*Journal of Network Intelligence*
**Taiwan Ubiquitous Information**
 ISSN 2414-8105 (Online)
Volume **8**, Number **1**, February **2023**

# Blockchain-based Biddings Protocol for Cyber-Physical System

Yi Luo[1,2,*], Mi Zhou[1,2], Houpeng Hu[3], Yong Xiao[1,2], Jiaxiang Ou[3], Bin Qian[1,2], Yanhong Xiao[3]

[1] Electric Power Research Institute, CSG, Guangzhou 510663, China
[2] Guangdong Provincial Key Laboratory of Intelligent Measurement and Advanced Metering of Power Grid Guangzhou 510663, China
luoyi_csg@outlook.com, zhoumi@csg.cn, xiaoyong@csg.cn, qianbin@csg.cn

[3] Guizhou Power Grid Co., LTD, Guizhou 550000, China
huhp@gzsyy.csg.cn, oujx@gzsyy.csg.cn, Xiaoyh@gzdy.csg.cn

*Corresponding author: Yi Luo

Received July 29, 2022, revised September 10, 2022, accepted December 23, 2022.

Abstract. *Government procurement or public procurement in an Cyber-Physical System (CPS) such as electricity infrastructure procurement often demands open bidding. In organizing such biddings, a challenge is to make the bidding fair and verifiable for anyone. Traditionally, the government agency conducts the whole bidding in which all bidders submit their sealed bid to the agency. The confidentiality and fairness of the bidding are based on the trust of the government agency. However, in an IoE participants are mutual untrusted and well-designed bidding should be verifiable and Third Trusted Party (TTP)- free. As a result, some bidding schemes based on blockchain have been proposed to achieve the requirement of TTP-free. Nevertheless, these blockchain-based schemes may not fit the IoE systems well for it is difficult to balance the efficiency, gas cost and design requirements. To address these challenges, we propose a novel bidding scheme that is fully decentralized and meets seven design requirements for an IoE system. Moreover, we tested our scheme in the live networks of Ethereum and compared it with a similar state-of-art bidding scheme in the same experiment environment. The results show that our scheme is more efficient and needs less gas cost.*

**Keywords:** Internet of Energy, Bidding, Blockchain, Verifiable, Efficient

1. **Introduction.** Many countries today are upgrading their energy systems to a new type of smart grid [1, 2] called the Cyber-Physical System (CPS) [3, 4]. CPS adds new capabilities to physical systems by embedding computing and communication deeply into physical processes and making them interact closely with them. The CPS system is as small as a pacemaker and as big as a national grid. This allows energy production to move forward more efficiently and cleanly with the least amount of waste. With the growth of distributed energy resources, Internet of Energy (IoE) needs a fully P2P energy market and more decentralized management. And bidding or auction is a common activity in such an open market [5], such as a P2P energy exchange. In this paper, we focus on how to build a practical biding scheme for IoE.

Traditionally, there is an auctioneer or administrator in a bidding scheme [6] and the credibility often relies on such a central trusted third-party. However, a centralized bidding scheme in IoE will face many challenges, as there are many mutual untrusted participants. For example, how to guarantee that the auctioneer will not reveal the submitted bid

content to any bidders before the bidding is completed. In this context, many blockchain-based bidding protocols are proposed due to the features of immutability, public ledger facility, and decentralization of blockchain technology [7] [8]. Separately, they focused on the problem in different application scenarios. For example, [9] proposed the blockchain-based scheme concentrating on crowdsourcing. In 2017, Huang et al. [10] proposed an auction mechanism to achieve max-min fairness in a real crowdsensing system. To prevent collusion coalitions among sellers, buyers, and auctioneers, [11] presented a decentralized collusion-resistant e-auction in 2018. Another challenge is to tackle the conflict between the inherent transparency and lack of privacy on the blockchain. To tackle this, [12] gave a verifiable sealed-bid auction that utilized various cryptographic primitives.

Generally, these blockchain-based protocols have provided methods to address the problem caused by the trusted third party [13, 14, 1, 15]. Nevertheless, there are still limitations in them for an IoE system. Firstly, not all of them are fully decentralized. Although one can use the blockchain to store the data of bidding, it is difficult to protect the privacy of the bidders and make the validity of the bidding be public verifiable simultaneously. As a result, some of the bidding schemes depend on an off-Chain centralized component to provide proof of validity and fairness. Secondly, some of the schemes are infeasible in reality due to blockchain's limit in transaction size. A practical bidding scheme should take the capacity of the blockchain into account. Finally, a useful bidding scheme should not take too much gas cost as bidding is a basic component in IoE and an expensive scheme will be abandoned. To address these challenges, we propose a full distributed bidding scheme via Ethereum blockchain which is efficient and at a low cost. Specifically, our contributions are summarized as follows:

- We propose a decentralized bidding scheme for the internet of energy based on blockchain. Too far as we know, our IoE decentralized practical verifiable Bidding Scheme named IoEPVB is the first work to consider the key features of the IoE, public verifiable, sealed-bid and fully decentralized.
- Our scheme address six design requirements that are thought to be difficult to attain simultaneously. We give a theoretical analysis to show how the proposed scheme fulfills requirements and resist potential attacks [16, 17, 18].
- We implement experiments in the two live networks of the Ethereum blockchain and reimplement a similar blockchain-based sealed-bid scheme Galal in the same environment for comparison. Compared with the baseline blockchain-based sealed-bid scheme Galal, our scheme enjoys less gas cost and more efficiency. The experiment shows that IoEPVB is a decentralized, effective, fair and low-cost bidding scheme for IoE.

