


## Article

# A Privacy-Preserving, Two-Party, Secure Computation Mechanism for Consensus-Based Peer-to-Peer Energy Trading in the Smart Grid

Zhihu Li <sup>1</sup>, Haiqing Xu <sup>2</sup>, Feng Zhai <sup>1,3</sup>, Bing Zhao <sup>1</sup>, Meng Xu <sup>1</sup> and Zhenwei Guo <sup>4,5,\*</sup> <sup>1</sup> China Electric Power Research Institute, Beijing 100081, China<sup>2</sup> State Grid Corporation of China, Beijing 100031, China<sup>3</sup> School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China<sup>4</sup> Hangzhou Innovative Institute, Beihang University, Hangzhou 310051, China<sup>5</sup> Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University, Hangzhou 311121, China

\* Correspondence: zhenweigu0724@163.com

**Abstract:** Consumers in electricity markets are becoming more proactive because of the rapid development of demand–response management and distributed energy resources, which boost the transformation of peer-to-peer (P2P) energy-trading mechanisms. However, in the P2P negotiation process, it is a challenging task to prevent private information from being attacked by malicious agents. In this paper, we propose a privacy-preserving, two-party, secure computation mechanism for consensus-based P2P energy trading. First, a novel P2P negotiation mechanism for energy trading is proposed based on the consensus + innovation (C + I) method and the power transfer distribution factor (PTDF), and this mechanism can simultaneously maximize social welfare and maintain physical network constraints. In addition, the C + I method only requires a minimum set of information to be exchanged. Then, we analyze the strategy of malicious neighboring agents colluding to attack in order to steal private information. To defend against this attack, we propose a two-party, secure computation mechanism in order to realize safe negotiation between each pair of prosumers based on Paillier homomorphic encryption (HE), a smart contract (SC), and zero-knowledge proof (ZKP). The energy price is updated in a safe way without leaking any private information. Finally, we simulate the functionality of the privacy-preserving mechanism in terms of convergence performance, computational efficiency, scalability, and SC operations.

**Keywords:** P2P negotiation mechanism; consensus + innovation method; homomorphic encryption; zero-knowledge proof; two-party; secure computation; blockchain; smart contract



**Citation:** Li, Z.; Xu, H.; Zhai, F.; Zhao, B.; Xu, M.; Guo, Z. A Privacy-Preserving, Two-Party, Secure Computation Mechanism for Consensus-Based Peer-to-Peer Energy Trading in the Smart Grid. *Sensors* **2022**, *22*, 9020. <https://doi.org/10.3390/s22229020>

Academic Editors: Peter Han Joo Chong and Omprakash Kaiwartya

Received: 22 August 2022

Accepted: 14 November 2022

Published: 21 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Recently, renewable distributed energy resources (DERs) [1], electric vehicles (EVs) [2–6], and energy storage systems (ESSs) have turned traditional consumers into prosumers; therefore, they can share energy locally to optimize the load and costs. Many households are now equipped with renewable generators, such as solar panels or wind turbines, which can provide energy in order to satisfy their own demand. The use of these DERs can help more DERs be absorbed into the grid in order to further reduce pollution. However, consumers who participate in the electricity market are required to behave more proactively and are, thus, known as prosumers. The increase in the number of prosumers naturally implies the need for a decentralized energy-trading mechanism that allows prosumers to freely trade with each other without a central supervising entity. Therefore, the network architecture is also changing from centralized to decentralized. A fully decentralized network architecture can be defined as a peer-to-peer (P2P) network in which the participants in the network share a portion of their own resources with one another. These shared resources can be accessed directly by

other peers without the intervention of a mediating entity [7]. A formal definition of P2P networks can be found in [8]. In this context, P2P trading mechanisms have emerged as a next-generation energy-management technique that enables prosumers to actively participate in the energy market.

Although the P2P mechanism provides better scalability, reliability, and resilience, growing privacy concerns are hindering its widespread adoption. In a P2P network, it is expected that prosumers will trade their energy with each other without any influence from a central coordinator, which makes P2P platforms a trustless and unreliable system. In addition, P2P energy trading requires a significant amount of data to be exchanged in order to compute the optimal energy amounts and prices for all sellers and buyers [9]. Disclosing such local data for computation would be damaging to their privacy. For instance, local generation reveals the generation capacities and time series of generation patterns [10], and the local demand load reveals consumption patterns [11,12].

Therefore, protecting prosumers' privacy and encouraging them to cooperate are challenges in such an environment with a lack of trust and security. Different technologies have been used to solve these problems. Blockchain has emerged as a promising, user-friendly, and efficient technology for the implementation of secure and reliable P2P energy-trading mechanisms. Existing studies have exploited a large variety of blockchain-enabled platforms to ensure secure and transparent P2P energy trading [13–22]. It makes communication transparent for prosumers and allows them to make decisions about energy dispatches in a decentralized and untrusted environment. In blockchain, security mainly means that data are stored on all nodes, are resistant to single points of failure, and are unalterable. Existing blockchain-based energy-trading studies mainly used blockchain to store and protect the final trading results. In addition, smart contracts (SCs) play a very important role in P2P energy trading, as they control the energy transactions between two peers by following predefined rules [17,19,23].

Homomorphic encryption (HE) is a form of encryption that allows for computations on ciphertexts, which generate an encrypted result that, when decrypted, matches the results of the operations as if they had been performed on the plaintext [24]. HE can be further categorized into two classes: semi-HE and fully HE. Semi-HE methods are schemes that only support a subset of the encrypted arithmetic. For example, the Paillier algorithm only supports arithmetic that uses addition; therefore, it is also known as additive semi-HE. On the contrary, fully HE schemes support all encrypted arithmetic. The main advantage of HE is that the security is very high, since it is based on cryptographic techniques, while the most commonly known drawback of HE-based methods is the increased computing power that is required for their complex encryption and decryption operations. Some works have studied the application of HE technology to energy systems. A novel private collaborative distributed energy-management system (P-CoDEMS) was proposed in order to solve the problem of AC optimal power flow (ACOPF) in a distributed and private manner in [25]. Yi et al. integrated HE, blockchain, and other technologies to implement a secure energy trading system [26]. Liu et al. adopted the Paillier method to protect the privacy of ADMM-based distributed DC optimal power flow in [27]. The Paillier-based distributed optimization method was generalized for all gradient-based distributed optimization in [28], and it was reported to be applied to a distributed transactive problem in [29].

However, to our knowledge, existing works did not adequately consider privacy issues in the negotiation process for fully decentralized P2P energy-trading mechanisms. In the P2P energy-trading market, there are multiple agents who negotiate energy trades with each other, and the objective of the market mechanism is to determine the trading prices and amounts for each pair of agents. Thus, in this paper, we propose a privacy-preserving, two-party—instead of multi-party—secure computation mechanism for the negotiation process for each pair of agents. The novel privacy-preserving P2P energy-trading framework combines the technologies of blockchain, SCs, HE, and zero-knowledge proof (ZKP). In detail, we first propose a P2P negotiation mechanism that uses a combination of the

consensus + innovation (C + I) method with a power transfer distribution factor (PTDF) model. Then, we analyze the privacy disclosure risk of this mechanism in the case of collusive attacks from neighboring agents. To avoid this risk, a secure, two-party computation framework is designed for updating the energy price between each pair of agents. Finally, the simulation results demonstrate the performance of market convergence, and the line-limit constraints, scalability, and encryption/decryption computation are maintained. The main contributions are the following:

- We propose a novel P2P negotiation mechanism that incorporates the power transfer distribution factor (PTDF) model into the consensus + innovation (C + I) method, which can simultaneously maximize social welfare and comply with physical line constraints. By introducing line prices into the update process, agents are encouraged not to transfer power over congested lines.
- Although the C + I method exchanges a minimum amount of information, there is still a risk of revealing private information. We analyze how individual private information (e.g., coefficients of generation, utility functions, and power limits) can be stolen and computed through a collusion attack by a group of collusive neighboring agents in the context of the P2P negotiation mechanism based on the C + I method.
- The security objective and novelty of this paper are to protect the information exchanged between each pair of agents in the energy-trading negotiation process. We propose a novel, secure, two-party computation mechanism for the energy price update between each pair of agents based on the SC and Paillier encryption algorithm, which is known as an efficient additive HE method. Moreover, we propose a ZKP protocol to prove that the decrypted plaintext matches the ciphertext computed by SC.

The rest of the paper is organized as follows: Section 1 presents the formulation of the P2P energy-trading and social welfare maximization problem. Section 2 proposes the SC-based P2P negotiation mechanism for energy trading, followed by the two-party, secure computation framework in Section 3. The numerical results are presented in Section 4. Finally, in Sections 5 and 6, the discussions, conclusions, and future perspectives are drawn.

## 2. Problem Formulation

A typical P2P architecture for electricity markets is shown in Figure 1, which consists of simultaneous negotiation of the price and energy of multilateral trades based on predefined trading rules. It can be seen that a P2P mechanism for electricity markets is much more decentralized than existing centralized markets, where all agents must submit all their information, e.g., cost or utility function, power limits, and uncertainty information, to the market operator (MO), who centrally determines the dispatches of energy. In contrast, in P2P markets, all agents can freely negotiate the prices and quantities with each other for multilateral trading.

