, "Achieving differential privacy of data disclosure in the smart grid," in *INFOCOM, 2014*, 2014, pp. 504–512.

[9] X. Liao, P. Srinivasan, D. Formby, and A. R. Beyah, "Di-prida: Differentially private distributed load balancing control for the smart grid," *IEEE Transactions on Dependable and Secure Computing*, 2017.

[10] F. Zhou, J. Anderson, and S. H. Low, "Differential privacy of aggregated DC optimal power flow data," *arXiv preprint arXiv:1903.11237*, 2019.

[11] A. Karapetyan, S. K. Azman, and Z. Aung, "Assessing the privacy cost in centralized event-based demand response for microgrids," *CoRR*, vol. abs/1703.02382, 2017. [Online]. Available: http://arxiv.org/abs/1703.02382

[12] F. Fioretto, C. Lee, and P. Van Hentenryck, "Constrained-based differential privacy for private mobility," in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2018.

[13] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi, "Broadening the scope of differential privacy using metrics," in *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 2013, pp. 82–102.

[14] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *ACM symposium on Theory of computing*, 2009.

[15] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 901–914.

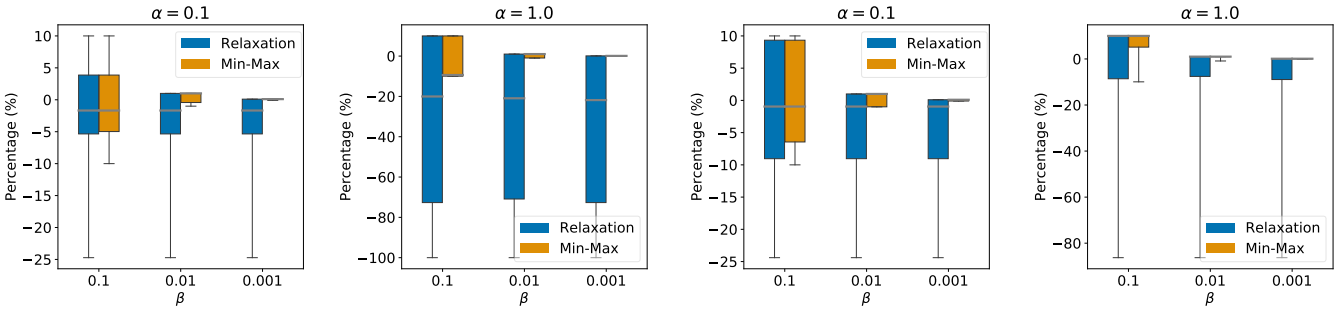
[16] J. Anderson, F. Zhou, and S. H. Low, "Disaggregation for networked power systems," in *2018 Power Systems Computation Conference (PSCC)*, June 2018, pp. 1–7.

[17] F. Koufogiannis, S. Han, and G. J. Pappas, "Optimality of the laplace mechanism in differential privacy," *arXiv preprint arXiv:1504.00065*, 2015.

[18] F. Fioretto and P. Van Hentenryck, "Constrained-based differential privacy: Releasing optimal power flow benchmarks privately," in *Proceedings of Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*, 2018, pp. 215–231.

[19] P. Hansen, B. Jaumard, and G. Savard, "New branch-and-bound rules for linear bilevel programming," *SIAM Journal on scientific and Statistical Computing*, vol. 13, no. 5, pp. 1194–1217, 1992.

[20] L. Vicente, G. Savard, and J. Júdice, "Descent approaches for quadratic bilevel programming," *Journal of Optimization Theory and Applications*, vol. 81, no. 2, pp. 379–399, 1994.



(a) NESTA-57 bus system: $\alpha = 0.1$ (left) and $\alpha = 1.0$ (right).

(b) NESTA-189 bus system: $\alpha = 0.1$ (left) and $\alpha = 1.0$ (right).

Fig. 7: Percentage difference on the AC-OPF optimal dispatch costs after obfuscation by $\mathcal{M}_R$ and $\mathcal{M}_M$.

[21] C. Coffrin, D. Gordon, and P. Scott, "Nesta, the NICTA energy system test case archive," *CoRR*, vol. abs/1411.0359, 2014. [Online]. Available: http://arxiv.org/abs/1411.0359

[22] C. Coffrin, R. Bent, K. Sundar, Y. Ng, and M. Lubin, "Powermodels.jl: An open-source framework for exploring power flow formulations," in *PSCC*, June 2018.

[23] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.