## 2. **Preliminaries.**

### 2.1. **Problem Statement.**

One of the current challenges in the IoE is creating a fully P2P energy market and a practical online bidding scheme is quite essential for the open energy market [19] [20]. Online bidding can be divided into three main types, namely **English Auction**, **Dutch Auction** and **Sealed Bid Auction** [6]. **English Auction** is a kind of price-rasing auction in which the price cast by all bidders should be higher than the previous one. At the end of the bidding, the highest bidder will win the bid. However, a specific bidder may make a bid at the very last minute before the bidding ends to leave no time for the other bidders to fight back. Conversely, the price in **Dutch Auction** will be reduced according to the initially set price reduction rules. For instance, the price of perishable items such as fruits and vegetables will be reduced until there is someone wanting to buy at that price. Sometimes, the prices of all bidders should be

kept secret before the deadline for the bid opening. This kind of bidding is usually called **Sealed Bid Auction** in which the prices of all bidders should be sealed. As IoE systems is a trustless environment, **Sealed Bid Auction** is quite suitable for it. Here we give a specific bidding example. Assume there is an electricity infrastructure procurement in the IoE system. To prevent fraud, waste, corruption, or local protectionism, such a procurement bidding is often demanded as open bidding and the prices of all bidders will not be compared before the deadline for the bid opening. And a government agency as a vendor of the bidding in an IoE system will create bidding. And all the eligible participants in the IoE can submit their bids. Usually, the lowest bidder will get the item. Generally, there are seven requirements for such a bidding scheme in IoE systems [6] [21]:

- **Eligibility**. The qualification of the bidders should be checked before the bidding start. For example, the government procurement may require an eligible bidder should be an enterprise that has a similar project experience, a healthy financial condition and so on.
- **Confidentiality**. For simplicity, we assume the lowest bidder will win the bid. As sealed bidding, all bidders should not be able to know the prices of other bidders before the deadline for the bid opening. As a government procurement, the vendor may collude with one of the bidders and accept bribes when the colluded bidder wins the bid. Therefore, the prices of all bidders should not be known by the vendor before the opening stage as well.
- **Non-repudiation**. Once a bidder has submitted a bid, he/she can not deny his/her commitment later [22, 23].
- **Non-changeability**. Once a bidder has submitted a bid, he/she can not change his/her price later.
- **Verifiability**. At the end of the bidding, anyone can verify the validity and fairness of the bidding result.
- **Collusion-resistance**. Even though the vendor of the bidding colludes with a bidder, the colluded bidder should not have any advantage to win the bid.
- **TTP-free**. Since participants in an IoE system are mutual untrusted, the bidding scheme should not import any kind of Third Trusted Party (TTP).

2.2. **Cryptographic Commitment and Bidding.** The bidding in IoE systems can work as follow: the participant in a system submits a secret sealed bid for the target item such as a government energy outsourcing project. The party who submitted the lowest bid gets the project according to all the bids in this project, and the the second-lowest bid is given to the price paid. In the situation that the participants do not know each other's bids until the auction is finished and do not collude together, the above mechanism like "Vickrey sealed bid auction" discussed in [24] and can be shown to have good properties in the game theory. In the situation that all the participants are offline, the auction house can implement such bidding in the same room by asking each participant pose their bid in a sealed packages. After receiving all the packages, the vendor opens them and gets the results. Participants can examine the packages and the documents inside it to verify that the bid was properly managed and are no need to trust auction houses.

To implement such sealed bidding online, we use a cryptographic primitive called cryptographic commitment [25] in this paper. By a cryptographic commitment scheme, a participant is able to choose a bid value and generate a corresponding commitment for his choice such that he can no longer change his mind. Before he reveals his choice, no one can learn any knowledge of his choice from this sealed commitment. However, he has to reveal his choice at the end and convince others that the open choice is bound

with the afore commitment. Formally, there are two properties should be satisfied by a cryptographic commitment:

- **Hiding:** Hiding requires that the commitment $C$ reveal no information about the original choice. To be more specific, the distribution on $C$ bound with the bid choice $B_0$ is indistinguishable from the distribution on $C$ when bound with bid choice $B_1$.
- **Binding:** Let $C$ be any commitment output by a bidder. Binding requests that the bidder can open the commitment as some bid choice $B$ then he cannot open it as $\hat{B}$. This means that once the bidder commits to a choice $B$ he can open it as $B$ and nothing else.

