Privacy-preserving Computation for Large-scale Security-Constrained Optimal Power Flow Problem in Smart Grid

XIANGYU NIU\textsuperscript{1}, (Student Member, IEEE,) HUNG KHANH NGUYEN\textsuperscript{2}, JINYUAN SUN\textsuperscript{1}, (Member, IEEE,) and ZHU HAN\textsuperscript{2}, (Fellow, IEEE)

\textsuperscript{1}Department of Electrical Engineering and Computer Science, The University of Tennessee, Knoxville, Tennessee 37996. (e-mail: xniu@vols.utk.edu, jyun@utk.edu)\textsuperscript{2}H. K. Nguyen and Zhu Han are with Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004 USA. (e-mail: hknnguyen9@uh.edu, zhan2@uh.edu)

Corresponding author: Xiangyu Niu (e-mail: xniu@vols.utk.edu).

ABSTRACT In this paper, we present a distributed privacy-preserving quadratic optimization algorithm to solve the Security Constrained Optimal Power Flow (SCOPF) problem in the smart grid. The SCOPF problem seeks the optimal dispatch subject to a set of postulated constraints under the normal and contingency conditions. However, due to the large problem size and real-time requirement, a fast and robust technique is required to solve this problem. Moreover, due to privacy concerns, it is important that the data remains confidential and processed on local computers. Therefore, a fully privacy-preserving algorithm is proposed which performs computation directly over the encrypted SCOPF problem. The SCOPF is decomposed into smaller subproblems corresponding to individual pre-contingency and post-contingency cases using the Alternating Direction Method of Multipliers (ADMM) and gradient projection algorithms. Both algorithms are presented for solving the SCOPF problem in a privacy-preserving and distributed manner. Security analysis shows that our algorithm can preserve both system confidentiality and data privacy. Performance evaluations validate the correctness and effectiveness of the proposed algorithm.

INDEX TERMS Optimal Power Flow, Privacy-preserving, Security, Cloud Computing, Smart Grid, Distributed Systems.

I. INTRODUCTION

In order to adequately supply the connected load while minimizing the operating costs, system operators need to solve the optimal power flow (OPF) problem subject to physical constraints and control limits of the power system [1]. However, due to the large-scale interconnected topology of transmission and distribution networks, the OPF model can not ensure the demand-supply balance condition when the power system experiences unexpected failure and disconnection of components such as generators, transmission lines, transformers, etc., known as an outage or a contingency [2].

To address this issue, security requirements that ensure the power system continues its reliable operation during contingency scenarios need to be performed with the OPF problem, which is referred to as the security-constrained optimal power flow (SCOPF) problem in the literature [3], [4]. The optimal solution of the SCOPF problem produces the minimal cost generation dispatch while still assures that the power system remains balanced and no operational constraints are violated in both the normal state and contingencies [3], [4].

The contingency analysis is performed by independent system operators (ISOs) to ensure the reliable operation of power systems in both normal and contingency cases. By taking into account both pre-contingency constraints and post-contingency constraints in the SCOPF, the security of the power system can be significantly improved [5]. An illustration of the SCOPF is shown in Fig. 1. The system security level is improved by taking into account a number of contingencies in a dedicated selected contingency list. The SCOPF is commonly classified into two major types: the preventive model [3] and corrective model [4]. The preventive SCOPF formulation seeks the minimum cost dispatch solution in
the normal state that requires the normal-state variables to be feasible for all pre-specified contingency conditions, i.e., the control variables are not allowed to reschedule in contingency scenarios. However, this model makes the solution to be more conservative and may incur, in general, a higher operation cost [3].

The corrective SCOPF model, on the other hand, permits system operators to adjust post-contingency control variables such as power generation outputs and line power flows within a certain limit to eliminate any violation caused by the contingency. Due to the capability of adjusting control variables, the corrective SCOPF model often produces the optimal solution that has a lower total generation cost than the preventive model [6].

Despite the economical benefit of the corrective SCOPF model, its formulation generates a significant number of variables, which sharply increase the problem size when numerous contingencies are taken into account. The large-scale problem may result in excessive memory usage and unacceptable computation time [7]. On one hand, researchers have been investigating possible ways to optimize a variety of optimization problems [8]. On the other hand, new technologies, such as edge computing and cloud computing, have demonstrated their huge potential of tremendously speeding up intensive computation while reducing the cost [9], [10]. Hence, outsourcing the SCOPF problem to the cloud has emerged as a promising solution to the aforementioned challenges. Nonetheless, the fact that the operation takes place entirely at a third party will inevitably raise privacy concerns about data sensitivity. Sensitive power grid data can be captured by cyber attackers and used to initiate more sophisticated attacks (e.g. false data injection attack [11]) which could have potentially catastrophic consequences. Alternatively, the generic secure multi-party computation can implement any algorithm in principle, allowing the utility companies to take advantage of the aforementioned outsourcing paradigms while protecting the privacy of their operational data.

In this paper, we present a decentralized structure of outsourcing paradigm and a distributed privacy-preserving algorithm to demonstrate the feasibility of solving the corrective SCOPF problem without losing data privacy. The basic idea of the proposed scheme is to let each substation encrypt its private data after which a third party performs the SCOPF algorithm over the encrypted data without decrypting it. The third party then sends the encrypted result to the ISO company, which can be decrypted using the pre-distributed secret key. This is accomplished by leveraging additive Homomorphic Encryption (such as the Paillier cryptosystem [12]). However, according to the additive homomorphism property, we cannot directly solve the SCOPF problem using any available methods. In this work, we leverage both alternating direction method of multipliers (ADMM) [13] and gradient projection algorithm [14] to transform the SCOPF problem into a solvable problem for the additive homomorphic cryptosystem.

Note that even though the proposed scheme is based on ADMM and gradient projection, it can also be easily extended to other sophisticated optimization algorithms. Also, our proposed method is not limited to solve the SCOPF problem; it can also be applied to any other optimization problems that involve ADMM or gradient projection such as Internet congestion control and power system state estimation.

By presenting privacy-preserving computation for the large-scale SCOPF problem the main contributions of this paper can be summarized as follows:

- This work is the first to consider a privacy-preserving method for the SCOPF problem. Although there are a number of papers published to solve OPF problems in a privacy-preserving manner. With additional contingencies, solving SCOPF problems is much more costly. With a strong secure guarantee against semi-honest attackers that is proposed in this paper, ISO can confidently outsource the computation-heavy SCOPF problem to public cloud computing services.
- Decompose the SCOPF problem into independent sub-problems using ADMM and gradient projection which can be solved asynchronously.
- Propose privacy-preserving ADMM and gradient projection algorithms using additive homomorphic encryption. Both algorithms can be applied to other quadratic optimization problems with minor modifications.

We organize the rest of this paper as follows. Section II reviews the related works. Section III introduces the architecture and threat model. Section IV explains the components of privacy-preserving SCOPF. The section continues to present the optimization techniques and summarizes our proposed privacy-preserving SCOPF scheme along with security analysis in Section V. Section VI presents the experiment results of our proposed scheme. Finally, we conclude this paper in Section VII.
II. RELATED WORK

Due to the large-scale nature of the SCOPF problem, recent research has tried to propose different methodologies to handle the SCOPF problem. The contingency filtering technique has been developed in [15], [16] to discard contingencies that do not affect the optimal solution. An exact method to obtain the global optimal solution using a branch-and-bound algorithm is proposed in [7]. The authors in [17] use benders decomposition techniques to handle each contingency separately and check the feasibility of the optimization problem to achieve computational efficiency. The studies in [6], [18] apply the ADMM decomposition method to design a parallel computing framework, which can be executed simultaneously on multiple computers to reduce the latency.

[19], [20] try to solve the OPF problem using differential-privacy (DP) algorithms. Those methods target a multi-agent distributed computation scenario and use conventional OPF methods with different types of DP methods. In comparison, our paper has a stronger adversary model (honest-but-curious adversaries) thus cannot leverage DP methods. Another paper also considers the privacy of OPF in a multi-party computation scenario in [21]. This paper uses obfuscation to guarantee the privacy which is not computationally indistinguishable against probabilistic polynomial-time adversaries.

