Hindawi
Security and Communication Networks
Volume 2022, Article ID 6911463, 12 pages
https://doi.org/10.1155/2022/6911463

Research Article

Toward Privacy-Preserving Blockchain-Based Electricity Auction for V2G Networks in the Smart Grid

Weijian Zhang,[1] **Wen Yang**,[2] **Cen Chen**,[2] **Nuannuan Li**,[2] **Zijian Bao**,[3] **and Min Luo**[3]

[1]*State Grid Henan Electric Power Company, Zhengzhou, China*
[2]*State Grid Henan Electric Power Research Institute, Zhengzhou, China*
[3]*Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China*

Correspondence should be addressed to Min Luo; mluo@whu.edu.cn

Received 26 January 2022; Accepted 19 May 2022; Published 16 June 2022

Academic Editor: Jie Cui

With the development of electric vehicle (EV) technology, EV has become a key component in the future smart grid. Due to the sheer large number of EVs on the road, the emerging vehicle-to-grid (V2G) technology, which allows for two-way electrical flows between EVs and the power grid, is gaining traction. However, establishing a fair and private electricity exchange scheme has gradually become a critical challenge. The emergence of blockchain technology offers a novel approach for resolving this issue. In this study, we overview the opportunities and challenges of blockchain in the smart grid. Then, we provide a privacy-preserving blockchain-based electricity auction scheme for V2G networks in smart grid. In particular, we exploit PS group signatures to keep the privacy of EVs or charging stations and leverage blockchain to provide automated auction execution. With our mechanism, the identity of EV/charging station is conditionally protected. In case of an emergency, the trusted authority (i.e., the group manager) can open the identity. Meanwhile, we present the security analysis to prove our scheme's security. Finally, we implement the experiment to evaluate efficiency. The experimental results show that our proposal is efficient and suitable for V2G networks.

1. Introduction

The smart grid [1,2], also known as "power grid 2.0," is the intellectualization of the electrical grid. It uses sophisticated sensing and measurement technology, advanced equipment technology, control methods, and decision support system technology to create an integrated, high-speed two-way communication network. It has the tremendous potential of making the electricity system more secure, dependable, cost-effective, and efficient. It can dispatch and control all components of the network according to its own needs and realize the intelligence, transparency, automation, and controllability of the grid.

Predictably, the smart grid will be widely used in all aspects of people's life. Among them, vehicle-to-grid (V2G) system is envisaged as a key component of the smart grid [3]. The research on EVs is in full bloom, reconstructing the ecological chain of the automobile industry. For example, Tesla, the largest electric vehicle company in the United States, has a market value of trillion [1].

In particular, our paper adopts the V2G network model as shown in Figure 1. The EVs can get electricity from the charging stations or other EVs. They can also sell their surplus power resources to get paid. The aggregators are responsible for the interaction and power arrangement of EVs and charging stations in the smart grid. Electricity resources can be flexibly scheduled between *vehicle-to-vehicle* and *vehicle-to-charging* stations to maximize energy use. Both approaches are helpful for providing available and cheap renewable energy sources.

However, electricity distribution is a difficult problem for smart grid applications. An auction scheme is expected to alleviate this problem [4,5]. Using an auction is a straightforward idea and seems to be easy. However, we have to face new challenge #1: in a normal auction system, there is always a centralized manager to conduct an auction scheme.

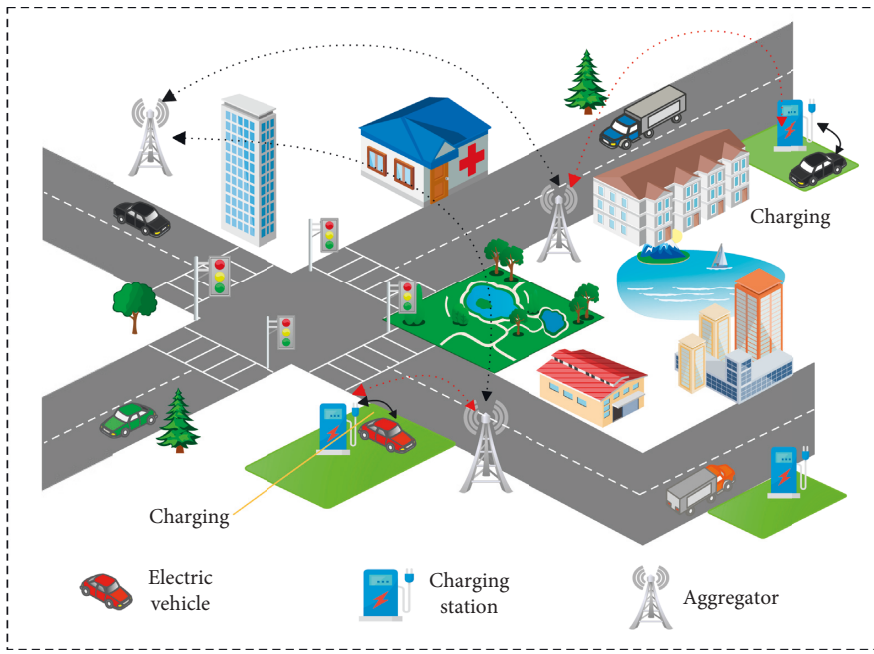

Figure 1: V2G architecture.

Although this way is efficient and convenient, there exist two problems: (1) there may be a single point of failure and 2) the auctioneer may be malicious. The auctioneer can unite users and infringe on the interests of the compliant user, thereby destroying the trust between the sellers, buyers, and auctioneers. We ask the question that "is there exist a decentralized auction scheme without a single point of failure?"

Fortunately, the answer is yes. With the advent of blockchain technology, a decentralized auction is possible. The blockchain technology [6,7], which has been rapidly rising in recent years, has been used in a variety of applications, including medical, Internet of things, and digital finance. It can bring new opportunities and challenges to the auction scheme and the smart grid. Blockchain is a decentralized shared ledger and database, which stores and verifies data using a chain-based data structure. Each block is made up of a set of transactions that are committed by network peers using a predetermined consensus procedure. All participants work together to maintain and supervise the data storage. It has the characteristics of decentralization, transparency, immutability, and security properties. The smart contract [8] supports a program to execute securely in a decentralized environment, making blockchain a powerful tool for building a self-organized system. In particular, Hyperledger Fabric [9], as a widely known project based on blockchain technology, allows users to complete data calculations securely and reliably on the chain through the deployment of smart contracts.

However, things are not as simple as they seem. Challenge #2 emerges: the openness and transparency of blockchain data often conflict with privacy protection. For example, in the auction process, we do not want the identity of users, the dealing price of the auction, and other information to be disclosed to others. For instance, the Australian Information Commissioner released the survey results, confirming that Uber violated the privacy of more than one million Australians [10]. A series of privacy-preserving measures should be implemented. However, existing solutions, such as Zerocash [11] and Monero [12], cannot be directly applied. Thanks to the emergence of anonymous authentication [13], for example, *CL signature* [14], *BBS + signature* [15], and *PS signature* [16], they can greatly alleviate the privacy contradiction between users and servers. In particular, the user is authorized by a trusted third party, and then, the user can request services from the server. The server does not need to know a user's identity but only needs to know that the user has a legal identity.

1.1. Our Contributions. Our contributions are listed as follows:

(1) Although there have been several discussions about the use of blockchain in the smart grid, few literature studies summarize the topic comprehensively. One principal goal of this study was to investigate the role of blockchain in the smart grid and summarize its usage in the smart grid.

(2) By leveraging the PS group signature, we first propose a privacy-preserving blockchain-based electricity auction for V2G networks. In particular, the bidder can send the auction request to the manager anonymously. Then, the manager invokes the smart contracts on the blockchain to automate the auction protocol. Finally, the bidder and auctioneer complete the electricity transaction.

(3) We analyze the scheme's security properties and provide experiments to show our system's computation costs of the offchain and onchain parts.

1.2. Organization. The structure of this study is as follows: in Section 2, we review the related work. In Section 3, we provide the related building blocks. In Section 4, we summarize the opportunities and challenges of the combination of blockchain and the smart grid. In Section 5, we give the system model, threat model, and design goals. In Section 6, we present the detailed construction. In Section 7, we analyze the security of our scheme. In Section 8, we point out some points that are not considered in this study and give the possible work direction in the future. In Section 9, we present the experimental results, including computation costs of the offchain and onchain parts. In the end, we give the conclusion in Section 10.