Then we construct a cryptographic commitment protocol using a hash function SHA256 [26]. Let $H$ be a SHA256 hash function define over $(X, Y)$ where $X = M \times R$. Here $M$ is the message space for the commitment protocol, and $R$ is a finite nonce space that will be used for the bidding property. For $m \in M$, the commitment protocol contains two algorithms $(Commit, Open)$ is defined as:

$Commit(m) := \{o \xleftarrow{r} R, c \leftarrow H(m, o), output(c, o)\}$
$Open(m, c, o) := \{$ output **accept** if $c = H(m, o)\}$

3. **IoEPVB: The Framework.** In this section, IoEPVB, the proposed bidding scheme is described. As is shown in Fig. 1, the framework of our scheme can be divided into four phases, namely *Setup*, *Commit Bid*, *Open Bid* and *Verification*. The phase *Setup* is executed before the bidders start to bid while the phase *Verification* is executed after the bidders finish their offers. And the efficiency of these two phases has less impact on the bidders' experience. Therefore, we define the *Commit Bid* and *Open Bid* as a bidding cycle in order to compare the efficiency with other similar schemes later.

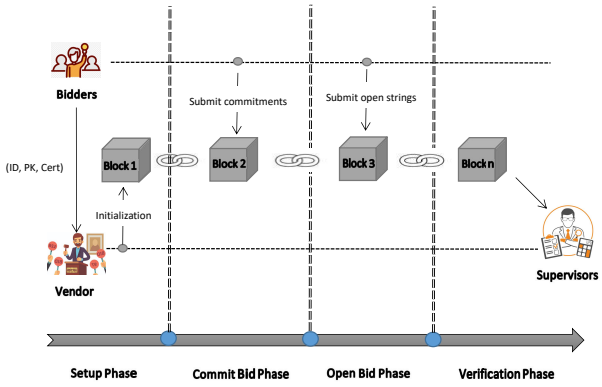


FIGURE 1 The framework of the Proposed Bidding Scheme

3.1. **Setup Phase.** In the setup phase, all bidders provide their verified document(s) to the vendor (such as a government agency) to show their qualifications to attend the bidding. Then the eligible bidders generate their key pairs with Elliptic Curve Public Key Cryptosystem on the curve of "secp256k1" [27]. Here we note a key pair for a bidder $i$ as $(PK_i, SK_i)$. All eligible bidders must send their public keys $PK$ to the vendor and keep their private key $SK$ secret at all times. Next, the vendor can create bidding by the smart contract and submit all the public information of the bidders to the blockchain by an interface *Initialization*.

The pseudocode of the algorithm *Initialization* in the smart contract is described in Algorithm 1. Here the input $Addr, ID, PK$ are the Ethereum account addresses, identifications and public keys of all bidders separately. Suppose there are $n_b$ bidders and

Table 1 The structure of a Bidder record

| Data Type | Description |
|---|---|
| address | bidder |
| string | PK |
| string | ID |
| string | orignalBid |
| string | commitStr |
| string | openStr |
| string | sigCommit |
| string | sigOpen |

*bids* is a mapping from an address to a bidder record which is shown in Table 1. The bidder record is the key data structure in the smart contract to record the information all the bidding. Line 1 in algorithm 1 make sure that only the vendor as the creator of the smart contract can invoke this procedure. The *caller* and *vendor* are Ethereum account addresses. Moreover, Line 3 in algorithm 1 make sure that once the public information of the bidders is recorded into the blockchain, no one can modify it. Thus, at the end of this phase, anyone can check the eligibility of the bidders from the public information on the blockchain.

**Algorithm 1** Initialization

**Input** $(Addr, ID, PK)$

```
1: require(caller == vendor);
2: for i=0 to n_b do
3:    require(bids[Addr[i]].bidder == 0);
4:    bids[Addr[i]].bidder = Addr[i];
5:    bids[Addr[i]].id = ID[i];
6:    bids[Addr[i]].pk = PK[i];
7: end for
```

3.2. **Commit Bid Phase.** After all bidders have checked the validity of the *Initialization*, they start to submit their own bid. As is shown in Algorithm 2, a bidder who tries to submit a bid will first generate a random string $iO$ and then use the cryptographic commitment protocol 2.2 to calculate a string $c$. Here $iM$ is an original bid value. Then the bidder generates a signature on $c$ using his secret key $iSK$. The tuple $(c, sig)$ can be sent into the blockchain according to the account address $iAddr$. Finally, all bidders kept their own opening string $iO$ secret until the next phase.

**Algorithm 2** Commitment

**Input** $(iM, iAddr, iSK)$

```
1: iO = RandomBytes(32);
2: c = SHA256(iM+iO);
3: sig = ecdsaSign(c, iSK);
4: require(bids[iAddr].bidder == caller and bids[iAddr].commitStr == 0);
5: bids[iAddr].commitStr = c;
6: bids[iAddr].sigCommit = sig;
7: save(iO);
```

3.3. **Open Bid Phase.** In the commit bid phase, all bidders will submit their own sealed bid and wait for all sealed bids to be confirmed by the blockchain. Then they enter into this phase. As is shown in Algorithm 3, all bidders will reveal their own original bid value $iM$ and opening string $iO$. Line 2 in Algorithm 3 make sure that once the bidder submits their opening string, they can not modify it later.