Previously, researchers mainly focused on the privacy of customer data in the smart grid. Giaconi et al. [22] study information leakage issues in a smart meter system and the privacy can be partially preserved by a low-complexity policy that can approach the theoretical lower bound. This scheme only guarantees a lower information leakage rate in limited scenarios. Other methods for protecting consumers’ privacy can be found in [23]–[25]. Additionally, only a few research works have been published to address sensitive operational data for the power system. [26] tries to securely outsource widely applicable linear programming (LP) computations by applying affine mapping on decision variables, which will transform the original vector space to a different one. [27] proposes a novel scheme that enables privacy-preserving multi-party spectral estimations, which conduct spectral estimation directly over the encrypted synchrophasors to limit privacy breaches. However, both works only deal with the minimal scenario which cannot be applied to solve more complicated problems such as SCOPF.

Furthermore, privacy-preserving computations have received significant attention in areas other than the smart grid. A popular approach that makes use of two-party computation is based on Yao’s garbled circuits. [28] shows the feasibility by designing a system that performs matrix factorization, a popular method used in a variety of modern recommendation systems. Several frameworks implement Yao’s garbled circuits and describe many applications [29]. However, due to the inherent serial property, recent work [30] shows that the garbled circuit is $2^{14}$ times slower than computing in plaintext. To deal with the slowness of the garbled circuit, some works introduced hybrid approaches, such as combining Homomorphic Encryption and garbled circuits for regression [28], face [31] and fingerprint recognition [32], and combining secret sharing with garbled circuits for learning a decision tree [33]. Nonetheless, the performance of those hybrid approaches is largely affected by network connection, which does not apply to our scheme.

III. SYSTEM ARCHITECTURE

A. SYSTEM MODEL

Our system is designed for one or multiple entities who want to solve the large-scale SCOPF problem with a limited computational resource. The system model of the proposed scheme is captured in Fig. 2 where the notation is given in Table 1. The proposed system model involves four different entities: Control Center (CC), Balancing Authority (BA), Server (S), and Cryptographic Provider (CP).

Control Center refers to the operator of the regional power system who solves the SCOPF problem to minimize generation costs, market surplus, and losses, etc. An independent system operator (ISO) firstly prepares a SCOPF problem $F$ which contains the object function and constraints and sends the initiation request to each balancing authority. In our scheme, the control center initiates a SCOPF problem and sends it to the server.

Balancing Authority is normally a substation in the power system $F$ which contains local operation data and contingencies. In our scheme, after receiving a request command from the control center, balancing authorities generate encrypted local data and upload it to the server.

Server only stores the public key $K_{pub}$. Upon receiving the encrypted SCOPF problem $F_c$ from the ISO, the server executes the privacy-preserving algorithm over $F_c$, and finally returns encrypted optimal $[P^{g,c}]$ to the ISO. The

### TABLE 1: Notation Definitions

| $\mathcal{G}$ | The set of generators |
|--------------|----------------------|
| $\mathcal{N}$ | The set of buses |
| $\mathcal{B}$ | The set of branches |
| $\theta^c \in \mathbb{R}^{|\mathcal{N}|}$ | The vector of voltage angles |
| $P^{g,c} \in \mathbb{R}^{|\mathcal{G}|}$ | The vector of real power flows |
| $f^g_i \in \mathbb{R}$ | The generation cost function |
| $P^{g,0}_i$ | The placeable real power of each individual generation unit $i$ for pre-contingency configuration |
| $B_{bus}^{g,c} \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{N}|}$ | The power network system admittance matrix |
| $V_{bus}^{g,c} \in \mathbb{R}^{|\mathcal{B}| \times |\mathcal{N}|}$ | The branch admittance matrix |
| $P_{dc} \in \mathbb{R}^{\mathcal{N}}$ | The real power demand |
| $C_{g,c} \in \mathbb{R}^{|\mathcal{N}|}$ | The sparse generator connection matrix, whose element $(i,j)$ element is 1 if generator $j$ is located at bus $i$ and 0 otherwise |
| $F_{max}$ | The vector for the maximum power flow |
| $P_{max}^g$ | The upper bound of real power generation |
| $P_{max}^{g,c}$ | The lower bound of real power generation |
| $\Delta_e$ | The pre-defined maximal allowed variation of power outputs |
| $K_{priv}$ | The private key of a cryptosystem |
| $K_{pub}$ | The public key of a cryptosystem |
| $\mathcal{F}$ | A Paillier cryptosystem |
| $\mathcal{F}_c$ | The plaintext of a SCOPF problem, denoted as a collection of base case and all the contingencies |
| $F_c$ | The encrypted SCOPF problem stored in the server |
| $[x]$ | The encryption form of $x$ using Paillier cryptosystem |

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119618, IEEE Access
Control flow

Balancing Authority

Area n

output an incorrect result. However, the server is motivated
server that tries to disrupt the optimization algorithm to
a correct model. Thus, we are not concerned with a malicious
adversary (honest-but-curious adversaries) [34]. The specific require-
party computation framework for semi-honest adversaries (or
compute the optimal state of the power system.

Contribution of our system is to ensure that the server learns
nothing about the power flow while still being capable of
computing the optimal state of the power system.

Cryptographic Provider is a third party that initializes
the system by assigning setup parameters to each party and
is needed for a short one-round online step in each iteration
while the server computes the model.

B. DESIGN GOALS AND THREAT MODEL

To enable secure, efficient, and accurate solving SCOPF over
the encrypted problem under the above model. Our goal is
to ensure the security of our algorithms using the secure two-
party computation framework for semi-honest adversaries (or
honest-but-curious adversaries) [34]. The specific require-
ments are summarized as follows:

- **Data Privacy**: The server and cryptographic provider
could not reveal the power flow using the statistical
information during the computation process.

- **System Confidentiality**: Given an encrypted SCOPF
problem $F_c$. The server and cryptographic provider are
not able to recover key information of the power grid,
such as power demand, bus limits, and generation load.

- **Efficiency**: The scheme aims to achieve efficiency by
offloading the computation to the server and by using
the parallel approach.

- **Accuracy**: The difference between the result calculated
by the proposed parallel scheme and traditional central-
ized method will not exceed the threshold $\epsilon$.

In our system, we assume that the Server is able to produce
a correct model. Thus, we are not concerned with a malicious
server that tries to disrupt the optimization algorithm to
output an incorrect result. However, the server is motivated
to learn information about private data stored in the server
since this data can potentially be sold to other parties, e.g.,
the black market. In our scenario, consider the server is
compromised by a semi-honest adversary. The adversary
aims to learn the SCOPF problem and the optimization result
as much as possible by analyzing all the input and output of
this party. That is to say, the server can conduct a ciphertext-
only attack (COA) [34] in this model. However, the adver-
sary cannot prevent this party from executing the algorithm
faithfully. Hence, we do not consider an adversary who will
intentionally corrupt the operation to generate misleading
results.

IV. PRELIMINARIES

To better understand our scheme, in this section, we review
the SCOPF problem and necessary components of the pro-
posed algorithm.

A. SCOPF PROBLEM FORMULATION

The corrective SCOPF problem finds the optimal dispatch so-
lation for the power network while satisfying security criteria
in which the system operator is allowed to re-adjust control
variables after a contingency occurs. This capability gives the
system operator a time window to adjust control variables in
order to eliminate any violations caused by the contingency.

The general formulation of the corrective SCOPF problem
can be compactly formulated as follows [4]:

\[
\begin{align*}
\min_{x^0, \ldots, x^C; u^0, \ldots, u^C} & \quad f^0(x^0, u^0) \\
\text{subject to} & \quad g^0(x^0, u^0) = 0, \\
& \quad h^0(x^0, u^0) \leq 0, \\
& \quad g^c(x^c, u^c) = 0, \quad \forall c \in C, \\
& \quad h^c(x^c, u^c) \leq 0, \quad \forall c \in C, \\
& \quad |u^0 - u^c| \leq \Delta_c, \quad \forall c \in C,
\end{align*}
\]

where $C = \{1, 2, \ldots, C\}$ is the set of postulated contingencies,
superscript $c$ denotes variables and constraints associ-
ated with the $c$-th contingency, superscript $0$ represents the
base case (pre-contingency state), $x$ and $u$ denote state vari-
ables and control variable. The constraints in (2) and (3) de-
note the set of equality and inequality constraints associated
with the operation of the power system such as transmission
line limits, power flow equations, etc., in the base case. Simi-
larly, (4) and (5) represent the operational constraints of the
power system when switching into contingency states. The
last constraints in (6) are the coupling constraints between
the base case and post-contingency, which means that the
deviation of control variables between the normal state and
contingency states must be within the allowable adjustment
limit, denoted by $\Delta_c$.

