Confidentiality-Preserving Optimal Power Flow for Cloud Computing

Alex R. Borden, Daniel K. Molzahn, Parmeswaran Ramanathan, Bernard C. Lesieutre
Department of Electrical and Computer Engineering
University of Wisconsin-Madison
Madison, WI 53706, USA
borden@wisc.edu, molzahn@wisc.edu, parmesh@ece.wisc.edu, lesieutre@wisc.edu

Abstract—**In the field of power system engineering, the optimal power flow problem is essential in planning and operations. With increasing system size and complexity, the computational requirements needed to solve practical optimal power flow problems continues to grow. Increasing computational requirements make the possibility of performing these computations remotely with cloud computing appealing. However, power system structure and component values are often confidential; therefore, the problem cannot be shared. To address this issue of confidential information in cloud computing, some techniques for masking optimization problems have been developed. The work of this paper builds upon these techniques for optimization problems but is specifically developed for addressing the DC and AC optimal power flow problems. We study the application of masking a sample OPF using the IEEE 14-bus network.**

I. Introduction

The optimal power flow (OPF) problem is used to determine an optimal operating point in electric power systems. It takes a number of different forms depending on the particular objective and the scale of interest (planning vs. operations, economics, reliability, etc.). The mathematical representation varies from a linear program (DC OPF [1]) to a nonlinear, nonconvex mixed-integer program (security constrained AC OPF). Generally, all variants include an objective function (commonly quadratic or piece-wise linear), physical network constraints (the power flow equations) and imposed engineering limits (voltage magnitude, active and reactive power generation, transmission line-flow, etc.). The problem can be large with thousands of decision variables and tens of thousands of constraints. In this context, advances in the field of computing are of considerable interest.

An emerging paradigm in computer science and engineering is cloud computing [2]. Cloud computing provides subscribers shared access to powerful remote computing platforms; therefore, the potential to solve OPF problems remotely with cloud computing is an appealing possibility. The full AC OPF problem is nonlinear and nonconvex, and with realistic power system models being very large, potentially having tens of thousands of buses, the OPF problem seems a promising candidate for remotely solving in the cloud.

It is well recognized however, that security in cloud computing is a significant concern [3],[4]. With a shared computing platform comes the possible risk of attackers obtaining data sent to the cloud. In the case of power systems, this data is often confidential. Leaks of confidential data can be financially damaging and potentially threatening to national security. For this reason, cloud computing is currently not well suited for power system applications without further security advances.

This confidentiality motivates the need to improve OPF problem security masking. The masking process obscures the problem data such that an attacker with access to the masked problem cannot obtain confidential information. The masking process preserves the ability to obtain the original optimal solution. Knowledge of the masking process details are required in order to extract the original solution from the masked solution.

Existing research has investigated techniques for masking optimization problems [5],[6]. In [5], the authors outline a systematic approach for masking a general linear program. The approach in [5] seems well suited for the linear DC OPF problem; however, some additions are needed. The approach in [5] only specifies a linear objective function whereas quadratic cost functions are necessary for many practical OPF problems. Furthermore, existing literature does not discuss dual solutions to the original unmasked problem. The dual variables in the OPF problem are important to power system operations with some of them being the locational marginal prices in market contexts. The method in [5] also does not obscure the number and type of facilities present in the problem.

An additional computational concern is that the approach in [5] destroys problem sparsity, making solutions of large OPF problems computationally intractable. A masking approach that preserves sparsity in integer programs is described in [6]. This approach serves as inspiration to a similar approach for preserving sparsity in the masked OPF problems presented in this paper.

This paper presents a confidentiality preserving optimal power flow for cloud computing. We address several issues pertaining to the OPF including dual variable calculations in Section III, controlling the sparsity of a linear program for computational ease in Section IV-B, imposing quadratic cost functions in Section IV-0, obscuring the number of system facilities in Section V, and masking nonlinear constraints in Section VI. We initially focus on the linear DC OPF and its relation to the existing literature on cloud computing security and then consider the nonlinear AC OPF with an example of both in Section VII.

II. DC Optimal Power Flow Problem Overview

The DC OPF uses a power flow model that is a linear approximation of the nonlinear power flow equations. There are four main approximations made in the DC OPF: the bus voltage magnitudes are all equal to one, the voltage angle differences are small so that $\cos(\delta_k - \delta_m) \approx 1$ and $\sin(\delta_k - \delta_m) \approx \delta_k - \delta_m$, the resistance for each branch is negligible and set to zero and all shunt elements are neglected. Reactive power at the loads and generators are not explicitly considered.

The DC OPF can be written with linear constraints and quadratic cost function having the following form:

$$\min_{P_g,\,\delta} \; \frac{1}{2} P_g^T \boldsymbol{D} P_g + d^T P_g \tag{1}$$

$$s.t. \qquad -P_g + \boldsymbol{B}\delta = -P_L$$

$$\delta_{ref} = 0$$

$$P_{g,min} \le P_g \le P_{g,max}$$

$$-P_{flow,max} \le diag(b_{br})\boldsymbol{A_{inc}}\delta \le P_{flow,max}$$

In the above formulation for the DC OPF, the optimization variables are P_g as the vector of generator power injections and δ as the vector of bus voltage angles. There is a quadratic cost function, where $\boldsymbol{D}$ is a diagonal matrix of generator quadratic cost coefficients and d is a vector of generator linear cost coefficients.

The first equality constraint enforces power balance at each bus. Here the bus susceptance matrix $\boldsymbol{B}$ is the imaginary part of the bus admittance matrix with shunt elements neglected. Reflecting common power system topology, the matrix $\boldsymbol{B}$ is typically sparse. The vector P_L contains the bus active power loads. It is important to note that in the formulation of (1), the power generated is in the delivering reference frame (i.e., P_g is nonnegative), and the bus loads are in the receiving reference frame (i.e., P_L is also nonnegative). The second equality constraint enforces the bus voltage angle at the reference bus to be zero.