2. Related Work

At present, there have been several research works on auction schemes in the smart grid. Hahn et al. [17] provided a smart contract-based decentralized transactive energy auctions. The auction method uses a second-price auction, ensuring that bidders make honest bids. They implemented the contracts on the Ethereum blockchain. Wang et al. [18] presented a decentralized electricity transaction mode of microgrid based on blockchain and the double auction mechanism. They designed an adaptive aggressiveness strategy to allow traders to alter their bids in real time in response to the market changes. Ramachandran et al. [19] introduced a hybrid optimization method for decentralized energy resource management. Based on risk and competitive equilibrium price prediction, they implemented a profit-maximizing adaptive bidding technique. To cut the cost, a hybrid immune system-based particle swarm optimization is utilized to generate the model, which assumes actual power market pricing. Stubs et al. [20] proposed multitier double auctions for smart energy distribution grids using blockchain technology. The scheme can reduce blockchain workload by aggregating energy usage and generation. Edge computing is also used to improve reliability and response time. Ma et al. [21] proposed an efficient pricing method that can prevent users' cheating. They provided an enhanced Arrow-d'Aspremont-Gerard-Varet (AGV), a complex auction mechanism, to ensure truthfulness. Using an incentive mechanism, the user's payment is linked to their credit for consumption. Wen et al. [22] provided an effective search encryption auction system for marketing in smart grid. The scheme employs public key encryption with keyword search technologies to allow energy sellers to query relevant offers while preserving the anonymity of energy purchasers. For convenience, their system also supports conjunctive keyword search. Li et al. [23] concentrated on the problem of creating a secure online power market. They suggested an online double auction technique with differential privacy based on two building blocks: a Laplace-based winner selection mechanism and an exponential-based allocation algorithm. Zhou et al. [24] presented an auction mechanism for the geo-distributed cloud. By combining principles from the Gibbs sampling method and the alternating direction approach of the multiplier, they proposed a decentralized social welfare maximization algorithm.

3. Preliminary

In this section, we review the building blocks of our scheme, such as *bilinear pairing*, *PS signature*, *commitment*, *hashed ElGamal encryption*, *blockchain*, *smart contract*, *Hyperledger Fabric*, and *auction scheme*.

3.1. Notions

Definition 1 (Bilinear Pairing). Let $(\mathbb{G}, \mathbb{G}_T)$ be a bilinear map such that $\tilde{e}$: $\mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$, where p is the order for both $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$. Note that we use type 3 bilinear pairing, where $\mathbb{G}_1 \neq \mathbb{G}_2$, and there is no efficient computable homomorphism between them. The bilinear map should satisfy the following properties:

(1) Bilinear: given any two elements $a, b \in Z_q^*$ and $\forall x \in \mathbb{G}_1, y \in \mathbb{G}_2$, $e(x^a, y^b) = e(x, y)^{ab}$.
(2) Nondegenerate: for $\forall x \in \mathbb{G}_1, y \in \mathbb{G}_2$, $e(x, y) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ represents the identity element in $\mathbb{G}_T$.
(3) Efficient Computability: for $\forall x \in \mathbb{G}_1, y \in \mathbb{G}_2$, $e(x, y)$ is efficiently computable.

3.2. PS Signature. PS signature was proposed by Pointcheval et al. in [16]. It utilizes type 3 bilinear pairing to construct a randomized signature. In particular, this original signature σ can be randomized to a new randomized signature $\sigma\prime$, which can be applied to many privacy-preserving application scenarios, and achieve well performance simultaneously. The detailed algorithms are as follows.

Definition 2 (PS Signature). *It consists of 4 probabilistic polynomial time ($\mathcal{PPT}$) algorithms.*

(1) Setup(1^n): given a system security parameter 1^n, a set of public parameters $\text{pp} = \{\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, p\}$ are outputs. We denote $\mathbb{G}_1^* = \mathbb{G}_1 / \{1_{\mathbb{G}_1}\}$, and p is the order of $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$.
(2) Keygen(pp): this algorithm randomly chooses $\tilde{g} \in \mathbb{G}_2$ and $(x, y) \in \mathbb{Z}_p^2$, then computes $(\tilde{X}, \tilde{Y}) \leftarrow (\tilde{g}^x, \tilde{g}^y)$, sets sk as (x, y), and sets pk as $(\tilde{g}, \tilde{X}, \tilde{Y})$.
(3) Sign(m, sk): given a message $m \in \{0, 1\}^*$, it randomly chooses $h \in \mathbb{G}_1^*$ and then computes $\sigma = (h, h^{(x+y \cdot m)})$.
(4) Verify(m, σ, pk): given a message $m \in \{0, 1\}^*$, a signature σ, and a public key pk , it parses σ as (σ_1, σ_2) and checks whether $\sigma_1 \neq 1_{\mathbb{G}_1}$ and $e(\sigma_1, \tilde{X} \cdot \tilde{Y}^m) = e(\sigma_2, \tilde{g})$. If true, it outputs 1; otherwise, 0.

PS signature is EUF-CMA under the LRSW assumption [25]. Meanwhile, a group signature can be easily obtained from the PS signature [16].

Definition 3 (Group Signature Based on PS Signature). It consists of 6 $\mathcal{PPT}$ algorithms.

(1) GSetip(1^n): the group manager runs Setup and Keygen to obtain (sk, pk), where $\text{sk} = (x, y)$,

$\mathrm{pk} = (\widetilde{g}, \widetilde{X}, \widetilde{Y})$, and then, it sets $\mathrm{gsk} := \mathrm{sk}$ and $\mathrm{gpk} := (\mathrm{pk}, g)$.

(2) KPI join $(i, 1^n)$: the user i generates its private/public key pair $(\mathrm{sk}_i, \mathrm{pk}_i)$ and then sends pk_i to the certificate authority.

(3) G Join: the user randomly chooses s_i, generates $(\delta, \widetilde{\delta}) \leftarrow (g^{s_i}, \widetilde{Y}^{s_i})$ and a signature $\theta \leftarrow \mathrm{Sign}(\mathrm{sk}_i, \delta)$, and then sends them to the group manager. The group manager checks whether θ is valid and $e(\delta, \widetilde{Y}) = e(g, \widetilde{\delta})$. Then, the user gives the zero-knowledge proof that he owns the s_i. After that, the group manager generates a random number r and computes $\sigma \leftarrow (\sigma_1, \sigma_2) \leftarrow (g^r, (g^x \cdot \delta^y)^r)$, which is a valid signature on s_i. In the end, the group manager stores $(i, \delta, \theta, \widetilde{\delta})$ in a secret register and sends $\widetilde{\sigma}$ and $e(\sigma_1, \widetilde{Y})$ to the user, where $\mathrm{gsk}_i = (s_i, \sigma, e(\sigma_1, \widetilde{Y}))$.

(4) G Sign (gsk_i, m): the user needs to randomize σ using a random number t and computes $(\sigma_1', \sigma_2') \leftarrow (\sigma_t^1, \sigma_2^t)$ along with a signature of knowledge of s_i. The detailed steps are as follows: the user randomly chooses $k \in \mathbb{Z}_p$ and computes $c \leftarrow \mathscr{H}(\sigma_1', \sigma_2', e(\sigma_1, \widetilde{Y})^{k \cdot t}, m)$, where $\mathscr{H}$ is a secure hash function. Finally, the user computes $s \leftarrow k + c \cdot s_i$ and outputs $(\sigma_1', \sigma_2', c, s)$ as the group signature μ on the message m.

(5) G Verify (gpk_i, m, μ): to verify whether the signature $(\sigma_1', \sigma_2', c, s)$ is valid, the verifier computes $T \leftarrow e(\sigma_{1\prime}, \widetilde{X})^c \cdot e(\sigma_{2\prime}, \widetilde{g})^{-c} \cdot e(\sigma_{1\prime}, Y)^s$ and $c = \mathscr{H}(\sigma_1', \sigma_2', T, m)$. If it is valid, it outputs 1; otherwise, 0.