**Algorithm 3** Opening

**Input** $(iO, iAddr, iM, iSK)$

```
1: sig = cdsaSign(iO, iSK);
2: require(bids[iAddr].bidder == caller and bids[iAddr].openStr == 0);
3: bids[iAddr].openStr = iO;
4: bids[iAddr].orignalBid = iM;
5: bids[iAddr].sigOpen = sig;
```

3.4. **Verification Phase.** When the open bid phase is completed, everyone can check the fairness and correctness of the bidding. As is shown in 4, anyone can query the data about the whole bidding to verify the validity. First, the signatures of each bidder's commitment and the open string will be verified. According to the cryptographic commitment protocol, one can calculate a string *vcommitStr* and compare it with the commitment.

**Algorithm 4** Verification

**Input** $(Addr, PK, ID)$
**Output** WinnerID

```
1: highestbid = 0;
2: WinnerID = 0;
3: for i=0 to n_b do
4:    rsigCommit = bids[Addr[i]].sigCommit;
5:    rsigOpen = bids[Addr[i]].sigOpen;
6:    rcommitStr = bids[Addr[i]].commitStr;
7:    rm = bids[Addr[i]].orignalBid;
8:    ropenStr = bids[Addr[i]].openStr;
9:    if ecdsaVerify(rsigCommit, rcommitStr, PK[i]) and
      ecdsaVerify(rsigCommit, ropenStr, PK[i]) then
10:      vcommitStr = SHA256(rm + ropenStr);
11:      if vcommitStr == rcommitStr and rm > highestbid then
12:         highestbid = rm;
13:         WinnerID = ID[i];
14:      end if
15:   end if
16: end for
17: return WinnerID;
```

The verification bidding begins with the verification of the commitment of each bidder and the signature of the opening conditions. According to the encryption commitment algorithm, a string *vcommitStr* can be calculated and compared with the commitment. Finally, the identity of the highest bidder can be found. Lines 1-2 of the algorithm defines two temporary variables, and then for all bidders (Line 3), the algorithm turns on the commitment signature (*sigCommit*) in the bidding data structure of the blockchain,

opens the conditional signature commitment string ($sigCommit$), The original bid information ($orignalBid$) and open information string information ($Openstr$) are taken from the blockchain (Lines 4-8). Then, the algorithm uses the public key corresponding to the bidder to verify the validity of the signature of the commitment and the opening condition (Line 9). If it passes, the information of the current original bid and the opening string is calculated by the encrypted commitment algorithm designed by the invention to obtain a commitment value (Line 10). Finally comparing this value with block chain achieve the promise of value are the same and check whether the bidders bid is higher than before (Line 11), if it is true, then the chain chain of the block is set to the current block. Finally, the bidding and bidding information update to the temporary variable (Lines 12-13), the algorithm is finally return to the highest bid bidder status information.

## 4. **Security Analysis.**

4.1. **The Security of Cryptographic Commitment.** In our scheme, we use a cryptographic commitment protocol based on a collision resistant hash function SHA256. A cryptographic commitment protocol is secure if it is both hiding and binding. Then we show that the cryptographic commitment protocol is secure:

- Firstly, if $H$ is collision resistant then the protocol is a binding commitment. Suppose an efficient adversary $A$ outputs two pairs $(m_1, o_1)$ and $(m_2, o_2)$, where $m_1 \neq m_2$, but $Open(m_1, c, o_1) = Open(m_2, c, o_2) =$ **accept**, for some commitment string $c$. Then $H(h_1, o_1) = c = H(m_2, o_2)$ is a collision for $H$.
- Secondly, we want $H$ to satisfy a certain statistical property for satisfying the hiding property. Here in our scheme, $H$ is input hiding since for all $m_1, m_2 \in M$, the distribution $\{H(m_1, o)\}$ is statistically indistinguishable from the distribution $\{H(m_2, o)\}$, where $o \xleftarrow{r} R$. So no adversary $A$ can break the semantic security of the derived commitment protocol.

4.2. **The Validity of the design.** In this part, we show that our proposed scheme fulfills all the design requirements. Note that these requirements are widely discussed in a secure scheme [28, 29, 30, 31, 32, 33].

- **Eligibility**. In the "Initialization" phase of our proposed scheme, only the eligible bidders are recorded into the smart contract. Moreover, once the information of the eligible bidders is initialized, they can not be modified and everyone can verify its validity before starting to bid. Therefore, an illegal bidder has no right to bid.
- **Confidentiality**. Before the opening stage, all the commitment strings of the bidders are hidden by the cryptographic commitment protocol. Since the cryptographic commitment protocol is secure, no one can learn any knowledge about others' original bids.
- **Non-repudiation**. The commitment and opening strings submitted by a bidder are ECDSA signed by his/her secret key corresponding with his identification. Thus, a bidder can not deny his/her commitment.
- **Non-changeability**. The design of the smart contract guarantees that a bidder/can not change his/her submitted bid later.
- **Verifiability**. Since the cryptographic commitment protocol is secure, anyone can verify whether all the opening strings are bound to the corresponding commitment strings.
- **Collusion-resistance**. Since all the commitment strings of the bidders are confidential until the opening stage, the bidder who colluded with the vendor has no advantage to win the bid.

- **TTP-free**. Our scheme is fully decentralized for we have not introduced any kind of trusted third party.