The first inequality constraint limits power generation for each generator to be within its lower and upper bounds. The last constraint limits the power flow in both directions on each branch to be less than a maximum flow $P_{flow,max}$. Here the vector b_{br} contains the branch susceptances and $diag(b_{br})$ is the diagonal matrix with the vector b_{br} on the diagonal. The matrix $\boldsymbol{A_{inc}}$ is the bus-to-branch incidence matrix; this matrix has number of rows equal to the number of branches and number of columns equal to the number of buses. Each row has +1 in the column corresponding to the branch's "from" bus and -1 in the column corresponding to the branch's "to" bus.

III. Masking Primal and Dual Linear Programs

Recent research details a method for masking a linear program [5]. In this section we will briefly summarize this method and further develop a method for recovering the unmasked dual variables. In Section IV we adopt and extend this method for application to the DC OPF problem.

The primal notation in this section is adopted from [5]. We start from the standard linear program formulation of the primal (2a) and dual (2b) problems.

$$\min_{x} \; c^T x \tag{2a}$$

$$s.t. \qquad \boldsymbol{M_1} x = b_1$$

$$\boldsymbol{M_2} x \le b_2$$

$$x \ge 0$$

$$\max_{u,v} \; b_1^T u + b_2^T v \tag{2b}$$

$$s.t. \qquad \boldsymbol{M_1^T} u + \boldsymbol{M_2^T} v \le c$$

$$v \le 0$$

A random positive monomial matrix $\boldsymbol{Q}$ (i.e., a matrix containing exactly one non-zero entry per row and column) and a random positive vector r are used to hide the cost vector c and the optimization variable vector x.

$$\min_{x} \; c^T \boldsymbol{Q}(\boldsymbol{Q}^{-1}x + r) \tag{3}$$

$$s.t. \qquad \boldsymbol{M_1 Q}(\boldsymbol{Q}^{-1}x + r) = b_1 + \boldsymbol{M_1 Q}\, r$$

$$\boldsymbol{M_2 Q}(\boldsymbol{Q}^{-1}x + r) \le b_2 + \boldsymbol{M_2 Q}\, r$$

$$\boldsymbol{Q}^{-1}x + r \ge r$$

Substituting the masked variable $z = \boldsymbol{Q}^{-1}x + r$ and introducing the random positive diagonal matrix $\boldsymbol{S}$ yields the following primal and dual problems:

$$\min_{z} \; c^T \boldsymbol{Q} z \tag{4a}$$

$$s.t. \qquad \boldsymbol{M_1 Q} z = b_1 + \boldsymbol{M_1 Q}\, r$$

$$\boldsymbol{M_2 Q} z \le b_2 + \boldsymbol{M_2 Q}\, r$$

$$\boldsymbol{S} z \ge \boldsymbol{S} r$$

$$\max_{u,v} \; (b_1 + \boldsymbol{M_1 Q} r)^T u + (b_2 + \boldsymbol{M_2 Q} r)^T v \tag{4b}$$

$$s.t. \qquad (\boldsymbol{M_1 Q})^T u + (\boldsymbol{M_2 Q})^T v \le (c^T Q)^T$$

$$v \le 0$$

The inequality constraints in (4a) are converted to equality constraints through the introduction of slack variables z_{sl}. Denote the optimization variable vector z' as the prior vector z augmented with the slack variables, $z'^T = [z^T \; z_{sl}^T]$. The cost function vector is augmented with zero entries corresponding to the slack variables, $c'^T = [c^T \boldsymbol{Q} \;\; 0 \ldots 0]$. The dual variable vectors u and v are consolidated into a single variable vector $u'^T = [u^T \; v^T]$.

The constraint notation is simplified by defining

$$\boldsymbol{M}' = \begin{pmatrix} \boldsymbol{M_1 Q} & \boldsymbol{0} \\ \boldsymbol{M_2 Q} & \\ -\boldsymbol{S} & \boldsymbol{A} \end{pmatrix} \ , \ b' = \begin{pmatrix} b_1 + \boldsymbol{M_1 Q} r \\ b_2 + \boldsymbol{M_2 Q} r \\ -\boldsymbol{S} r \end{pmatrix}$$

where $\boldsymbol{A}$ is a random positive monomial matrix. The formulations in (4a) and (4b) can be rewritten as seen in (5a) and (5b).

$$\min_{z'} \ c'^T z' \tag{5a}$$
$$s.t. \qquad \boldsymbol{M}' z' = b'$$
$$z' \geq 0$$

$$\max_{u'} \ b'^T u' \tag{5b}$$
$$s.t. \qquad \boldsymbol{M'}^T u' \leq c'$$

Lastly the matrix $\boldsymbol{M'}$ and vector b' are hidden using a nonsingular matrix $\boldsymbol{P}$ and a random positive monomial matrix $\boldsymbol{T}$ with $\boldsymbol{M}'' = \boldsymbol{P M' T}$, $b'' = \boldsymbol{P} b'$ and $c''^T = c'^T \boldsymbol{T}$. The $\boldsymbol{P}$ matrix takes linear combinations of the rows in the constraint equations. The $\boldsymbol{T}$ matrix scales and permutes the columns of the constraint matrix $\boldsymbol{M'}$ and cost function vector c'^T. The new primal optimization variable vector is $z'' = \boldsymbol{T}^{-1} z'$ and new dual optimization variable vector is $u'' = (\boldsymbol{P}^T)^{-1} u'$. The linear program is now in its final masked primal (6a) and dual (6b) forms.

$$\min_{z''} \ c''^T z'' \tag{6a}$$
$$s.t. \qquad \boldsymbol{M}'' z'' = b''$$
$$z'' \geq 0$$

$$\max_{u''} \ b''^T u'' \tag{6b}$$
$$s.t. \qquad \boldsymbol{M}''^T u'' \leq c''$$

The original optimal primal variable vector x^* can be recovered after solving masked problem (6a) with $\boldsymbol{T} z''^* = z'^* = \left[z^{*T} \ z_{sl}^{*\,T} \right]^T$ and $x^* = \boldsymbol{Q}(z^* - r)$. The original optimal dual variable vectors u^* and v^* can be solved by $\boldsymbol{P}^T u''^* = u'^* = \left[u^{*T} \ v^{*T} \right]^T$.