(6) G Open (gmsk, m, μ): when we need to open one user's identity, the group manager searches in the list $(i, \delta_i, \theta_i, \widetilde{\delta}_i)$ and checks whether $e(\sigma_{2\prime}, \widetilde{g}) \cdot e(\sigma_{1\prime}, \widetilde{X})^{-1} = e(\sigma_{1\prime}, \widetilde{\delta}_i)$ until he gets a match. He then outputs a corresponding (i, δ_i, θ_i) with a proof of knowledge $\widetilde{\delta}_i$.

3.3. Cryptographic Commitment. A commitment scheme enables a user to commit to a specific statement, which is hidden from others during the *commit* phase, but visible to the users during the ***open*** phase. The following two properties belong to a commitment scheme:

(1) *Binding*: after committing to a statement, the committer is unable to alter it.

(2) *Hiding*: before the committer opens the commitment, the receiver knows nothing about the committed statement.

We give the Pedersen commitment [26] as follows:

(1) Setup (1^n): given a system security parameter 1^n, a set of public parameters $\mathrm{pp} = \mathscr{G}, p, g, h$ is outputs, where p is the order of $\mathscr{G}$, and g, h are the generators of $\mathscr{G}$.

(2) Commit $(m; r)$: on inputs a message $m \in \mathbb{Z}_p$, this algorithm randomly chooses $r \in \mathbb{Z}_p$ and outputs $c \leftarrow g^m h^r$.

(3) Open (c, m, r): if $c = g^m h^r$, this algorithm outputs 1; otherwise, 0.

3.4. Hashed ElGamal Encryption. The ElGamal encryption system is a asymmetric key encryption system based on the Diffie–Hellman key exchange [27]. Here, we use a variant of ElGamal encryption, called hashed ElGamal encryption [28]. It includes the 4 $\mathscr{PPT}$ algorithms listed as follows:

(1) *Setup* $((1^n))$: given a security parameter 1^n, it outputs $\mathrm{pp} = (\mathbb{G}, g, h, p, \mathscr{H})$, where g, h are generators of the cyclic group $\mathbb{G}$ of prime order p, and $\mathscr{H}$ is a hash function $\{0,1\}^* \longrightarrow (0,1)^n$.

(2) *Keygen* (pp): it outputs $(\mathrm{sk}, \mathrm{pk})$, where $\mathrm{sk} \leftarrow \mathbb{Z}_p$ and $\mathrm{pk} = g^{\mathrm{sk}}$.

(3) *Encrypt (pk,m)*: it chooses $r \leftarrow \mathbb{Z}_p$, computes $c_1 = g^r$, $c_2 = \mathscr{H}(\mathrm{pk}^r) \oplus m$, and outputs $c = (c_1, c_2)$.

(4) *Decrypt (sk,c)*: it computes $m = c_2 \oplus \mathscr{H}(c_1^{sk})$.

3.5. Blockchain and Smart Contract. Blockchain is a peer-to-peer decentralized ledger that is based on the Bitcoin concept [29]. It is a multi-technology application paradigm that includes encryption, game theory, decentralized systems, and other technologies. With a certain consensus algorithm, all nodes in the blockchain retain a consistent record. The ledger, as illustrated in Figure 2, is a series of data blocks that include various transactions sent by users in a peer-to-peer network, with the last block always including the hash of the preceding block. The following are the key characteristics of blockchain.

(1) *Decentralization*: as blockchain technology adopts a decentralized structure, there is no centralized management organization. Every node in it has the same rights and obligations. They jointly maintain the data ledger stored in the system.

(2) *Immutability*: once the information is verified and added to the blockchain, it will be stored permanently. It is impossible to update the data in a single block without affecting all following blocks. For example, in the Bitcoin system, unless the attacker has more than 50% of the whole network computing power, it is impossible to regenerate blocks to tamper with the data. Generally, we assume that the data in the blockchain cannot be tampered with.

(3) *Openness and Transparency*: once the transaction is packaged into a block, the block will be broadcast to all nodes, achieving data synchronization. Each node can trace back all the transaction information of any parties in the past.

(4) *Security*: the security of all entries in the blockchain is guaranteed using cryptographic algorithms, such as digital signatures and encryption algorithms.

Furthermore, thanks to Ethereum [8], a well-known blockchain project, the notion of smart contracts has been revitalized with the advent of the *Blockchain 2.0* era. Smart contracts are computer programs that are recorded on a blockchain and run automatically when certain criteria are met. For example, when a stock price is less than a certain value, a predefined smart contract can automatically execute

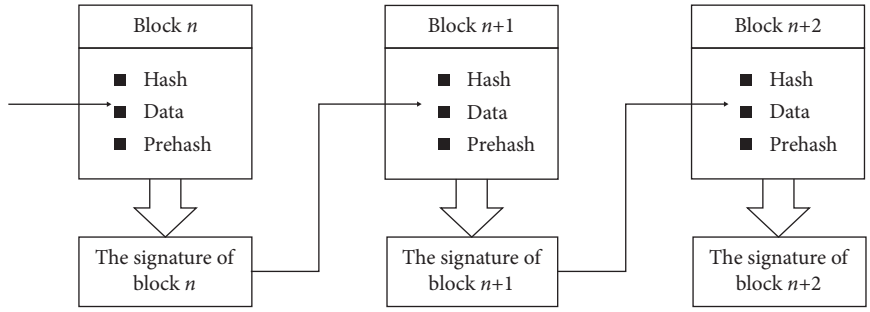

Figure 2: Blockchain structure.

the operation of buying the stock. The smart contract has been extensively used in the Internet of things, product traceability, supply chain finance, insurance, and so on. Based on the access mechanism, blockchain can be generally categorized into public, private, and consortium blockchains. In a public blockchain, each node is free to enter or leave at will. However, nodes without authorization cannot access the data in either a private or consortium blockchain. Consortium blockchain offers a stronger decentralized nature, since there are multiple institutions involved, rather than one in a private blockchain.

3.6. Hyperledger Fabric. Hyperledger Fabric is an open-source, enterprise-level, permission-based consortium blockchain platform [9]. It is underpinned by the modular architecture and offers excellent confidentiality, scalability, flexibility, and extensibility. There are three kinds of nodes in Hyperledger Fabric: the endorsement nodes, the order nodes, and the normal nodes. The endorsement nodes are responsible for endorsing and executing transactions. The order nodes are in charge of packaging transactions into blocks, and the normal nodes always publish the transactions to endorse nodes and receive new blocks from order nodes. This architecture avoids a bottleneck, thanks to the decoupling of node functions, which makes Fabric more efficient. Meanwhile, to protect privacy, private "subnets" are used to communicate among multiple specific network members, which are defined as a channel. Channel contains one or more organizations, which are the interest entities with collaborative relationships. Moreover, in Hyperledger Fabric, the consensus algorithm is designed to be pluggable. Fabric provides some alternative algorithms, such as Solo, Kafka, and Raft. Because there is just one order in which to sort messages and construct blocks in Solo mode, it is most commonly utilized in a testing environment. Raft is a sorting service that supports crash fault tolerance (CFT), which means it can only tolerate half of the fault nodes. Kafka is similar to Raft; however, it has a higher computational cost.

3.7. Auction Scheme. The auction can achieve an effective allocation of electricity resources and ensure the transparency and fair of the process [30]. There are some mainstream auction mechanisms.

(1) *The First-Price Sealed Auctions (FSAs)*: the bidder delivers the bid to the auctioneer in a sealed envelope. After that, the auctioneer opens the envelope and identifies the highest bidder.

(2) *The Second-Price Sealed Auctions (SSAs)*: the process is similar to FSA. The winner only needs to pay the second highest bid, which eliminates the bidder's concerns about the difference between the first and second prices.

(3) *The Open Ascending Bid Auctions (English Auctions)*: the bidders increase the bids gradually until no one wants to pay more than the current highest bid. The highest bidder gets the auction item at his price.

(4) *The Open Descending Bid Auctions (Dutch Auctions)*: the auctioneer gradually reduces the price from a preset high price until there is a bidder willing to pay the current price.

4. Opportunities and Challenges of Blockchain in the Smart Grid

Blockchain technology has been widely concerned in industry and academia and has become one of the new infrastructures in the digital age. It is expected to resolve some issues in the smart grid, promoting some research for the combination of blockchain and the smart grid.

We summarize the possible usage of blockchain as follows:

A Decentralized Database with Immutability: compared with the traditional database, blockchain can be regarded as a special kind of database, which only supports the adding operation. We can use blockchain to store critical data in the smart grid.

