symmetry

MDPI

Article

A Privacy-Preserving Consensus Mechanism for ADMM-Based Peer-to-Peer Energy Trading

Zhihu Li [1,*], Bing Zhao [1], Hongxia Guo [2], Feng Zhai [1] and Lin Li [2]

[1] Department of Metrology, China Electric Power Research Institute, Beijing 100192, China; zhaob@epri.sgcc.com.cn (B.Z.); zhaifeng@epri.sgcc.com.cn (F.Z.)
[2] Marketing Service Center (Metrology Center), State Grid Shandong Electric Power Company, Jinan 250001, China; guohongxia@sd.sgcc.com.cn (H.G.); igginman_rox@163.com (L.L.)
* Correspondence: lizhihu@epri.sgcc.com.cn

Abstract: In the electricity market, prosumers are becoming more and more prevalent due to the fast development of distributed energy resources and demand response management, which also promote the appearance of peer-to-peer (P2P) trading mechanisms for energy. Optimization-based methods are efficient tools to design the P2P energy trading negotiation mechanism. However, the main drawback for market mechanisms based on optimization methods is that the incentive compatibility cannot be satisfied, which means participants can obtain more profit by providing untruthful biddings. To overcome this challenge, a novel consensus mechanism based on Proof of Solution (PoSo) is proposed for P2P energy trading. The optimization results will be verified by neighboring agents according to the KKT conditions in a fully decentralized and symmetric manner, which means agents will check each other's solutions. However, the verification process may leak the private information of agents, and a privacy-preserving consensus mechanism is designed using Shamir's secret sharing method. After that, we explore a method to realize that trusted agents can recover the right information even under the misbehavior of malicious agents by inheriting the philosophy of Practical Byzantine Fault Tolerance (PBFT). The numerical results demonstrate the effectiveness and efficiency of our proposed consensus mechanisms. In more detail, (1) when the message delivery success rate is not lower than 0.7, the consensus mechanisms almost guarantee success; (2) if the proportion of untrusted agents satisfies $4f + 1 \leq N_{\omega_n}$, the proposed method guarantees the correctness of the consensus verification results; (3) the communication times among agents can be highly reduced by more than 60% by only verifying the optimality of the received results for the first three and last few iterations.

Keywords: P2P energy trading mechanism; consensus alternating ADMM; proof of solution consensus mechanism; Shamir's secret sharing; practical byzantine fault tolerance

Citation: Li, Z.; Zhao, B.; Guo, H.; Zhai, F.; Li, L. A Privacy-Preserving Consensus Mechanism for ADMM-Based Peer-to-Peer Energy Trading. *Symmetry* **2023**, *15*, 1561. https://doi.org/10.3390/sym15081561

Academic Editors: Remigiusz Wiśniewski and Aniruddha Bhattacharjya

Received: 14 June 2023
Revised: 4 July 2023
Accepted: 11 July 2023
Published: 10 August 2023

1. Introduction

The ever-increasing distributed energy resources (DERs) and energy system management (ESM) [1] are changing the approach of power system operation and turning traditional producers and consumers into prosumers. The increase in prosumers means the need for a full decentralized energy trading mechanism that permits prosumers to negotiate and trade with each other freely without a central organization or institute. Thus, the network architecture is also changing to decentralized. The peer-to-peer (P2P) network is defined to be a fully decentralized and symmetric architecture as in [2], in which participants share their resources with each other without the intervention of an intermediary entity [3]. It is in this context, as a new generation of energy management framework, P2P trading mechanism [4] encourages prosumers to participate in the energy market actively.

A traditional distributed energy system mainly adopts centralized optimization, which can directly obtain the optimal energy trading scheduling. However, with the rapid development of renewable energy, a large amount of accessed distributed energy generation

equipment makes the centralized optimization method consume a lot of time and computing resources. As a result, the energy trading mechanism is moving towards a more open and decentralized direction. Compared with centralized optimization, distributed optimization can realize parallel computation while maintaining the optimality of trading results, which is a better method to implement and design a new energy trading mechanism. Ref. [5] used the generalized fast dual ascent method to propose a P2P energy trading mechanism that considers the safety constraints of distribution network; Ref. [6] proposed a primal dual gradient algorithm to clear the energy market in a P2P manner; Ref. [7] improved the method to a primal dual sub-gradient algorithm for P2P trading; Ref. [8] proposed relaxed consensus + innovation (RCI) that aims to solve multilateral and bilateral energy economic dispatch in a completely decentralized manner. All the above researches are based on deriving the first-order function of a Lagrange equation of individual optimization problems, and then updating the variables by using different types of gradient descent methods. The advantage is that the calculation speed of single iteration is fast, but the disadvantage is that convergence requires many iterations and the communication cost is high. Compared with other methods, the Alternating Direction Method of Multipliers (ADMM) [9] is widely used because of its completely distributed architecture, less iterations for convergence and better potential for parallel computing performance. Standard ADMM [10,11] is an efficient tool for the design of P2P energy trading negotiation mechanism, but usually a coordinator is required to help with the convergence; to overcome this drawback to construct a completely decentralized market, Ref. [12–16] improved the standard ADMM to consensus ADMM (CADMM), which has a fully decentralized scheme without a central coordinator.

However, the main challenge of optimization-based methods is encouraging prosumers to cooperate in such an untrusted environment. Most of the optimization-based methods use a locational marginal pricing (LMP) mechanism, which cannot satisfy "incentive compatibility" (Each player can maximize his goal according to his true preferences.) and "market efficiency" (Market efficiency can be maximized when outcomes maximize social welfare.) Market participants can exercise "market power" by offering untruthful prices and quantities. When profit maximizing producers adopt strategies to manipulate prices, LMP may cause a large loss in economic efficiency. To address this challenge, the existing P2P energy trading mechanisms mainly adopt the VCG mechanism [17–19] to satisfy incentive compatibility. The VCG mechanism induces truth-telling behavior in a dominant strategy equilibrium, that is, it is profit-maximizing for each player to reveal his true marginal cost, no matter what action the other players take. However, the main problem is that it still cannot satisfy the property of revenue adequacy, which refers to a condition in which the market operator never incurs a financial deficit. Thus, we are trying to find another method to build an incentive-compatible P2P electricity market.

Fortunately, we find that consensus mechanisms in the blockchain are efficient tools to ensure the correctness and authenticity of submitted results. It defines how parties can agree on the state and behavior in a system. The best known and most widely used consensus mechanism is Proof of Work (PoW) [20], which is mostly used in Bitcoin and Ethereum. However, PoW is also known for wasting a lot of time and effort to solve a difficult but meaningless mathematical puzzle. The criticism of PoW has led to other consensus mechanisms such as Proof of Stake (PoS) [21], Practical Byzantine Fault Tolerance (PBFT) [22] and Delegated Byzantine Fault Tolerance (DBFT) [23]. These consensus mechanisms enable various applications in finance, supply chain management and energy [24–26].

However, existing consensus mechanisms are not able to support optimization methods and problems, and it is challenging to design a suitable and efficient consensus mechanism for an optimization-based P2P energy trading mechanism. In this work, a novel consensus mechanism on the basis of Proof of Solution (PoSo) is proposed [27], which simulates PoW by replacing meaningless mathematical puzzles with meaningful optimization problems. Additionally, we further improve the consensus mechanism by reducing the

number of verification and implement privacy protection. Compared with previous works, the main contributions are listed below.

- First of all, a negotiation mechanism using consensus ADMM is designed for P2P energy trading. Social welfare can be maximized in a fully decentralized manner. After that, a PoSo-inspired consensus mechanism is developed for this P2P energy trading mechanism. After solving individual local optimization problems to obtain an optimal solution, agents will broadcast the equivalent accumulated Karush–Kuhn–Tucker (KKT) conditions to neighbor agents for optimality validation. It is worth noting that, similar to PoW, checking whether a given solution satisfies the KKT condition for optimality is much easier than obtaining the solution. Compared to the original work, the biggest contribution of the proposed P2P energy trading consensus mechanism is that no central delegate is required, and agents do not need to share their private information with a central leader. In addition, agents who fail in validation are subject to a significant penalty attached to the objective function. The novel consensus mechanism can overcome the deficiency of the LMP mechanism, and create an incentive compatible P2P energy market.
- However, our proposed consensus mechanism requires a mass of communication time due to many iterations to converge, and the private information cannot be protected. To solve these two problems, first we propose to reduce the number of consensus verification by only verifying the solutions in a few iterations at the beginning and the end. Second, we proposed a privacy-preserving consensus mechanism that is resistant to a proportion of untrusted agents. The private information is divided into encrypted pieces by using Shamir's secret sharing scheme, which will be distributed to corresponding neighboring agents. Then, a PBFT-based method is designed to realize that trusted agents can recover the right results even under the misbehavior of malicious agents who may manipulate the encrypted pieces to cause incorrect results.

The rest of the paper is organized as follows: Section 2 presents related works, Section 3 presents the formulation of P2P energy trading and the negotiation mechanism for the P2P energy market, followed by the consensus mechanism and improvements for it in Section 4. Numerical results are presented in Section 5. Finally, in Sections 6 and 7, discussions, conclusions, and future perspectives are drawn.