IV. Masking a DC OPF Problem

In this section we specifically apply the masking techniques developed in Section III to the DC OPF problem outlined in Section II. We first specify the composition of the matrices in (2a) and (2b). We detail a method for constructing the $\boldsymbol{P}$ matrix used in (6a) and (6b). We then extend the masking to include a quadratic cost function which is required for typical OPF problems.

A. *Problem Setup*

The DC OPF problem in (1) must be formulated in terms of a standard linear program as in (2a) and (2b). Note that P_g and δ in (1) are free variables; however, the standard linear program formulation in (2a) requires nonnegative variables. Therefore P_g and δ are represented as the difference of two nonnegative variables, $P_g = P_g^+ - P_g^-$ and $\delta = \delta^+ - \delta^-$ where $P_g^+, P_g^-, \delta^+, \delta^- \geq 0$. In (2a), the optimization variable vector $x^T = \left[P_g^{+T} \quad P_g^{-T} \quad \delta^{+T} \quad \delta^{-T} \right]$. There are $2n_g + 2n_b$ elements in x, where n_g is the number of generators and n_b is the number of buses.

We temporarily neglect the quadratic terms of the cost function in (1); the quadratic cost terms will be revisited in Section IV-0. In (2a), the linear cost coefficient vector $c^T = [d^T \quad -d^T \quad 0 \dots 0]$. Here d is the linear generator cost coefficients in (1), and there are $2n_b$ number of zeros.

In the equality constraints of (2a), the matrix

$$\boldsymbol{M_1} = \begin{bmatrix} -\boldsymbol{E_g} & \boldsymbol{E_g} & \boldsymbol{B} & -\boldsymbol{B} \\ 0 \ \dots \ 0 & & e_{ref} & -e_{ref} \end{bmatrix}$$

has $n_b + 1$ rows. The $n_b \times n_g$ matrix $\boldsymbol{E_g}$ has a single +1 entry in each column for the rows corresponding to buses with generators and has zeros elsewhere. The $n_b \times n_b$ matrix $\boldsymbol{B}$ is the bus susceptance matrix with shunt elements neglected. The row vector e_{ref} has $n_b - 1$ zeros and a single +1 in the column corresponding to the reference bus angle. In the equality constraints of (2a), the column vector $b_1 = [-P_L^T \quad 0]^T$. Here P_L is the vector of bus loads; therefore, the first n_b rows of equality constraints enforce power balance at each bus. The final row of $\boldsymbol{M_1}$ and the zero in b_1 enforce the reference bus angle, $\delta_{ref} = \delta_{ref}^+ - \delta_{ref}^- = 0$.

In the inequality constraints of (2a), the matrix

$$\boldsymbol{M_2} = \begin{bmatrix} \boldsymbol{I_g} & -\boldsymbol{I_g} & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{I_g} & \boldsymbol{I_g} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & diag(b_{br})\boldsymbol{A_{inc}} & -diag(b_{br})\boldsymbol{A_{inc}} \\ \boldsymbol{0} & \boldsymbol{0} & -diag(b_{br})\boldsymbol{A_{inc}} & diag(b_{br})\boldsymbol{A_{inc}} \end{bmatrix}$$

has $2n_g + 2n_{br}$ rows where n_{br} is the number of branches. Here I_g is the $n_g \times n_g$ identity matrix. As described in Section II, the vector b_{br} contains the branch susceptances, and $diag(b_{br})$ is the diagonal matrix with the vector b_{br} on the diagonal. The matrix $\boldsymbol{A_{inc}}$ is the bus-to-branch incidence matrix, and $diag(b_{br})\boldsymbol{A_{inc}}$ is an $n_{br} \times n_b$ matrix. In the inequality constraints of (2a), the column vector $b_2 = \left[P_{g,max}^T \quad -P_{g,min}^T \quad P_{flow,max}^T \quad P_{flow,max}^T \ \right]^T$. Vectors $P_{g,max}$ and $P_{g,min}$ are the upper and lower power generation limits, and $P_{flow,max}$ is the branch power flow limits.

The dual optimization variables in (2b) are important in OPF problems. The optimization variable vector u corresponds to the equality constraints of the primal problem, and v corresponds to the inequality constraints. The locational marginal prices (LMPs) are the Lagrange multipliers of the power balance equality constraints; therefore, the first n_b Lagrange multipliers in u are the LMPs. The final Lagrange multiplier in u corresponds to the reference bus angle equality constraint. In Section III, the method for recovering the dual variables was shown.

B. *Constructing a $\boldsymbol{P}$ Matrix for a DC OPF*

The masking steps of equations (2a) through (5a) are straightforward as the construction of matrices $\boldsymbol{Q}, \boldsymbol{A}, \boldsymbol{S}$ and $\boldsymbol{T}$ and vector r only require generation of random positive numbers and permutations. Sufficiently large random numbers will mask the magnitudes of all values. However, creation of the $\boldsymbol{P}$ matrix used in (6a) is not as straightforward in the case of the DC OPF problem.

The construction of matrix $\boldsymbol{M'}$ in (5a) was shown in Section III. To clearly illustrate the typical sparse structure of $\boldsymbol{M'}$ in a DC OPF problem, let matrices $\boldsymbol{Q}$ and $\boldsymbol{A}$ used in the construction of $\boldsymbol{M'}$ be random positive diagonal matrices rather than random positive monomial matrices. A spy plot displaying the nonzero elements of the $\boldsymbol{M'}$ matrix for the IEEE 14-bus test system [7] is shown below in Figure 1.

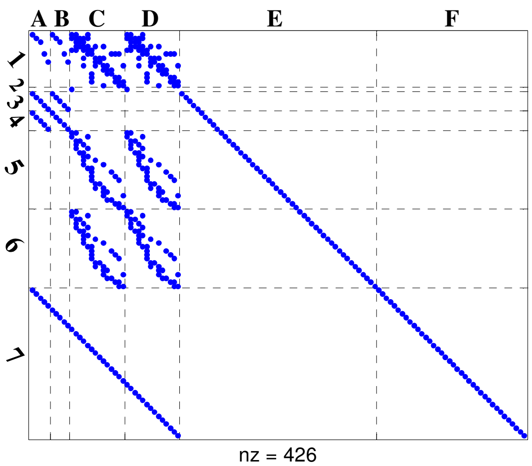

Figure 1: Nonzero elements of $\boldsymbol{M'}$ in 14-bus system

The typical structure of $\boldsymbol{M'}$ for the DC OPF can be seen in Figure 1 with obvious distinguishable sections; grid lines and labels were inserted to emphasize these sections. Given Figure 1, an attacker could only recover the original system's numerical values by knowing matrices $\boldsymbol{Q}, \boldsymbol{A}$ and $\boldsymbol{S}$ and vector r. An attacker could, however, easily identify the topology of the system by viewing the matrix in Figure 1.