Automated Smart Contracts in Decentralized Environment: a smart contract is a piece of code that can be executed automatically by multiple consensus nodes in a decentralized environment, opening up new possibilities for electricity management in the smart grid. For example, a self-organized electricity auction system can be built by smart contracts, the electricity sellers and buyers only need to submit the demand information to the chain, and the smart contracts can automatically match the demand of both sides according to a predetermined algorithm.

Incentive Mechanisms: traditional electricity management requires a centralized organization, which may

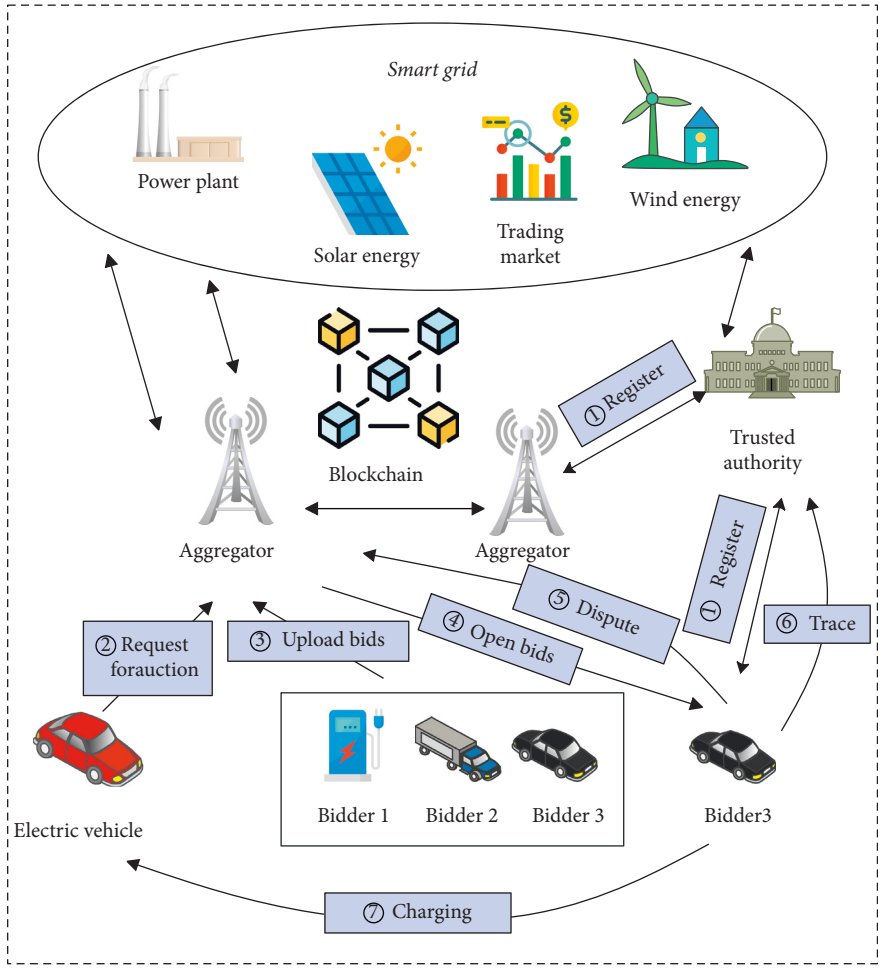

Figure 3: System model (*Note*. Steps ∗∗∗∗5 and ∗∗∗∗6 are only performed when there is a dispute or the identity needs to be opened).

cause a single point of failure. The management may corrupt for the sake of interests, which leads to unfair distribution of electricity. Thus, the current electricity management lacks a well-designed incentive mechanism to promote the benign behavior of electricity management organizations. Therefore, creating a blockchain-based electricity management system with economic incentives could be a promising direction.

Then, we list the points that need to be considered when using blockchain in the smart grid.

Efficiency: compared with the centralized structure, the decentralized structure and consensus process of blockchain will dramatically reduce the efficiency. The most extreme example is Bitcoin's processing speed of *7 transactions/per second (TPS)*. Generally speaking, we have certain requirements for transaction processing speed. As a result, how to choose a suitable form of blockchain based on the actual needs of real contexts and how to configure the blockchain's underlying data structure are challenges that need our attention.

Supervision: the allocation, sharing, and use of electricity in the smart grid should be recorded in the blockchain. To supervise the rational use of electricity and prevent some malicious acts, we must design appropriate mechanisms.

Privacy Protection: we need to protect the privacy of users, such as hiding the identity of the bidders, the auctioneers, and the auction price.

Selection of Blockchain Types: we have mentioned that blockchains can be divided into three types: public, private, and consortium. The public chain data are completely open, and any node can access the blockchain at any time, while the consortium chain is only for members of a specific group, which is more suitable for the scenario with permission control. Hence, it is more suitable for the scenarios in the smart grid.

5. Problem Overview

In this section, we present the specific system architecture in our paper. Then, we give the threat models and design goals of our system. Last, we show the detailed construction.

5.1. System Model. Figure 3 illustrates the system model in our paper. There are 5 entities in our model: *EV*, *charging station*, *trusted authority*, *aggregator*, and *blockchain*.

EV. EVs, owned by users, are mobile and geographically separated. Users want to charge their EVs or sell surplus electricity to other EVs in the smart grid. That is, an EV can be either a seller or a buyer of electricity. They communicate with the aggregator through the privacy-preserving method, publish their own needs on the blockchain, match the supplier through the auction mechanism, and complete the electricity transaction.

Charging Station. The charging station can charge EVs in their region. They are scattered all over the city.

Trusted Authority (TA). The TA is responsible for initializing the whole system and provides charging services for EVs and charging stations. TA will be offline, except in case of an emergency when we need to trace the identity of an entity (*i.e.*, EVs or charging stations).

Aggregator. The aggregators are responsible for coordinating EVs and charging stations. They act as the decision center to dispatch energy for the V2G network. They have sufficient computing power and storage capacity and jointly maintain a blockchain. In particular, they play the role of auctioneer. They can assist EVs in releasing requirements on the blockchain and matching transactions through auction protocols.

Blockchain. Blockchain is regarded as a tamper-resistant ledger in which smart contracts can provide decentralized program execution. We deploy functions for auctions on smart contracts, thereby ensuring the security of the auction protocol.

5.2. Threat Model. In this study, we assume that (1) the trusted authority is fully trusted and (2) the aggregators are honest but curious. That is, they can execute the protocol honestly but may infer the EV's private information. (3) We also consider that there exists an external adversary (abbreviated as $\mathcal{A}$) that can eavesdrop on the communication channels between the parties. They can access and record transactions on the blockchain. Moreover, (4) an $\mathcal{A}$ may impersonate the EV or the charging station to trick other parties in the electricity auction phase.

5.3. Design Goals. The design goals in our study are listed as follows:

(1) *EV/Charging Station Authentication*. The EV/charging station should be authenticated when they participate in the auction scheme. There is no $\mathcal{PPT}$ $\mathcal{A}$ who can forge their identities.

(2) *EV/Charging Station Privacy*. The EV/charging station should be protected. Even a malicious aggregator or external $\mathcal{A}$ cannot know their true identity, except the statement that they have legal identities.

(3) *Auction Privacy*. The bidding information is hidden, and only the bidding information of the final winner is published on the blockchain.

(4) *Traceability*. The TA can trace the identity of a malicious EV/charging station when needed.

(5) *Accountability*. The scheme needs to ensure the accountability of the auction agreement. That is, the bidder with the cheapest bid must become the winner, and the auctioneer cannot maliciously modify the results. Anyone who doubts the auction results can question the results and draw conclusions.

6. Our Construction

6.1. High-Level Description. Let us briefly give a high-level overview of the scheme. We adopt the *first-price sealed auction*, since its satisfactory performance in real life, and it can be easily combined with privacy-preserving technologies.

(1) *System Initialization*. The TA initializes the whole system and generates public parameters. All aggregators jointly generate the initialization parameters of blockchain and then maintain such a blockchain.

(2) *Register*. EVs, charging stations, and aggregators need to register with TA. Each entity needs to have an account and corresponding amount on the blockchain, namely account:=$\{\mathrm{pk}_{\mathrm{address}}, \mathrm{value}\}$.

(3) *Request for Auctions*. The EV sends its request (*i.e.*, buying the electricity) to the aggregator through the anonymous authentication method. After receiving the message, the aggregator sends the request information to the smart contracts on the blockchain.