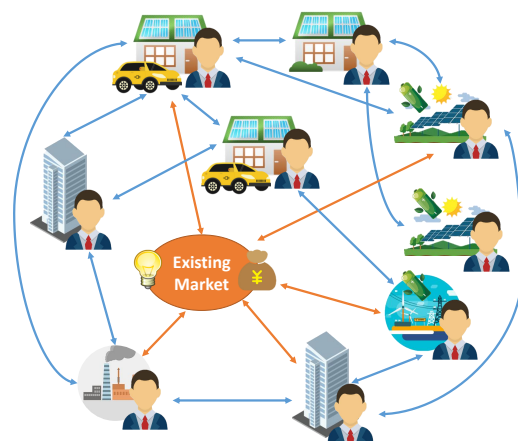


Figure 1. P2P energy-trading market architecture.

### 2.1. Peer-to-Peer Trading

In this paper, we build a market with a set  $\Omega$  of agents defined as either producers or consumers. The market-clearing mechanism proposed below is for a day-ahead market to allocate the supply and demand of energy. It is assumed that all agents are supposed to be rational and truthful, as in [30], which means that they always make decisions to maximize individual benefits. A similar model of the P2P energy-trading process was proposed in our previous work [31,32].

First, the power injection  $E_n$  of each agent  $n \in \Omega$  is divided into a sum of bilaterally traded quantities with a set of neighboring agents  $m \in \omega_n$  as

$$E_n = \sum_{m \in \omega_n} E_{nm}, \quad \forall n \in \Omega \quad (1)$$

A positive value of  $E_n$  represents surplus energy and a negative value means required energy. Before P2P energy trading, each prosumer will individually calculate the value of  $E_n$  according to the power generation and consumption and then decide to be a buyer or seller in the trading. A positive value of  $E_{nm}$  represents a sale/production, and a negative value means a purchase/consumption. To lighten notations,  $E_n = \{E_{n1}, \dots, E_{nm}, m \in \omega_n\}$  is used to represent the whole set of transactions of agent  $n$ . The power of an agent  $n$  is constrained as below:

$$\underline{E}_n \leq E_n \leq \overline{E}_n, \quad \forall n \in \Omega \quad (2)$$

Each agent is restrained to either producer or consumer ( $E_n \overline{E}_n \geq 0$ ). Hence, the decision variables are constrained to be positive ( $E_{nm} \geq 0$ ) if it is a producer and negative ( $E_{nm} \leq 0$ ) if it is a consumer, as follows:

$$\begin{cases} E_{nm} \geq 0, & \forall (n, m) \in (\Omega_p, \omega_n) \\ E_{nm} \leq 0, & \forall (n, m) \in (\Omega_c, \omega_n) \end{cases} \quad (3)$$

where  $\Omega_p$  and  $\Omega_c$  are the sets of energy producers and consumers, respectively.

Finally, the market equilibrium between energy production and consumption is represented by a set of balance constraints of each pair of agents

$$E_{nm} + E_{mn} = 0, \quad \forall (n, m) \in (\Omega, \omega_n) \quad (4)$$

### 2.2. Line Flow Constraints of Power Network

In this paper, PTDF is used to compute the power flow of lines and to label the lines used for power transfer in each transaction [33,34]. The PTDF for line  $l$  is denoted by  $\varphi_{ij}^l$  and indicates the fraction of the energy generated by the agents on bus  $i$  that is transmitted over line  $l$  to the agents on bus  $j$ . The PTDF is calculated by  $\varphi_{ij}^l = \psi_i^l - \psi_j^l$ , where  $\psi_i^l$ ,  $\psi_j^l$  are injection shift factors (ISF) in line  $l$  for bus  $i$  and  $j$ . The ISF is an approximation of the sensitivity matrix and quantifies the redistribution of power through each branch after a change in generation or load on a particular bus. The ISF matrix is represented by  $\Psi \triangleq [\psi_i^l] \in \mathbb{R}^{L \times N}$ , where  $N$  is the number of buses and  $L$  is the number of lines. This matrix can be obtained using  $\Psi \triangleq B'AC^{-1}$  by a diagonal branch susceptance matrix ( $B'$ ), a branch-node incidence matrix ( $A$ ), and a reduced nodal susceptance matrix ( $C$ ). In the matrix  $A$ ,  $a_i^T$  is the  $l^{th}$  row where a line exists between bus  $i$  and  $j$ .

$$A \triangleq [a_1, a_2, \dots, a_L] \in \mathbb{R}^{L \times N}, \quad a_i^T \triangleq [0 \dots 0 \overset{i}{1} 0 \dots 0 \overset{j}{-1} 0 \dots 0] \quad (5a)$$

$$B' \triangleq \text{diag}[b_1, b_2, \dots, b_L] \in \mathbb{R}^{L \times L}, \quad C \triangleq A^T B' A \in \mathbb{R}^{N \times N} \quad (5b)$$

By having PTDF matrix and traded energy between prosumers, the power flow in line  $l$  can be computed by (6)

$$P_l = \sum_{n \in \Omega_p} \sum_{m \in \Omega_c} \varphi_{ij}^l E_{nm} \quad (6)$$

In the above Equation (6), the producer  $n$  is at bus  $i$  and the consumer  $m$  is at bus  $j$ . Their traded power  $E_{nm}$  has an impact on the flow of the line  $l$ . If the value is below or above the boundaries, the line prices  $(\bar{v}_l, \underline{v}_l)$  are sent to the agents using that particular line to transfer power to avoid overflow or congestion.

Since the agents in the power grid use the conventional grid to transmit energy, both social welfare and line flow constraints should be considered. Here, line flow constraints are added as a constraint to the objective function to model the physical network in energy trading. To avoid damage to the transmission lines, the real power flow  $P_l$  in each line  $l$  is bounded by the maximum capacity  $P_l^{max}$  with respect to the heat they can dissipate.

$$-P_l^{max} \leq P_l \leq P_l^{max}, \quad \forall l \in \mathcal{L}. \quad (7)$$

### 2.3. Social Welfare Maximization Problem

To simplify the formulation of the process, we model the production cost and consumer utility functions as quadratic functions of the power set-point, as below:

$$C_n(E_n) = a_n E_n^2 + b_n E_n + c_n, \quad (8)$$

where  $a_n$ ,  $b_n$ , and  $c_n$  are predetermined positive constants. From above, the P2P market has the objective to maximize the social welfare of all agents under the constraints. The problem can be equivalently formulated as a cost minimization problem, as below:

$$\min \sum_{n \in \Omega} C_n(E_n) \quad (9a)$$

$$\text{s.t. } \underline{E}_n \leq E_n \leq \bar{E}_n \quad \forall n \in \Omega \quad (9b)$$

$$E_{nm} \geq 0 \quad \forall (n, m) \in (\Omega_p, \omega_n) \quad (9c)$$

$$E_{nm} \leq 0 \quad \forall (n, m) \in (\Omega_c, \omega_n) \quad (9d)$$

$$E_{nm} + E_{mn} = 0 \quad \forall (n, m) \in (\Omega, \omega_n) \quad (9e)$$

$$-P_l^{max} \leq \sum_{n \in \Omega_p} \sum_{m \in \Omega_c} \varphi_{ij}^l E_{nm} \leq P_l^{max} \quad \forall l \in \mathcal{L} \quad (9f)$$

Since the social welfare maximization (or cost minimization) problem is a convex optimization problem, it has a unique optimum that can be achieved by a plethora of centralized methods. However, this requires the disclosure of all the agents' information. It is better to design a P2P negotiation mechanism that can achieve optimal dispatches of the above optimization problem (9).

## 3. Blockchain-Based P2P Negotiation Mechanism for Energy Trading

In this section, we first design a novel P2P negotiation mechanism for energy trading inspired by the consensus-based approach proposed in [35]. We then present the implementation of P2P energy trading using blockchain and SC.

### 3.1. C + I-Based Decentralized Negotiation Mechanism

The decentralized negotiation mechanism for P2P energy trading is based on the C + I method, which consists of updates to the primary energy quantity variables, updates to the dual variables, and convergence criteria. The main reason for choosing the C + I method to design the market-clearing algorithm is that the information exchanged between agents is minimal compared to other methods, such as the ADMM method [36,37] and the primal-dual gradient [33]. Since the shared information is very small, the communication overhead is lower and the risk of leakage of private information is also lower. Compared with the previous results in [35], the first difference is that the physical line flow constraints of the power grid are considered in our model. Line prices are introduced to induce agents to spontaneously adjust their power generation or consumption, as shown in (13). The second difference is that SC is used to implement the mechanism, including updating the

energy quantities and prices, calculating the power flows, updating the line prices, convergence checking, storing the transaction results, and querying. Therefore, compared with previous work, the mechanism we developed is a more realistic and practical decentralized negotiation algorithm for P2P energy trading.

### 3.1.1. Local Optimization Problem

For each agent  $n$  in bus  $i$ , the local optimization problem at a given iteration  $k$  is

$$\min C_n(E_n) - \sum_{m \in \Omega_n} \lambda_{nm}^k E_{nm} + \sum_{l \in \mathcal{L}} \sum_{n, m \in i, j}^{m \in \omega_n} \varphi_{ij}^l (\bar{v}_l^k - \underline{v}_l^k) E_{nm} \quad (10a)$$

$$\text{s.t. } \underline{E}_n \leq E_n \leq \bar{E}_n \quad (10b)$$

$$E_{nm} \geq 0 \quad \forall m \in \omega_n \quad \text{if } n \in \Omega_p \quad (10c)$$

$$E_{nm} \leq 0 \quad \forall m \in \omega_n \quad \text{if } n \in \Omega_c \quad (10d)$$

where  $\lambda_{nm}$  are the dual variables of the equilibrium conditions (4) and define the traded energy prices  $E_{nm}$ .  $\lambda_n = \{\lambda_{n1}, \dots, \lambda_{nm}\}$  is used to represent the total traded energy prices between neighboring agents.