The final step of the masking process requires the left multiplication of $\boldsymbol{M'}$ by a nonsingular matrix $\boldsymbol{P}$ and right multiplication of $\boldsymbol{M'}$ by a random positive monomial matrix $\boldsymbol{T}$, specifically $\boldsymbol{M''} = \boldsymbol{PM'T}$ in (6a). This left multiplication by $\boldsymbol{P}$ results in linear row combinations of $\boldsymbol{M'}$. Using Figure 1 as example, it can be seen that the columns in sections A-D are much more dense than the columns in E and F. Naively taking random row combinations of $\boldsymbol{M'}$ would increase the density of the columns in A-D much more than columns in E and F. By counting the number of denser columns, an attacker could determine the number of generators plus buses, $n_g + n_b$. Having the number of generators plus buses, an attacker knowing the DC OPF formulation could calculate the number of system branches n_{br}. The nonsingular matrix $\boldsymbol{P}$ must carefully be constructed to prevent against this.

The structure of $\boldsymbol{M'}$ can be exploited to perform custom tailored linear row operation tricks for the DC OPF that more evenly distribute column densities. The details of the DC OPF row operation tricks are left out of this report, but many different linear row operations can be performed using a preconditioning matrix $\boldsymbol{P_1}$ to be multiplied by $\boldsymbol{M'}$. There are numerous possible $\boldsymbol{P_1}$ matrices, and all are valid as long as $\boldsymbol{P_1}$ is nonsingular. One possible $\boldsymbol{P_1}$ matrix was generated to get $\boldsymbol{P_1M'}$ in Figure 2 .

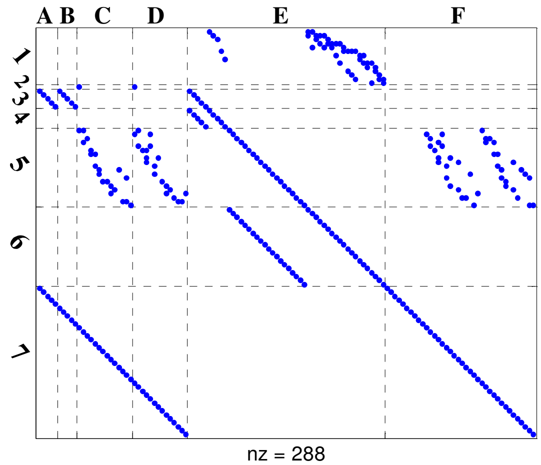

Figure 2: Nonzero elements of $\boldsymbol{P_1M'}$ in 14-bus system

The column density of $\boldsymbol{P_1M'}$ in Figure 2 has been more evenly distributed compared to $\boldsymbol{M'}$ in Figure 1, and the number of nonzero elements has also been reduced from 426 to 288. This new matrix is better suited for performing row combinations that will obscure sensitive information like the number of generators plus buses.

To further obscure system topology while maintaining sparsity, we define a second matrix $\boldsymbol{P_2}$ to be left multiplied by $\boldsymbol{P_1M'}$ and a density parameter $d_{\%}$ between zero and one which controls the tradeoff between sparsity and security. The matrix $\boldsymbol{P_2}$ is constructed in a way that enforces every row and column density percentage of $\boldsymbol{P_2P_1M'}$ to be greater than or equal to the parameter value $d_{\%}$ defined by the user. In Figure 3, each row and column density percentage is greater than or equal to $d_{\%} \approx 0.05$ for this example. An attacker viewing the matrix in Figure 3 could not recover any system topology information now by simply observing column or row density.

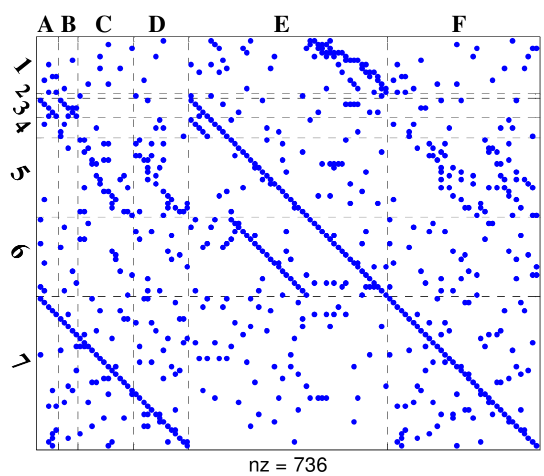

Figure 3: Nonzero elements of $\boldsymbol{P_2P_1M'}$ in 14-bus system

Recall matrices $\boldsymbol{Q}$ and $\boldsymbol{A}$ used in the construction of $\boldsymbol{M'}$ here were diagonal matrices rather than monomial matrices.Therefore the underlying diagonal matrix spanning sections 3E through 7F would be permuted and not so obvious. The diagonal matrix $\boldsymbol{S}$ however, in sections 7A-D, would evidently still be present. The size of the diagonal matrix $\boldsymbol{S}$ reveals the number of buses plus generators, which can consequently reveal the number of branches as well to a knowledgeable attacker. One final step of scaling and permuting the rows and columns of the matrix in Figure 3 is therefore performed to completely obscure the matrix structure. A matrix $\boldsymbol{P_3}$ is created for scaling and permuting rows, and a matrix $\boldsymbol{T}$ is created for scaling and permuting columns. Altogether, the final obscured matrix is $\boldsymbol{M''} = \boldsymbol{P_3 P_2 P_1 M' T} = \boldsymbol{PM'T}$, as seen in Figure 4.