## 5. Implementation And Performance Evaluation.

5.1. **Implementation.** The proposed IoEPVB is designed to provide a practical blockchain-based bidding mechanism that is compatible with an Internet of Energy system. Our implementation has Web APIs and smart contract programs. The Web APIs are written in JavaScript to provide interfaces for users to interact with the smart contract. For example, a bidder may use Web APIs to submit his/her bid commitment. The smart contract is written in Solidity to realize the basic functions of our protocol. There are several software libraries we used to finish our programs. We used the library "ethers.js" [34] to interact with the blockchain in a succinct way. To implement the basic cryptographic primitives in the scheme, we used "Crypto.js" [35] which is a standard crypto library written in JavaScript. The hash algorithm in our protocol is SHA256 [26]. The digital signature protocol in our scheme has been implemented with Elliptic Curves Digital Signature Algorithm (ECDSA) on the "secp256k1" [27] curve.

Our scheme has been implemented in the Ethereum blockchain. The live networks in the Ethereum blockchain can be divided into "mainnet" and "testnet". "mainnet" is a real Ethereum network that deals with real money while "testnet" are other live networks that don't deal with real money but do mimic the real world scenario well. Since we want to execute plenty of test cases to show the performance of our scheme, we have executed the programs in two live testnets of the Ethereum blockchain. They are "Ropsten" [36] and "Rinkeby" [37]. The web3 [38] application programs called by the bidders and the tenderer has been executed in a PC with the OS of Ubuntu Desktop 18.04 64x, intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30GHz and memory of 4G.

5.2. **Efficiency Analysis.** To evaluate our scheme, we run a similar protocol proposed by Galal et al. [12] for comparison. The compared protocol we called Galal has been implemented with the same environment mentioned above for precision. As is shown in Fig. 2 and 3, the efficiency comparisons in "Ropsten" and "Rinkeby" are given. Here we compare the total time for finishing a bidding cycle in the two schemes. The performance of our scheme represented in the red line is more efficient than that of Galal's scheme. Note that a bidding cycle includes the phases which should be executed online and waited for feedback timely. Therefore, the "Deployment", "Initialization", "Verification" and "Prove" phases in the two schemes separately should be ignored. Because they can be executed before or after the bidding cycle.

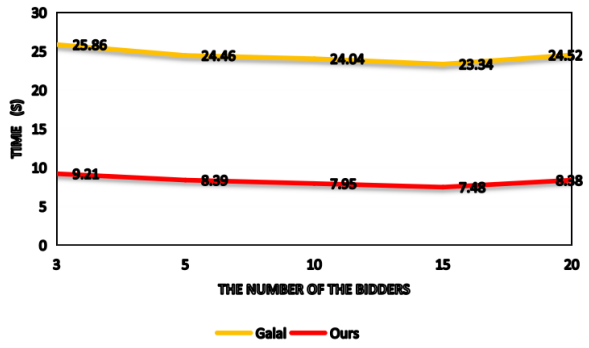


FIGURE 2 Total Time For Finishing a Bidding Cycle In Rinkeby Network

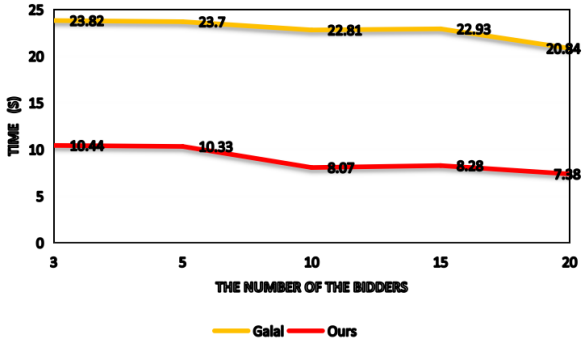


FIGURE 3 Total Time For Finishing a Bidding Cycle In Ropsten Network

As a practical scheme, our scheme should perform stably when the number of bidders increases. To show stability, we give the performance of different phases in our scheme when the number of bidders increases. As is shown in Fig. 4 to 7 , the gradient of the curves represented the stability, namely the more smoothy, the more stable. As we can see, almost all the efficiency of the phases except the "Verification" phase are stable and the time cost of them wouldn't increase sharply with the addition of the number of bidders. Although the running time of the "Verification" phase is proportionable with the number of bidders, she is outside of the bidding cycle and can be executed thereafter. Verification in a minute is acceptable apparently.