### 3.1.2. Primal Variable Updates

Updates to the energy quantities of agent  $n$  are based on the Karush–Kuhn–Tucker (KKT) conditions of the local optimization problem. The relaxed Lagrangian function of the local optimization problem (10) at iteration  $k$  can be expressed as follows:

$$L_n^{loc} = C_n(E_n) - \sum_{m \in \Omega_n} \lambda_{nm}^k E_{nm} + \sum_{l \in \mathcal{L}} \sum_{n, m \in i, j}^{m \in \omega_n} \varphi_{ij}^l (\bar{v}_l^k - \underline{v}_l^k) E_{nm} + \bar{\mu}_n (E_n - \bar{E}_n) - \underline{\mu}_n (E_n - \underline{E}_n) \quad (11)$$

According to the first-order optimality conditions of the Lagrangian problem, for all trades between agents  $n \in \Omega$  and  $m \in \omega_n$ , we have

$$a_n E_n + b_n - \lambda_{nm}^k + \sum_{n, m \in i, j}^{l \in \mathcal{L}} \varphi_{ij}^l (\bar{v}_l^k - \underline{v}_l^k) + \bar{\mu}_n^k - \underline{\mu}_n^k = 0 \quad (12)$$

Then, we can obtain that

$$E_n^{k+1} = \frac{\lambda_{nm}^k - \sum_{n, m \in i, j}^{l \in \mathcal{L}} \varphi_{ij}^l (\bar{v}_l^k - \underline{v}_l^k) - \bar{\mu}_n^k + \underline{\mu}_n^k - b_n}{a_n} \quad (13)$$

According to the complementary conditions  $\bar{\mu}_n \times \bar{E}_n = \underline{\mu}_n \times \underline{E}_n = 0$ , the above update (13) can be equivalently transformed to another more concise form, as below:

$$E_n^{k+1} = \max \left\{ \min \left\{ \frac{\lambda_{nm}^k - \sum_{n, m \in i, j}^{l \in \mathcal{L}} \varphi_{ij}^l (\bar{v}_l^k - \underline{v}_l^k) - b_n}{a_n}, \bar{E}_n \right\}, \underline{E}_n \right\} \quad (14)$$

In this way, the dual variables  $\{\bar{v}_l^k, \underline{v}_l^k\}$  is omitted and the update process is simpler. Then, the primal variables  $\{E_{nm}, m \in \omega_n\}$  are updated as below (here for a producer):

$$E_{nm}^{k+1} = \left[ E_{nm}^k + f_{nm}^k (E_n^{(m),k+1} - E_n^{(m),k}) \right]^+ \quad (15)$$

where  $f_{nm}$  is an asymptotically proportional factor defined as

$$f_{nm}^k = \frac{|E_{nm}^k| + \delta^k}{\sum_{l \in \omega_n} (|E_{nl}^k| + \delta^k)} \quad (16)$$

with  $\delta^k$  a positive constant. The operator  $[\cdot]^+ = \max(0, \cdot)$  in (15) is used to enforce the sign constraint of the decision variables and is replaced in the case of a consumer by operator  $[\cdot]^- = \min(0, \cdot)$ .

### 3.1.3. Dual Variable Updates

The price for a given trade is calculated individually by each agent. After convergence, a consensus has to be reached on these prices (i.e.,  $\lambda_{nm} = \lambda_{mn}$ ). The energy price  $\lambda_{nm}^{k+1}$  will be updated in this form:

$$\lambda_{nm}^{k+1} = \lambda_{nm}^k - \beta^k (\lambda_{nm}^k - \lambda_{mn}^k) - \alpha^k (E_{nm}^k + E_{mn}^k). \quad (17)$$

Price convergence is ensured in the price update by a consensus term. The last term, the innovation term, ensures energy equilibrium between agents.  $\alpha^k$  and  $\beta^k$  are sequences of positive factors set by the individuals such that each excitation is persistent so that the series of each sequence converge. The tuning of these parameters ( $\alpha^k$  and  $\beta^k$ ) is key to the convergence performance of the algorithm and usually requires a trade-off between convergence speed and adaptation to changes in setting. Performance could be improved by using an adaptive parameter. The calculations steps (13)–(16) are all performed locally without communicating with others. Only in step (17) does agent  $n$  need to receive information  $\{E_{mn}^k, \lambda_{mn}^k\}$  from agent  $m$  to update the energy price  $\lambda_{nm}^{k+1}$ .

Finally, the line manager (LM) will be responsible for calculating the power flows in each line by (6), and the line prices  $(\bar{v}_l^{k+1}, \underline{v}_l^{k+1})$  will be updated as

$$\bar{v}_l^{k+1} = [\bar{v}_l^k + \phi^k (P_l^{k+1} - P_l^{max})]^+ \quad (18a)$$

$$\underline{v}_l^{k+1} = [\underline{v}_l^k - \phi^k (P_l^{k+1} + P_l^{max})]^+ \quad (18b)$$

where  $\phi^k$  is the tuning parameter.

### 3.1.4. Condition of Convergence

The above decentralized algorithm converges as long as the following conditions are met:

$$\sum_{n \in \Omega} \sum_{m \in \omega_n} |E_{nm}^{k+1} - E_{nm}^k| \leq \chi^E \quad (19a)$$

$$\sum_{n \in \Omega} \sum_{m \in \omega_n} |\lambda_{nm}^{k+1} - \lambda_{nm}^k| \leq \chi^\lambda \quad (19b)$$

$$\sum_{l \in \mathcal{L}} |\bar{v}_l^{k+1} - \bar{v}_l^k| + |\underline{v}_l^{k+1} - \underline{v}_l^k| \leq \chi^v \quad (19c)$$

where  $\chi^E$ ,  $\chi^\lambda$  and  $\chi^v$  are stopping criterion predetermined by market operator.

## 3.2. Implementation of P2P Energy Trading by Smart Contracts

An illustration of the blockchain-based P2P trading architecture is shown in Figure 2. The process is described below.

- In the first step, all agents initiate a pair of energy prices and quantities in parallel and send it to neighboring agents. Then, each agent updates its quantities and prices for its neighbors using (15) and (17), respectively. The update process is automatically performed by SC, which is installed on each agent.

- After updating each agent, all agents send their traded energy to LM, which calculates the power flows and line prices on each line using (6) and (18), also from SC.
- Then, LM sends the line flow prices to the corresponding agents using the particular line for power transmission. By applying these line usage price signals, the agents will try to trade energy with nearby ones, which can reduce power losses.
- After each iteration, each agent and LM send the updated results to MO, who will check if the stopping criteria are met (19).
- Finally, after the market converges, MO collects all transactions and stores them in the blockchain.

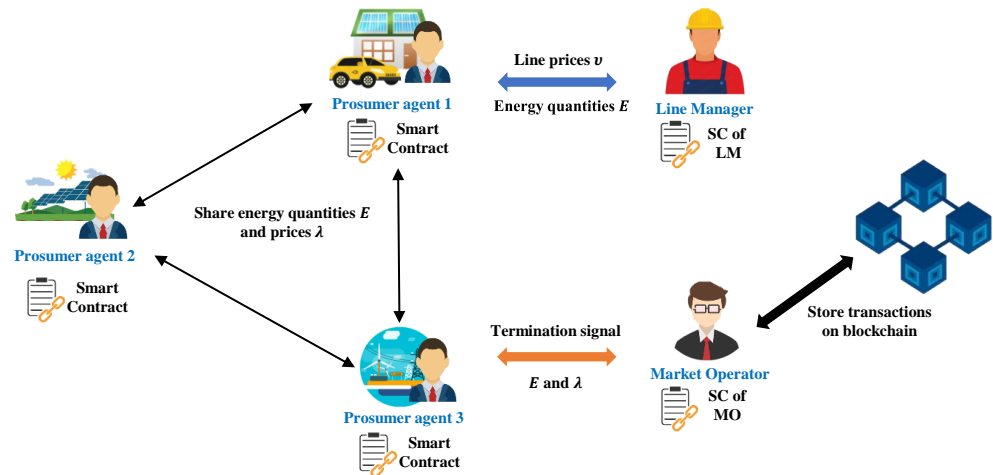


Figure 2. Blockchain-based P2P energy-trading market architecture.

#### 4. Privacy-Friendly P2P Computation Framework

We have formulated a decentralized negotiation algorithm between agents based on the C + I method, but there are still obvious shortcomings. During the negotiation process, agents need to share the updated energy and price data with neighboring agents, and privacy may be lost during the process. Malicious attackers can obtain private information by studying the updated energy and prices. Therefore, developing a privacy-friendly information exchange scheme is the prerequisite for P2P energy trading. In this paper, we propose a privacy-friendly, two-party, secure computation scheme, mainly using HE technology, SC, and ZKP to realize secure information exchange between agents. To our knowledge, none of the existing work uses HE for P2P energy trading. Previous works mainly use HE to solve the AC optimal power flow (ACOPF) problem [25], DC optimal power flow [27], and gradient-based distributed optimization [28]. Our work is the first attempt to combine the HE method with a consensus-based approach and to apply it to the P2P energy-trading mechanism. In the proposed scheme, encryption is implemented by the Paillier cryptosystem [38].