(4) *Upload Bids*. When charging stations find the request on the blockchain, they commit their supply requests, including the commitment value of electricity bid, to the aggregator. Note that we use the charging station as the representative for the convenience of expression. Other EVs with surplus power can also participate in the bidding.

(5) *Open Bids*. After a specific time node, the aggregator (also as the auctioneer) needs to open all bids, select the bid which is the cheapest, and send it to the blockchain. Then, the auctioneer helps the EV and the winner to establish a secure channel for electricity exchange.

(6) *Dispute*. To ensure accountability, anyone who needs to spend a small part of the token can question the result of an auction. That is, the winner's bid is not the cheapest. Accordingly, the auctioneer needs to give the corresponding zero-knowledge proof that the winner's bid is cheaper than the challenger's bid and send it to the blockchain. If the proof is not given, the challenger can obtain a certain token as a reward from the auctioneer's deposit.

(7) *Trace*. The TA can trace the identity of a malicious EV/charging station when needed.

6.2. Detailed Process

6.2.1. System Initialization. The TA chooses a system security parameter 1^n and outputs a set of public parameters $\mathrm{pp} = \{\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h, \tilde{g}, p, \mathscr{H}\}$, where g and h are generators of cyclic group $\mathbb{G}_1$, $\tilde{g}$ is the generator of $\mathbb{G}_2$, and p is

the order of $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$. $\mathscr{H}$ is the hash function. TA randomly chooses $\tilde{g} \in \mathbb{G}_2$ and $(x, y) \in \mathbb{Z}_p{}^2$ and then computes $(\tilde{X}, \tilde{Y}) \leftarrow (\tilde{g}^x, \tilde{g}^y)$, and sk is (x, y), and pk is $(\tilde{g}, \tilde{X}, \tilde{Y})$. Aggregators jointly determine blockchain parameters, such as *blockchain type selection*, *block generation speed,* and *block size*. Aggregators can jointly maintain a blockchain due to their considerable computing and storage capacity.

6.2.2. Register. The EV $\mathrm{id}_{\mathrm{ev}}$ generates its private/public key pair $(\mathrm{sk}_{\mathrm{ev}}, \mathrm{pk}_{\mathrm{ev}})$ and then sends $\mathrm{pk}_{\mathrm{ev}}$ to the TA. The EV randomly chooses s_{ev}, generates $(\delta, \tilde{\delta}) \leftarrow (g^{s_{\mathrm{ev}}}, \tilde{Y}^{s_{\mathrm{ev}}})$ and a signature $\theta \leftarrow \mathrm{Sign}(\mathrm{sk}_{\mathrm{ev}}, \delta)$, and then sends them along with $\mathrm{id}_{\mathrm{ev}}$ to the TA. The TA checks whether θ is valid and then whether $e(\delta, \tilde{Y}) = e(g, \tilde{\delta})$. Then, the EV gives the zero-knowledge proof that he owns the s_{ev}. After that, the TA generates a random number r and computes $\sigma \leftarrow (\sigma_1, \sigma_2) \leftarrow (g^r, (g^x \cdot \delta^y)^r)$. It is a valid signature on s_{ev}. In the end, the TA stores $(\mathrm{id}_{\mathrm{ev}}, \delta, \theta, \tilde{\delta})$ in a secret register and sends σ and $e(\sigma_1, \tilde{Y})$ to the EV. Finally, the EV's group public key is δ, and group private key is $\mathrm{gsk}_{\mathrm{ev}} = (s_{\mathrm{ev}}, \sigma, e(\sigma_1, \tilde{Y}))$.

Similarly, the charging station $\mathrm{id}_{\mathrm{cs}}$ generates its private/public key pair $(\mathrm{sk}_{\mathrm{cs}}, \mathrm{pk}_{\mathrm{cs}})$, randomly chooses s_{cs}, and generates $(\delta_{\mathrm{cs}}, \tilde{\delta}_{\mathrm{cs}}) \leftarrow (g^{s_{\mathrm{cs}}}, \tilde{Y}^{s_{\mathrm{cs}}})$. Through the similar interaction with TA, the charging station obtains its own group public key is $\delta\imath$, and group private key is $\mathrm{gsk}_{\mathrm{cs}} = (s_{\mathrm{cs}}, \sigma_{\mathrm{cs}}, e(\sigma_{1\,\mathrm{cs}}, \tilde{Y}))$. The TA stores $(\mathrm{id}_{\mathrm{cs}}, \delta_{\mathrm{cs}}, \theta_{\mathrm{cs}}, \tilde{\delta}_{\mathrm{cs}})$ in his secret register. The aggregator generates its private/public key pair $(\mathrm{sk}_{\mathrm{ag}}, \mathrm{pk}_{\mathrm{ag}})$ and then sends $\mathrm{pk}_{\mathrm{ag}}$ to the TA. The TA stores it in the secret register.

Meanwhile, any entity that wants to participate in electricity trading (*i.e.*, auction) needs to have an account on the blockchain, and there are a certain number of tokens in the account for purchasing or paying electricity. We use account $:= \{\mathrm{pk}_{\mathrm{address}}, \mathrm{value}\}$ to abstract the account. The aggregator, as an auctioneer, also needs to deposit enough tokens in the smart contract of the blockchain, which is in charge of the TA. When the illegal behavior of the auctioneer is detected, it can be punished.

6.2.3. Request for Auctions. When an EV needs to be charged, it sends a charging request CR to the aggregator, using the anonymous authentication based on PS group signatures. The specific operations are as follows:

The EV sets the required parameters $\mathrm{PA} := \{\mathrm{id}, \mathrm{eq}, \mathrm{model}, t_1, t_2, \mathrm{pr}\}$: id ID represents the unique serial number of the auction, eq is the required electric quantity, model is the charging model, t_1 is the deadline for accepting bids, t_2 is the deadline for the whole auction, and pr is the maximum price of a kilowatt-hour accepted by himself.

The EV first randomly chooses a k_{ev} and the public key $K = \tilde{g}^{k_{ev}}$, where $(k_{\mathrm{ev}}, K_{\mathrm{ev}})$ is the temporary public-private key pair for the future usage.

Then, the EV uses hashed ElGamal encryption to encrypt PA, and the ciphertext is (c_1, c_2), where $c_1 = \tilde{g}^{r\prime}$ and $c_2 = \mathscr{H}(\tilde{X}^{r\prime}) \oplus (\mathrm{PA} \| K_{\mathrm{ev}})$. Then, the EV uses the $GSign((\mathrm{gsk}_{\mathrm{ev}}))$ to sign the message $\mathrm{PA} \| K_{\mathrm{ev}}$. In particular, the EV needs to randomize σ using a random number t and computes $(\sigma_1', \sigma_2') \leftarrow (\sigma_1^t, \sigma_2^t)$.

Then, he randomly chooses $k \in \mathbb{Z}_p$ and computes $c \leftarrow \mathscr{H}(\sigma_1', \sigma_2', e(\sigma_1, \tilde{Y})^{k \cdot t}, \mathrm{PA} \| K_{\mathrm{ev}})$, where $\mathscr{H}$ is a secure hash function. Finally, the EV computes $s \leftarrow k + c \cdot s_{\mathrm{ev}}$ and outputs $(\sigma_1{}', \sigma_2{}', c, s)$ as the group signature μ on the message $\mathrm{PA} \| K_{\mathrm{ev}}$.

Then, the EV keeps the $(k_{\mathrm{ev}}, K_{\mathrm{ev}})$ in his secret register and sends to the aggregator the request tuple for auctions $\mathrm{CR} := \{\sigma_1', \sigma_2', c_1, c_2, c, s, \mathrm{eq}, \mathrm{model}, t_1, \mathrm{pr}\}$.

When the aggregator receives the request CR, he decrypts the ciphertext (c_1, c_2) and checks whether the PS group signature is valid. If it is valid, he releases the auction information to the blockchain. The specific operations are as follows:

The aggregator decrypts the ciphertext (c_1, c_2), by computing $\mathrm{PA} = c_2 \oplus \mathscr{H}(c_1^x)$.