In the SCOPF problem, equality constraints $g^c$ are the
nodal load-flow equations, and inequality constraints $h^c$ are
the plant and transmission system operating limits. Con-
straints (2)-(3) stand for the economic dispatch and enforce
the feasibility of the pre-contingency state while constraints (4)-(5) stand for the post-contingency state. The superscript \(c = 0\) corresponds to the pre-contingency configuration, while \(c = 1, \ldots, C\) corresponds to the \(c\)-th post-contingency state. \(\Delta_c\) is the maximal allowance of the control variables between the pre-contingency and post-contingency states. Note that there are some variations on the objective function and constraints of the SCOPF problem, and we focus on the conventional formulation in this paper.

Based on the standard form of the SCOPF problem, there are some variations on the objective function and constraints for the SCOPF problem in alternating current (AC) and direct current (DC) power networks. For the sake of computational tractability of the proposed privacy scheme, we consider a DC power network with the objective function to minimize the total generation cost while ensuring the security requirements for the power system. Therefore, the corrective SCOPF problem can be simplified as follows [18]:

\[
\min_{\theta^0, \ldots, \theta^C, \mathbf{P}^g, 0, \ldots, \mathbf{P}^g, C} \sum_{i \in \mathcal{G}} f_i^g (\mathbf{P}^g, 0) \tag{7}
\]

subject to

\[
\mathbf{B}^{g, 0} \mathbf{\theta} + \mathbf{P}^{d, 0} - C^{g, 0} \mathbf{P}^g, 0 = 0, \tag{8}
\]

\[
\mathbf{B}^{g, c} \mathbf{\theta} + \mathbf{P}^{d, c} - C^{g, c} \mathbf{P}^g, c = 0, \tag{9}
\]

\[
|\mathbf{B}^{g, 0} \mathbf{\theta} + \mathbf{F}_{\text{max}}| \leq 0, \tag{10}
\]

\[
|\mathbf{B}^{g, c} \mathbf{\theta} + \mathbf{F}_{\text{max}}| \leq 0, \tag{11}
\]

\[
\mathbf{P}^{g, 0} \leq \mathbf{P}^g, 0 \leq \mathbf{P}^{g, 0}, \tag{12}
\]

\[
\mathbf{P}^{g, c} \leq \mathbf{P}^g, c, \tag{13}
\]

\[
\mathbf{P}^{g, 0} - \mathbf{P}^{g, c} \leq \Delta_c, \tag{14}
\]

\(i \in \mathcal{G}, \quad c = 1, \ldots, C\), \tag{15}

where \(\mathbf{P}^g, 0\) is the generation output of each individual generator for pre-contingency configuration. \(f_i^g (\mathbf{P}^g, 0)\) represents the generation cost function of the generator using the following function

\[
f_i^g (\mathbf{P}^g, 0) = a_i (\mathbf{P}^g, 0)^2 + b_i \mathbf{P}^g, 0 + c_i, \tag{16}
\]

where \(a_i, b_i\) and \(c_i\) are the cost coefficients. The constraints (8) and (9) are the nodal load-flow equations. The inequality constraints (10) and (11) are the transmission line limits. Constraints (12) and (13) are the power generation limits. Constraints (14) enforce the maximum adjusting limits for the generation units when switching into post-contingency states. The key notations of the above problem can be found in Table 1.

The problem in (7)-(15) is convex and can be solved by a central controller using convex optimization techniques [35]. However, due to numerous contingencies being incorporated into the model, the formulated problem becomes a large-scale optimization problem, which makes the centralized computational framework impractical. We will propose a parallel computation algorithm using the ADMM decomposition technique in the next section.

### B. PAILLIER CRYPTOSYSTEM

Paillier cryptosystem is a public key based additive homomorphic cryptosystem first proposed in [12] and further generalized by Damgård and Jurik [36]. Here, additive homomorphic cryptosystem means it can compute the sum of two values in the encrypted domain, which means given the encryption of \(a\) and \(b\), we can get encryption \(a + b\) without decryption. In this case, the Paillier cryptosystem is very useful for privacy-preserving applications. In this subsection, we will illustrate the key components of the Paillier cryptosystem.

1) Key Generation

First, two big prime numbers \(p\) and \(q\) are selected to compute \(n = pq\) and \(\lambda = \text{lcm}(p - 1, q - 1)\). Then choose a random integer \(g\) where \(g \in \mathbb{Z}_n^*\) and the order of \(g\) is a multiple of \(n\). The private key is \(\lambda\) and the public key is the tuple \((n, g)\).

2) Encryption

To encrypt a value \(m\) where \(m < n\) in the typical setting, we need to select a random value \(r \in \mathbb{Z}_n^*\). Then the ciphertext can be computed as: \(c = g^m \cdot r^n \mod n^2\). Note that in our scenario, \(m\) is not necessarily positive. Therefore, we divide the encryption space in two parts and then \(m\) is in the range of \((-n/2, n/2)\).

3) Decryption

To decrypt a ciphertext \(c\), the plaintext message can be computed as:

\[
m = D(c) = \frac{L(c^\lambda \mod n^2)}{L(g^\lambda \mod n^2)} \mod n, \tag{17}
\]

where function \(L\) is defined as \(L(u) = \frac{u - 1}{n}\).

4) Additive Homomorphism

Given the ciphertext of \(m_1\) and \(m_2\), the product of two ciphertexts will decrypt to the sum of their corresponding plaintexts:

\[
D(E(m_1, r_1) \cdot E(m_2, r_2) \mod n^2) = m_1 + m_2 \mod n. \tag{18}
\]

Moreover, given a ciphertext and a plaintext, we can compute the sum of the corresponding plaintexts:

\[
D(E(m_1, r_1) \cdot g^{m_2} \mod n^2) = m_1 + m_2 \mod n. \tag{19}
\]

5) Homomorphic Multiplication of Plaintext

By raising the ciphertext to a constant \(k\), we can get the encryption of the product of the plaintext and the constant:

\[
D(E(m_1, r_1)^k \mod n^2) = km_1 \mod n. \tag{20}
\]

However, given the encryptions of two plaintexts, there is no direct way to compute the product of these messages without knowing the private key.
C. THE ADMM METHOD

ADMM is a powerful algorithm that is proposed to solve small local subproblems, which are coordinated to find a solution to a global problem by blending the benefits of dual decomposition and augmented Lagrangian methods for constrained optimization and also widely adopted in power system [6], [18], [37]. The general form of ADMM is described as follows:

\[
\min_{x,z} \ f(x) + g(z) \tag{21}
\]
subject to \(Ax + Bz = c,\)

where \(x \in \mathbb{R}^n, z \in \mathbb{R}^m, A \in \mathbb{R}^{p \times n}, B \in \mathbb{R}^{p \times m} \) and \(c \in \mathbb{R}^p.\) Functions \(f\) and \(g\) are closed, convex and proper. The scaled augmented Lagrangian can be expressed as:

\[
\mathcal{L}_\rho(x, z, \mu) = f(x) + g(z) + \frac{\rho}{2} \|Ax + Bz - c + \mu\|_2^2, \tag{23}
\]

where \(\rho > 0\) is the penalty parameter and \(\mu\) is the scaled dual variable. Using the scaled dual variable, \(x\) and \(z\) can be updated in a Gauss-Seidel fashion. At each iteration \(k,\) the update process can be expressed as:

\[
x^{k+1} = \arg \min_x f(x) + \frac{\rho}{2} \|Ax + Bz^k - c + \mu^k\|_2^2, \tag{24}
\]
\[
z^{k+1} = \arg \min_z g(z) + \frac{\rho}{2} \|Ax^{k+1} + Bz - c + \mu^k\|_2^2. \tag{25}
\]

Finally, the scale dual variable is updated by:

\[
\mu^{k+1} = \mu^k + Ax^{k+1} + Bz^{k+1} - c. \tag{26}
\]

V. PRIVACY-PRESERVING SCOPF

The basic idea of this algorithm is to use the Paillier cryptosystem to solve multiplication, addition, and subtraction without leaking any information about the input. However, the objective function of the SCOPF problem is not necessarily linear and usually in quadratic form. To address this challenge, we will reformulate the SCOPF problem using the ADMM and gradient projection algorithms in this section.