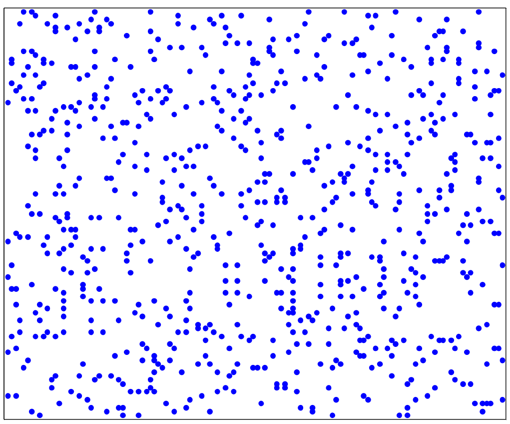

Figure 4: Nonzero elements of $\boldsymbol{M''} = \boldsymbol{PM'T}$ in 14-bus system

The matrix $\boldsymbol{M''} = \boldsymbol{PM'T}$ in Figure 4 is the constraint matrix in (6a) and (6b). The three step process of creating the matrix $\boldsymbol{P}$ outlined above hides the original structure and values of $\boldsymbol{M'}$ while maintaining sparsity. No columns or rows have too great or too low of a density which prevents against attacks of that nature. According to [5], the numerical values of the masked problem are sufficiently secure. The transformation from $\boldsymbol{M'}$ to $\boldsymbol{M''}$ may not completely hide system structure. A sophisticated attacker may be able to extract topological information knowing that the masked problem was derived from a DC OPF. Further work investigating structural security is ongoing.

C. Quadratic Cost Function

In this section we outline the details of masking a quadratic cost function which is typically required in OPF problems. In (1), a quadratic cost function was shown:

$$\min_{P_g,\,\delta} \ \tfrac{1}{2} P_g^T \boldsymbol{D} P_g + d^T P_g$$

Rewriting this cost function in terms of an optimization variable vector $x^T = \begin{bmatrix} P_g^{+T} & P_g^{-T} & \delta^{+T} & \delta^{-T} \end{bmatrix}$, as was done in Section IV-A, changes the primal and dual problems to those in (7a) and (7b).

$$\begin{aligned} \min_{x} \quad & \tfrac{1}{2} x^T \boldsymbol{C} x + c^T x \\ s.t. \quad & \boldsymbol{M_1} x = b_1 \\ & \boldsymbol{M_2} x \le b_2 \\ & x \ge 0 \end{aligned} \tag{7a}$$

$$\begin{aligned} \max_{u,v,w} \quad & -\tfrac{1}{2} \begin{bmatrix} u \\ v \\ w \end{bmatrix}^T \begin{bmatrix} M_1 C^{-1} M_1^T & M_1 C^{-1} M_2^T & M_1 C^{-1} \\ M_2 C^{-1} M_1^T & M_2 C^{-1} M_2^T & M_2 C^{-1} \\ C^{-1} M_1^T & C^{-1} M_2^T & C^{-1} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \\ & + \begin{bmatrix} b_1 + \boldsymbol{M_1 C^{-1}} c \\ b_2 + \boldsymbol{M_2 C^{-1}} c \\ \boldsymbol{C^{-1}} c \end{bmatrix}^T \begin{bmatrix} u \\ v \\ w \end{bmatrix} - \tfrac{1}{2} c^T \boldsymbol{C^{-1}} c \\ s.t. \quad & v \le 0 \\ & w \ge 0 \end{aligned} \tag{7b}$$

Like before, $c^T = [d^T \ -d^T \ 0 \ldots 0]$, but there is now also the $(2n_g + 2n_b) \times (2n_g + 2n_b)$ quadratic cost matrix $\boldsymbol{C}$.

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{D} & -\boldsymbol{D} & \boldsymbol{0} \\ -\boldsymbol{D} & \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$

Following the masking procedure outlined in Section III (3) through (6a) and (6b) gives the final masked primal (8a) and dual (8b) problem for a quadratic cost function.

$$\begin{aligned} \min_{z''} \quad & \tfrac{1}{2} z''^T \boldsymbol{C''} z'' + c''^T z \\ s.t. \quad & \boldsymbol{M''} z'' = b'' \\ & z'' \ge 0 \end{aligned} \tag{8a}$$

$$\begin{aligned} \max_{u'',v''} \quad & -\tfrac{1}{2} \begin{bmatrix} u'' \\ v'' \end{bmatrix}^T \begin{bmatrix} \boldsymbol{M'' C''^{-1} M''^T} & \boldsymbol{M'' C''^{-1}} \\ \boldsymbol{C''^{-1} M''^T} & \boldsymbol{C''^{-1}} \end{bmatrix} \begin{bmatrix} u'' \\ v'' \end{bmatrix} \\ & + \begin{bmatrix} b'' + \boldsymbol{M'' C''^{-1}} c'' \\ \boldsymbol{C''^{-1}} c'' \end{bmatrix}^T \begin{bmatrix} u'' \\ v'' \end{bmatrix} - \tfrac{1}{2} c''^T \boldsymbol{C''^{-1}} c'' \\ s.t. \quad & v'' \ge 0 \end{aligned} \tag{8b}$$

Most of the variables above were already defined in Section III with a few important distinctions. In the linear cost function of Section III, $c'^T = [c^T \boldsymbol{Q} \quad 0 \ldots 0]$ with zero entries corresponding to the slack variables; however, now due to the quadratic cost function this changes to $c'^T = [(c^T \boldsymbol{Q} - r^T \boldsymbol{Q^T C^T Q}) \quad 0 \ldots 0]$ and $c''^T = c'^T \boldsymbol{T}$. There is also the new quadratic cost matrix

$$\boldsymbol{C'} = \begin{bmatrix} \boldsymbol{Q^T C Q} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$

with zero entries corresponding to the slack variables, and $\boldsymbol{C''} = \boldsymbol{T^T C' T}$. As before the original optimal primal solution to (7a) can be obtained by $\boldsymbol{T} z''^* = z'^* = \begin{bmatrix} z^{*T} & z_{sl}^{*\,T} \end{bmatrix}^T$ and $x^* = \boldsymbol{Q}(z^* - r)$. The original optimal dual solution to (7b) can be obtained by $\boldsymbol{P^T} u''^* = u'^* = \begin{bmatrix} u^{*T} & v^{*T} & w^{*T} \end{bmatrix}^T$.