2. Related Work

2.1. Consensus ADMM

ADMM is an efficient tool to design distributed and parallel algorithms for machine learning problems at a large scale [9]. It can be applied to convex optimization problems with special equality constraints as follows:

$$\begin{aligned} &\text{minimize } f(x) + g(z) \\ &\text{subject to } Ax + Bz = c. \end{aligned} \tag{1}$$

ADMM solves the above problem based on the augmented Lagrangian function, which is formulated as below:

$$L_\rho(x, y, z) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|^2. \tag{2}$$

Here, y is a dual variable or Lagrangian multiplier for the equality constraint ($Ax + Bz = c$), and ρ is a positive penalty parameter. ADMM obtains optimal solutions by updating the variables in an alternating way as below:

$$x_{t+1} = \arg\min_x L_\rho(x, z_t, y_t) \tag{3a}$$

$$z_{t+1} = \arg\min_z L_\rho(x_{t+1}, z, y_t) \tag{3b}$$

$$y_{t+1} = y_t + \rho(Ax_{t+1} + Bz_{t+1} - c), \tag{3c}$$

With the development of high performance computing, ADMM is an important choice to solve the above constraint problems. At this point, the global consistency optimization strategy comes into play. The objective function is decomposed into N sub-objective functions or subsystems. Each one tries to compute a local solution x_i that is constrained to be equal to the global variable z. The global consensus optimization problem can be formulated as follows:

$$\begin{aligned} &\min \sum_{i=1}^{N} f_i(x_i) \\ &\text{s.t. } x_i = z. \end{aligned} \tag{4}$$

Here, x_i is the local solution for the i-th subsystem, and f_i is the loss function of the i-th subsystem. All local x_i solutions construct the global variable z, and z links and connects the data of the various subsystems. The augmented Lagrangian function of problem (4) is formulated as:

$$L_\rho(x_1, \ldots, x_N, z, y) = \sum_{i=1}^{N} (f_i(x_i) + y^T(x_i - z) + (\rho/2)\|x_i - z\|^2). \tag{5}$$

ADMM addresses it by updating variables as below:

$$(x_i)_{t+1} = \arg\min_{x_i} (f_i(x_i) + (y_i)_t^T(x_i - z_t) + (\rho/2)\|x_i - z_t\|^2) \tag{6a}$$

$$z_{t+1} = \frac{1}{N} \sum_{i=1}^{N} ((x_i)_{t+1} + (1/\rho)(y_i)_t) \tag{6b}$$

$$(y_i)_{t+1} = (y_i)_t + \rho((x_i)_{t+1} - z_{t+1}) \tag{6c}$$

The above equations reveals that the iterative steps of ADMM can be executed in parallel. This algorithm can be further simplified by computing the averages of x and y. The z-update and y-update can be written as

$$z_{t+1} = \overline{x}_{t+1} + (1/\rho)\overline{y}_t \tag{7a}$$

$$\overline{y}_{t+1} = \overline{y}_t + \rho(\overline{x}_{t+1} - z_{t+1}) \tag{7b}$$

Substituting (7a) into (7b) can obtain that $\overline{y}_{t+1} = 0$ and $z_{t+1} = \overline{x}_{t+1}$. Finally, we derive the consensus ADMM, and it can be simply formulated as

$$(x_i)_{t+1} = \arg\min_{x_i} \left(f_i(x_i) + (y_i)_t^T(x_i - \overline{x}_t) + (\rho/2)\|x_i - \overline{x}_t\|^2 \right) \tag{8a}$$

$$(y_i)_{t+1} = (y_i)_t + \rho((x_i)_{t+1} - \overline{x}_{t+1}) \tag{8b}$$

2.2. Consensus Mechanisms for P2P Energy Trading

For traditional blockchain, there are several mature consensus mechanisms. When combined with energy trading, it is also likely to adopt a more traditional mechanism or slightly modified one to meet the requirements of the system. Several common consensus mechanisms, e.g., Proof-of-Work (PoW), Proof-of-Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT) [28], are introduced as follows.

A comparison table, Figure 1, shows the merits and demerits of different consensus mechanisms used in P2P energy trading.

Consensus Mechanism	Idea	Right of accounting	Advantages	Disadvantages
PoW	Based on the computing resources	Through the node's computing resources to capture	1. Easy to implement 2. The security is high and the attack is difficult	1. Huge energy consumption 2. Long confirmation time
PoS	Based on the stake	The node with the highest equity gets the accounting right	1. Reduce the consumption of computing resources 2. Improve block generation efficiency	1. Complex protocols 2. Make the rich richer and the poor poorer
PBFT	Based on State Machine Replication	Most of the slave node responses are the final results	1. No tokens are required for rewards 2. Improving consensus efficiency	1. Only 1/3 corrupt nodes can be tolerated 2. Complex consensus confirmation processes
PoSo	Based on optimization problem solution	Delegates take turns to be an interim leader	1. Replace meaningless puzzles with useful optimization problems 2. Reduce computing resources consumption	1. Sacrifice decentralization 2. Private information is easy to leak during the consensus process

Figure 1. Comparison of different consensus mechanisms used in P2P energy trading.

2.2.1. Proof-of-Work

In the PoW mechanism, consensus proof is achieved by solving a difficult problem, which is to find a nonce to form a hash value that satisfies the condition. Since solving this problem requires a large amount of computing resources, but it can be easily verified by other nodes, the double-spending can be avoided to some extent. However, PoW is also known for wasting a lot of time and effort to solve a difficult but meaningless mathematical puzzle [29]. Since PoW was originally used in Bitcoin, the most common consensus mechanism for blockchain-based energy transactions is PoW. A blockchain-based edge-as-a-service framework is proposed in [30] for secure energy trading between electric vehicles (EVs), and PoW is used to help EVs reach consensus. In [31], the blockchain is designed to accommodate the decentralized nature of P2P markets and is developed using the PoW consensus mechanism.

2.2.2. Proof-of-Stake

The criticism of PoW has led to other consensus mechanisms. In PoS systems, instead of competing, miners maintain a set of verifiers that participate in the block creation process. Each node verifies that it has an interest in the grid, and this interest is used to determine the likelihood that the node will add the next transaction block to the blockchain. The system requires participants to prove ownership of the currency. However, this mechanism tends to make the rich richer and the poor poorer. Compared with the PoW mechanism, the PoS mechanism does not need to search for a suitable nonce to solve a difficult puzzle, which saves energy and improves efficiency. In [32], the demand response program is verified based on the PoS consensus mechanism. Each distributed energy producer in the grid can either act as a verifier of energy transactions or become the next valid block miner, and each verifier holds a certain stake. The authors of [33] proposed that by using PoS in P2P energy trading, miners sacrifice part of their stake to compensate for power losses and narrow the price gap in traditional prosumer-to-grid trading. In addition, the proposed model also improves the income of producers and saves the cost of consumers through the designed pricing mechanism, which is helpful to increase the social welfare.

2.2.3. Practical Byzantine Fault Tolerance

The PBFT algorithm is a replication algorithm that can tolerate Byzantine faults. The algorithm is feasible in an asynchronous environment. PBFT is suitable for small networks because the consensus process requires three rounds of voting, and each round of voting is broadcast. Because it does not have the same mining process as PoW, it saves a lot of computing resources. When the proportion of malicious nodes is less than 1/3, the

correct consensus can be reached. In [34], relying on blockchain smart contract technology, a distributed reputation system RBT is designed to realize distributed and automated reputation management. It is used to implement a PBFT-based delegated consensus and reputation-based auction in a P2P energy trading system. Considering that most user nodes are trusted, a fast PBFT algorithm is proposed in [35] to achieve efficient consensus for P2P energy trading. The consensus algorithm is executed to store transactions, support data traceability, and improve transaction efficiency.

2.2.4. Proof of Solution

In PoW-based blockchains, participants are competing to solve a difficult but meaningless mathematical puzzle in order to become the interim leader. The winner will become the leader and announces the block containing the latest state of the network. Other participants will accept a block if and only if it passes validation. A block is deemed to be valid if the puzzle was correctly solved and the latest state of the network was generated by following strict rules. It is worth noting that the math problems are difficult to solve but easy to verify, so other participants will not expend a lot of effort to verify them.

The novel consensus mechanism PoSo is inspired by PoW. The difference is that PoSo replaces meaningless mathematical puzzles with useful and meaningful optimization problems. Like the solutions of mathematical puzzles in PoW, the solutions of optimization problems in PoSo are difficult to find, but easy to verify. Specifically, if the optimization problem is convex, then x^* is an optimal solution if and only if x^* satisfies the KKT conditions. Obviously, it is much easier to verify whether a given x^* satisfies the optimality condition than to find it. However, unlike PoW, PoSo does not spend huge amounts of computing resources on nonsensical problems. These advantages expand the use of blockchain in various applications that require optimization, but do not have a central trusted authority to run it.

As shown in Figure 2, in a PoSo scenario, all participants select a delegation of participants interested in managing the collaborative network. An interim leader, elected by the delegates, is charged with coming up with the best solution. The other delegates are followers who validate and exchange the solutions they receive. If followers agree on a solution, they broadcast the solution to every participant. Participants trust a solution if it is approved by a majority of delegates. Any leader who is deemed incompetent or dishonest will be removed from the delegation and replaced by another representative.

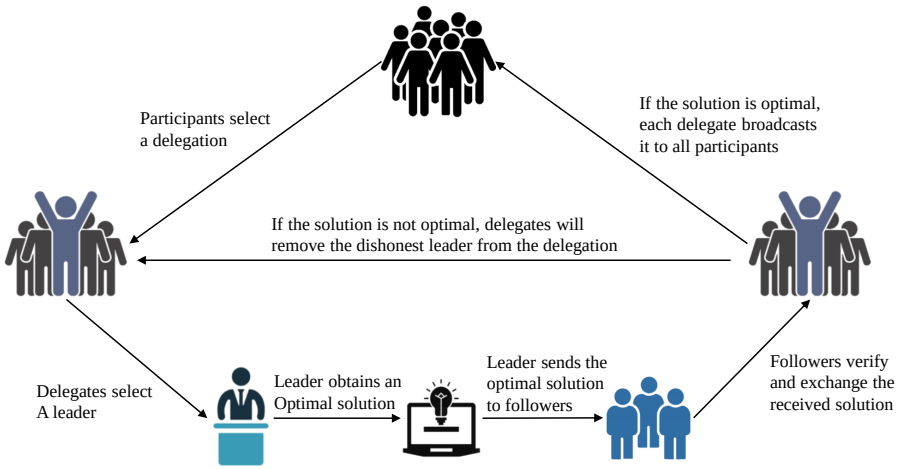

Figure 2. Flow chart of PoSo.