A. REFORMULATING THE PROBLEM

The SCOPF problem in (7)-(15) contains a large number of constraints. However, we can separate constraints (8)-(13) into the normal state and each contingency state. The only constraints in (14) are coupling between the normal state and contingency state. In order to make constraints in (14) to be separable, we define auxiliary variables

\[
P^{g,c} = P^{g,c}_o, \forall c \in C, \tag{27}
\]

where each auxiliary variable \(P^{g,c}_o\) can be interpreted as a local copy of \(P^{g,c}\) at the normal state. Then, constraint (14) now can be rewritten separately for the normal state and each contingency as

\[
|P^{g,0} - P^{g,c}_o| \leq \Delta_c. \tag{28}
\]

Algorithm 1 Distributed SCOPF

Input: \(B_{bus}^g, B_f^g, A^{gc}, P^{d,e}, P^{g,c}, P^{g,c}, \Delta_c\)

Initialize: \(\theta^c, P^{g,e,c}, \mu^c, \rho^c, k = 0\)

1: while not converge do
2: \(P^{g,0}\)-update
3: \(P^{g,0,k+1} = \arg \min_{P^{g,0}} \frac{1}{2} \sum_{c=1}^C \mu^c P^{g,c}_o + \frac{\rho}{2} \sum_{c=1}^C \|P^{g,c} - P^{g,c}_o\|_2^2\)
\[s.t. \{P^{g,0}, P^{g,c}_o\} \in \mathcal{F}^0\]

4: \(P^{g,e,c}\)-update, distributively at each node:
5: \(P^{g,e,c,k+1} = \arg \min_{P^{g,e,c}} \mu^c P^{g,c}_o + \frac{\rho}{2} \|P^{g,c} - P^{g,c}_o\|_2^2\)
\[s.t. \{P^{g,e,c}\} \in \mathcal{F}^c\]
6: \(\mu^{k+1} = \mu^k + \rho(P^{g,c} - P^{g,c}_o)\)
7: Adjust penalty parameter \(\rho^c\) when necessary
8: \(k = k + 1\)
9: end while
10: return \(\theta^c, P^{g,e,c}, c = 0, ..., C\)

The constraints (8), (10), (12), and (28) are now consisting of variables in the normal state only, while constraints (9), (11), and (13) contains variables in contingencies. To facilitate the presentation, we define the feasible sets in the normal state, \(\mathcal{F}^0,\) as well as the constraints in the contingency state, \(\mathcal{F}^c,\)

\[
\mathcal{F}^0 = \{P^{g,0}, P^{g,c}_o\}|(8), (10), (12), (28)\}, \mathcal{F}^c = \{P^{g,c} - (9), (11), (13)\} \forall c \in C. \tag{29}
\]

Then, the problem in (7)-(15) can be rewritten as

\[
\min_{\theta^0, ..., \theta^C, P^{g,0}, ..., P^{g,c}} \sum_{i \in G} f_i^g(P^{g,0}_i) \tag{29}
\]
subject to
\[
\{P^{g,0}, P^{g,c}_o\} \in \mathcal{F}^0, \{P^{g,c} \in \mathcal{F}^c, \forall c \in C, \tag{30}
\]
\[
P^{g,c}_o = P^{g,c}_o, \forall c \in C. \tag{31}
\]

The augmented Lagrangian can then be calculated as:

\[
\mathcal{L}_\rho(P^{g,0}, ..., P^{g,c}, P^{g,0}_o, ..., P^{g,c}_o, \mu^1, ..., \mu^C) \tag{32}
\]
\[
= \sum_{i \in G} f_i^g(P^{g,0}_i) + \sum_{c=1}^C \mu^c (P^{g,c}_o - P^{g,c}_o) \tag{33}
\]
\[
+ \frac{\rho}{2} \sum_{c=1}^C \|P^{g,c}_o - P^{g,c}_o\|_2^2.
\]

Based on the Lagrangian function, we can decompose the problem in (7)-(15) into \(C + 1\) subproblems as follows: \(P^{g,0}\)-update; \(P^{g,e,c}\)-update; dual variables update. The subproblems are updated iteratively and can be processed distributively on the server. At each iteration, the \(P^{g,0}\)-update needs to coordinate with all the nodes to solve (34). The updated \(P^{g,0}\) will then be distributed to each computing node for \(P^{g,e,c}, \mu^c\) and dual variable update.
1) \( P^{g,0} \)-update

At the \( k^{th} \) iteration, the \( P^{g,0} \)-update solves the base scenario with square regularization terms enforce by the coupling constraints and expressed as:

\[
P^{g,0}[k+1] = \arg \min_{P^{g,0}} \mathcal{L}_\rho(P^{g,0})
\]

\[
= \arg \min_{P^{g,0}} \sum_{i \in \mathcal{G}} f_i(P^{g,0}) - \sum_{c=1}^{C} \mu^c P^{g,c}_o + \frac{\beta}{2} \|P^{g,c} - P^{g,c}_o\|_2^2
\]

s.t. \( \{P^{g,0}, P^{g,c}_o\} \in \mathcal{F}^0 \).

2) \( P^{g,c} \)-update

The remaining \( C \) subproblems are associated with variables in contingency scenarios. Each contingency can be solved in parallel at different computing nodes as

\[
P^{g,c}[k+1] = \arg \min_{P^{g,c}} \mathcal{L}_\rho(P^{g,c})
\]

\[
= \arg \min_{P^{g,c}} \sum_{i \in \mathcal{G}} f_i(P^{g,c}) - \sum_{c=1}^{C} \mu^c P^{g,c}_o + \frac{\beta}{2} \|P^{g,c} - P^{g,c}_o\|_2^2
\]

s.t. \( \{P^{g,0}, P^{g,c}_o\} \in \mathcal{F}^c \).

3) \( \mu \)-update

The computation of updating dual variables are also linear and can be performed locally at \( e^{th} \) computing utility as:

\[
\mu^c[k+1] = \mu^c[k] + \rho(P^{g,c} - P^{g,c}_o).
\]

B. SOLVING SUBPROBLEMS

In this section, we explain how to solve each subproblem using Homomorphic Encryption in a simple way. We formulate each subproblem as a quadratic optimization problem:

\[
\text{arg min}_{x} \quad x^T \cdot A \cdot x + b^T \cdot x
\]

subject to

\[
B \cdot x = d,
\]

\[
u \leq C \cdot x \leq v,
\]

\[
s \leq x \leq t.
\]

The linear inequality constraints are difficult to deal with basic operations in their current form. So we further simplify this problem by introducing a dummy variable \( y = C \cdot x \). Then, we obtain the problem:

\[
\text{arg min}_{x} \quad x^T \cdot A \cdot x + b^T \cdot x
\]

subject to

\[
B \cdot x = d,
\]

\[
C \cdot x - y = 0,
\]

\[
u \leq y \leq v,
\]

\[
s \leq x \leq t.
\]

This subproblem itself can be solved by ADMM. We denote the augmented Lagrangian:

\[
\mathcal{L}_\beta(x, y; w_1, w_2) := x^T \cdot A \cdot x + b^T \cdot x + \frac{\beta}{2} \|Bx - d - w_1\|_2^2 + \frac{\beta}{2} \|Cx - y - w_2\|_2^2,
\]

where \( w_1 \) and \( w_2 \) are scaled Lagrange multipliers, and \( \beta \) is a penalty parameter. The iteration of ADMM is

\[
x^{k+1} = \arg \min_x \mathcal{L}_\beta(x^k, y^k; w_1^k, w_2^k) \quad \text{subject to} \quad s \leq x \leq t;
\]

\[
y^{k+1} = \arg \min_y \mathcal{L}_\beta(x^{k+1}, y^k; w_1^k, w_2^k) \quad \text{subject to} \quad u \leq y \leq v;
\]

\[
w_1^{k+1} = w_1^k - (Bx^{k+1} - d);
\]

\[
w_2^{k+1} = w_2^k - (Cx^{k+1} - y^{k+1}).
\]

The x-subproblem does not have a closed-form solution. It is solved by another iteration of gradient projection: \( x \leftarrow \text{proj}_{[s,t]}(x - \alpha \nabla_x \mathcal{L}_\beta(x, y; w_1, w_2)) \). Since \( \mathcal{L} \) is quadratic, \( (x - \alpha \nabla_x \mathcal{L}_\beta(x, y; w_1, w_2)) \) is linear and reduces to matrix-vector multiplications and vector-vector sums/differences.