There are two security goals for the privacy-friendly P2P computational framework. The first is to protect individual private information  $F_{nm}^k = \{E_{nm}^k, \lambda_{nm}^k\}$  from attacks and acquisition by malicious neighboring agents. The second task is to guarantee that the third party (not the agents) follows the energy price update rules (17) during operation.

##### 4.1. Collusion Attack

To perform C + I updates, a minimum amount of information must be exchanged. At each iteration of the process, the set  $F_{nm}^k$  of information sent from one agent  $n \in \Omega$  to a neighboring agent  $m \in \omega_n$  at iteration  $k$  must be the following:

$$F_{nm}^k = \{E_{nm}^k, \lambda_{nm}^k\} \quad (20)$$



The internal production/consumption parameters  $(a_n, b_n, \bar{E}_n, \underline{E}_n)$  of all agents need not be shared to achieve optimality.

However, this mechanism cannot protect individual privacy. Consider a specific scenario in which the neighboring agents of agent  $n$  conspire to obtain the internal production/consumption parameters of agent  $n$ , as shown in Figure 3a. We will introduce two attack strategies to derive the parameters  $(\bar{E}_n, \underline{E}_n)$  and  $(a_n, b_n)$ , respectively.

1. If agent  $n$  is a producer, all neighboring agents (consumers) can intentionally increase the purchase price  $\lambda_{mn}$  little by little until  $E_{nm}$  remains unchanged between two iterations. In this case, the output of agent  $n$  has reached the upper bound  $\bar{E}_n$ . After that, all neighboring agents can communicate with each other to sum all  $E_{nm}$  and obtain the private information  $\bar{E}_n$ . Similarly, a group of malicious neighboring agents can cooperatively lower the purchase price to obtain the lower bound  $\underline{E}_n$ .
2. Since the neighboring agents of agent  $n$  have received the information about the power boundaries, the group of neighbors for the power update (13) can construct a set  $\lambda_n$  such that the output does not reach  $(\bar{E}_n, \underline{E}_n)$ , (means  $\bar{\mu}_n = \underline{\mu}_n = 0$ ). Under this construction, the update (13) can be simplified as follows:

$$E_n^{k+1} = \frac{\lambda_{nm}^k - \sum_{n,m \in i,j}^l \varphi_{ij}^l (\bar{v}_i^k - \underline{v}_i^k) - b_n}{a_n}, \quad \forall m \in \omega_n \tag{21}$$

By substituting two iteration results  $\{\lambda_n^k, E_n^{k+1}\}, \{\lambda_n^{k+1}, E_n^{k+2}\}$  (where  $E_n$  can be obtained by summing up all  $E_{nm}$ ) into (21),  $a_n$  can be solved by randomly choosing a trade with neighbor  $m$ , as below:

$$a_n = \frac{(\lambda_{nm}^{k+1} - \lambda_{nm}^k) - (V_{nm}^{k+1} - V_{nm}^k)}{(E_n^{k+1} - E_n^k)} \tag{22}$$

where  $V_{nm}^k = \sum_{n,m \in i,j}^l \varphi_{ij}^l (\bar{v}_i^k - \underline{v}_i^k)$ , and this is all public information. After obtaining  $a_n, b_n$  can be readily calculated by (21).

Thus, although very little information needs to be shared in C + I updates, there is still the risk of loss of privacy in the event of a clandestine attack by a group of malicious neighboring agents. There is a need to develop a privacy-protection mechanism for P2P negotiations between agents.

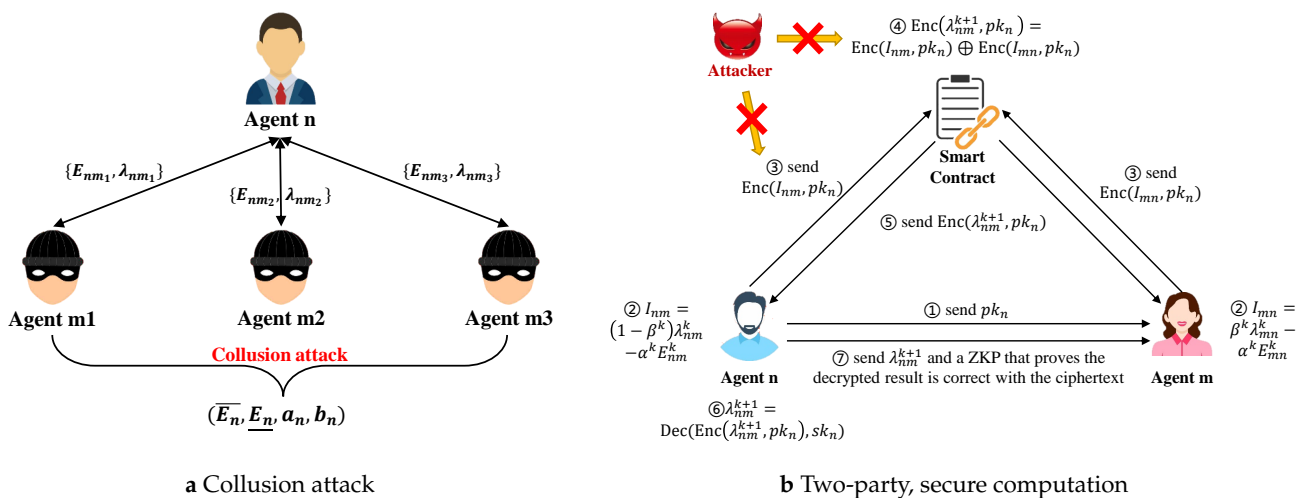


Figure 3. Collusion attack for malicious neighboring agents and two-party, secure computation between two agents.

#### 4.2. Homomorphic Encryption/Decryption Mechanism

The Paillier algorithm implementation scheme is detailed below [39].

*Key generation:* Two prime numbers  $p$  and  $q$  are randomly chosen to satisfy  $\gcd(pq, (p-1)(q-1)) = 1$ , where  $\gcd$  stands for the greatest common divisor. Then,  $N = p * q$  and  $\lambda = \text{lcm}(p-1, q-1)$  are founded, where  $\text{lcm}$  stands for the least common multiple. We randomly pick  $g \in Z_{N^2}^*$  to satisfy  $\gcd(L(g^\lambda \bmod N^2), N) = 1$  and ensure there exists

$$\mu = (L(g^\lambda \bmod N^2))^{-1} \bmod N \quad (23)$$

where  $L(x) = \frac{x-1}{N}$ . The public key is found as  $N, g$ , and the private key is found as  $\lambda, \mu$ .

*Encryption Function (Enc):* Let the plaintext message be  $m \in Z_N$  and the public key be  $pk$ ; then, the encrypting function is

$$\text{Enc}(m, pk) = g^m \cdot r^N \bmod N^2 \quad (24)$$

where  $r$  is a random pad  $r \in Z_{N^2}^*$ .

*Decryption Function (Dec):* Let the ciphertext be  $c$  and the secret key be  $sk$ , the plaintext can be computed as follows:

$$m = \text{Dec}(c, sk) = \frac{L(c^\lambda \bmod N^2)}{L(g^\lambda \bmod N^2)} \bmod N = L(c^\lambda \bmod N^2) * \mu \bmod N. \quad (25)$$

**Property 1. (Additive Homomorphic):** *The additive homomorphic property allows the user to operate the message in its ciphertext directly. Assume the two plaintexts are  $m_1, m_2$  and the key pair is  $sk_i, pk_i$ ; then, we have*

$$\begin{aligned} c_1 &= \text{Enc}(m_1, pk_i) \equiv g^{m_1} \cdot r_1^N \bmod N^2 \\ c_2 &= \text{Enc}(m_2, pk_i) \equiv g^{m_2} \cdot r_2^N \bmod N^2 \end{aligned} \quad (26)$$

Obviously, we have  $c_1 * c_2 \equiv g^{m_1+m_2} \cdot (r_1 \cdot r_2)^N \bmod N^2$ ; thus, we can conclude that

$$m_1 + m_2 \bmod N = \text{Dec}(\text{Enc}(m_1, pk_i) \oplus \text{Enc}(m_2, pk_i), sk_i) = \text{Dec}(c_1 * c_2, sk_i). \quad (27)$$

**Property 2. (Non-Deterministic):** *The non-deterministic means that a given plaintext can be encrypted into a very large set of possible ciphertexts. This property prevents an adversary from associating ciphertext with observed information.*

#### 4.3. Two-Party, Secure Computation

A privacy-preserving, two-party, secure computation framework is designed using HE, ZKP, and SC, as shown in Figure 3b. Before submitting the transaction data to SC, the agents use the public keys generated by the Paillier encryption algorithm to encrypt the aggregated transaction data. The data are in the form of ciphertext, which does not reveal any private information of the agents even if an attacker obtains it. The result of the ciphertext operation matches the result of the plaintext operation. Compared to standard public key encryption, it is the simpler method with the same result, but there is no guarantee that agent  $n$  follows the rules to compute  $\lambda_{nm}^{k+1}$ . Agent  $n$  can increase  $\lambda_{nm}^{k+1}$  to make more profit but runs the risk of not offering enough goods in real time. The combination of HE and SC costs more computational resources but can guarantee the update of energy prices, fend off privacy attacks, and restore the computation result to the blockchain for verification.