3. Negotiation Mechanism for P2P Energy Trading

3.1. Problem Formulation

An illustration of P2P energy markets is shown in Figure 3 [36]. Prosumers will negotiate the simultaneous price and energy for multi-lateral transactions based on predefined trading rules. It can be seen that the P2P energy market is much more decentralized compared to the existing centralized market, where all participants must provide and share all private information (such as cost or utility functions, power boundaries, and generation uncertainty) to a central coordinator, who centrally determines energy dispatches. In a P2P market, all agents are free to negotiate prices and quantities for multilateral transactions without leaking any private information.

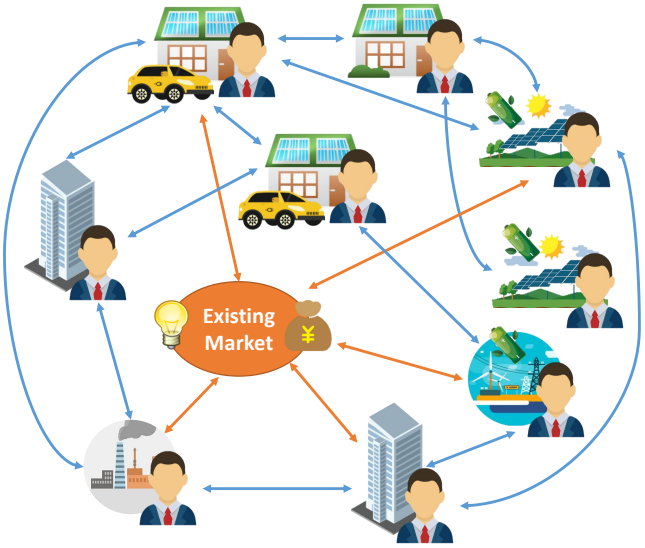

Figure 3. P2P energy trading market architecture.

There are a few agents in the P2P energy market who act as sellers if they have surplus renewable energy generation, such as wind generators or solar panel; or as buyers to meet their energy demand, such as EV charging. In this section, we first model the P2P energy trading process, and the social welfare maximization problem is built.

First of all, before the negotiation process, each prosumer will first determine the role (seller or buyer) according to the power generation and consumption. The power E_n of each agent $n \in \Omega$ is split into the number of bilateral transactions with a set of neighboring agents $m \in \omega_n$ as

$$E_n = \sum_{m \in \omega_n} E_{nm}, \quad \forall n \in \Omega. \tag{9}$$

A positive E_n indicates residual energy, a negative value indicates energy demand; thus, a positive E_{nm} indicates sale or production, and a negative one means purchase or consumption. For simplicity, $\mathbf{E}_n = \{E_{n1}, \ldots, E_{nm}, m \in \omega_n\}$ is used to represent all transactions of agent n between neighboring agents. The power of agent n shall be bounded as follows:

$$\underline{E_n} \leq E_n \leq \overline{E_n}, \quad \forall n \in \Omega \tag{10}$$

For renewable resources producer, the lower bound $\underline{E_n}$ is set to zero, and the upper bound $\overline{E_n}$ is set to the forecast power generation.

Each agent is restricted to be either a producer or a consumer, that is $\underline{E_n}\overline{E_n} \geq 0$. Thus, the variable is 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:

$$\begin{cases} E_{nm} \geq 0, & \forall (n,m) \in (\Omega_p, \omega_n) \\ E_{nm} \leq 0, & \forall (n,m) \in (\Omega_c, \omega_n) \end{cases} \tag{11}$$

where Ω_p and Ω_c denote the set of producers and consumers, respectively. Finally, a set of equilibrium constraints is used to represent the market equilibrium among agents:

$$E_{nm} + E_{mn} = 0, \quad \forall (n,m) \in (\Omega, \omega_n). \tag{12}$$

To simplify the presentation of the process, quadratic functions are commonly used to model the producer generation cost and consumer utility as follows:

$$C_n(E_n) = a_n E_n^2 + b_n E_n, \tag{13}$$

where a_n and b_n are predetermined positive constants according to the equipment and consumers' preference.

Finally, the objective of a P2P energy market is to maximize the social welfare of all agents while satisfying these constraints. The problem can be equivalently formulated as a cost minimization problem as below:

$$\min \sum_{n \in \Omega} C_n(E_n) \tag{14a}$$

$$\text{s.t. } \underline{E_n} \leq E_n \leq \overline{E_n} \qquad \forall n \in \Omega \tag{14b}$$

$$E_{nm} \geq 0 \qquad \forall (n,m) \in (\Omega_p, \omega_n) \tag{14c}$$

$$E_{nm} \leq 0 \qquad \forall (n,m) \in (\Omega_c, \omega_n) \tag{14d}$$

$$E_{nm} + E_{mn} = 0 \qquad \forall (n,m) \in (\Omega, \omega_n) \tag{14e}$$

The above problem is a convex optimization problem, which has a unique optimal solution and can be obtained by centralized methods such as interior point methods. However, the centralized approach needs to leak the private information of all agents, which is unacceptable in practical applications. It is necessary to design a decentralized negotiation mechanism to achieve the optimal scheduling of the above optimization problem (14) without leaking any individual information.

Remark 1. *If the product differentiation or preference is considered, the bilateral trading costs are calculated as linear functions of the quality traded with each neighboring agents* $\tilde{C}_n(E_n) = \sum_{m \in \omega_n} c_{nm} E_{nm}$. *The bilateral trading coefficients* c_{nm} *are affected by emissions, transport distance, size of prosumers, etc. The different preference or value* c_{nm} *will influence the prices between agents. For example, the long distance between agent n and m will generate a high transmission fee, and cause a price between n and m, which will be* $\lambda_{nm} + c_{nm}$, *where* λ_{nm} *represents the traded price provided from agent n to m for amount* E_{nm}. *Since we focus on the consensus mechanism in this paper, the product differentiation is not considered, but the proposed method is still workable when including it.*

3.2. Negotiation Mechanism

In this section, a decentralized negotiation mechanism based on consensus ADMM is proposed for P2P energy trading, which is shown in Figure 4. Agents will solve individual optimization problems to determine the optimal biddings, and then communicate and share the updated energy quantities and prices to neighbor trading agents. The negotiation process will run iteratively until convergence, which is decided by the Market Operator (MO). The MO collects updated transaction information and sends a termination signal after the balance between each pair of agents is met. The proposed negotiation mechanism focuses on a deterministic clearing algorithm for a single time period of the previous day, and it can be easily extended to multiple time periods [15,16].

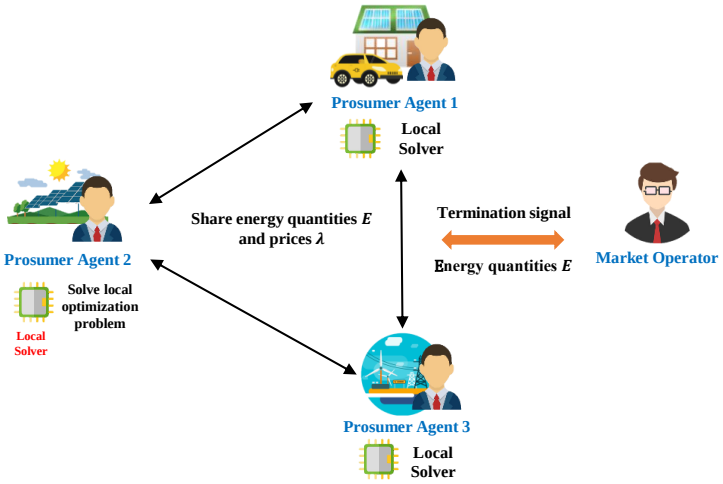

Figure 4. Schematic diagram of the P2P energy trading negotiation mechanism.

Local Update for Each Agent: At the beginning, each agent randomly calculates and broadcasts the initial energy price and quantity in parallel. Each agent will then iteratively update the energy amounts and prices of its neighbors until equilibrium.

Then, at a given iteration k, each agent $n \in \Omega$ will update its amount of energy traded with neighbor agents by solving an optimization problem as follows.

$$\begin{aligned} \min_{\mathbf{E}_n^{k+1}} \quad & C_n(E_n) + \sum_{m\in\omega_n}\left[\lambda_{nm}^k\left(F_{nm}^k - E_{nm}\right) + \frac{\rho}{2}\left(F_{nm}^k - E_{nm}\right)^2\right] \\ \text{s.t.} \quad & E_n = \sum_{m\in\omega_n} E_{nm} \\ & \underline{E_n} \le E_n \le \overline{E_n} \\ & E_{nm} \ge 0, \quad \forall (n,m) \in (\Omega_p, \omega_n) \\ & E_{nm} \le 0, \quad \forall (n,m) \in (\Omega_c, \omega_n) \end{aligned} \tag{15}$$

where λ_{nm} is the traded price for amount E_{nm}, and it is also the dual variable for the balance constraints (12); $F_{nm} = \frac{E_{nm} - E_{mn}}{2}$ represents the median value of the energy amount between n and m. After obtaining the updated energy amount, each agent will update the energy prices λ_{nm}^{k+1} as follows:

$$\lambda_{nm}^{k+1} = \left[\lambda_{nm}^k - \frac{\rho}{2}\left(E_{nm}^{k+1} + E_{mn}^{k+1}\right)\right]^+, \tag{16}$$

where $[.]^+$ denotes the $\max(.,0)$.