To summarize, the numerical operations of the above algorithms include and only include:

- matrix-vector multiplications;
- vector-vector addition and subtraction;
- component-wise min and max.

C. GRADIENT PROJECTION ALGORITHM

In the previous section, we reformulate the SCOPF problem to a set of subproblems that are solvable by additive Homomorphic Encryption. However, at the end of each subproblem, \( P^{g,0} \) and \( P^{g,c} \) are updated using the gradient projection algorithm. The gradient projection algorithm solves the bound-constrained optimization problems by projecting the result of the gradient descent to the feasible set:

\[
x_{k+1} = \text{proj}_{[s,t]}[x_k - \alpha \nabla f(x_k)].
\]

For the bound-constrained problems, the projection can be easily computed by setting:

\[
\text{proj}_{[s,t]}(x) = \text{mid}(x, s, t),
\]

where \( \text{mid} \) is the median element of a set that cannot be directly solved by additive Homomorphic Encryption.

We now describe our privacy-preserving gradient projection algorithm. One challenging part is how to perform comparison in the ciphertext domain. There are a few approaches to perform comparison efficiently. One naive way is sending the projection operation back to the ISO and calculating offline. Although this method is very efficient, it requires the ISO to keep online. Two methods that do not need the ISO to keep online are using specialized homomorphic encryption [38], or using garbled circuits [39]. Based on [40], the former is more efficient for comparison of encrypted values,
Algorithm 2: Gradient Projection Algorithm

Input: $S$: data $[x_k]$, lower boundary $[s]$, upper boundary $[t]$. Input: CP: private key $K_{priv}$
Output: $[x_{k+1}]$

1: $S: [x_k] \leftarrow [x_k] - \alpha \cdot \nabla \mathcal{L}_x(x)$
2: $S: [d_l] \leftarrow [x_k] \cdot [s]^{-1} \mod n^2$
3: $S: [d_h] \leftarrow [x_h] \cdot [t]^{-1} \mod n^2$
4: S choose two random integers $r_l, r_h \leftarrow (-2^l, 2^l)$
5: $S: [z_l] \leftarrow [d_l] r_l \mod n^2$
6: $S: [z_h] \leftarrow [d_h] r_h \mod n^2$
7: $S$ sends $[z_l]$ and $[z_h]$ to $B$
8: CP decrypts $[z_l]$ and $[z_h]$
9: CP checks $z_l < 0$ and $z_h > 0$ and encrypts them using $K_{pub}$
10: CP sends $S: [z_l < 0]$ and $[z_h > 0]$
11: if $z_l > 0 \oplus (r_l < 0)$ then
12: \hspace{1em} $[x_{k+1}] = [s]$
13: else if $z_h < 0 \oplus (r_h > 0)$ then
14: \hspace{1em} $[x_{k+1}] = [t]$
15: else
16: \hspace{1em} $[x_{k+1}] = [x_k]$
17: end if

The second is more efficient for comparison of unencrypted values. In our system, the lower and upper boundaries are normally power generation or bus limits which are sensitive information for the power grid. In this case, we are using homomorphic encryption to solve the comparison.

The idea is to exploit the homomorphic property to obscure the inputs with an additive mask. Here $r_l, r_h \leftarrow (-2^l, 2^l)$ are in the message space of homomorphic encryption $E$ and they follow that: $[x] \cdot (r_l, r_h) = (x \cdot r_l, x \cdot r_h)$. To evaluate (52) over encrypted $x$, we need first calculate $[x_k] - \alpha \cdot \nabla \mathcal{L}_x(x)$ using its homomorphic property. Then, the server chooses a random mask $r_l, r_h$, obscures the difference $[x - s; x - t]$ as above, and sends the masked value to the Cryptographic Provider. The Cryptographic Provider can apply its decryption key and determine if a number is positive or negative. $proj_{[s,t]}(x)$ is hence solved by the cloud by simply checking the value of $(r_l; r_h)$ the result sent from the Cryptographic Provider.

The privacy-preserving gradient projection algorithm details in Algorithm 2. Here, we use $[x]$ to denote the encryption of $x$ under the Paillier cryptosystem. Note that the computations in this algorithm are in the ciphertext domain and $[a] \cdot [b]^{-1} \mod N^2$ in the ciphertext domain is equal to $a \cdot b^{-1}$ in the plaintext domain. Detailed secure proof of Algorithm 2 is in Section V-G.

D. DEALING WITH FLOATING POINT NUMBERS

Given its Homomorphic property, the Paillier cryptosystem has been tested in many scenarios. However, one limitation of the Paillier cryptosystem is that it can only work with integers. Although we can test our scheme only on an integer model, the SCOPF problem usually uses floating-point numbers in the real world. Hence when we evaluate our scheme, we must adapt it accordingly.

In practice, as discussed in Section IV-B, the Paillier cryptosystem involves only additions and multiplications. Therefore, in [27], the authors use a solution by multiplying each floating number with a constant $K$. However, the selection of $K$ must guarantee that we do not overflow the Paillier cryptosystem plaintext space during the entire operation. To guarantee this, the selection of $K$ must satisfy:

$$0 < K < \sqrt{\frac{2^{1024}}{2 \cdot m_{max}}}$$

where $m_{max}$ is the largest plaintext number in the system. The square root is due to the fact that for multiplications, the Paillier cryptosystem requires integers for both sides of the operation. Fortunately, due to constraints (10)-(13) introduced in the SCOPF problem, the range of the encrypted values is small enough to ensure that the selection of $K$ can maintain high accuracy of the scheme.

Recall that, in our scheme, the global minima is found by iterations $P^{g,c} \leftarrow P^{g,c} - \alpha \cdot \nabla \mathcal{L}(P^{g,c})$ and at each iteration, the magnitude of $x$ will increase by $K$. This is because both $\alpha$ and $\nabla \mathcal{L}(P^{g,c})$ are scaled to integers by multiplying $K$. Moreover, as mentioned in Section IV-B, the Paillier cryptosystem can only deal with multiplication directly, not division. As a consequence, the plaintext space will be overflowed after several rounds of iterations. The simplest way to rescale $P^{g,c}$ is to send it to an authorized party who has the secret key to decrypt it and sends the rescaled value back to the cloud. However, in reality, it is costly to find a trusted third party for this job since this party needs to keep online during the whole process. Another practical solution is to use a garbled circuit for operations that are not able to be solved by Homomorphic Encryption. In our case, the garbled circuit is considered computationally complex compared with Homomorphic Encryption if we only use the garbled circuit to solve divisions.

In [41], an efficient way to compute $[x \div d]$ from $[x]$ and $d$ is proposed by additive blinding. Because $B$ is not allowed to learn the value $x$, it is additively blinded by a random number $r$ which is the statistical security parameter. It leads to Algorithm 3, where the random number $r$ is chosen as large as possible to ensure the best statistical hiding of $x$.

E. PRIVACY-PRESERVING SCOPF

Our Privacy-Preserving SCOPF algorithm can fall into 4 phases:

- **Phase 1: Key Generation and Distribution.** After the system is activated, Cryptographic Provider initializes the Paillier cryptosystem as described in Section IV-B with a public key $K_{pub}$ and a private key $K_{priv}$. The Cryptographic Provider sends the private key to the control center and keeps a copy to itself and assigns the public key to server $S$ and balancing authorities $B,A$.