Looking at the update steps, only the energy price update (17) will use the information  $F_{nm}^k = \{E_{nm}^k, \lambda_{nm}^k\}$  received from neighbor  $m$ . Thus, the energy price update is implemented by the Paillier encryption algorithm since it satisfies additive homomorphic. The HE-based secure two-party computation algorithm is described below.

- Agent  $n$  generates an individual public key  $pk_n$  and a secret key  $sk_n$ . The public key is sent to agent  $m$  for encryption.
- Agent  $n$  performs an aggregation operation  $I_{nm} = (1 - \beta^k)\lambda_{nm}^k - \alpha^k E_{nm}^k$ , and an encryption  $Enc(I_{nm}, pk_n)$  is sent to SC on Agent  $n$ .
- Agent  $m$  also first performs an aggregation operation  $I_{mn} = \beta^k \lambda_{mn}^k - \alpha^k E_{mn}^k$  and an encryption  $Enc(I_{mn}, pk_n)$  using agent  $n$ 's public key and sends it to SC.
- After collecting the information from two agents, SC computes  $Enc(I_{nm}, pk_n) \oplus Enc(I_{mn}, pk_n)$ . From (17), we have  $\lambda_{nm}^{k+1} = I_{nm} + I_{mn}$ . Thus, according to the additive homomorphic encryption property, the result is  $Enc(\lambda_{nm}^{k+1}, pk_n)$ , which will be sent to agent  $n$  and  $m$ .
- Agent  $n$  executes  $Dec(Enc(\lambda_{nm}^{k+1}, pk_n), sk_n)$  to obtain the decryption  $\lambda_{nm}^{k+1}$  and sends it to Agent  $m$ .
- Agent  $n$  generates and sends a ZKP to Agent  $m$  to prove that the plaintext  $\lambda_{nm}^{k+1}$  is correct with the ciphertext  $Enc(\lambda_{nm}^{k+1}, pk_n)$  computed by SC. Details of the construction of the ZKP are provided in Appendix A.

**Remark 1.** Another challenge is to verify the authenticity of the message  $Enc(I_{nm}, pk_n)$ . To solve this problem, we can take advantage of digital signatures. Agent  $n$  first uses a one-way hash function to obtain a 128-bit digest  $H(Enc(I_{nm}, pk_n))$  and then encrypts the digest with its private key to obtain the encrypted digest  $D_n = Enc(H(Enc(I_{nm}, pk_n)), sk_n)$ . The message  $Enc(I_{nm}, pk_n)$ , the encrypted digest  $D_n$ , and the public key  $pk_n$  are packed and sent to SC. SC verifies the authenticity of the message by checking that the digest of the message processed by the hash function matches the decryption of the received encrypted digest with the public key, i.e.,

$$H(Enc(I_{nm}, pk_n)) \stackrel{?}{=} Dec(D_n, pk_n) \quad (28)$$

#### 4.4. Security and Privacy Analysis

Given the two security goals, to achieve the first goal, we first perform an information aggregation operation for agent  $n$  and  $m$ , respectively ( $I_{nm} = (1 - \beta^k)\lambda_{nm}^k - \alpha^k E_{nm}^k$  and  $I_{mn} = \beta^k \lambda_{mn}^k - \alpha^k E_{mn}^k$ ). By using aggregation operations, even if attackers obtain the information, they cannot reveal the original information. Then, agent  $n$  uses public key  $pk_n$  to encrypt  $I_{nm}$  and sends  $pk_n$  to neighboring agent  $m$  to encrypt  $I_{mn}$ . The information is encrypted with the public key of agent  $n$ , so even if the information is obtained by malicious attackers, the original data cannot be recovered without the private key. The information is encrypted with agent  $n$ 's public key, so it is undeniable that agent  $n$  can recover  $I_{mn}$ . However, agent  $n$  can only obtain the value of  $I_{mn}$ ; there is no way for agent  $n$  to recover the original private information  $\{E_{mn}^k, \lambda_{mn}^k\}$  from  $I_{mn}$  since the aggregation operation is performed locally in agent  $m$ .

To achieve the second goal, the third party is traditionally required to provide zero-knowledge proof of the additional operation. However, this can lead to a higher computational cost for generating the proof. In this work, HE ensures that the decryption value of the result of the ciphertext computation is equal to the result of the plaintext computation, and we use secure SC to realize the ciphertext computation  $Enc(I_{nm}, pk_n) \oplus Enc(I_{mn}, pk_n)$ . Thus, the combination of SC and HE can ensure the correctness of the result  $Enc(\lambda_{nm}^{k+1}, pk_n)$ . Moreover, we design a ZKP protocol to prove that the decrypted result is correct with the ciphertext computed by SC using Paillier's algorithm.

Through the above analysis, it is concluded that using a combination of HE, SC, and ZKP to build the two-party secure operation is a very useful and efficient way to satisfy the security goals of P2P energy trading.

## 5. Results

This section presents numerical results for performance evaluation of the proposed privacy-preserving, P2P negotiation mechanism using different case studies. The case studies were conducted on a computer with an Intel Core i7 processor running at 2.90 GHz

and 32 GB RAM. We use Ganache to set up a private Ethereum Homestead blockchain test network. Remote procedure calls via Web3.py/HTTP allow the Python scripts to communicate with the SCs. The Solidity language is used to develop the SCs, which is a special language for SCs on Ethereum.

### 5.1. Simulation Setup

For illustration and discussion, a small distribution network with seven agents is considered as in [33]. The convergence performance, line congestion management, and encryption algorithm performance are shown in Figures 4–6, respectively. Then, we investigate the impact of the number of agents on convergence performance, as measured by the number of iterations and computation time, and the results are shown in Figure 7. The results verify that our proposed mechanism is feasible for networks with a large number of agents. For line congestion management, the verification results in networks with 13 nodes are sufficient to prove the feasibility of the proposed mechanism in large networks. Finally, regarding the performance of the encryption algorithm, increasing the number of nodes has little impact on the computational performance since the method is used for the negotiation process between two agents.

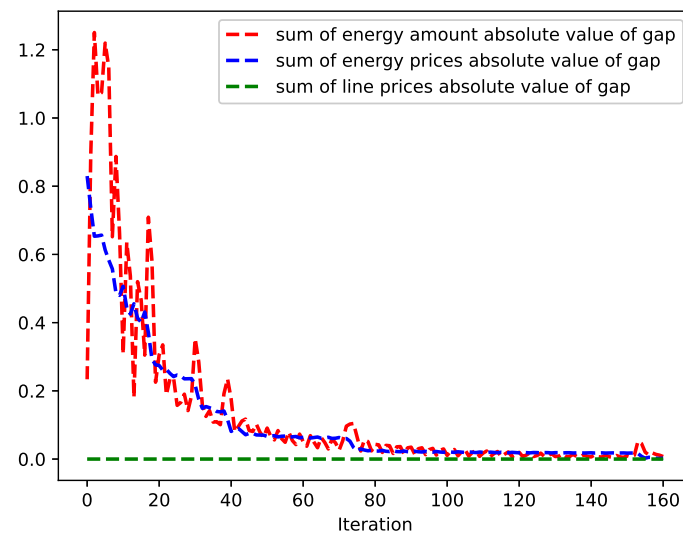


Figure 4. Convergence of the algorithm.