Local Update for Market Operator: Since $C_n(E_n)$ is strictly convex, it suffices to guarantee that our algorithm converges to the global optimum point [37]. The algorithm converges when the total primary residuals (the squared difference between each pair of traded amount) and total dual residuals (the squared difference between the amount of two successive iterations) is smaller than the global stopping criterion as follows:

$$R^{k+1} \triangleq \sum_{n\in\Omega}\sum_{m\in\omega_n} (E_{nm}^{k+1} + E_{mn}^{k+1})^2 \le \chi^r, \tag{17a}$$

$$T^{k+1} \triangleq \sum_{n\in\Omega}\sum_{m\in\omega_n} (E_{nm}^{k+1} - E_{nm}^{k})^2 \le \chi^t, \tag{17b}$$

where χ^r and χ^t are predetermined positive and very small constants. The MO must be trusted, and can be the community manager or the authorized representative of all agents. The MO will collect the energy of all transactions to check whether the algorithm converges

by (17), and broadcast a termination signal to all agents if the convergence conditions are met.

4. Privacy-Preserving P2P Energy Trading Consensus Mechanism

4.1. PoSo-Based Consensus Mechanism

In last section, we propose a negotiation mechanism for P2P energy trading, which is based on the marginal price. However, under the LMP mechanism, the agents who dominate the prices can earn more profit by taking some dishonest strategies. To solve this problem, in this section, a new consensus mechanism is proposed for P2P energy trading based on PoSo, which is an effective way to prevent agents from dishonest behavior.

Reviewing the PoSo flow chart again, a delegate is first selected from participants, and a temporary leader is further elected from the delegate. The leader will solve the social welfare maximization problem to propose an optimal solution, while the followers in the delegate verify the optimality to decide whether to accept it or not. Although this mechanism has good computational efficiency, it is not suitable for P2P energy trading due to the unwillingness of prosumers to share private information with the central leader.

Therefore, to solve this problem, an improved and revised version of the PoSo mechanism is designed to fit our proposed CADMM-based P2P energy trading negotiation mechanism [38], which can well protect the private information. Each agent solves individual optimization problems (15) to obtain an optimal solution, and neighbor agents verify the equivalent KKT conditions to prove its optimality. Then, in turn, the agent will also check the solution of neighbor agent. Thus, the consensus mechanism is running in a symmetric manner.

The main advantage is that no authorization is required and the agent does not need to convey private information to the central leader. All agents can solve the local optimization problem in parallel and broadcast the results to neighboring agents for optimality verification.

In more detail, the KKT conditions for local updates (15) are listed as in (18).

$$\begin{aligned}
&\partial C_n\Big(\sum_{m\in\omega_n} E_{nm}\Big) - \lambda^k_{nm} + \rho(E_{nm} - F^k_{nm}) - \underline{\mu}_n + \overline{\mu}_n - \delta_{nm} = 0, \forall (n,m) \in (\Omega_p, \omega_n)\\
&\partial C_n\Big(\sum_{m\in\omega_n} E_{nm}\Big) - \lambda^k_{nm} + \rho(E_{nm} - F^k_{nm}) - \underline{\mu}_n + \overline{\mu}_n + \delta_{nm} = 0, \forall (n,m) \in (\Omega_c, \omega_n)\\
&\underline{E_n} \le \sum_{m\in\omega_n} E_{nm} \le \overline{E_n}\\
&E_{nm} \ge 0, \quad \forall m \in \omega_n, \quad \text{if} \quad n \in \Omega_p\\
&E_{nm} \le 0, \quad \forall m \in \omega_n, \quad \text{if} \quad n \in \Omega_c\\
&-\underline{\mu}_n \sum_{m\in\omega_n} E_{nm} = 0, \quad \overline{\mu}_n \sum_{m\in\omega_n} E_{nm} = 0\\
&\delta_{nm} E_{nm} = 0, \quad \forall m \in \omega_n
\end{aligned} \tag{18}$$

The above equations construct a nonlinear problem with primal variables $\mathbf{E}_n$ and dual variables $\{\underline{\mu}_n, \overline{\mu}_n, \delta_{nm}\}$. $\{\underline{\mu}_n, \overline{\mu}_n\}$ are for constraints $\underline{E_n} \le E_n \le \overline{E_n}$, and δ_{nm} is for $0 \le E_{nm}$ $(0 \ge E_{nm})$.

However, if we choose to check all of these equations, it will cause a very large computational and communication burden. Alternatively, we choose to verify the cumulative KKT condition as below:

$$\begin{aligned}
N_{\omega_n}\Big(2a_n \sum_{m\in\omega_n} E_{nm} + b_n\Big) &= \sum_{m\in\omega_n} \lambda^k_{nm} - \rho\Big(\sum_{m\in\omega_n} E_{nm} - \sum_{m\in\omega_n} F^k_{nm}\Big) + N_{\omega_n}\Delta\mu_n + \sum_{m\in\omega_n} \delta_{nm}, \forall n \in \Omega_p\\
N_{\omega_n}\Big(2a_n \sum_{m\in\omega_n} E_{nm} + b_n\Big) &= \sum_{m\in\omega_n} \lambda^k_{nm} - \rho\Big(\sum_{m\in\omega_n} E_{nm} - \sum_{m\in\omega_n} F^k_{nm}\Big) + N_{\omega_n}\Delta\mu_n - \sum_{m\in\omega_n} \delta_{nm}, \forall n \in \Omega_c
\end{aligned} \tag{19}$$

After each update, agent n will send a information set $\mathbb{I}_n^k$ to neighboring agents for verification. The set contains information as below:

$$\mathbb{I}_n^k = \left\{ a_n, b_n, E_n^{k+1} = \sum_{m \in \omega_n} E_{nm}^{k+1}, \sum_{m \in \omega_n} \lambda_{nm}^k, F_n^k = \sum_{m \in \omega_n} F_{nm}^k, \Delta\mu_n, \sum_{m \in \omega_n} \delta_{nm} \right\} \tag{20}$$

If the cumulative KKT conditions (19) are met, the optimality of the solution is also successfully verified, and results are exchanged among other neighbor agents until the majority of them are approved. An incentive and punishment mechanism is also designed. In detail, if the verification fails, the dishonest agent n will be subject to a heavy fine, which is expressed as adding a penalty indicator function $\Delta p(v)$ in the objective function. If the result is optimal, $\Delta p(v) = \Delta p(0) = 0$; on the contrary, $\Delta p(v) = \Delta p(1) = -P$. Here, P is a predefined penalty that will be distributed equally among all verifiers who participate in the consensus process. This new consensus mechanism can motivate agents to participate honestly in the energy market and achieve an incentive compatible market. An illustrative example is shown in the Figure 5, and the consensus mechanism is summarized in Algorithm 1.

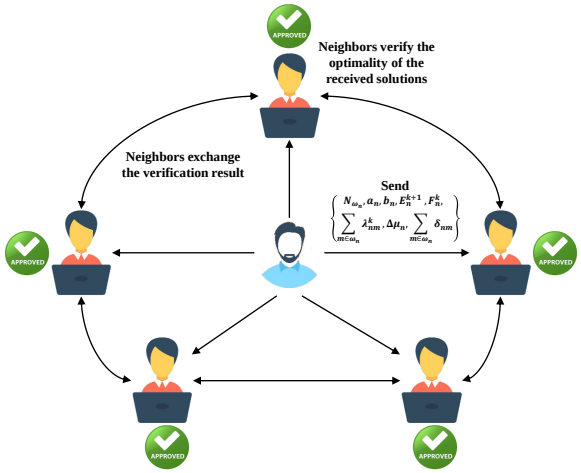

Figure 5. Diagram of the consensus mechanism for P2P energy trading.

Algorithm 1: The improved PoSo-based consensus mechanism for P2P energy trading.

for *Agent n in* Ω **do**
  Agent n solves individual optimization problem to obtain the optimal solutions $\{\sum_{m\in\omega_n} E_{nm}^{k+1}, \Delta\mu_n^{k+1}, \sum_{m\in\omega_n} \delta_{nm}^{k+1}\}$
  Agent n sends the solutions together with $\{N_{\omega_n}, a_n, b_n, \sum_{m\in\omega_n} \lambda_{nm}^k, \sum_{m\in\omega_n} F_{nm}^k\}$ to neighbor verifiers.
  for *Agent m in* ω_n **do**
    Validate the optimality of the received solution by checking if the cumulative KKT conditions (19) hold.
    if *The verification passes* **then**
      Forward $v_m = 0$ to other neighbors.
    else
      Forward $v_m = 1$ to other neighbors.
  if *Over 50% of the verification results* $v_m = 0$ **then**
    Complete the consensus verification, set $v = 0$, and $\Delta p(v) = 0$.
  else
    Complete the consensus verification, set $v = 1$, and $\Delta p(v) = -P$.

Similar to other blockchain consensus mechanisms, the proposed consensus mechanism is only effective if the number of honest neighbor agents exceeds the number of dishonest agents. An effective consensus mechanism indicates that all participants take actions based on the correct and identical optimal solution. As long as there are honest neighbor agents in the verifiers list, all honest agents will eventually see the optimal solution, so the consensus verification succeeds. In other words, as long as there are a majority of honest neighbor agents, the optimality of the received solution can be successfully verified. When more than 50% of the neighbor agents obtain the same verification result, anyone can trust the result because a non-optimal solution is impossible to be accepted by more than half of the neighbor agents. However, if dishonest agents dominate the verifiers set, they may also control the verification outcome.