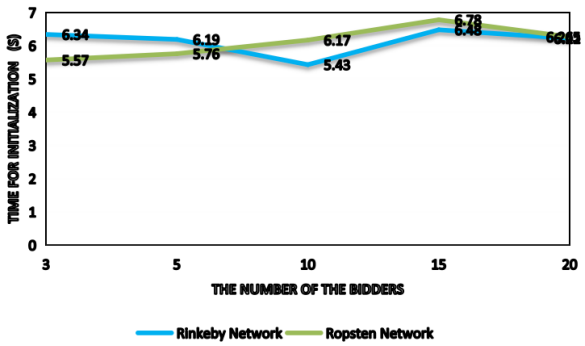


FIGURE 4 Time For Initialization

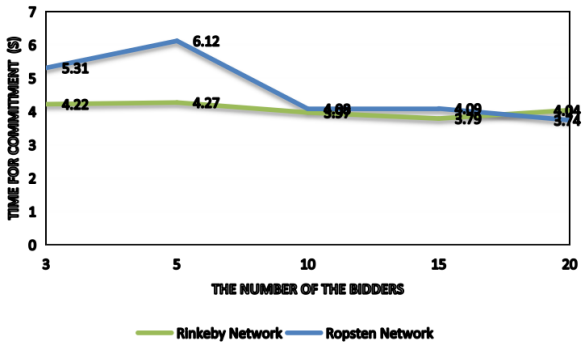


FIGURE 5 Time For Commitment

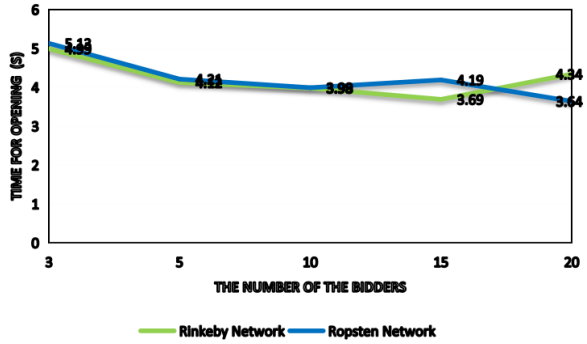


FIGURE 6 Time For Opening

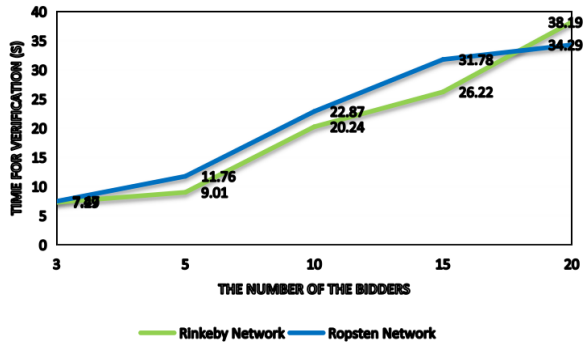


FIGURE 7 Time For Verification

5.3. **Gas Cost Aanlysis.** Another evaluation is about the gas cost for a scheme based on blockchain. Similarly, we compare our scheme with Galal's [12]. We set the parameters of "gasLimit" and "gasPrice" to be 6400000 and 5 Gwei [39] separately in the two schemes. As is shown in Fig 8 and 9, the total gas cost of our scheme represented by the red line is only about 25% of that in Galal's no matter in Rinkeby Network or Ropsten Network. We can also find that the gas cost of the same scheme has little change in different networks.

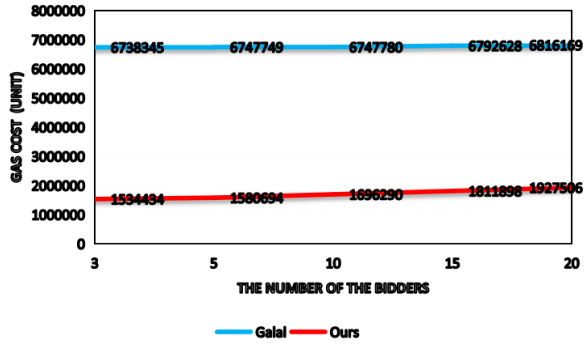


FIGURE 8 Total Gas Cost In Rinkeby Network

Table 2 and 3 give a close view for the gas cost of the two schemes. We show the gas cost in each phase of the two schemes for comparison. Then we can find that the "Deployment" and "Prove" phases in Galal's scheme are quite expensive. As a less cost scheme, our scheme has a costless phase "Verification". To achieve this, we first take the blockchain as a distributed database and avoid heavy computation on-chain. Actually, in the "Verification", we only query the tamper-proof data from the blockchain and then

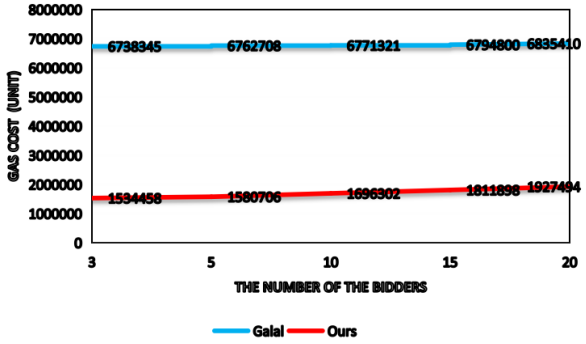


FIGURE 9 Total Gas Cost In Ropsten Network

anyone can prove its validity off-chain. Thus, the gas cost of the "Verification" phase is zero.