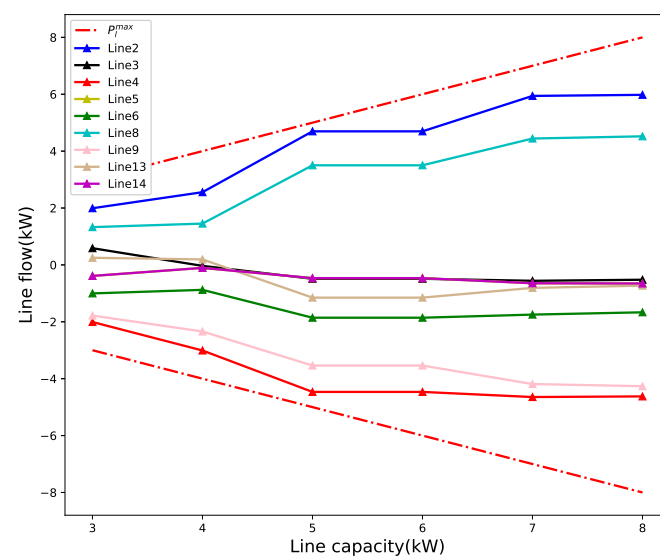
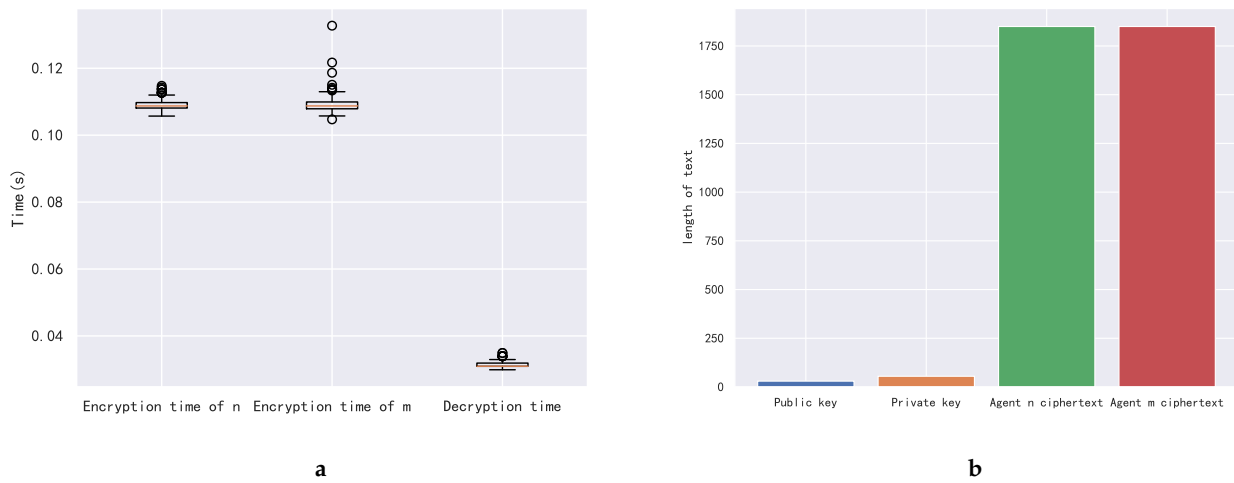
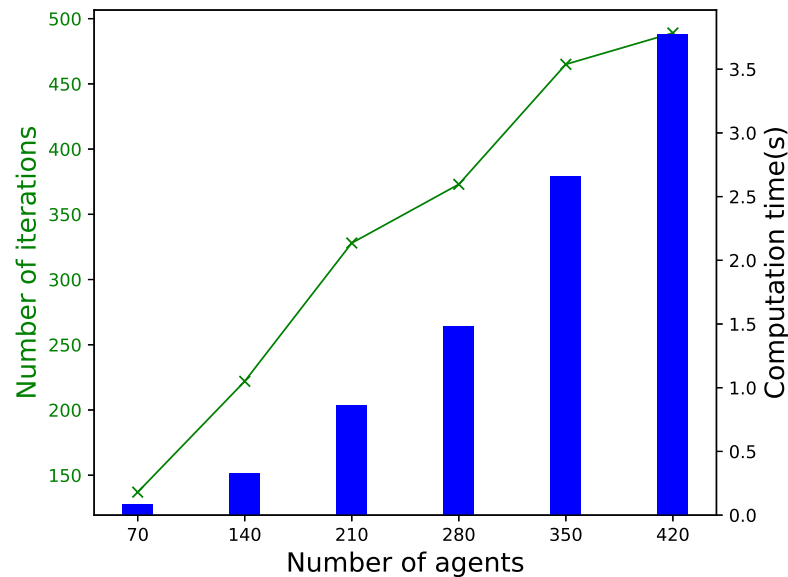


Figure 5. Power flow in different lines under different line capacities.



**Figure 6.** Encryption algorithm computation performance. (a) Agent encryption and decryption time. (b) The size of the public/private keys and ciphertext.



**Figure 7.** Impact of number of agents on computation time and number of iterations for convergence.

There are seven agents in the power network, consisting of four sellers and three buyers. The test system is a 13-node network, as shown in Figure 8. The sellers are located at buses 2, 5, 8, and 10, and the buyers are located at buses 3, 4, and 9. Bus 1 is the reference bus. The connections indicate the physical electrical connections, and the communication network is assumed to have a connected network for the communication of all agents. The parameters of sellers and buyers are listed in Table 1. We set the susceptance of each branch to  $b_1 = b_2 = \dots = b_L = 10\text{s}$ . All stopping criteria  $\chi$  are set to  $10^{-4}$ . The tuning parameters are chosen as follows:

$$\delta^k = 0.1, \quad \beta^k = \frac{0.1}{k^{0.1}}, \quad \alpha^k = \frac{0.1}{k^{0.01}}, \quad \phi^k = 10 \quad (29)$$

and the stopping criteria are set to

$$\chi^E = 0.01, \quad \chi^\lambda = 0.01, \quad \chi^v = 0.01 \quad (30)$$

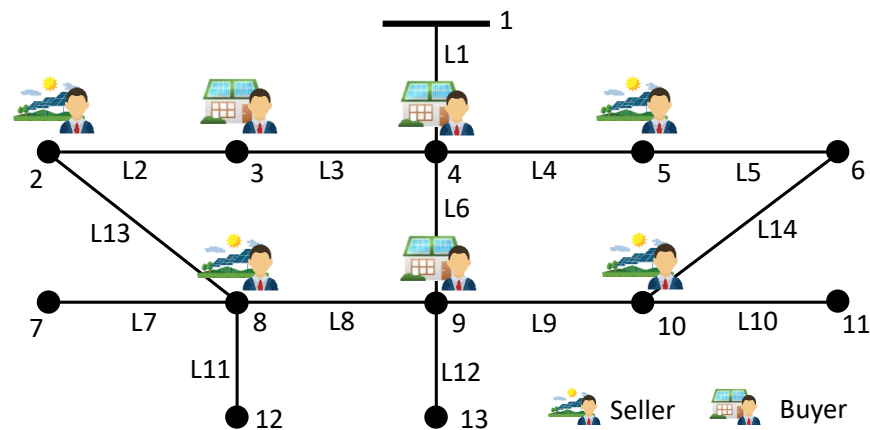


Figure 8. Test system schematic.

Table 1. Sellers' and buyers' parameters of a simple case study.

| Agent | Bus | $a_n$ (USD/kW <sup>2</sup> ) | $b_n$ (USD/kW) | $E_n$ (kW) | $\bar{E}_n$ (kW) |
|-------|-----|------------------------------|----------------|------------|------------------|
| S1    | 2   | 0.04                         | 1              | 1          | 7                |
| S2    | 5   | 0.046                        | 1              | 1          | 4                |
| S3    | 8   | 0.04                         | 1              | 1          | 6                |
| S4    | 10  | 0.05                         | 1              | 1          | 5                |
| B1    | 3   | 0.05                         | 3              | −7         | −1               |
| B2    | 4   | 0.056                        | 3              | −6         | −1               |
| B3    | 9   | 0.05                         | 3              | −8         | −1               |

### 5.2. Convergence Performance of the Negotiation Mechanism

In this case study, the maximum line capacity  $P_l^{max}$  for all lines is set to 10. The convergence process of the algorithm is shown in Figure 4, from which it can be seen that all trading between sellers and buyers converges after about 160 iterations. Although the consensus-based algorithm requires a minimum amount of information to be exchanged, the main drawback is that the number of iterations to converge can be higher than other methods. It can be seen that the sum of the absolute values of the gap of energy quantity and prices decreases with oscillation, while the sum of the absolute values of the gap of line prices remains at zero since no line is congested. The final traded energy quantities and prices are shown in Table 2. It is noticeable that the results of S1 and S3 are the same because their parameters  $a_n$  and  $b_n$  are the same. For B1 and B3, the purchase prices are the same, but the quantities of B3 are higher because the demand of B3 is higher ( $-8 < -7$ ).

Table 2. Final traded quantities and prices of energy.

|    | B1                  | B2                  | B3                  |
|----|---------------------|---------------------|---------------------|
| S1 | 1.75 kW/1.21 USD/kW | 1.50 kW/1.19 USD/kW | 2.00 kW/1.24 USD/kW |
| S2 | 1.33 kW/2.74 USD/kW | 1.27 kW/2.72 USD/kW | 1.37 kW/2.74 USD/kW |
| S3 | 1.75 kW/1.21 USD/kW | 1.50 kW/1.19 USD/kW | 2.00 kW/1.24 USD/kW |
| S4 | 1.67 kW/2.66 USD/kW | 1.50 kW/1.24 USD/kW | 1.75 kW/2.65 USD/kW |

### 5.3. Performance of Line Congestion Management

The impact of line capacity limit on power flow is investigated. The maximum line capacity for these lines ranges from 3 to 8 kW. In the test system, the results are shown only for lines with non-zero power flow. The results are shown in Figure 5, and it is confirmed that the power flows in these lines are always within the maximum line capacity, which

means that the proposed algorithm can meet the line flow constraints in the P2P power grid. If there is a congested line in the network, agents will avoid trading over the congested lines because they have to pay additional network charges

#### 5.4. Performance of Scalability

In the real world, the P2P energy trading mechanism will be used in power networks with a large number of agents, and the number of transactions will be significant. The computation time and the number of iterations are two key factors that measure the scalability of the mechanism. To demonstrate the scalability of our mechanism, we add more agents to each bus. The parameters of the agents are chosen randomly, while the tuning parameters ( $\delta^k$ ,  $\beta^k$ , and  $\alpha^k$ ) are carefully designed for tolerable performance. The line capacity is chosen large enough to make no congestion happens. Figure 7 shows the effects of the number of agents (between 70 and 420) on the two factors. It can be seen that both the computation time and the number of iterations increase approximately linearly with the number of agents. The performance of computational time is excellent (under 4 s for 420 agents), but more iterations (almost 450) cost. The results show that our proposed mechanism is feasible for networks with a large number of agents.