4.2. Protect the Private Information and Resist Malicious Agents

In the consensus verification process, agents have to send private information $\{a_n, b_n\}$ to neighboring agents. However, there may be dishonest nodes who will collect individual privacy to take strategy and earn more profit. Additionally, they may not complete the verification work or deliberately disapprove the optimality of the received solutions. Thus, the proposed method should resist untrusted agents who have this misbehavior. In this section, we propose a privacy-preserving consensus mechanism that is resistance to untrusted agents, who may collect individual privacy or destroy the consensus verification process. Here, we assume that there are always no more than f of N agents are untrusted and malicious in the neighboring agents set.

4.2.1. Algorithm

The proposed privacy-preserving consensus mechanism is illustrated in Figure 6. The agent m_5 is malicious, who will not forward any messages.

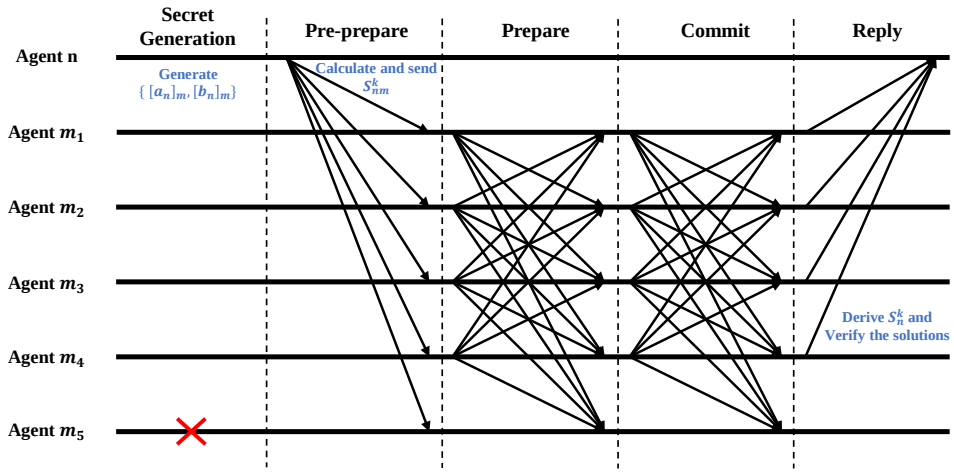

Figure 6. Flowchart of the privacy-preserving consensus mechanism.

Briefly speaking, an agent generates N_{ω_n} pieces of encrypted data and separately submits encrypted aggregated information to each corresponding neighboring agent. Then, we use a PBFT-based method [39] to realize that the trusted agents can recover the right result even under the misbehavior of malicious agents.

In detail, first of all, in the secret generation phase, before the negotiations begin, agent n would generate pieces of the encrypted a_n and b_n using Shamir's Secret Sharing scheme [40]. It constructs polynomials

$$f_{a_n}(\xi) = a_n + d_{a_n,1}\xi + d_{a_n,2}\xi^2 + \ldots + d_{a_n,f}\xi^f, \tag{21a}$$

$$f_{b_n}(\xi) = b_n + d_{b_n,1}\xi + d_{b_n,2}\xi^2 + \ldots + d_{b_n,f}\xi^f, \tag{21b}$$

where d_{a_n}, d_{b_n} are randomly generated. Then, for each neighboring agent $m \in \omega_n$, agent n generates encrypted information $\{f_{a_n}(m), f_{b_n}(m)\}, m = 1, 2, \ldots, N_{\omega_n}$. Here, we use $\{[a_n]_m, [b_n]_m\}$ to denote $\{f_{a_n}(m), f_{b_n}(m)\}$ for simplicity.

The encryption algorithm has two important properties as follows:

(i) The algorithm is a $(f+1, N_{\omega_n})$ threshold scheme. The actual information $\{a_n, b_n\}$ of agent n can be decrypted using (22) only if at least $f+1$ encrypted messages are collected.

$$a_n = \sum_{m=1}^{f+1} [a_n]_m \prod_{l=1, l\neq m}^{f+1} \frac{-l}{l-m}, \tag{22a}$$

$$b_n = \sum_{m=1}^{f+1} [b_n]_m \prod_{l=1, l\neq m}^{f+1} \frac{-l}{l-m}. \tag{22b}$$

(ii) It is additively homomorphic. That is to say, only when the sum of at least $f+1$ agent n encrypted information is collected, can (23) be used to decrypt the sum of agent n information.

$$a_n + b_n = \sum_{m=1}^{f+1} ([a_n]_m + [b_n]_m) \prod_{l=1, l\neq m}^{f+1} \frac{-l}{l-m}. \tag{23}$$

As far as we know, Shamir's secret sharing is the simplest scheme that satisfy the above properties in the domain of real numbers. Our approach remains valid for other schemes with the same properties, such as the Brickell scheme [41].

We then try to explore a way to realize that trusted agents can always recover correct information even under the misbehavior of malicious agents. This algorithm inherits the idea of PBFT. Specifically, in the Pre-prepare phase, agent n obtains the updated solution $\sum_{m\in\omega_n} E_{nm}^{k+1}$ after iteration k, agent n calculates aggregated piece information S_{nm}^k as follows:

$$S_{nm}^k = N_{\omega_n} \left(2[a_n]_m \sum_{m\in\omega_n} E_{nm}^{k+1} + [b_n]_m \right) + \rho \sum_{m\in\omega_n} E_{nm}^{k+1}, \quad \forall m \in \omega_n, \tag{24}$$

and sends them to corresponding neighboring agents; then, in the Prepare phase, each agent m will broadcast S_{nm}^k to all agents in ω_n; after that, in the Commit phase, when receiving S_{nm}^k from no less than $3f$ agents, each agent send a Commit message to others; finally, in the Reply phase, when receiving Commit messages from no less than $3f+1$ agents (including itself), agent m derives the correct aggregated result $S_n^k = N_{\omega_n}(2a_n \sum_{m\in\omega_n} E_{nm}^{k+1} + b_n) + \rho \sum_{m\in\omega_n} E_{nm}^{k+1}$ by executing Algorithm 3. In short, the agent enumerates all possible decryption results, and most of these possible results are considered correct.

After the result reconstruction process, each neighboring agent validates the optimality of the received solutions $\mathbf{E}_n^k$ by (19) according to the correct results. Algorithms 2 and 3 introduce the procedure of the privacy-preserving consensus mechanism for agent n and correct result decryption method.

4.2.2. Security Analysis

Theorem 1: If the proportion of untrusting neighboring agents is less than 1/4, i.e., $4f+1 \leq N_{\omega_n}$, the proposed method guarantees the correctness of the consensus verification result.

Proof 1: The Reply phase ensures the correctness of the aggregation result S_n^k. In detail, when an agent receives no less than $3f+1$ Commit messages, at least $2f+1$ Commit messages were sent by trusted agents. Therefore, a trusted agent can obtain at least C_{2f+1}^{f+1} correct results in Algorithm 3. Untrusted agents may tamper with pieces of their encrypted messages to cause trusted agents to decrypt incorrect results. However, even if the untrusted agents send f of $3f+1$ Commit messages, the trusted agent exports incorrectly aggregated data up to C_{2f}^{f+1} times [42]. Therefore, the trusted agent always outputs the correct result S_n^k in Algorithm 3.

Algorithm 2: Privacy-preserving consensus mechanism for agent n.

//Secret Generation
Agent n generates the piece of encrypted private information $\{[a_n]_m, [b_n]_m\}$ of corresponding neighboring agents.
//Pre-prepare phase
Agent n Calculate $S_{nm}^k = N_{\omega_n}(2[a_n]_m \sum_{m\in\omega_n} E_{nm}^{k+1} + [b_n]_m) + \rho \sum_{m\in\omega_n} E_{nm}^{k+1}$, and broadcast it to all agents in ω_n.
//Prepare phase
for *Each agent m in* ω_n **do**
    When receive S_{nm}^k, broadcast it to other agents in ω_n.
//Commit phase
for *Each agent m in* ω_n **do**
    when receiving S_{nm}^k from no less than $3f$ agents, send a Commit message to others.
//Reply phase
for *Each agent m in* ω_n **do**
    when receiving Commit messages from no less than $3f+1$ agents (including itself), derive the correct result S_n^k by executing Algorithm 3.
Validate the optimality of the received solutions by (19) according to the correct S_n^k.

Algorithm 3: Correct aggregated result decryption method.

Let **D** denote the set of agents who sent Commit messages.
Construct all subsets of **D**: $\Gamma_1, \ldots, \Gamma_u$, where each subset has $f+1$ agents. Obviously, $u = \binom{3f+1}{f+1}$.
for $l = 1 : u$ **do**
    Calculate $s_l = N_{\omega_n}(2a_n \sum_{m\in\omega_n} E_{nm}^{k+1} + b_n) + \rho \sum_{m\in\omega_n} E_{nm}^{k+1}$ according to S_{nm}^k provides by agents in Γ_l.
Output the major result in $\{s_1, \ldots, s_u\}$ as the correct result.

4.3. Improve the Efficiency of the Consensus Mechanism

In a distributed optimization method-based P2P energy trading mechanism, it usually takes many iterations to converge, which will cause a mass of communication time for the consensus mechanism. Additionally, since agents have to exchange information with each other to reach a consensus, the communication complexity is $O(k \times N^3)$, where k is the iterations for convergence, and N is the numbers of agents in the market. It is undeniable that our algorithm is inefficient when dealing with a large number or prosumers, but there are two ways to improve efficiency.