To verify whether the signature $(\sigma_1', \sigma_2', c, s)$ is valid, the aggregator computes $T \leftarrow e(\sigma_{1\prime}, \tilde{X})^c \cdot e(\sigma_{2\prime}, \tilde{g})^{-c} \cdot e(\sigma_{1\prime}, \tilde{Y})^s$ and $c = \mathscr{H}(\sigma_1', \sigma_2', T, \mathrm{PA})$. If it is valid, the aggregator sends the auction information $\{\mathrm{eq}, \mathrm{model}, t_1, \mathrm{pr}\}$ to the smart contracts deployed on the blockchain; otherwise, the aggregator rejects the request.

6.2.4. Upload Bids. The charging stations find the request on the blockchain, and they commit their supply requests, including the commitment value of electricity bid, to the aggregator before t_1. The specific operations are as follows:

The charging station chooses the price v of a kilowatt-hour, randomly chooses r, computes the commitment $\mathrm{cm}: = g^r h^v$, and sets the bid as $\mathrm{bid}: = \{\mathrm{id}, \mathrm{pk}_{\mathrm{address}}, \mathrm{cm}\}$, where $i\ d$ represents the auction number participating in the bidding and $\mathrm{pk}_{\mathrm{address}}$ indicates the address of the charging station on the blockchain. At the same time, some funds need to be sent to the aggregator as deposits.

6.2.5. Open Bids. All bidders open the committed value to the auctioneer, and the auctioneer gets the highest bid and uploads the winner's identity (*i.e.*, *address*) to the blockchain before t_2. The winner can sell his electricity with the auction requester. Suppose the requester has α tokens, the smart contract will transfer β tokens $((\beta = v * \mathrm{eq}))$ in the requester's deposit to the winner's account, and the rest $((\alpha - \beta))$ will be returned to the requester's account. Then, the winner needs to prove that he has a legal identity; *i.e.*, he needs to sign the auction results.

The charging station first randomly chooses a k_{cs} and the public key $K_{\mathrm{cs}} = \tilde{g}^{k_{\mathrm{cs}}}$, where $(k_{\mathrm{cs}}, K_{\mathrm{cs}})$ is the temporary public-private key pair for the future usage.

Then, the charging station uses hashed ElGamal encryption to encrypt $\mathrm{bid} \| K_{\mathrm{cs}}$, and the ciphertext is $(c_{1\prime}, c_{2\prime})$, where $c_{1\prime} = \tilde{g}^{r\prime\prime}$ and $c_{2\prime} = \mathscr{H}(\tilde{X}^{r\prime\prime}) \oplus (\mathrm{bid} \| K_{\mathrm{cs}})$.

Then, the charging station gives a zero-knowledge proof that $\mathcal{SOK} = \{(\mathrm{sk}_{\mathrm{address}}): \mathrm{pk}_{\mathrm{address}} = g^{\mathrm{sk}_{\mathrm{address}}}\}(\mathrm{bid}\|K_{\mathrm{cs}})$.

Then, the charging station uses the *GSign* $((\mathrm{gsk}_{\mathrm{cs}}))$ to sign the message K_{cs}. In particular, the charging station needs to randomize σ_{cs} using a random number $t\prime$ and computes $(\sigma_{1\,\mathrm{cs}}{}', \sigma_{2\,\mathrm{cs}}{}') \leftarrow (\sigma_{1\,\mathrm{cs}}^{t\prime}, \sigma_{2\,\mathrm{cs}}^{t\prime})$. Then, he randomly chooses $k\prime \in \mathbb{Z}_p$ and computes $c\prime \leftarrow \mathcal{H}(\sigma_{1\,\mathrm{cs}}{}', \sigma_{2\,\mathrm{cs}}{}', e(\sigma_{1\,\mathrm{cs}}, \widetilde{Y})^{k\prime \cdot t\prime}, \mathrm{bid}\|K_{\mathrm{cs}})$, where $\mathcal{H}$ is a secure hash function. Finally, the EV computes $s\prime \leftarrow k\prime + c\prime \cdot s_{\mathrm{cs}}$ and outputs $(\sigma_{1\,\mathrm{cs}}{}', \sigma_{2\,\mathrm{cs}}{}', c\prime, s\prime)$ as the group signature $\mu\prime$ on the message $\mathrm{bid}\|K_{\mathrm{cs}}$.

Then, the charging station keeps the $(k_{\mathrm{cs}}, K_{\mathrm{cs}})$ in his secret register and sends to the aggregator the tuple $\{\sigma_{1\,\mathrm{cs}}{}', \sigma_{2\,\mathrm{cs}}{}', c_{1\prime}, c_{2\prime}, c\prime, s\prime, \mathcal{SOK}\}$.

When the aggregator receives the tuple, he decrypts the ciphertext $(c_{1\prime}, c_{2\prime})$ and checks whether the PS group signature is valid. If it is valid, he sends K_{cs} to the EV and sends K_{ev} to the charging station. Then, the charging station and the EV can establish trusted communication for power supply operation. The specific operations are as follows:

The aggregator decrypts the ciphertext $(c_{1\prime}, c_{2\prime})$, by computing $\mathrm{PA} = c_{2'} \oplus \mathcal{H}(c_1^x)$.

To verify whether the signature $(\sigma_{1\,\mathrm{cs}}', \sigma_{2\,\mathrm{cs}}', c\prime, s\prime)$ is valid, the aggregator computes $T\prime \leftarrow e(\sigma_{1\prime\,\mathrm{cs}}, \widetilde{X})^{c\prime} \cdot e(\sigma_{2\prime\,\mathrm{cs}}, \widetilde{g})^{-c\prime} \cdot e(\sigma_{1\prime\,\mathrm{cs}}, \widetilde{Y})^{s}$ and $c = \mathcal{H}(\sigma_{1\,\mathrm{cs}}', \sigma_{2\,\mathrm{cs}}', T\prime, \mathrm{bid}\|K_{\mathrm{cs}})$.

If it is valid, the aggregator sends K_{cs} to the EV and sends K_{ev} to the charging station. The EV computes $k = K_{\mathrm{cs}}^{k_{\mathrm{ev}}}$, and the aggregator computes $k = K_{\mathrm{ev}}^{k_{\mathrm{cs}}}$ for secure communications. They can conduct offline power supply operation after negotiation.

Handling Malicious Winners. There is a situation that when a malicious charging station wins the auction, it does not carry out subsequent operations. To prevent this situation, the aggregator can call the smart contract to deduct the deposit of the charging station.

6.2.6. Dispute. To ensure the accountability of the auction scheme, we allow anyone to question the auction results. That is, the price of the winner's bid is not the lowest price.

One can choose any bid participating in the auction to compare with the winner's price and upload the request to the smart contract by consuming a small amount of token.

The auctioneer generates a zero-knowledge proof to prove that the value in the designated bid commitment is higher than the winner's price. If the auctioneer is unable to make the proof within the specified time, the smart contract will deduct a certain proportion of the auctioneer's deposit as a punishment.

6.2.7. Trace. When we need to open someone, the TA searches in the list $(\mathrm{id}_i, \delta_i, \theta_i, \widetilde{\delta}_i)$ and checks whether $e(\sigma_2, \widetilde{g}) \cdot e(\sigma_1, \widetilde{X})^{-1} = e(\sigma_1, \widetilde{\delta}_i)$ until he gets a match. He then outputs a corresponding (i, δ_i, θ_i) with a proof of knowledge $\widetilde{\delta}_i$.

7. Security Analysis

In this section, we briefly analyze the properties of the scheme.

EV/Charging Station Authentication. As the PS group signature is EUF-CMA under LRSW assumption, all anonymous authentications of EVs and charging stations use PS group signatures, namely $(\sigma_1', \sigma_2', c, s)$ and $(\sigma_{1\,\mathrm{cs}}{}', \sigma_{2\,\mathrm{cs}}{}', c\prime, s\prime)$. So, we can reduce the authentication security to the signature's security.

EV/Charging Station Privacy. The EVs and charging stations use the anonymous method to send auction requests and supply requests to the aggregator. Each signature will be randomized with random numbers, so their identity information will not be disclosed.

Auction Privacy. We use the Pedersen commitment $\mathrm{cm} := g^r h^v$ to hide the information of the biddings. As the hiding properties of a cryptography commitment scheme, there is no $\mathcal{PPT}$ $\mathcal{A}$ who can know the hidden value v of a commitment. Only the bidding information of the final winner is published on the blockchain. Therefore, thanks to the FSA model, our scheme can protect the privacy of bidders as much as possible.