#### 5.5. Encryption Algorithm Computation Performance Analysis

In this section, we analyze the trade-off between privacy and computational cost. In the original decentralized negotiation mechanism, where no homomorphic encryption is applied, the computation time of each agent for each iteration is so small that it is negligible. To ensure privacy, a privacy-preserving mechanism based on homomorphic encryption is proposed to be used at each iteration. Agents need to encrypt their private information and to submit it to SC to perform ciphertext computation. The Paillier homomorphic encryption used in the simulation comes from the phe (Partially Homomorphic Encryption) library in Python. Figure 6a shows the encryption and decryption time of the agents. The encryption time of agent  $n$  and  $m$  is close to each other and is about 0.11 s. The decryption time is much lower compared to the encryption time and is about 0.03 s. Figure 6b shows the public/private key and the size of the ciphertext. The size of the ciphertext is slightly larger than 1750, while the public/private keys are much smaller.

#### 5.6. Computational Performance under Different Mechanisms

In this section, we investigate the computational performance under four different mechanisms. (1) P2P trading is performed without a privacy-preserving mechanism. (2) P2P trading runs under the Paillier HE mechanism. The agents each encrypt their bid information  $\{\lambda_{nm}, E_{nm}\}$  and send it to a program to perform cipher computation. (3) P2P trading runs under the two-party, secure computation mechanism without ZKP. (4) P2P trading runs under the two-party, secure computation mechanism with ZKP. The computation time for each agent in one iteration is displayed in Figure 9. It can be seen that the computation efficiency is very high without any privacy mechanism. The time spent on the second mechanism is higher than for the third because more information needs to be encrypted, which is very time-consuming. The efficiency of the third mechanism is at a medium level and is acceptable. The agents only need to encrypt the aggregated information  $\{I_{nm}, I_{mn}\}$ , which can greatly reduce the time consumption. Finally, the fourth mechanism is the most ineffective one because the ZKP protocol is very time-consuming and, most of the time, is for computing the inverse element by the expand Euclid algorithm ( $M = N^{-1} \bmod \phi(N)$ ). This problem will be studied in our future work.





## 6. Discussion

The most valuable achievement of our proposed mechanism is to provide a privacy-preserving, two-party, secure computation mechanism for the P2P negotiation mechanism between each pair of agents. The agents cannot know each other's actual bidding information. However, operational efficiency has been sacrificed for privacy protection. A lot of time and computing power are spent on encrypting and decrypting information. In addition, the introduction of SC further extends the time to achieve convergence.

Therefore, our future work will mainly focus on how to increase the computational efficiency under the privacy-friendly mechanism. The first way is to develop a P2P negotiation mechanism that uses a more efficient decentralized optimization algorithm. For example, the consensus ADMM algorithm [31,32], which can guarantee convergence with a smaller number of iterations. The challenge is to combine the consensus ADMM with the HE mechanism. Another way to increase efficiency is to reduce the amount of information to be encrypted or protected. As we analyzed in Section 3.1, in the C + I method, private information is revealed and disclosed only in the collusion attack by all neighboring agents. If we carefully select a part of the exchanged information to be encrypted, the private information can also be protected. We can perform the two-party secure computation with only one neighboring agent, and that is enough to protect private information from attacks. With this strategy, the computation cost can be reduced from  $O(N^2\Delta T)$  to  $O(N\Delta T)$ , where  $\Delta T$  is the sum of the encryption and decryption time of the two-party secure computation.

## 7. Conclusions

In the P2P energy market, agents must exchange a large amount of information to reach consensus on the final trade. However, this fully decentralized negotiation may lead to the disclosure of private information. In this paper, we propose a privacy-preserving, two-party, secure computation mechanism for P2P energy trading that leverages many technologies. We first design a P2P negotiation mechanism based on the C + I method and the PTDF model. This mechanism can maximize social welfare while satisfying the physical line flow constraints. Then, for this mechanism, we analyze the two collusion attack strategies to obtain private information from a group of malicious neighboring agents. To protect against this kind of attacks, a two-party, secure computation mechanism is proposed for each pair of agents to update the energy prices. The agents first aggregate their bid price and bid quantity and then encrypt the information with the public key generated by the Paillier algorithm. Then, the computation of the ciphertext is automatically performed by SC, and the correctness of the decryption is proved by a ZKP protocol. The simulation results demonstrate the performance of convergence, line congestion management, scalability, computation efficiency, and SC operations.

**Author Contributions:** Conceptualization, Z.L. and Z.G.; Methodology, Z.L. and Z.G.; Validation, F.Z.; Investigation, B.Z.; Software, M.X.; Writing—original draft, Z.L. and H.X.; Writing—review and editing, Z.G.; Project administration, H.X.; Funding acquisition, Z.L. and Z.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the 2020 Industrial Internet Innovation and Development Project—For the Power Industry Industrial Internet Network Trust Support Platform Project (grant number: JL71-20-017), the Populus Euphratica Found grand number (grant number: CCF-HuaweiBC2021009), and the Open Research Fund of Key Laboratory of Cryptography of Zhejiang Province (grand number: ZCL21007).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Nomenclature

|                                      |   |
|--------------------------------------|---|
| $C_n(\cdot)$                         | Production cost or utility function of agent $n$                          |
| $i, j$                               | Indices for buses   |
| $n, m$                               | Indices for agents  |
| $l$                                  | Index for power lines   |
| $\underline{E}, \bar{E}$             | Boundaries of power   |
| $a_n, b_n, c_n$                      | Coefficients of the quadratic function of agent $n$                       |
| $\mathcal{L}$                        | Set of lines  |
| $\mathcal{N}$                        | Set of buses  |
| $\Omega$                             | Set of agents   |
| $\Omega_p$                           | Set of energy producers   |
| $\Omega_c$                           | Set of energy consumers   |
| $\omega$                             | Set of neighboring agents   |
| $\lambda_{nm}$                       | Energy prices provided by $n$ to $m$                                      |
| $E_n$                                | Power injection or total traded quantity of agent $n$                     |
| $E_{nm}$                             | Traded energy quantity from $n$ to $m$                                    |
| $P_l$                                | Power flow of line $l$  |
| $\varphi_{ij}$                       | Power transfer distribution factor of line $l$ connecting bus $i$ and $j$ |
| $\psi_i^l$                           | Injection shift factor in line $l$ for bus $i$                            |
| $A$                                  | Branch to node incidence matrix   |
| $B'$                                 | Diagonal branch susceptance matrix  |
| $C$                                  | Reduced nodal susceptance matrix  |
| $P_l^{max}$                          | Maximum capacity of line $l$  |
| $(\bar{v}_l, \underline{v}_l)$       | Upper bound and lower bound prices of $l$                                 |
| $(\bar{\mu}_n, \underline{\mu}_n^k)$ | Dual variables for power boundaries                                       |
| $f_{nm}$                             | Asymptotically proportional factor  |
| $(\alpha^k, \beta^k)$                | Sequences of positive factors at iteration $k$                            |
| $\phi^k$                             | Tuning parameter  |
| $\chi^E, \chi^\lambda, \chi^v$       | Stopping criterion  |
| $I_{nm}$                             | Aggregation Information   |
| $pk_n, sk_n$                         | Public key and secret key of agent $n$                                    |

## Appendix A

The problem is how agent  $n$  can prove that the decrypted result  $\lambda_{nm}^{k+1}$  is correct with the ciphertext  $Enc(\lambda_{nm}^{k+1}, pk_n)$  computed by SC. This can be carried out using a zero-knowledge proof to prove that a Paillier ciphertext is an encryption of zero. For simplicity, let  $c = Enc(\lambda_{nm}^{k+1}, pk_n)$  be the original ciphertext, and let  $d$  be the decryption that agent  $n$  sends to agent  $m$ . Then, both  $n$  and  $m$  can each locally use the homomorphic property to compute a ciphertext  $c'$  equal to the value of  $c$  minus the encrypted  $d$ , i.e.,  $c' = c - E(d)$ . Note: If  $c$  is an encryption of  $d$ ; then,  $c'$  is an encryption of zero, since  $c' = E(d) - E(d) = E(d - d) = E(0)$ .

Thus, it suffices for  $n$  with zero knowledge to prove that  $c'$  is an encryption of zero (or, put another way, that  $c' = r^N \bmod N^2$ ). This can be carried out very efficiently using an improved method described in Section 5.2 of “A Generalization of Paillier’s Public-Key System with Applications to Electronic Voting” by Damgard and Jurik [40]. In this method, the inverse element is computed using the expand Euclid algorithm. The protocol is described in detail in Algorithm A1, and for a detailed proof, we refer to Lemma 3 in Section 5.2 of [40].

**Algorithm A1:** Protocol for proving  $c'$  is an encryption of zero**Input:**  $N = p * q, c'$ 

- 1 Prover  $P$  calculate:  $M = N^{-1} \bmod \phi(N)$  and  $r = c'^M \bmod N$  such that  $c' = E(0, r)$
- 2  $P$  chooses  $v$  at random in  $Z_{N^2}^*$  and sends  $a = E(0, v)$  to Verifier  $V$
- 3  $V$  chooses  $e$ , a random  $t$  bit number, and sends  $e$  to  $P$ .
- 4  $P$  sends  $z = vr^e \bmod N$  to  $V$ .
- 5  $V$  checks that  $c', a, z$  are prime to  $N$  and  $E(0, z) = ac^{le} \bmod N^2$ , and accepts if and only if this is the case.