First, it can be discovered that the verification equation contains the previous results $\{\sum_{m\in\omega_n} \lambda_{nm}^k, \sum_{m\in\omega_n} F_{nm}^k\}$. It means that if a dishonest agent constructs an incorrect solution at a certain iteration to pass the verification, it probably will not pass in the next iteration unless it proceeds to solve the local problems using the sub-optimal solutions, but it may result in economic loss. Therefore, the optimal solutions are linked to each other to form a "chain", and it is sufficient to verify the optimality of only the first and last few iterations. To be specific, for each agent n, we only verify the solutions of the first three iterations and last few iterations according to the residuals (17). When the maximal value of R^{k+1} and T^{k+1} is less than $\alpha\chi$, the consensus verification process is activated. The smaller α means less consensus verification. The α is chosen to be 10 in this work. By using this low-frequency consensus mechanism, the computational burden can be highly reduced.

Second, in the energy community, prosumers usually have frequent or fixed trading partners. Thus, it is better to divide the network into different shards according the

transaction history, and prosumers only need to communicate with others in the same shard. In this way, the communication complexity can be highly reduced, and more shards, lower complexity. How to reasonably allocate prosumers to different shards is the focus of our future work.

5. Results

This section provides numerical results for the performance evaluation of the proposed P2P energy trading privacy-preserving consensus mechanism through different case studies. A 13-node meshed distributed network with five sellers and seven buyers is considered as in [6] for illustration and discussion. The parameters $\{a_n, b_n, \underline{E_n}, \overline{E_n}\}$ of agents are listed in the Table 1 and are chosen randomly for simulation. In the network, sellers have renewable energy generation, such as solar power or wind generator. The power output lower bounds of sellers or producers are all set to 0, and the power output upper bounds $\overline{E_n}$ are the forecast renewable power generation in the real time. Buyers are houses equipped with controllable appliances, and the upper bounds are all set to -1, which means the minimal energy demand is 1 kW. As shown in Figure 7, each agent is connected to a bus. Bus 1 is the reference bus. Sellers are located on buses 2, 5, 8, 10 and 11, while buyers are on buses 3, 4, 6, 7, 9, 12 and 13. All agents can communicate with each other through a connected virtual communication network, and solid lines constitute a physical electrical network. The stopping conditions χ^r and χ^t are both set to 10^{-5} in the simulation.

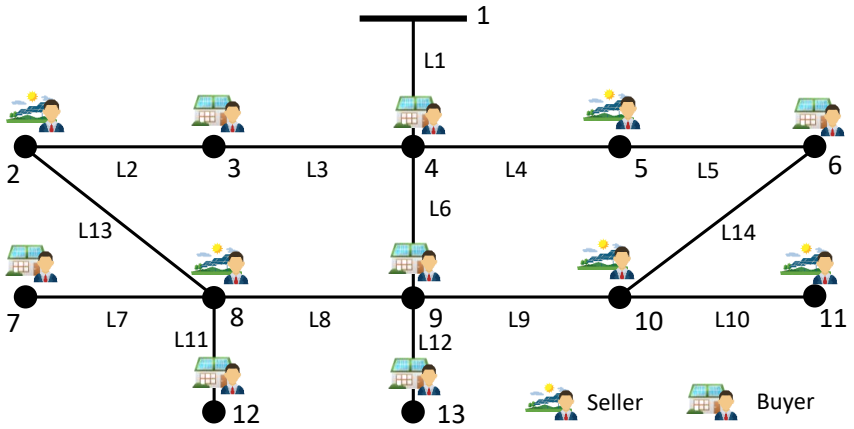

Figure 7. Schematic diagram of the test system.

Table 1. Agents' parameters of a simple case study.

Agent	Bus	a_n	b_n	$\underline{E_n}$ [kW]	$\overline{E_n}$ [kW]
S1	2	0.04	2.1	0	7
S2	5	0.046	2.5	0	4
S3	8	0.06	3.2	0	6
S4	10	0.03	4	0	8
S5	11	0.04	3	0	10
B1	3	0.056	3	-7	-1
B2	4	0.075	4	-9	-1
B3	9	0.042	5	-8	-1
B4	3	0.056	6	-5	-1
B5	4	0.036	4.5	-6	-1
B6	9	0.025	7	-6.5	-1
B7	9	0.04	5.5	-7.5	-1

5.1. Convergence Performance of the Negotiation Mechanism

For the convergence performance of the negotiation mechanism, we use the value of total primary and dual residuals to measure it. The convergence process of the negotiation algorithm is shown in Figure 8, from which it can be seen that the total local primary and dual residuals keep decreasing with oscillation, and all transactions between sellers and buyers converge after 36 iterations. Compared with other distributed optimization algorithms, such as consensus + innovation [8] and gradient-based methods [6], the ADMM-based algorithm has a very good convergence speed.

Table 2 shows the final power transaction quantities among agents of the case study. Since there is no line constraint, the power transactions prices among agents will converge to the same value, which is 4.29 $/kW. We also use the Power Transfer Distribution Factors (PTDF) model to calculate the flows on each line. Figure 9 shows the line flows of the test system.

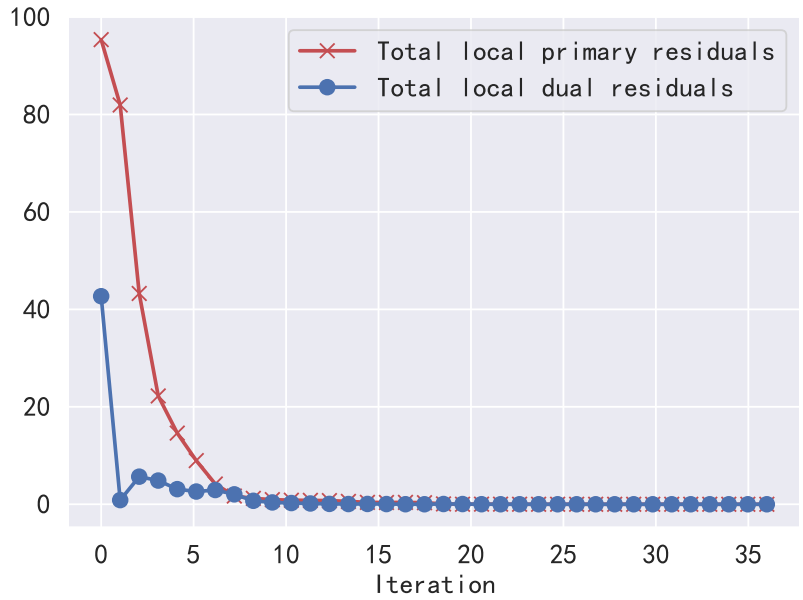

Figure 8. Convergence of the negotiation algorithm.

Table 2. Final power transactions quantities among agents.

	S1	S2	S3	S4	S5
B1	0.19 kW	0.19 kW	0.19 kW	0.23 kW	0.19 kW
B2	0.15 kW	0.19 kW	0.19 kW	0.34 kW	0.13 kW
B3	2.09 kW	1.27 kW	1.45 kW	0.82 kW	2.38 kW
B4	0.62 kW	0.43 kW	1.05 kW	1.21 kW	1.69 kW
B5	0.59 kW	0.01 kW	0.53 kW	0.43 kW	1.32 kW
B6	1.46 kW	0.80 kW	1.23 kW	0.98 kW	2.04 kW
B7	1.91 kW	1.11 kW	1.36 kW	0.87 kW	2.26 kW

5.2. Performance of the Consensus Mechanism

After each agent completes the verification for the optimality of received solutions, they need to share the consensus results to other neighboring agents. In this process, the message may be lost due to packet dropout, node crash, or malicious attack. Thus, we assume that the message delivery has a success rate, which is set to constant for all agents. In this section, we demonstrate the performance of the consensus mechanism under different message delivery success rates. The results are shown in Figure 10, from which we can find that the consensus success rate keeps increasing with the increase in the message

delivery success rate, and when the message delivery success rate is not lower than 0.7, the consensus mechanisms almost guarantee success.

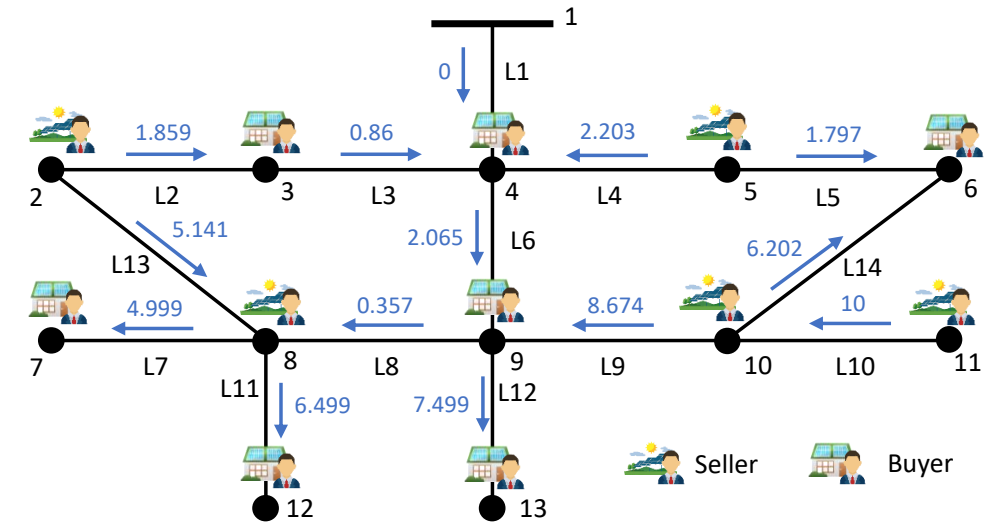

Figure 9. Flows on each line of the test system.