Traceability. The TA can trace the identity of a malicious EV/charging station when needed. The regulatory authority can send the signature that needs to be traced to the TA, and then, we use the *GOpen* algorithm to trace it. That is, the TA searches in the list $(i, \delta_i, \theta_i, \widetilde{\delta}_i)$ and checks whether $e(\sigma_{2\prime}, \widetilde{g}) \cdot e(\sigma_{1\prime}, \widetilde{X})^{-1} = e(\sigma_{1\prime}, \widetilde{\delta}_i)$ until he gets a match. Finally, he finds the corresponding (i, δ_i, θ_i).

Accountability. The accountability of our scheme is based on the premise of rational participants. One can choose any bid participating in the auction to compare with the winner's price and upload the request to the smart contract by consuming a small amount of token. The auctioneer needs to prove that the value in the designated bid commitment is higher than the winner's price. If he cannot, he will be deducted from a certain deposit. We assume that all auctioneers are rational. They do not want to be deducted because of cheating, so they execute the agreement honestly.

8. Future Work

In this section, we discuss some shortcomings of this study and possible future work directions.

8.1. Privacy-Preserving Payment on Blockchain. Although the scheme in this study takes into account the privacy of EV/charging station's identity, the payment process on the blockchain is not private, which means that it is possible to

Table 1: Experimental results on Miracle Library.

Notions	Description	Values (ms)
T_{bp}	Bilinear pairing	8.34
$T_{add\,1}$	Point addition in $\mathbb{G}_1$	0.01
$T_{mul\,1}$	Point multiplication in $\mathbb{G}_1$	2.82
$T_{add\,2}$	Point addition in $\mathbb{G}_2$	0.02
$T_{mul\,2}$	Point multiplication in $\mathbb{G}_2$	2.31
T_{mul}	Multiplication operation in $\mathbb{G}_T$	0.01
T_{exp}	Exponentiation operation in $\mathbb{G}_T$	0.58
$T_{\mathscr{H}}$	Hash function	0.03

Table 2: Computation costs of offchain part.

Stages		Computation costs
System initialization		$2\ T_{mul\,2}$
Register	EV	$1\ T_{add\,1} + 5\ T_{mul\,1}\ 1\ T_{mul\,2} + 3\ T_{bp}$
	Charging station	$1\ T_{add\,1} + 5\ T_{mul\,1}\ 1\ T_{mul\,2} + 3\ T_{bp}$
	Aggregator	$1\ T_{mul\,1}$
Request for auctions		$2\ T_{mul\,1} + 4\ T_{mul\,2} + 4\ T_{exp} + 2\ T_{mul} + 4\ T_{bp} + 4\ T_{\mathscr{H}}$
Upload bids		$t \cdot (1\ T_{add\,1} + 2\ T_{mul\,1})$, where t is number of bidders
Open bids		$1\ T_{add\,1} + 5\ T_{mul\,1} + 6\ T_{mul\,2} + 4\ T_{exp} + 2\ T_{mul} + 4\ T_{bp} + 3\ T_{\mathscr{H}}$
Trace		$d \cdot (1\ T_{exp} + 1\ T_{mul} + 3\ T_{bp})$, where d is the average number of matching queries

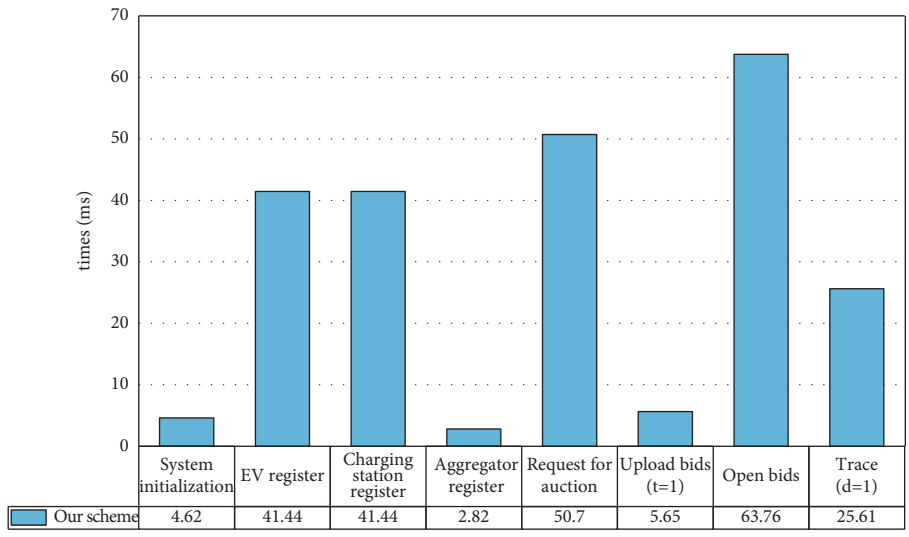

Figure 4: Computation costs of offchain part.

obtain EV or aggregator identity through the correlation of payment. There is some work in this area [31], and our scheme can adopt these strategies. In the future, we can also study the privacy protection payment system suitable for V2G and smart grid.

8.2. Light Client. In this study, the blockchain is jointly maintained by the aggregator. EVs and charging stations can view the blockchain information without writing information to the blockchain. Considering the storage and computing power of EV and charging station, as well as the development of vehicle networking, this assumption is feasible. However, the light client mode may be more suitable for the existing scheme. EVs and charging stations, as light clients, only need to maintain a small amount of data (*i.e.*, block header information) to verify the correctness of auction information on the blockchain. A series of studies on light clients [32,33] can be transplanted into our scheme.

8.3. More Efficient Auction Protocol. Although the FSA model is used in this scheme, there are actually more efficient auction protocols [30]. Combined with the scenarios of smart grid and V2G, considering the dynamic mobility of EVs, how to design a more efficient and secure auction scheme is also a potential future research direction.

9. Implementation

In this section, we evaluate the scheme by testing the onchain and offchain parts, respectively.

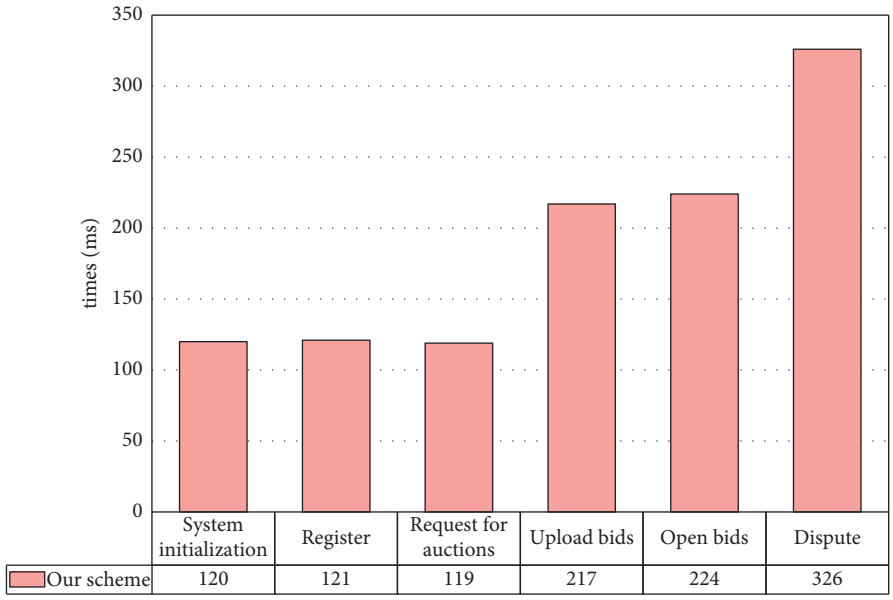

Figure 5: Experimental results of onchain part.

9.1. Experimental Environments. The experiment is carried out on a laptop with an i5-10400 Processor, 8G bytes RAM, and Windows 10 operating system. We utilize the Miracle Library [34] to implement cryptographic primitives. Hyperledger Fabric v1.4 is used in our experiment, and it is a stable version with long-term support. The smart contracts are developed in GoLang (v1.15.2) and tested using Fabric's dev-mode.

9.2. Evaluation Findings of Offchain Part. For time costs of anonymous authentication, we test the time overhead of the primary operations in Table 1 using the Miracle Library. We use a supersingular elliptic curve, with order divisible by $q = 2^{159} + 2^{17} + 1$ and security multiplier $k = 2$. The prime p is 512 bits. We count the number of main operations of our scheme in Table 2 and give the computation costs of the offchain part in our scheme in Figure 4.