V. FURTHER OBSCURING

In this section we explore further methods of masking an OPF. Further obscuring methods may be necessary in order to hide the number and type of facilities in the system.

A. Adding Constraints to the Cost Function

Even in the fully obscured problem, there remains sensitive information that could be extracted. By counting the number of nonzero entries in the masked linear cost coefficient c'' (or the masked quadratic cost coefficient $\boldsymbol{C}''$), an attacker could determine the number of generators present in the system. If the zero entries in c'' and $\boldsymbol{C}''$ were filled in with nonzero entries then that particular attack could be prevented. This can be accomplished by adding the linear constraints to the cost function, or by adding the linear constraints squared in the case of a quadratic cost function. Take for example a problem with quadratic cost function and linear constraints as seen in (9).

$$\begin{aligned} \min_x \quad & \tfrac{1}{2}x^T\boldsymbol{C}x + c^T x \qquad (9) \\ s.t. \quad & \boldsymbol{M}x = b \\ & x \geq 0 \end{aligned}$$

Denote the i^{th} constraint as $m_i x = b_i$, and rearrange it to get $m_i x - b_i = 0$. The constraint squared is the following:

$$x^T m_i^T m_i x - 2b_i m_i x + b_i^2 = 0$$

Since the above equals zero, it can be added to the objective function without changing the optimal solution.

$$\begin{aligned} \max_x \quad & \tfrac{1}{2}x^T(\boldsymbol{C}x + m_i^T m_i)x + (c^T - 2b_i m_i)x + b_i^2 \qquad (10) \\ s.t. \quad & \boldsymbol{M}x = b \\ & x \geq 0 \end{aligned}$$

The optimal solutions to problems (9) and (10) are equivalent. However the optimization problem of (10) has a squared constraint added to the objective function which produces more nonzero entries in the objective function. This approach can be used to prevent against attacks that count the number of nonzero entries in the objective function. This approach would protect against counting the number of generators present in an OPF problem.

B. Fictitious Buses, Generators and Loads

The approach in Section V-A masks the number of generators in the cost function, but the total number of rows and columns in $\boldsymbol{M}''$ contains information on the number of system facilities. To be exact the number of rows in $\boldsymbol{M}''$ equals $3n_b + 4n_g + 2n_{br} + 1$ and the number of columns equals $4n_b + 6n_g + 2n_{br}$. An attacker with knowledge of just one of the variables, n_b, n_g or n_{br} could then calculate the two other unknown variables.

To obscure the number of rows and columns fictitious buses, generators and loads can be created. A fictitious bus can be created by splitting an existing line. This way of adding fictitious buses does not alter the solution of the OPF, but naively adding fictitious generators can alter the solution. However adding a fictitious generator with a very large cost would not be dispatched and therefore does not affect the solution of the OPF. Alternatively an offsetting fictitious load and generator pair at the same bus with equivalent upper and lower generation limits will also not affect the solution of the OPF.

VI. MASKING A NONLINEAR CONSTRAINTS

Previous work that focused on masking linear programs [5], can be extended to masking nonlinear constraints such as the one shown below in (11) with quadratic cost function.

$$\begin{aligned} \min_x \quad & \tfrac{1}{2}x^T\boldsymbol{C}x + c^T x \qquad (11) \\ s.t. \quad & f_{eq}(x) = 0 \\ & f_{ineq}(x) \leq 0 \\ & x \geq 0 \end{aligned}$$

Substitute the masked variable $z = \boldsymbol{Q}^{-1}x + r$, and neglect the constant term created in the cost function.

$$\begin{aligned} \min_z \quad & \tfrac{1}{2}z^T\boldsymbol{Q^T C Q}z + (c^T\boldsymbol{Q} - r^T\boldsymbol{Q^T C^T Q})z \qquad (12) \\ s.t. \quad & f_{eq}\big(\boldsymbol{Q}(z - r)\big) = 0 \\ & f_{ineq}(\boldsymbol{Q}(z - r)) \leq 0 \\ & z \geq r \end{aligned}$$

The nonlinear inequality constraints in (12) are converted to equality constraints through the introduction of slack variables z_{sl}. Denote the optimization variable vector z' as the prior vector z augmented with the slack variables, $z'^T = [z^T \; z_{sl}^T]$. Define the linear cost coefficient vector $c'^T = [(c^T\boldsymbol{Q} - r^T\boldsymbol{Q^T C^T Q}) \quad 0 \ldots 0]$ with zero entries corresponding to the slack variables. Define the matrix

$$\boldsymbol{C}' = \begin{bmatrix} \boldsymbol{Q^T C Q} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$

with zero entries corresponding to the slack variables. The nonlinear equality constraint notation is defined as

$$f'_{eq}(z') = \begin{bmatrix} f_{eq}\big(\boldsymbol{Q}(z - r)\big) \\ \begin{bmatrix} f_{ineq}\big(\boldsymbol{Q}(z - r)\big) \\ -\boldsymbol{S}z - \boldsymbol{S}r \end{bmatrix} + \boldsymbol{A}z_{sl} \end{bmatrix} = 0$$

Here $\boldsymbol{A}$ is a random positive monomial matrix and $\boldsymbol{S}$ is a random positive diagonal matrix. The formulation in (12) can then be rewritten as (13).

$$\begin{aligned} \min_{z'} \quad & \tfrac{1}{2}z'^T\boldsymbol{C}'z' + c'^T z' \qquad (13) \\ s.t. \quad & f'_{eq}(z') = 0 \\ & z' \geq 0 \end{aligned}$$