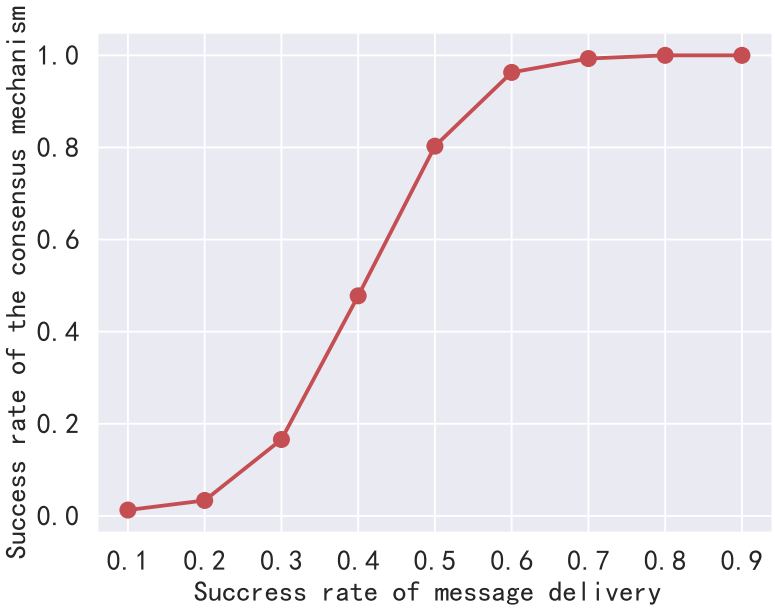

Figure 10. Success rate of the consensus mechanism under different success rate of message delivery.

5.3. Performance of the Privacy-Preserving Consensus Mechanism

In this paper, we propose a privacy-preserving consensus mechanism to protect the private information of agents. Agent n first constructs the pieces of encrypted information $\{[a_n]_m, [b_n]_m\}$, and then calculates $S_{nm}^k = N_{\omega_n}(2[a_n]_m \sum_{m\in\omega_n} E_{nm}^{k+1} + [b_n]_m) + \rho \sum_{m\in\omega_n} E_{nm}^{k+1}$, which will be sent to corresponding neighboring agents. By using the Shamir's secret sharing method, the private information $\{a_n, b_n\}$ and traded energy quantities $\sum_{m\in\omega_n} E_{nm}^{k+1}$ can be well protected. Table 3 shows the plaintext S_n^k and ciphertext S_{nm}^k of a seller in a certain iteration. It can be seen that the ciphertext is totally different from the plaintext, and an attacker cannot obtain the true value.

Table 3. The plaintext and ciphertext using Shamir's Secret Sharing.

Neighboring Agent	Plaintext S_n	Ciphertext S_{nm}
B1	36.60	7,272,133.495513787
B2	36.60	14,544,230.391027808
B3	36.60	21,816,327.286541834
B4	36.60	29,088,424.182055857
B5	36.60	36,360,521.07756989
B6	36.60	43,632,617.973083906
B7	36.60	50,904,714.86859793

Then, we assume that there are malicious agents who will manipulate the shared information S_{nm}^k to make the recovery value wrong. To be specific, the malicious agents will add a random value to the S_{nm}^k, and broadcast it to other neighboring agents during the consensus process. Table 4 shows the recovery results set $\{s_1, \ldots, s_u\}$ of a trusted agent with and without malicious attack. It can be seen from the table that, under one malicious agent attack, the major result in $\{s_1, \ldots, s_u\}$ is still the right result. However, if there are two malicious agents in the neighboring agents' set, one trusted agent may receive two fake messages S_{nm}^k from the two malicious agents, then in the six (C_4^2) recovery results, there will be five wrong results, and it is impossible to obtain the right result from the recovery set as shown in the third column in Table 4. Therefore, if the proportion of untrusted agents satisfy $4f + 1 \leq N_{\omega_n}$, the proposed method guarantees the correctness of the consensus verification results. Figure 11 shows the success rate of the consensus mechanism under different numbers of malicious agents. With the increase in malicious agents, the success rate decreases quite fast. When $f = 3$, i.e., there are three malicious agents in the neighboring agents, the success rate is only about 20%.

Table 4. The recovery results set $\{s_1, \ldots, s_u\}$ with and without malicious agents.

No Malicious Agent ($f = 0$)	One Malicious Agent ($f = 1$)	Two Malicious Agent ($f = 2$)
36.60	−3644.287	−8518.492
36.60	−29,410.494	−947.18
36.60	−781.375	36.6
36.60	36.60	52,052.002
36.60	36.60	−1674.418
36.60	36.60	−182.018

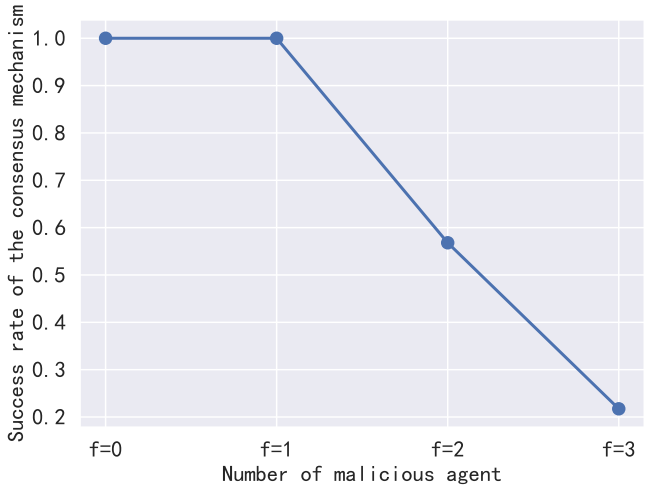

Figure 11. Success rate of privacy-preserving consensus mechanism under different number of malicious agents.

5.4. Computational Efficiency Improvement for the Consensus Mechanism

The computational efficiency improvement for the consensus mechanism is shown in Figure 12. We use the communication times among agents to measure the performance. It can be seen that after taking the strategy, i.e., only verify the optimality of the received results for the first three and last few iterations, the communication times among agents can be highly reduced by more than 60%, from 15,120 to 5880 ($\alpha = 100$) and 3360 ($\alpha = 10$).

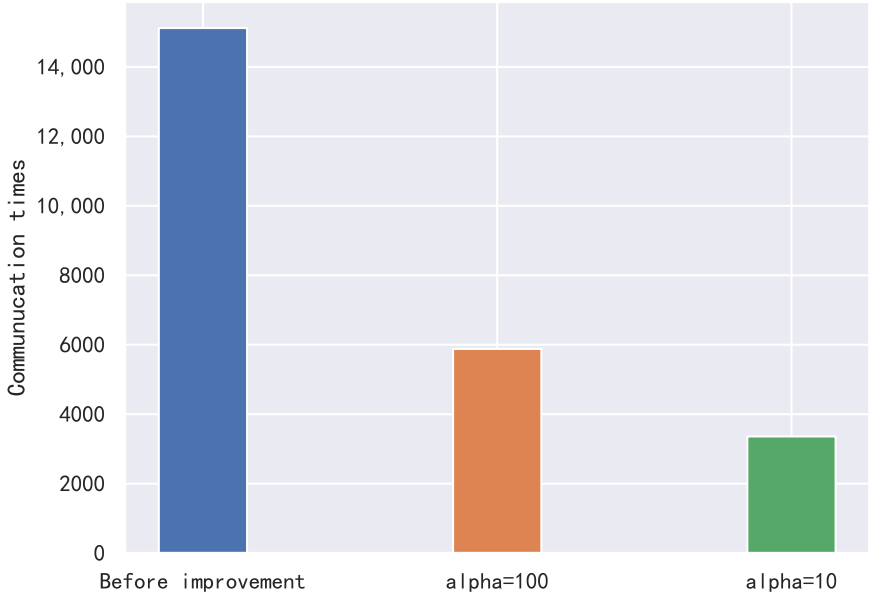

Figure 12. Communication times forthe consensus mechanism under different values of α.

6. Discussion

The most valuable achievement of this paper is the privacy-preserving consensus mechanism, which provides a secure and efficient method to ensure the incentive compatibility of an optimization-based P2P energy trading mechanism. Using our method, all agents can spontaneously and collaboratively guarantee the optimality of the solutions, i.e., all agents will obey the negotiation mechanism to decide the energy biddings. However, our consensus mechanism has some limitations. First, it is only applicable to optimization problems where the objective function and constraints are continuously differentiable. It is not suitable for integer programming (IP) problems because the optimal solution cannot be verified using KKT conditions. Second, for privacy-preserving methods, the proportion of untrusted representatives is less than 1/4. While it can effectively protect privacy, it performs slightly worse on dishonesty tolerance. In our future work, we will focus on improving the consensus mechanism to handle more complex optimization problems, increase the tolerance of dishonesty, and apply it to blockchain-based systems to prove their effectiveness and performance.

7. Conclusions

For an optimization-based P2P energy trading mechanism, it is a challenge that to encourage agents to honestly cooperate in a trustless environment. In this work, an improved and revised PoSo-inspired consensus mechanism is proposed for P2P energy trading. The solutions solved by each agent will be verified by neighboring agents based on the KKT conditions, and a privacy-preserving method is designed to protect agents' private information using Shamir's Secret Sharing scheme. A PBFT-based method is designed to recover the correct result even under the malicious attacks from untrusted agents. In the simulation, we demonstrate the convergence of the negotiation mechanism, effectiveness of consensus mechanism, privacy protection performance and computational efficiency improvement for the consensus mechanism. The results prove that our proposed consensus

mechanism can effectively verify the correctness of the solutions without a central authority and well-protect agents' privacy under a proportion of malicious attackers.