9.3. Evaluation Findings of Onchain Part. For time costs on the blockchain, we evaluate the time costs of the steps in a local network with Raft modes. We set the same local private network configuration for each mode, including one channel and two organizations, and each organization has two peers.

We implement the steps on the blockchain, which correspond to our previous definitions. The results are shown in Figure 5. To ensure the accuracy of the experimental data, all the time consumption is obtained by executing the code 100 times to get the average value. In *system initialization* phase, we need to deploy smart contracts on the blockchain. In *register* phase, we need to initialize the accounts of EV, charging station, and aggregator and store certain tokens in the account. In *request for auction* phase, the aggregator uploads the information for bidding, *i.e.*, the auction request $\mathrm{PA} := \{\mathrm{id}, \mathrm{eq}, \mathrm{model}, t_1, t_2, \mathrm{pr}\}$. In *upload bid* phase, the charging station uploads the commitment for bidding. In *open bid* phase, the aggregator uploads the winner's address to the smart contract. In *dispute* phase, the challenger needs to send a challenge request to the blockchain, while the auctioneer needs to generate a zero-knowledge proof and then send it to the smart contract for verification.

In summary, we perform the experimental results of the offchain and onchain parts in our scheme. The results show that our proposal is efficient and suitable for V2G networks in the smart grid.

10. Conclusion

In this study, we systematically summarized the opportunities and challenges of blockchain in the smart grid. Then, we proposed a privacy-preserving blockchain-based electricity auction scheme for V2G networks in the smart grid under the FSA model, which inspires the applications of blockchain in the smart grid. Our scheme can ensure the privacy of EVs and charging stations, while using smart contracts to provide reliability. The experimental results showed the feasibility and practicality of our scheme.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The work was supported by the State Grid Henan Electric Power Company Science and Technology Project (No. 52170220009S).

References

[1] M. Z. Gunduz and R. Das, "Cyber-security on smart grid: threats and potential solutions," *Computer Networks*, vol. 169, p. 107094, 2020.

[2] Y. Guo, Z. Wan, and X. Cheng, "When Blockchain Meets Smart Grids: A Comprehensive Survey," *High-Confidence Computing*, vol. 2, no. 2, Article ID 100059, 2022.

[3] W. Han and Y. Xiao, "Privacy preservation for v2g networks in smart grid: a survey," *Computer Communications*, vol. 91-92, pp. 17–28, 2016.

[4] N. Fabra, N.-H. Fehr, and D. Harbord, "Designing electricity auctions," *The RAND Journal of Economics*, vol. 37, no. 1, pp. 23–46, 2006.

[5] J. Nicolaisen, V. Petrov, and L. Tesfatsion, "Market power and efficiency in a computational electricity market with discriminatory double-auction pricing," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 5, pp. 504–523, 2001.

[6] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, pp. 841–853, 2020.

[7] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *Journal of Network and Computer Applications*, vol. 126, pp. 45–58, 2019.

[8] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, 2014.

[9] E. Androulaki, A. Barger, V. Bortnikov et al., "Hyperledger Fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the European Conference on Computer Systems (EUROSYS)*, pp. 1–15, Crete, Greece, 17 April 2018.

[10] A. I. Commissioner, "Australia: Oaic Finds Uber Interfered with Privacy Rights," 2021, https://www.dataguidance.com/news/australia-oaic-finds-uber-interfered-privacy-rights.

[11] E. B. Sasson, A. Chiesa, C. Garman et al., "Zerocash: decentralized anonymous payments from bitcoin," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, pp. 459–474, IEEE, Berkeley, CA, USA, 18 May 2014.

[12] N. Van Saberhagen, "CryptoNote v2," 2013, https://en.bitcoinwiki.org.

[13] L. Nguyen and R. Safavi-Naini, "Dynamic k-times anonymous authentication," in *Applied Cryptography and Network Security,* pp. 318–333, Springer, Berlin, Heidelberg, 2005.

[14] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Advances in Cryptology - CRYPTO 2004,* pp. 56–72, Springer, Berlin, Heidelberg, 2004.

[15] M. H. Au, W. Susilo, and Y. Mu, "Constant-size dynamic k-taa," in *Proceedings of the International Conference on Security and Cryptography for Networks (SCN)*, pp. 111–125, Springer, Maiori, Italy, 6 September 2006.

[16] D. Pointcheval and O. Sanders, "Short randomizable signatures," in *Proceedings of the Cryptographers' Track at the RSA Conference (CTRSA)*, pp. 111–126, Springer, San Francisco, CA, USA, 29 February 2016.

[17] A. Hahn, R. Singh, C.-C. Liu, and S. Chen, "Smart contract-based campus demonstration of decentralized transactive energy auctions," in *Proceedings of the IEEE Power & Energy Society Innovative Smart Grid Technologies Conference*, pp. 1–5, IEEE, Washington, DC, USA, 23 April 2017.

[18] J. Wang, Q. Wang, and N. Zhou, "A Decentralized Electricity Transaction Mode of Microgrid Based on Blockchain and Continuous Double Auction," in *Proceedings of the IEEE Power & Energy Society General Meeting*, pp. 1–5, IEEE, Portland, OR, USA, Agust 2018.

[19] B. Ramachandran, S. K. Srivastava, C. S. Edrington, and D. A. Cartes, "An intelligent auction scheme for smart grid market using a hybrid immune algorithm," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4603–4612, 2010.

[20] M. Stübs, W. Posdorfer, and S. Momeni, "Blockchain-based multi tier double auctions for smart energy distribution grids," in *Proceedings of the IEEE International Conference on Communications Workshops*, pp. 1–6, IEEE, Dublin, Ireland, 7 June 2020.

[21] J. Ma, J. Deng, L. Song, and Z. Han, "Incentive mechanism for demand side management in smart grid using auction," *IEEE Transactions on Smart Grid*, vol. 5, no. 3, pp. 1379–1388, 2014.

[22] M. Wen, R. Lu, J. Lei, H. Li, X. Liang, and X. S. Shen, "SESA: an efficient searchable encryption scheme for auction in emerging smart grid marketing," *Security and Communication Networks*, vol. 7, no. 1, pp. 234–244, 2014.

[23] D. Li, Q. Yang, W. Yu, D. An, Y. Zhang, and W. Zhao, "Towards differential privacy-based online double auctioryjrtn for smart grid," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 971–986, 2019.

[24] Z. Zhou, F. Liu, Z. Li, and H. Jin, "When smart grid meets geo-distributed cloud: an auction approach to datacenter demand response," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pp. 2650–2658, IEEE, Hong Kong, China, 26 April 2015.

[25] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf, "Pseudonym systems," in *Proceedings of the International Workshop on Selected Areas in Cryptography*, pp. 184–199, Springer, Kingston, ON, Canada, 9 August 1999.

[26] Q. Gang, W. Hong, W. Shimin, and X. Guozhen, "Information-theoretic secure verifiable secret sharing over rsa modulus," *Wuhan University Journal of Natural Sciences*, vol. 11, no. 6, pp. 1849–1852, 2006.

[27] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[28] M. Abdalla, M. Bellare, and P. Rogaway, "The oracle diffie-hellman assumptions and an analysis of dhies," in *Proceedings of the Cryptographers' Track at the RSA Conference (CTRSA)*, pp. 143–158, Springer, San Francisco, CA, USA, 8 April 2001.

[29] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[30] V. Krishna, *Auction Theory*, Academic Press, Cambridge, CB, USA, 2009.

[31] S. Jain, N. J. Ahuja, P. Srikanth et al., "Blockchain and Autonomous Vehicles: Recent Advances and Future Directions," *IEEE Access*, vol. 9, 2021.

[32] B. Bünz, L. Kiffer, L. Luu, and M. Zamani, "Flyclient: super-light clients for cryptocurrencies," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, pp. 928–946, IEEE, San Francisco, CA, USA, 18 May 2020.

[33] P. Chatzigiannis, F. Baldimtsi, and K. Chalkias, *SoK: Blockchain Light Clients*, Cryptology ePrint Archive, 2021.

[34] S. software ltd, "Miracle," 2021, https://github.com/miracl/MIRACL.