TABLE 2. Gas Cost Comparision in Rinkeby Network

| **Paper** | **Functionality** | **Gas unit** | **Sum** |
|---|---|---|---|
| Galal | Deployment | 3,854,477 | 6,747,780 |
| | Bid | 105,495 | |
| | Open | 549,728 | |
| | Prove | 2,111,645 | |
| | Finalize | 85,108 | |
| | Dispute | 41,327 | |
| Ours | Deployment | 1,006,042 | 1,696,290 |
| | Initialization | 255,181 | |
| | Commit | 206,371 | |
| | Open | 228,696 | |
| | Verification | 0 | |

TABLE 3. Gas Cost Comparision in Ropsten Network

| **Paper** | **Functionality** | **Gas unit** | **Sum** |
|---|---|---|---|
| Galal | Deployment | 3,854,477 | 6,771,321 |
| | Bid | 105,495 | |
| | Open | 549,728 | |
| | Prove | 2,111,676 | |
| | Finalize | 108,618 | |
| | Dispute | 41,327 | |
| Ours | Deployment | 1,006,042 | 1,696,302 |
| | Initialization | 255,193 | |
| | Commit | 206,371 | |
| | Open | 228,696 | |
| | Verification | 0 | |

6. **Conclusion.** In this paper, we propose a new practical decentralized online bidding scheme. To the best of our knowledge, our scheme is the first one to take the bidding requirements for an IoE system into account. We combine a cryptographic commitment

mechanism, the ECDSA signature mechanism and blockchain technology to fulfiltl all the requirements such as confidentiality, Non-repudiation etc. Our scheme stores and checks the commitments from bidders on the Ethereum blockchain by a smart contract. In order to integrate into an IoE system easily, we have implemented the Web3 interface of the bidding scheme. Compared with a similar bidding scheme, our scheme enjoys less gas cost and show higher efficiency.

**Acknowledgment.** This research was partially supported by National Key Research and Development Program of China (Grant Nos. 2019YFE0118700), Science and Technology Project of China Southern Power Grid Corporation (Grant No. 066600KK52200016).

## REFERENCES

[1] C.-M. Chen, X. Deng, S. Kumar, S. Kumari, and S. Islam, "Blockchain-based medical data sharing schedule guaranteeing security of individual entities," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–10, 2021. [Online]. Available: https://doi.org/10.1007/s12652-021-03448-7

[2] T.-Y. Wu, Y.-Q. Lee, C.-M. Chen, Y. Tian, and N. A. Al-Nabhan, "An enhanced pairing-based authentication scheme for smart grid communications," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, 2021. [Online]. Available: https://doi.org/10.1007/s12652-020-02740-2

[3] "Internet of energy for electric mobility home page," http://www.artemis-ioe.eu/ioe_project.htm, accessed: 2021-09-30.

[4] Q. Mei, H. Xiong, Y.-C. Chen, and C.-M. Chen, "Blockchain-enabled privacy-preserving authentication mechanism for transportation cps with cloud-edge computing," *IEEE Transactions on Engineering Management*, 2022. [Online]. Available: https://doi.org/10.1109/TEM.2022.3159311

[5] A. Miglani, N. Kumar, V. Chamola, and S. Zeadally, "Blockchain for internet of energy management: Review, solutions, and challenges," *Computer Communications*, vol. 151, pp. 395–418, 2020.

[6] Y.-H. Chen, L.-C. Huang, I.-C. Lin, and M.-S. Hwang, "Research on blockchain technologies in bidding systems," *International Journal of Network Security*, vol. 22, no. 6, pp. 897–904, 2020.

[7] H. S. Galal and A. M. Youssef, "Trustee: full privacy preserving vickrey auction on top of ethereum," in *International Conference on Financial Cryptography and Data Security*. Springer, 2019, pp. 190–207.

[8] M. Xiao, K. Ma, A. Liu, H. Zhao, Z. Li, K. Zheng, and X. Zhou, "Sra: Secure reverse auction for task assignment in spatial crowdsourcing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 4, pp. 782–796, 2019.

[9] M. Kadadha, R. Mizouni, S. Singh, H. Otrok, and A. Ouali, "Abcrowd an auction mechanism on blockchain for spatial crowdsourcing," *IEEE Access*, vol. 8, pp. 12 745–12 757, 2020.

[10] H. Huang, Y. Xin, Y.-E. Sun, and W. Yang, "A truthful double auction mechanism for crowdsensing systems with max-min fairness," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2017, pp. 1–6.

[11] S. Wu, Y. Chen, Q. Wang, M. Li, C. Wang, and X. Luo, "Cream: A smart contract enabled collusion-resistant e-auction," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1687–1701, 2018.

[12] H. S. Galal and A. M. Youssef, "Succinctly verifiable sealed-bid auction smart contract," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, 2018, pp. 3–19.

[13] C.-M. Chen, X. Deng, W. Gan, J. Chen, and S. Islam, "A secure blockchain-based group key agreement protocol for iot," *The Journal of Supercomputing*, vol. 77, no. 8, pp. 9046–9068, 2021.

[14] M. Yavari, M. Safkhani, S. Kumari, S. Kumar, and C.-M. Chen, "An improved blockchain-based authentication protocol for iot network management," *Security and Communication Networks*, vol. 2020, p. 8836214, 2020.

[15] S. M. Sajjad, M. R. Mufti, M. Yousaf, W. Aslam, R. Alshahrani, N. Nemri, H. Afzal, M. A. Khan, and C.-M. Chen, "Detection and blockchain-based collaborative mitigation of internet of things botnets," *Wireless Communications and Mobile Computing*, vol. 2022, p. 1194899, 2022.