A random positive monomial matrix $\boldsymbol{T}$ scales and permutes the optimization variables with $z'' = \boldsymbol{T}^{-1}z'$. The objective function is modified as $c''^T = c'^T\boldsymbol{T}$ and $\boldsymbol{C}'' = \boldsymbol{T^T}\boldsymbol{C}'\boldsymbol{T}$. A nonsingular matrix $\boldsymbol{P}$ creates linear row combinations of the nonlinear constraints in $f'_{eq}(z')$. The nonlinear constraints are rewritten as

$$f'_{eq}(\boldsymbol{T}\boldsymbol{T}^{-1}z') = f'_{eq}(\boldsymbol{T}z'')$$

$$\boldsymbol{P} \cdot f'_{eq}(\boldsymbol{T}z'') = f''_{eq}(z'')$$

The final masked primal nonlinear program is (14).

$$\min_{z''} \frac{1}{2} z''^T \boldsymbol{C}'' z'' + c''^T z'' \tag{14}$$

$$s.t. \qquad f''_{eq}(z'') = 0$$

$$z'' \geq 0$$

As before the original optimal primal solution to (11) can be obtained by $\boldsymbol{T}z''^* = z'^* = \left[z^{*T}\ z_{sl}^{*\,T}\right]^T$ and $x^* = \boldsymbol{Q}(z^* - r)$. The original optimal dual solution can be obtained by $\boldsymbol{P}^{\boldsymbol{T}} u''^* = u'^* = \left[u^{*T} \quad v^{*T} \quad w^{*T}\right]^T$.

VII. Numeric Example

The IEEE 14-bus network [7] is used for the example in the following section. It is presented as both the DC OPF and the nonlinear AC OPF to show the successful recovery of the original optimal solution from both masked problems.

The IEEE 14-bus network has already partially been used as a DC OPF example in Section IV-B, Figures 1-4. It will be examined more closely here. The 14-bus network has 5 generators and 20 branches. The standard IEEE 14-bus network does not have binding branch flow limits, therefore some branch flow limits were tightened to enforce binding branch flow constraints in this example.

A quadratic cost function is assumed in the DC OPF having the formulation shown in (1). The masking process requires the generation of random positive monomial matrices $\boldsymbol{Q}$ and $\boldsymbol{A}$, random positive diagonal matrix $\boldsymbol{S}$ and random positive vector r. The original primal variable x is substituted with $x = \boldsymbol{Q}(z - r)$. All inequality constraints are converted to equality constraints via introduction of slack variables creating constraint matrix $\boldsymbol{M'}$ as in (5a).

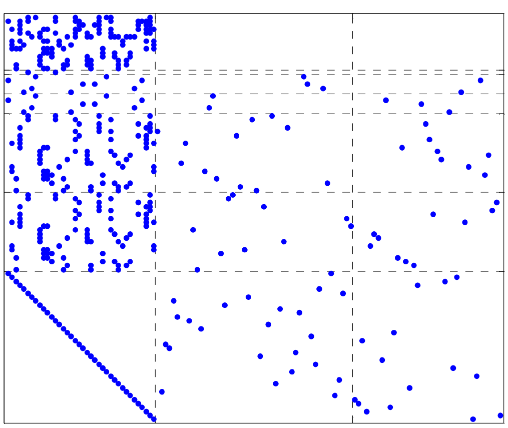

Figure 5: Nonzero elements of $\boldsymbol{M'}$ in 14-bus system

The sparsity pattern of matrix $\boldsymbol{M'}$ for the 14-bus network was shown in Figure 1, but there the matrices $\boldsymbol{Q}$ and $\boldsymbol{A}$ used in construction of $\boldsymbol{M'}$ were diagonal rather than monomial. Properly masking the OPF requires $\boldsymbol{Q}$ and $\boldsymbol{A}$ to be monomial matrices, which produces a matrix $\boldsymbol{M'}$ as seen in Figure 5. The sparsity pattern of $\boldsymbol{M'}$ in Figure 5 is more typical of those seen in (5a) compared to Figure 1. A nonsingular matrix $\boldsymbol{P}$ is carefully constructed using the steps outlined in Section IV-B, and a random positive monomial matrix $\boldsymbol{T}$ is generated. The constraint matrix $\boldsymbol{M''} = \boldsymbol{PM'T}$ is shown in the spy plot of Figure 6. The masked primal DC OPF is in the form of (8a).

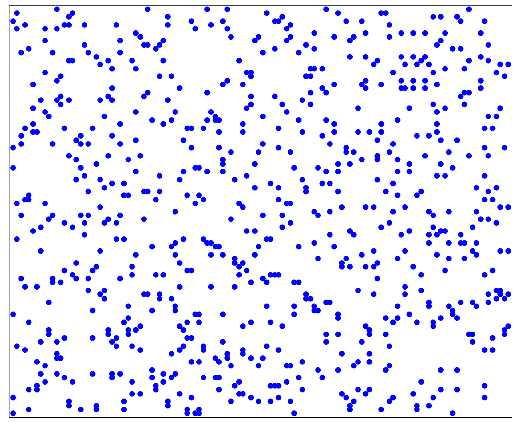

Figure 6: Nonzero elements of $\boldsymbol{M''} = \boldsymbol{PM'T}$ in 14-bus system

Further obscuring is performed by adding squared constraints to the cost function as outlined in Section V-A. The fully masked DC OPF is passed to a quadratic program solver such as *quadprog* in MATLAB, and the masked primal and dual variables are calculated. The original optimal primal variables can be solved with $\boldsymbol{T}z''^* = z'^* = \left[z^{*T}\ z_{sl}^{*\,T}\right]^T$ and $x^* = \boldsymbol{Q}(z^* - r)$. The original optimal objective value $f^* = \frac{1}{2} x^{*T} \boldsymbol{C} x^* + c^T x^*$, as in (7a). The primal variable vector $x^T = \left[P_g^{+T} \quad P_g^{-T} \quad \delta^{+T} \quad \delta^{-T}\right]$, so the final optimization variables are solved with $P_g = P_g^+ - P_g^-$ and $\delta = \delta^+ - \delta^-$.

The original optimal dual variables can be solved with $\boldsymbol{P}^{\boldsymbol{T}} u''^* = u'^* = \left[u^{*T} \quad v^{*T} \quad w^{*T}\right]^T$. In this 14-bus network, $n_b = 14$. The LMPs are the first n_b Lagrange multipliers in u, with the last Lagrange multiplier being associated with the reference bus angle constraint.