Author Contributions: Conceptualization, Z.L. and B.Z.; methodology, Z.L. and H.G.; writing—original draft, Z.L. and B.Z.; writing—review and editing, H.G. and L.L.; validation, F.Z.; investigation, L.L.; software, F.Z.; project administration, Z.L.; funding acquisition, Z.L. and B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Research Program of STATE GRID Corporation of China under grant 5700-202255294A-2-0-QZ.

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
n, m	Indices for agents
$\underline{E}, \overline{E}$	Boundaries of power
a_n, b_n	Coefficients of the quadratic function of agent n
Ω	Set of agents
Ω_p	Set of energy producers
Ω_c	Set of energy consumers
ω	Set of neighboring agents
λ_{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 agent n to m
F_{nm}	Intermediate value of traded energy quantity between agent n to m
ρ	Penalty factor
χ^r, χ^t	Stopping criterion
R	Total local primary residuals
T	Total local dual residuals
$\{\underline{\mu}_n, \overline{\mu}_n\}$	Dual variables for constraints $\underline{E_n} \leq E_n \leq \overline{E_n}$
δ_{nm}	Dual variables for constraints $0 \leq E_{nm}$ $(0 \geq E_{nm})$
$\Delta p(\cdot)$	Indicator function for consensus fine
v	Consensus verification result
f	Number of untrusted agents in the neighboring agents set
d_{a_n}, d_{b_n}	Randomly generated coefficient for Shamir's Secret Sharing functions
$[a_n]_m, [b_n]_m$	Encrypted piece of information for agent m
S^k_{nm}	Encrypted aggregated information from agent n to m
S^k_n	Decrypted aggregated result
α	Tuning parameter to control the number of consensus verification
$\mathbf{D}$	Set of agents who sent Commit messages
Γ	Subset of $\mathbf{D}$

References

1. Dorahaki, S.; Rashidinejad, M.; Ardestani, S.F.F.; Abdollahi, A.; Salehizadeh, M.R. An integrated model for citizen energy communities and renewable energy communities based on clean energy package: A two-stage risk-based approach. *Energy* **2023**, *277*, 127727. [CrossRef]
2. 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]
3. 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; IEEE: Piscataway, NJ, USA, 2001; pp. 101–102.

4. Dorahaki, S.; Rashidinejad, M.; Ardestani, S.F.F.; Abdollahi, A.; Salehizadeh, M.R. A Peer-to-Peer energy trading market model based on time-driven prospect theory in a smart and sustainable energy community. *Sustain. Energy Grids Netw.* **2021**, *28*, 100542. [CrossRef]
5. Feng, C.; Liang, B.; Li, Z.; Liu, W.; Wen, F. Peer-to-peer energy trading under network constraints based on generalized fast dual ascent. *IEEE Trans. Smart Grid* **2022**, *14*, 1441–1453. [CrossRef]
6. 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]
7. Mehdinejad, M.; Shayanfar, H.; Mohammadi-Ivatloo, B. Peer-to-peer decentralized energy trading framework for retailers and prosumers. *Appl. Energy* **2022**, *308*, 118310. [CrossRef]
8. 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]
9. Boyd, S.; Parikh, N.; Chu, E. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*; Now Publishers Inc.: Norwell, MA, USA, 2011.
10. 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]
11. Morstyn, T.; McCulloch, M.D. Multiclass energy management for peer-to-peer energy trading driven by prosumer preferences. *IEEE Trans. Power Syst.* **2018**, *34*, 4005–4014. [CrossRef]
12. 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]
13. 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]
14. Nguyen, D.H. Optimal solution analysis and decentralized mechanisms for peer-to-peer energy markets. *IEEE Trans. Power Syst.* **2020**, *36*, 1470–1481. [CrossRef]
15. 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]
16. 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]
17. Xu, Y.; Low, S.H. An efficient and incentive compatible mechanism for wholesale electricity markets. *IEEE Trans. Smart Grid* **2015**, *8*, 128–138. [CrossRef]
18. Jia, Y.; Wan, C.; Yu, P.; Song, Y.; Ju, P. Security Constrained P2P Energy Trading in Distribution Network: An Integrated Transaction and Operation Model. *IEEE Trans. Smart Grid* **2022**, *13*, 4773–4786. [CrossRef]
19. Exizidis, L.; Kazempour, J.; Papakonstantinou, A.; Pinson, P.; De Greve, Z.; Vallée, F. Incentive-compatibility in a two-stage stochastic electricity market with high wind power penetration. *IEEE Trans. Power Syst.* **2019**, *34*, 2846–2858. [CrossRef]
20. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Bus. Rev.* **2008**, 21260.
21. Kiayias, A.; Russell, A.; David, B.; Oliynykov, R. Ouroboros: A provably secure proof-of-stake blockchain protocol. In Proceedings of the Advances in Cryptology–CRYPTO 2017: 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, 20–24 August 2017; Proceedings, Part I; Springer: Berlin/Heidelberg, Germany, 2017; pp. 357–388.
22. Castro, M.; Liskov, B. Practical byzantine fault tolerance. In Proceedings of the OsDI, New Orleans, LA, USA, 22–25 February 1999; Volume 99, pp. 173–186.
23. Su, Z.; Wang, Y.; Xu, Q.; Fei, M.; Tian, Y.C.; Zhang, N. A secure charging scheme for electric vehicles with smart communities in energy blockchain. *IEEE Internet Things J.* **2018**, *6*, 4601–4613. [CrossRef]
24. Tushar, W.; Saha, T.K.; Yuen, C.; Smith, D.; Poor, H.V. Peer-to-peer trading in electricity networks: An overview. *IEEE Trans. Smart Grid* **2020**, *11*, 3185–3200. [CrossRef]
25. Tushar, W.; Saha, T.K.; Yuen, C.; Smith, D.; Ashworth, P.; Poor, H.V.; Basnet, S. Challenges and prospects for negawatt trading in light of recent technological developments. *Nat. Energy* **2020**, *5*, 834–841. [CrossRef]
26. Tushar, W.; Yuen, C.; Saha, T.K.; Morstyn, T.; Chapman, A.C.; Alam, M.J.E.; Hanif, S.; Poor, H.V. Peer-to-peer energy systems for connected communities: A review of recent advances and emerging challenges. *Appl. Energy* **2021**, *282*, 116131. [CrossRef]
27. Chen, S.; Mi, H.; Ping, J.; Yan, Z.; Shen, Z.; Liu, X.; Zhang, N.; Xia, Q.; Kang, C. A blockchain consensus mechanism that uses Proof of Solution to optimize energy dispatch and trading. *Nat. Energy* **2022**, *7*, 495–502. [CrossRef]
28. Wang, N.; Zhou, X.; Lu, X.; Guan, Z.; Wu, L.; Du, X.; Guizani, M. When energy trading meets blockchain in electrical power system: The state of the art. *Appl. Sci.* **2019**, *9*, 1561. [CrossRef]
29. Blom, F. A Feasibility Study of Blockchain Technology as Local Energy Market Infrastructure. Master's Thesis, NTNU, Trondheim, Norway, 2018.
30. Jindal, A.; Aujla, G.S.; Kumar, N. SURVIVOR: A blockchain based edge-as-a-service framework for secure energy trading in SDN-enabled vehicle-to-grid environment. *Comput. Netw.* **2019**, *153*, 36–48. [CrossRef]
31. Saini, V.K.; Purohit, C.S.; Kumar, R.; Al-Sumaiti, A.S. Proof of Work Consensus Based Peer to Peer Energy Trading in the Indian Residential Community. *Energies* **2023**, *16*, 1253. [CrossRef]
32. Pop, C.; Cioara, T.; Antal, M.; Anghel, I.; Salomie, I.; Bertoncini, M. Blockchain based decentralized management of demand response programs in smart energy grids. *Sensors* **2018**, *18*, 162. [CrossRef] [PubMed]

33. Yang, J.; Paudel, A.; Gooi, H.B.; Nguyen, H.D. A Proof-of-Stake public blockchain based pricing scheme for peer-to-peer energy trading. *Appl. Energy* **2021**, *298*, 117154. [CrossRef]
34. Wang, T.; Guo, J.; Ai, S.; Cao, J. RBT: A distributed reputation system for blockchain-based peer-to-peer energy trading with fairness consideration. *Appl. Energy* **2021**, *295*, 117056. [CrossRef]
35. Dong, J.; Song, C.; Liu, S.; Yin, H.; Zheng, H.; Li, Y. Decentralized peer-to-peer energy trading strategy in energy blockchain environment: A game-theoretic approach. *Appl. Energy* **2022**, *325*, 119852. [CrossRef]
36. 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. [CrossRef]
37. Boyd, S.; Boyd, S.P.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2010.
38. Guo, Z.; Qin, B.; Guan, Z.; Wang, Y.; Zheng, H.; Wu, Q. A High-Efficiency and Incentive-Compatible Peer-to-Peer Energy Trading Mechanism. *IEEE Trans. Smart Grid* **2023**. [CrossRef]
39. Ping, J.; Yan, Z.; Chen, S. A privacy-preserving blockchain-based method to optimize energy trading. *IEEE Trans. Smart Grid* **2022**, *14*, 1148–1157. [CrossRef]
40. Shamir, A. How to share a secret. *Commun. ACM* **1979**, *22*, 612–613. [CrossRef]
41. Brickell, E.F. Some ideal secret sharing schemes. *J. Comb. Math. Comb. Comput.* **1989**, *6*, 105–113.
42. Harn, L.; Lin, C. Detection and identification of cheaters in (t, n) secret sharing scheme. *Des. Codes Cryptogr.* **2009**, *52*, 15–24. [CrossRef]