[16] T.-Y. Wu, X. Guo, Y.-C. Chen, S. Kumari, and C.-M. Chen, "Sgxap: Sgx-based authentication protocol in iov-enabled fog computing," *Symmetry*, vol. 14, no. 7, p. 1393, 2022.

[17] T.-Y. Wu, Z. Lee, L. Yang, and C.-M. Chen, "A provably secure authentication and key exchange protocol in vehicular ad hoc networks," *Security and Communication Networks*, vol. 2021, p. 9944460, 2021.
[18] T.-Y. Wu, X. Guo, L. Yang, Q. Meng, and C.-M. Chen, "A lightweight authenticated key agreement protocol using fog nodes in social internet of vehicles," *Mobile Information Systems*, vol. 2021, p. 3277113, 2021.
[19] J. Cao and M. Yang, "Energy internet–towards smart grid 2.0," in *2013 Fourth International Conference on Networking and Distributed Computing*. IEEE, 2013, pp. 105–110.
[20] M. J. A. Baig, M. T. Iqbal, M. Jamil, and J. Khan, "Design and implementation of an open-source iot and blockchain-based peer-to-peer energy trading platform using esp32-s2, node-red and, mqtt protocol," *Energy Reports*, vol. 7, pp. 5733–5746, 2021.
[21] "Government procurement wikipedia," May 2022. [Online]. Available: https://en.wikipedia.org/wiki/Government_procurement
[22] T.-Y. Wu, Y.-M. Tseng, and C.-W. Yu, "Id-based key-insulated signature scheme with batch verifications and its novel application," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 7, pp. 4797–4810, 2012.
[23] T.-Y. Wu, Y.-M. Tseng, S.-S. Huang, and Y.-C. Lai, "Non-repudiable provable data possession scheme with designated verifier in cloud storage systems," *IEEE Access*, vol. 5, pp. 19 333–19 341, 2017.
[24] C. Barrot, S. Albers, B. Skiera, and B. Schäfers, "Vickrey vs. ebay: Why second-price sealed-bid auctions lead to more realistic price-demand functions," *International Journal of Electronic Commerce*, vol. 14, no. 4, pp. 7–38, 2010.
[25] I. Damgård, "Commitment schemes and zero-knowledge protocols," in *School organized by the European Educational Forum*. Springer, 1998, pp. 63–86.
[26] D. Rachmawati, J. Tarigan, and A. Ginting, "A comparative study of message digest 5 (md5) and sha256 algorithm," in *Journal of Physics: Conference Series*, vol. 978, no. 1. IOP Publishing, 2018, p. 012116.
[27] J. R. Shaikh, M. Nenova, G. Iliev, and Z. Valkova-Jarvis, "Analysis of standard elliptic curves for the implementation of elliptic curve cryptography in resource-constrained e-commerce applications," in *2017 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS)*. IEEE, 2017, pp. 1–4.
[28] T.-Y. Wu, X. Guo, Y. Chen, S. Kumari, and C. Chen, "Amassing the security: An enhanced authentication protocol for drone communications over 5g networks," *Drones*, vol. 6, no. 1, p. 10, 2021.
[29] C.-M. Chen, W. Fang, K.-H. Wang, and T.-Y. Wu, "Comments on "an improved secure and efficient password and chaos-based two-party key agreement protocol"," *Nonlinear Dynamics*, vol. 87, no. 3, pp. 2073–2075, 2017.
[30] A. Kumari, V. Kumar, M. Y. Abbasi, S. Kumari, P. Chaudhary, and C.-M. Chen, "Csef: cloud-based secure and efficient framework for smart medical system using ecc," *IEEE Access*, vol. 8, pp. 107 838–107 852, 2020.
[31] C.-M. Chen, C.-T. Li, S. Liu, T.-Y. Wu, and J.-S. Pan, "A provable secure private data delegation scheme for mountaineering events in emergency system," *Ieee Access*, vol. 5, pp. 3410–3422, 2017.
[32] V. Kumar, R. Kumar, S. Jangirala, S. Kumari, S. Kumar, and C.-M. Chen, "An enhanced rfid-based authentication protocol using puf for vehicular cloud computing," *Security and Communication Networks*, vol. 2022, p. 8998339, 2022.
[33] T.-Y. Wu, L. Yang, Q. Meng, X. Guo, and C.-M. Chen, "Fog-driven secure authentication and key exchange scheme for wearable health monitoring system," *Security and Communication Networks*, vol. 2021, p. 8368646, 2021.
[34] "etherjsonlinedocumentation," (Accessed on 03/05/2022). [Online]. Available: https://docs.ethers.io/v5/
[35] "crypto-js-npm," 2022 Mar. [Online]. Available: https://www.npmjs.com/package/crypto-js
[36] "Ropsten," Mar 2022. [Online]. Available: https://www.anyblockanalytics.com/networks/ethereum/ropsten/
[37] "Rinkeby," May 2022. [Online]. Available: https://www.anyblockanalytics.com/networks/ethereum/rinkeby/
[38] "Web3," Dec 2021. [Online]. Available: https://en.wikipedia.org/wiki/Web3
[39] "Gas and fees — ethereum.org," 2022 Mar. [Online]. Available: https://ethereum.org/en/developers/docs/gas/