- **Phase 2: System Initialization.** With the public key generated by the Cryptographic Provider, each area can
efficient and practical in real-world scenarios: The proposed algorithm has several strengths that make them
Input: $S: \{x\}$ and $d$.
Input: CP: $d$ and $K_{priv}$
Output: $\lfloor x \div d \rfloor$
1. $S$ chooses a random number $r$ of size $\log_2 n - 1$
2. $S$ computes $[x] = [x + r] = [x] \cdot [r] \mod n$ and sends $[x]$ to CP.
3. CP decrypts $[x]$, and computes $x \mod d$.
4. CP computes $c = \lfloor x \div d \rfloor$, encrypts it, and sends $[c]$ to $S$.
5. $S$ computes $[x \div d] = [(z \div d) - (r \div d)] = [c] \cdot [r \div d]^{-1} \mod n$.

Algorithm 3 Approximate Division Algorithm

Algorithm 4 Privacy preserving SCOPF
Input: ISO: $[P_{\theta}^0], [P_{\theta}^c], [P_{\theta}^d, c], B_{bus}^c, B_f^c, A_{g,c}$.
Initialize: $S: \{\theta, [P_{\theta}^c], [P_{\theta}^0], [\mu], \rho, k = 0$
while not converge do
$P_{\theta}^{0}-$update:
$S$ computes $\nabla_{P_{\theta}^0} L_{0}(P_{\theta}^{0+1})$
$S$ computes $P_{\theta}^{0+1} \leftarrow \text{proj}_{[0,1]}(P_{\theta}^{0+1} - \alpha \nabla_{P_{\theta}^0} L_{0}(P_{\theta}^{0+1}))$
/* Using Algorithm 2 */
$S$ computes $P_{\theta}^{0+1} / K$
/* Using Algorithm 3 */
$P_{\theta}^{0+1}$-update:
while $c < C$ do
$S$ computes $\nabla_{P_{\theta}^c} L_{c}(P_{\theta}^{c+1})$
$S$ computes $P_{\theta}^{c+1} \leftarrow \text{proj}_{[0,1]}(P_{\theta}^{c+1} - \alpha \nabla_{P_{\theta}^c} L_{c}(P_{\theta}^{c+1}))$
/* Using Algorithm 2 */
$S$ computes $P_{\theta}^{c+1} / K$
/* Using Algorithm 3 */
end while
while $c < C$ do
Update $\mu^{c+1} = \mu^c + \rho(P_{\theta}^c - P_{\theta}^0)$
end while
Adjust penalty parameter $\rho^c$ when necessary
$k = k + 1$
end while
return $[\theta], [P_{\theta}^c], c = 0, ..., C$
CC decrypts $[P_{\theta}^0]$ and $[P_{\theta}^c]$, divides it by $K$ to get final result.

F. DISCUSSION
The proposed algorithm has several strengths that make them efficient and practical in real-world scenarios:
• First, both balancing authorities and the control center do not need to stay online during the process. The ISO can leave the system after submitting the SCOPF problem and wait until final optimization is reached.
• Second, each local area can upload data in the encrypted form directly to the server to avoid the communication delay that costs by routing through the control center. The data integrity can be preserved using Internet layer security protocols such as IPSec.
• Furthermore, the system can be easily applied to solve SCOPF multiple times. Assuming that utility companies wish to perform optimization with different settings, it can initiate multiple instances. Consider an ISO wanting to know their system’s operation performance under different numbers of secure constraints. The server can test SCOPF with 20 and 40 secure constraints at the same time.
• Also, multiple estimations can be started when the ISO encrypt its sub-SCOPF problem and upload it to the server. The server then collects all the encrypted data and sets up the system by initialising the following variables: $[\theta^c], [P_{\theta}^c], [P_{\theta}^0], [\mu], \rho, k = 0$. It should notice that ADMM converges to the optimum geometrically for the convex optimization problem [42], and the convergence time will be significantly reduced by using the warm start technique [43].

• Phase 3: Privacy-preserving SCOPF. With the encrypted input, the server securely operates the privacy-preserving SCOPF Algorithm 4, which calls two supporting algorithms, i.e. Algorithms 2 and 3. Since all computations are carried out in the ciphertext domain, no information about the measurement is revealed.
• Phase 4: Result Decryption. The ISO receives the encrypted $\theta^c, P_{\theta}^c, c = 0, ..., C$ as a result of Algorithm 4 from the server. It then decrypts for the optimized variables using the private key. Also, since the uploaded power variables are multiplied by $K$, the optimized power variables should be divided by $K$ accordingly.

upload additional security constraints. In particular, the cryptographic provider does not need to refresh the public key too often since the public keys are long-lived, meaning that the ISO can submit more secure constants to the server without changing the previous ones.
• The computation is decoupled to subproblems $[P_{\theta}^c]$ to node $c$ and then iteratively update the dual variable to make sure the block-coupling constraints (28) are satisfied. In contrast with typical dual, or primal-dual decomposition schemes, the strength of ADMM resides in its noise-resilience as well as the very loose assumptions required to guarantee the robustness of our proposed method.

G. SECURITY ANALYSIS
In our model, what we are mainly trying to preserve is the value of $\theta^c, P_{\theta}^c, c = 0, ..., C$. Since the computation procedure of ADMM is publicly available, there is no need to hide the computation procedure. We will show the correctness of the algorithms, and then give proof of security in the honest-but-curious model. For correctness, we just modify the proof of [40].
• Definition. The two-party protocol securely computes the function $f$ if there exist two probabilistic polynomial-time
algorithms for every possible input $a, b, f$, it is computationally indistinguishable against probabilistic polynomial-time adversaries.

$$S_A(a, b, f_A(a, b)) \equiv_c V_A(a, b). \quad (55)$$

Here, $S_A$ means all the input of $A$, $\equiv_c$ means statistically indistinguishable for the adversaries, and $V_A$ means the view of $A$. Since we do not need to hide the computation procedure, we can reduce the secure definition to: if the input and output are statistically indistinguishable for the adversaries.

$$S_A(a, b) \equiv_c V_A(a, b). \quad (56)$$

**Proposition 1.** Algorithm 2 is correct and secure in the honest-but-curious model.

**Proof:** Since the process of upper and lower projections are identical here, we only try to prove the security of the process of lower boundary projection. $A$’s view is $V_A = ([x], [s], r, [x < s])$ and the output is $[x_{k+1}], [x - s]$. According to [44], it is sufficient to show that there exists a probabilistic polynomial-time algorithm $S$ such that $S(\pi, f(\pi))$ is computationally indistinguishable from $V$. By semantic security of the Pailler cryptosystem, each encryption is computationally indistinguishable, so this condition is easily verified if $S$ randomly generates $r$ of $\log_c N - 1$ bits. However, the following security is not guaranteed because $A$ will get the $x_k < s$ by comparing $[x_{k+1}]$ with $[s]$. But it does not compromise the privacy of $x$, since the boundary values $s$ and $t$ are both encrypted.

$$(\pi), [s], r, (x < s) \neq_c ([x], [s], r, (x > s)). \quad (57)$$

B’s view is $V_B = (SK_p, r \cdot (x - s))$ and the output is $r \cdot (x - s) > 0$. So if $r_1$ and $r_2$ are taken from the same distribution, independently from any other parameter, $V_B = ((x - s) < 0) = ((x - s) > 0)$. Here, since $r$ is taken randomly in $(-2^L, 2^L)$, the distribution of $(x - s > 0)$ and $(x - s < 0)$ are identical. Therefore, $([x], [s], r, (x < s))$ and $([x], [s], r, (x > s))$ are statistically indistinguishable. In this case, we conclude that Algorithm 2 is secure in the semi-honest model.