The DC OPF solution is shown in the table below. The recovered solution from the masked problem matches the solution from the unmasked problem as well as the optimal solution given by the DC OPF solver in MATPOWER [8].

$$f^* = \$9512.54$$

Bus #	Power Generated P_g	Bus Angles δ	LMPs
1	32.29 MW	0°	\$22.78 / MW
2	35.99 MW	-0.636°	\$38.00 / MW
3	100.00 MW	-1.241°	\$81.17 / MW
4		-2.331°	\$118.48 / MW
5		-1.728°	-\$34.58 / MW
6	65.72 MW	-0.635°	\$41.31 / MW
7		-2.331°	\$46.57 / MW

8	25.00 MW	-0.192°	$40.50 / MW
9		-3.907°	$133.83 / MW
10		-3.754°	$117.39 / MW
11		-2.418°	$80.02 / MW
12		-1.942°	$48.62 / MW
13		-2.264°	$54.34 / MW
14		-4.488°	$99.07 / MW

Table 1: DC OPF Optimal P_g, δ and LMPs

From Bus	To Bus	P_f	$\|P_{max}\|$	μ
5	4	25 MW	25 MW	$195.07 / MW
8	7	25 MW	25 MW	$6.07 / MW
7	9	25 MW	25 MW	$125.94 / MW

Table 2: DC OPF Binding Branch Flow Constraints

The nonlinear AC OPF for the 14-bus network can also be masked and solved for by following the steps in Section VI. The AC OPF formulation can be seen in [8]. The results of the full AC OPF are listed below, and the solution from the masked problem matches the solution from the unmasked problem as well as the solution given by the AC OPF solver in MATPOWER.

$$f^* = \$9497.73$$

Bus #	Power Generated P_g	Bus Angles δ	LMPs
1	37.57 MW	0°	$23.23 / MW
2	34.70 MW	-0.638°	$37.35 / MW
3	100.00 MW	-1.636°	$75.37 / MW
4		-2.358°	$110.83 / MW
5		-1.804°	-$27.64 / MW
6	64.60 MW	-0.993°	$41.29 / MW
7		-2.462°	$41.90 / MW
8	23.98 MW	-0.087°	$40.48 / MW
9		-3.987°	$136.74 / MW
10		-3.754°	$120.51 / MW
11		-2.511°	$81.70 / MW
12		-2.057°	$49.34 / MW
13		-2.285°	$56.43 / MW
14		-4.315°	$104.16 / MW

Table 3: AC OPF Optimal P_g, δ and LMPs

From Bus	To Bus	S_f	$\|S_{max}\|$	μ
5	4	25 MVA	25 MVA	$175.74 / MVA
7	8	25 MVA	25 MVA	$2.32 / MVA
7	9	25 MVA	25 MVA	$132.50 / MVA

Table 4: AC OPF Binding Branch Flow Constraints

VIII. Conclusion and Future Work

The optimal power flow (OPF) problem is a central problem in power systems optimization. The need to regularly solve this problem for large scale models motivates the use of cloud computing resources. However, concerns over the security of confidential power system models limits the potential use of cloud computing. In this paper we extend existing methods of masking optimization problems to the OPF problem. Specific contributions are the procedure for extracting the Lagrange multipliers from the masked dual problem, a method for preserving problem sparsity while ensuring a level of security in the masked problem, a method for masking a quadratic cost function, techniques for obscuring the number of system facilities and a proposal for masking nonlinear constraints.

Future work in this topic would investigate methods for controlling numeric conditioning while keeping problem sparsity. Without careful choice of operations, the approach in Section IV-B can lead to poor numeric conditioning. Additionally, further investigation of the security and characteristics of the masked AC OPF problem is needed.

A related masking application involves transforming the OPF problem of a given model into an OPF problem of a different model. In other words, masking the original OPF problem to yield a new problem that has the form of an OPF. This possibility would allow for increased sharing of confidential power system models for research purposes.

Furthermore the procedure of masking power system problems for cloud computing could be extended for use in multi-party computation. Here each party contributes a piece of the entire problem to collaboratively solve the problem involving all parties. From a power systems perspective, each party shares their masked confidential system model to collectively solve problems such as generation dispatch and transmission planning.

Acknowledgment

The authors acknowledge support of this work by U.S. Department of Energy under award #DE-SC0002319, as well as by the National Science Foundation under IUCRC award #0968833. Daniel Molzahn acknowledges the National Science Foundation Graduate Research Fellowship.

References

[1] J. Sun and L. Tesfatsion. DC Optimal Power Flow Formulation and Solution Using QuadProgJ. *ISU Economics Working Paper No. 06014.* Revised: March 1, 2010. [Online] Available at: http://www2.econ.iastate.edu/tesfatsi/dc-opf.jslt.pdf.

[2] Jadeja, Y.; Modi, K.; , "Cloud computing - concepts, architecture and challenges," *Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on* , vol., no., pp.877-880, 21-22 March 2012

[3] Kantarcioglu, M.; Bensoussan, A.; SingRu Hoe; , "Impact of security risks on cloud computing adoption," *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on* , vol., no., pp.670-674, 28-30 Sept. 2011

[4] Cachin, C.; Schunter, M.; , "A cloud you can trust," *Spectrum, IEEE* , vol.48, no.12, pp.28-51, December 2011

[5] Dreier, J.; Kerschbaum, F.; , "Practical Privacy-Preserving Multiparty Linear Programming Based on Problem Transformation," *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)* , vol., no., pp.916-924, 9-11 Oct. 2011

[6] H. Shojaei, A. Davoodi, and P. Ramanathan. Confidentiality Preserving Integer Programming for Global Routing. In *Proceedings of the 49th Annual Design Automation Conference,* pp. 709-716. ACM, 2012.

[7] Power Systems Test Case Archive, University of Washington Department of Electrical Engineering. [Online]. Available: http://www.ee.washington.edu/research/pstca/

[8] Zimmerman, R.D.; Murillo-Sánchez, C.E.; Thomas, R.J.; , "MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education," *Power Systems, IEEE Transactions on* , vol.26, no.1, pp.12-19, Feb. 2011