**References**

1. Hussain, S.; Kim, Y.C. Fault resilient communication network architecture for monitoring and control of wind power farms. In Proceedings of the 2016 18th International Conference on Advanced Communication Technology (ICACT), PyeongChang, Republic of Korea, 31 January–3 February 2016; pp. 685–692.
2. Hussain, S.; Thakur, S.; Shukla, S.; Breslin, J.G.; Jan, Q.; Khan, F.; Kim, Y.S. A two-layer decentralized charging approach for residential electric vehicles based on fuzzy data fusion. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 7391–7405. [[CrossRef](#)]
3. Hussain, S.; Mohammad, F.; Kim, Y.C. Communication network architecture based on logical nodes for electric vehicles. In Proceedings of the 2017 International Symposium on Information Technology Convergence, Shijiazhuang, China, 18–20 October 2017; pp. 19–21.
4. Hussain, S.; Kim, Y.S.; Thakur, S.; Breslin, J.G. Optimization of waiting time for electric vehicles using a fuzzy inference system. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 15396–15407. [[CrossRef](#)]
5. Cao, Y.; Kaiwartya, O.; Zhuang, Y.; Ahmad, N.; Sun, Y.; Lloret, J. A decentralized deadline-driven electric vehicle charging recommendation. *IEEE Syst. J.* **2018**, *13*, 3410–3421. [[CrossRef](#)]
6. Hassan, A.N.; Abdullah, A.H.; Kaiwartya, O.; Cao, Y.; Sheet, D.K. Multi-metric geographic routing for vehicular ad hoc networks. *Wirel. Netw.* **2018**, *24*, 2763–2779. [[CrossRef](#)]
7. Schollmeier, R. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In Proceedings of the First International Conference on Peer-to-Peer Computing, Linköping, Sweden, 27–29 August 2001; pp. 101–102.
8. Tushar, W.; Saha, T.K.; Yuen, C.; Morstyn, T.; McCulloch, M.D.; Poor, H.V.; Wood, K.L. A motivational game-theoretic approach for peer-to-peer energy trading in the smart grid. *Appl. Energy* **2019**, *243*, 10–20. [[CrossRef](#)]
9. Tushar, W.; Chai, B.; Yuen, C.; Smith, D.B.; Wood, K.L.; Yang, Z.; Poor, H.V. Three-party energy management with distributed energy resources in smart grid. *IEEE Trans. Ind. Electron.* **2014**, *62*, 2487–2498. [[CrossRef](#)]
10. Kursawe, K.; Danezis, G.; Kohlweiss, M. Privacy-friendly aggregation for the smart-grid. In *International Symposium on Privacy Enhancing Technologies Symposium*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 175–191.
11. Ács, G.; Castelluccia, C. I have a dream! (differentially private smart metering). In *International Workshop on Information Hiding*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 118–132.
12. Hong, Y.; Liu, W.M.; Wang, L. Privacy preserving smart meter streaming against information leakage of appliance status. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2227–2241. [[CrossRef](#)]
13. Shukla, S.; Thakur, S.; Hussain, S.; Breslin, J.G. A Blockchain-Enabled Fog Computing Model for Peer-To-Peer Energy Trading in Smart Grid. In *International Congress on Blockchain and Applications*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 14–23.
14. Dang, C.; Zhang, J.; Kwong, C.P.; Li, L. Demand side load management for big industrial energy users under blockchain-based peer-to-peer electricity market. *IEEE Trans. Smart Grid* **2019**, *10*, 6426–6435. [[CrossRef](#)]
15. Luo, F.; Dong, Z.Y.; Liang, G.; Murata, J.; Xu, Z. A distributed electricity trading system in active distribution networks based on multi-agent coalition and blockchain. *IEEE Trans. Power Syst.* **2018**, *34*, 4097–4108. [[CrossRef](#)]
16. Kang, J.; Yu, R.; Huang, X.; Maharjan, S.; Zhang, Y.; Hossain, E. Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains. *IEEE Trans. Ind. Inform.* **2017**, *13*, 3154–3164. [[CrossRef](#)]
17. Yang, X.; Wang, G.; He, H.; Lu, J.; Zhang, Y. Automated demand response framework in ELNs: Decentralized scheduling and smart contract. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *50*, 58–72. [[CrossRef](#)]
18. Wang, S.; Taha, A.F.; Wang, J.; Kvaternik, K.; Hahn, A. Energy crowdsourcing and peer-to-peer energy trading in blockchain-enabled smart grids. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1612–1623. [[CrossRef](#)]
19. Han, D.; Zhang, C.; Ping, J.; Yan, Z. Smart contract architecture for decentralized energy trading and management based on blockchains. *Energy* **2020**, *199*, 117417. [[CrossRef](#)]
20. AlSkaif, T.; Crespo-Vazquez, J.L.; Sekuloski, M.; van Leeuwen, G.; Catalão, J.P. Blockchain-based fully peer-to-peer energy trading strategies for residential energy systems. *IEEE Trans. Ind. Inform.* **2021**, *18*, 231–241. [[CrossRef](#)]
21. Zhang, M.; Eliassen, F.; Taherkordi, A.; Jacobsen, H.A.; Chung, H.M.; Zhang, Y. Demand-Response Games for Peer-to-Peer Energy Trading With the Hyperledger Blockchain. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *52*, 19–31. [[CrossRef](#)]
22. AlAshery, M.K.; Yi, Z.; Shi, D.; Lu, X.; Xu, C.; Wang, Z.; Qiao, W. A blockchain-enabled multi-settlement quasi-ideal peer-to-peer trading framework. *IEEE Trans. Smart Grid* **2020**, *12*, 885–896. [[CrossRef](#)]

23. Li, Y.; Yang, W.; He, P.; Chen, C.; Wang, X. Design and management of a distributed hybrid energy system through smart contract and blockchain. *Appl. Energy* **2019**, *248*, 390–405. [[CrossRef](#)]
24. Acar, A.; Aksu, H.; Uluagac, A.S.; Conti, M. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 1–35. [[CrossRef](#)]
25. Cheng, Z.; Ye, F.; Cao, X.; Chow, M.Y. A homomorphic encryption-based private collaborative distributed energy management system. *IEEE Trans. Smart Grid* **2021**, *12*, 5233–5243. [[CrossRef](#)]
26. Yi, H.; Lin, W.; Huang, X.; Cai, X.; Chi, R.; Nie, Z. Energy trading IoT system based on blockchain. *Swarm Evol. Comput.* **2021**, *64*, 100891. [[CrossRef](#)]
27. Liu, N.; Wang, C.; Cheng, M.; Wang, J. A privacy-preserving distributed optimal scheduling for interconnected microgrids. *Energies* **2016**, *9*, 1031. [[CrossRef](#)]
28. Lu, Y.; Zhu, M. Privacy preserving distributed optimization using homomorphic encryption. *Automatica* **2018**, *96*, 314–325. [[CrossRef](#)]
29. Lu, Y.; Lian, J.; Zhu, M. Privacy-preserving transactive energy system. In Proceedings of the 2020 American Control Conference (ACC), Denver, CO, USA, 1–3 July 2020; pp. 3005–3010.
30. Day, R.H. Rational choice and economic behavior. *Theory Decis.* **1971**, *1*, 229–251. [[CrossRef](#)]
31. Guo, Z.; Pinson, P.; Chen, S.; Yang, Q.; Yang, Z. Chance-constrained peer-to-peer joint energy and reserve market considering renewable generation uncertainty. *IEEE Trans. Smart Grid* **2020**, *12*, 798–809. [[CrossRef](#)]
32. Guo, Z.; Pinson, P.; Chen, S.; Yang, Q.; Yang, Z. Online optimization for real-time peer-to-peer electricity market mechanisms. *IEEE Trans. Smart Grid* **2021**, *12*, 4151–4163. [[CrossRef](#)]
33. Khorasany, M.; Mishra, Y.; Ledwich, G. A decentralized bilateral energy trading system for peer-to-peer electricity markets. *IEEE Trans. Ind. Electron.* **2019**, *67*, 4646–4657. [[CrossRef](#)]
34. Liu, M.; Gross, G. Role of distribution factors in congestion revenue rights applications. *IEEE Trans. Power Syst.* **2004**, *19*, 802–810. [[CrossRef](#)]
35. Sorin, E.; Bobo, L.; Pinson, P. Consensus-based approach to peer-to-peer electricity markets with product differentiation. *IEEE Trans. Power Syst.* **2018**, *34*, 994–1004. [[CrossRef](#)]
36. Baroche, T.; Pinson, P.; Latimier, R.L.G.; Ahmed, H.B. Exogenous cost allocation in peer-to-peer electricity markets. *IEEE Trans. Power Syst.* **2019**, *34*, 2553–2564. [[CrossRef](#)]
37. Moret, F.; Pinson, P. Energy collectives: A community and fairness based approach to future electricity markets. *IEEE Trans. Power Syst.* **2019**, *34*, 3994–4004. [[CrossRef](#)]
38. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 223–238.
39. Ma, Y.; Qiu, J.; Sun, X.; Tao, Y. A Multi-Stage Information Protection Scheme for CDA-Based Energy Trading Market in Smart Grids. *IEEE Trans. Smart Grid* **2021**, *13*, 2305–2317. [[CrossRef](#)]
40. Damgård, I.; Jurik, M.; Nielsen, J.B. A generalization of Paillier’s public-key system with applications to electronic voting. *Int. J. Inf. Secur.* **2010**, *9*, 371–385. [[CrossRef](#)]