**Proposition 2.** Algorithm 3 is correct and secure in the honest-but-curious model.

**Proof:** The view of $A$ consists of its encrypted value $[x]$, and $d$, its random number $r$, its output $[(x + r)d]$, and all intermediate messages received from $B$: the encrypted comparison $[e]$. Summarizing, $A$’s view is $V_A = ([x], [z \div d], r)$ and the output is $[x + r]$. $B$’s view is $V_B = (K_p r, x + r, z \div d)$ and the output is $[x \div d]$. So if $r_1$ and $r_2$ are taken from the same distribution, independently from any other parameter. Here, since $r$ is taken randomly in $(-2^L, 2^L)$, the distribution of $r_1 \div d$ and $r_2 \div d$ are identical. Therefore, follow the similar steps, we conclude with the computational indistinguishability of $V_A$ and $V_B$.

**Proposition 3.** Algorithm 4 is correct and secure in the honest-but-curious model.

**Proof:** Algorithm 4 is semantically secure, because the algorithm only calls Algorithm 2 and Algorithm 3, and does not include computations other than the underlying Pailler cryptosystem. Both the called algorithms are semantically secure by our analysis.

**VI. EVALUATION**

In this section, the numerical tests are given to evaluate the performance of the proposed algorithm. Two classical test systems are used to formulate the SCOPF problem: IEEE 57 bus, IEEE 118 bus, whose structures and characteristics are summarized in Table 2. Two kinds of contingencies are considered in numerical tests: branch outage and generator failure. The contingencies are artificially generated and the number of contingencies considered is listed in Table 2. We follow the physical limits on the equipment of test systems and assume every active generator can reschedule up to 50% of its maximum real power capacity.

The numerical tests are implemented in Java and run on a personal computer with a 4 core, 8 threads, 2.2GHz processor and 16GB memory.

The performance of the convergence and computing time of the proposed algorithm are investigated in the following parts.

**A. ACCURACY**

One of the most dominant features that affect the performance of our system is the number of $K$ used in the Pailler cryptosystem. Through synthetic experiments based on IEEE 57 bus case, Table 3 illustrates the trade-off between the number of bits of $K$, with the relative error and time. Suppose $r^{\ast}$ is the result of the value of object function at the $k$th iteration, and $r^{\ast}$ is the optimal solution. The relative error $e$ is defined as $e = |r^{\ast} - r^{\ast}|$. Relative time is defined as $t = t_k - t_{min}$, where $t_{min}$ is the lowest time in the test. It is shown that the larger selection of $K$ will increase the computation time while reducing the relative errors. However, when $K$ is larger than 20 bits, the relative error will significantly increase. This is because the plaintext space of the Pailler cryptosystem is overflowed when $K$ is larger than 20 bits. In this case, to balance between efficiency and accuracy, we choose 15 bits as the length of $K$ in the following experiment.

**B. CONVERGENCE RATE**

We then consider the convergence issue of the proposed algorithm. To better understand the convergence rate between different cases, the relative error is used here to demonstrate the results. The convergence performances are shown in Fig. 3. It shows that the proposed algorithm converges to the optimal values in all two cases after a few iterations.
We can see that with a larger scale of the test system and the number of contingencies, the proposed algorithm has a slower convergence rate, which is due to the fact that a larger system and the number of contingencies considered, lead to a larger optimization problem. By comparing the convergence rate with algorithms that optimizing the same IEEE 57-bus, IEEE 118-bus problems in plaintext space, we observe that the performance of our method aligns with ADMM-based method [45], and out-performs [46] which couples Dierential Evolutionary (DE) and Particle Swarm Optimization (PSO) together. For the IEEE 57 case, all three methods are able to converge at iteration 15.

C. PERFORMANCE

In this part, we compare the computing time of the proposed algorithm with the centralized approach to solve the SCOPF problem. Note that, both algorithms are performed over the same personal computer. The performance can be further optimized when outsourcing to the cloud computing platforms like Amazon EC2 which use a higher clock rate CPU and more machines.

The computing time to obtain the optimal solution is considered in both cases. Communication overhead is also presented. The results of the computing time performance and communication overhead are presented in Table 4. From the table, we can see that the computing time of our system is slower than the centralized algorithm. This is because our methods perform entirely over the encrypted data. However, compared with the performance of Yao’s garbled circuit [30], which is normally $2^{10}$ time slower than plaintext implementations, our system is significantly more efficient. In addition, the gap between centralized and proposed algorithms becomes smaller when the testing case is larger. This is due to the communication overhead between different processes during the simulations. A larger problem can be achieved on a large-scale test system because the communication overhead is negligible compared with the computing time of the optimization subproblem handled by each computing node.

TABLE 3: Trade-offs between the number of bits used for $K$, relative errors and time

| Number of Bites of $K$ | Relative Time | Relative Errors |
|------------------------|---------------|-----------------|
| 2                      | 0             | 0.1409          |
| 5                      | 0.0017        | 0.0389          |
| 10                     | 0.2152        | 0.0247          |
| 13                     | 0.2303        | 0.0075          |
| 18                     | 0.2877        | 0.0         |
| 20                     | 1.7819 e+04   | 0.0247          |
| 25                     | 1.7819 e+04   | 0.0247          |

IEEE case 57
IEEE case 118

FIGURE 3: Convergence performance

VII. CONCLUSION

In this paper, the original SCOPF problem is decoupled and divided into smaller subproblems. The subproblems are approximately the same size and optimized in a parallel fashion on distributed nodes. We then presented a practical approach that can solve the large-scale SCOPF problem directly over the encrypted SCOPF problem to guarantee the security and privacy of the system. The privacy-preserving ADMM and gradient projection algorithms were also proposed to support the scheme. The scheme is computationally secure and parallelizable based on the additive homomorphic cryptosystem. The numerical tests on IEEE buses were carried out, which showed that our proposed scheme is less than $2^4$ times slower than the non-privacy-preserving method. Moreover, security analysis proves that our scheme can preserve both system confidentiality and data privacy against semi-honest attackers. As a result, the SCOPF problem can be solved by entities with abundant computational resources to achieve better efficiency and economy using our scheme.

REFERENCES

[1] J. Carpentier, “Contribution a l’étude du dispatching economique,” Bulletin de la Societe Francaise des Electriciens, vol. 3, no. 1, pp. 431–447, 1962.
[2] D. S. Kirschen, “Power system security,” Power Engineering Journal, vol. 16, no. 5, pp. 241–248, Oct. 2002.
[3] O. Alsac and B. Stott, “Optimal load flow with steady-state security,” IEEE Transactions on Power Apparatus and Systems, vol. PAS-93, no. 3, pp. 745–751, May 1974.
[4] A. Monticelli, M. V. F. Pereira, and S. Granville, “Security-constrained optimal power flow with post-contingency corrective rescheduling,” IEEE Transactions on Power Systems, vol. 2, no. 1, pp. 175–180, Feb. 1987.
[5] F. Capitanescu, J. M. Ramos, P. Panciatici, D. Kirschen, A. M. Marcolini, L. Platbrood, and L. Wehenkel, “State-of-the-art, challenges, and future trends in security constrained optimal power flow,” Electric Power Systems Research, vol. 81, no. 8, pp. 1731 – 1741, 2011.
[6] D. T. Phan and X. A. Sun, “Minimal impact corrective actions in security-constrained optimal power flow via sparsity regularization,” IEEE Transactions on Power Systems, vol. 30, no. 4, pp. 1947–1956, Jul. 2015.
Y. Tong, J. Sun, and K. Sun, “Privacy-preserving spectral estimation in electric power grids,” IEEE Transactions on Power Systems, vol. 29, no. 2, pp. 863–872, Mar. 2014.

J. Zhao, S. Liu, M. Zhou, X. Guo, and L. Qi, “Modified cuckoo search algorithm to solve economic power dispatch optimization problems,” IEEE/CIAA Journal of Automatica Sinica, vol. 5, no. 4, pp. 794–806, 2018.

Y. Cai, Y. Guo, and Y. Fang, “Privacy-preserving scheme for power system state estimation,” IEEE/CIAA Journal of Automatica Sinica, vol. 5, no. 1, pp. 311–319, 2018.

J. Bi, H. Yuan, S. Duannu, M. C. Zhou, and A. Abusorrah, “Energy-optimized parallel computation offloading in mobile edge computing with genetic simulated-anneling-based particle swarm optimization,” IEEE Internet of Things Journal, pp. 1–1, 2020.

Y. Liu, P. Ning, and M. K. Reiter, “False data injection attacks against state estimation in electric power grids,” ACM Transactions on Information and System Security (TISSEC), vol. 14, no. 1, p. 13, 2011.

P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 1999, pp. 223–238.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” Foundations and Trends® in Machine Learning, vol. 3, no. 1, pp. 1–122, Jan. 2011.

D. P. Bertsekas, Nonlinear programming. Athena scientific Belmont, 1999.

A. J. Ardakani and F. Boufard, “Identification of umbrella constraints in dc-based security-constrained optimal power flow,” IEEE Transactions on Power Systems, vol. 28, no. 4, pp. 3924–3934, Nov. 2013.

F. Capitaniucu, M. Glavic, D. Ernst, and L. Wehenkel, “Contingency filtering techniques for preventive security-constrained optimal power flow,” IEEE Transactions on Power Systems, vol. 22, no. 4, pp. 1690–1697, Nov. 2007.

Q. Wang, J. D. McCalley, T. Zheng, and E. Litvinov, “A computational strategy to solve preventive risk-based security-constrained optimal power flow,” IEEE Transactions on Power Systems, vol. 28, no. 2, pp. 1666–1675, May 2013.

L. Liu, A. Khodaei, W. Yin, and Z. Han, “A distribute parallel approach for big data scale optimal power flow with security constraints,” in IEEE International Conference on Smart Grid Communications (SmartGridComm), Oct. 2013, pp. 774–778.

V. Dvorkin, P. Van Hentenryck, J. Kazempour, and P. Pinson, “Differentially private distributed optimal power flow,” in 2020 59th IEEE Conference on Decision and Control (CDC), 2020, pp. 2092–2097.

F. Zhou, J. Anderson, and S. H. Low, “Differential privacy of aggregated dc optimal power flow data,” in 2019 American Control Conference (ACC), IEEE, 2019, pp. 1307–1314.

D. Wu, B. C. Lesieutre, P. Ramanathan, and B. Kakumoori, “Preserving privacy of ac optimal power flow models in multi-party electric grids,” IEEE Transactions on Smart Grid, vol. 7, no. 4, pp. 2050–2060, 2016.

G. Giacconi, D. Gündüz, and H. V. Poor, “Smart meter privacy with renewable energy and an energy storage device,” IEEE Transactions on Information Forensics and Security, vol. 13, no. 1, pp. 129–142, 2018.

D. Engel and G. Eibl, “Wavelet-based multiresolution smart meter privacy,” IEEE Transactions on Smart Grid, vol. 8, no. 4, pp. 1710–1721, 2017.

Z. Baharlouei and M. Hashemi, “Efficiency-fairness trade-off in privacy-preserving autonomous demand side management,” IEEE Transactions on Smart Grid, vol. 5, no. 2, pp. 799–808, 2014.

Y. Gong, Y. Cai, Y. Guo, and Y. Fang, “A privacy-preserving scheme for incentive-based demand response in the smart grid,” IEEE Transactions on Smart Grid, vol. 7, no. 3, pp. 1304–1313, 2016.

C. Wang, K. Ren, and J. Wang, “Secure and practical outsourcing of linear programming in cloud computing,” Apr. 2011, pp. 820–828.

Y. Tong, J. Sun, and K. Sun, “Privacy-preserving spectral estimation in smart grid,” in IEEE International Conference on Smart Grid Communications (SmartGridComm), Nov. 2015, pp. 43–48.

V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joyce, N. Taft, and D. Boneh, “Privacy-preserving matrix factorization,” in Proceedings of the ACM SIGSAC conference on Computer & communications security, Nov. 2013, pp. 801–812.

R. L. Lagendijk, J. Erkin, and M. Barni, “Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation,” IEEE Signal Processing Magazine, vol. 30, no. 1, pp. 82–105, Jan. 2013.

K. Nayak, X. S. Wang, S. Ioannidis, U. Weinsberg, N. Taft, and E. Shi, “Graphps: Parallel secure computation made easy,” in IEEE Symposium on Security and Privacy, May 2015, pp. 377–394.

A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, “Efficient privacy-preserving face recognition,” in International Conference on Information Security and Cryptology. Springer, 2009, pp. 229–244.

D. Evans, Y. Huang, J. Katz, and L. Malka, “Efficient privacy-preserving biometric identification,” in Proceedings of the 17th conference Network and Distributed System Security Symposium, NDSS, Feb. 2011.

R. Agrawal and R. Srikant, “Privacy-preserving data mining,” in ACM Sigmod Record, vol. 29, no. 2, May 2000, pp. 439–450.

O. Goldreich, Foundations of cryptography: volume 2, basic applications. Cambridge university press, 2009.

S. Boyd and L. Vandenberghe, Convex Optimization. New York, NY, USA: Cambridge University Press, 2004.

I. Dangjärd and M. Jurik, “A generalisation, a simplification and some applications of paillier’s probabilistic public-key system,” in International Workshop on Public Key Cryptography. Springer, 2001, pp. 119–136.

Q. Yang, G. Chen, and T. Wang, “Admn-based distributed algorithm for economic dispatch in power systems with both packet drops and communication delays,” IEEE/CIAA Journal of Automatica Sinica, vol. 7, no. 3, pp. 842–852, 2020.

T. Veugen, “Comparing encrypted data,” Multimedia Signal Processing Group, Delft University of Technology, The Netherlands, and TNO Information and Communication Technology, Delft, The Netherlands, Tech. Rep. 2011.

M. Bellare, V. T. Hoang, S. Keelvedhi, and P. Rogaway, “Efficient garbling from a fixed-key blockcipher,” in Security and Privacy (SP), 2013 IEEE Symposium on, May 2013, pp. 478–492.

B. Bost, R. A. Popa, S. Tu, and S. Goldwasser, “Machine learning classification over encrypted data,” IACR Cryptology ePrint Archive, vol. 2014, p. 331, Feb. 2014.

T. Veugen, “Encrypted integer division.” in IEEE International Workshop on Information Forensics and Security, Dec. 2010.

W. Deng and W. Yin, “On the global and linear convergence of the generalized alternating direction method of multipliers,” Journal of Scientific Computing, vol. 66, no. 3, pp. 889–916, 2016.

J. Nocedal and S. Wright, Numerical optimization. Springer Science & Business Media, 2006.

G. Oded, “Foundations of cryptography. basic applications, vol. 2,” 2004.

L. Yang, C. Zhang, and J. Jian, “A parallel method for solving the dc security constrained optimal power flow problem,” IEEE Transactions on Smart Grid, vol. 48, no. 10, pp. 3672–3690, 2018.
HUNG KHANH NGUYEN received the B.S. degree from the Department of Electrical and Electronic Engineering, Ho Chi Minh City University of Technology, Vietnam, in 2010, the M.S. degree from the Department of Electronics and Radio Engineering, Kyung Hee University, Korea, in 2012, and the Ph.D. degree in Electrical and Computer engineering from University of Houston, TX, USA, in 2017. His research interests include big data analytic, resource allocation and game theory, distributed and parallel optimization, large scale data processing in smart grid and wireless network.

JINYUAN SUN received the BSc degree in computer information systems from Beijing Information Science and Technology University, China, in 2003, the MASc degree in computer networks from Ryerson University, Canada, in 2005, and the PhD degree in electrical and computer engineering from the University of Florida, in 2010. She was a network test developer with RuggedCom Inc., Ontario, Canada, 2005-2006. She was an assistant professor in the Department of Electrical Engineering and Computer Science, University of Tennessee Knoxville, from August 2010 to July 2016, and has been an associate professor since August 2016. Her research interests include security and privacy of cyber-physical systems, wireless networks, and mobile systems. She is a member of the IEEE and the ACM.

ZHU HAN (S'01–M'04-SM'09-F'14) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively. From 2000 to 2002, he was an RD Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a John and Rebecca Moores Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. Dr. Han was an IEEE Communications Society Distinguished Lecturer from 2015-2018, AAAS fellow since 2019 and ACM distinguished Member since 2019. Dr. Han is 1% highly cited researcher since 2017 according to Web of Science. Dr. Han is also the winner of 2021 IEEE Kiyo Tomiyasu Award, for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: “for contributions to game theory and distributed management of autonomous communication networks.”