symmetry

MDPI

*Article*

# Effective Consensus-Based Distributed Auction Scheme for Secure Data Sharing in Internet of Things

**Xuedan Jia [1], Xiangmei Song [1,*] and Muhammad Sohail [2]**

[1] School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China
[2] Department of Computer Software Engineering, MCS, National University of Sciences and Technology (NUST), Islamabad 46000, Pakistan
* Correspondence: jlsxm@ujs.edu.cn

**Abstract:** In a traditional electronic auction, the centralized auctioneer and decentralized bidders are in an asymmetric structure, where the auctioneer has more ability to decide the auction result. This asymmetric auction structure is not fair to the participants and not suitable for data auctions in the Internet of Things (IoT). The blockchain-based auction system, with participant equality and fairness, is typically symmetrical and particularly suitable for IoT data sharing. However, when applied to IoT data sharing in reality, it faces privacy and efficiency problems. In this context, how to guarantee privacy and break the inherent performance bottleneck of blockchain is still a major challenge. In this paper, a consensus-based distributed auction scheme is proposed for data sharing, which enforces privacy preservation and collusion resistance. A reverse auction-based decentralized data trading model is introduced to solve the trust problem without a centralized auctioneer, where bidders reach consensus on the auction result. Specifically, we devise a differentially private auction mechanism to incentivize data owners to participate in data sharing. An effective hybrid consensus algorithm is constructed among bidders to reach consensus on the auction result with improved security and efficiency. Theoretical analysis shows that the proposed scheme ensures the properties of privacy preservation, incentive compatibility and collusion resistance. Experimental results reveal that the proposed mechanism guarantees the data sharing efficiency and has certain scalability.

**Keywords:** distributed auction system; consensus mechanism; privacy preservation; data sharing; Internet of Things



**Citation:** Jia, X.; Song, X.; Sohail, M. Effective Consensus-Based Distributed Auction Scheme for Secure Data Sharing in Internet of Things. *Symmetry* **2022**, *14*, 1664. https://doi.org/10.3390/sym14081664

Academic Editors: Alexander Zaslavski, Liangmin Wang, Keyang Cheng and Haiqin Wu

Received: 22 July 2022
Accepted: 7 August 2022
Published: 11 August 2022



## 1. Introduction

Data is central to the Internet of Things (IoT). With more and more devices connected to the IoT, the amount of data generated has exploded. IoT data are collected to serve many types of applications, including smart homes, smart grids and smart transportation [1]. How to efficiently utilize the data becomes a critical issue [2]. This calls for the development of an efficient data trading market. Auctions are one of the most popular mechanisms to motivate the IoT nodes to share their data and further earn money from the data [3–6]. Building on the characteristic of decentralization, it requires a symmetric peer-to-peer (P2P) auction paradigm for IoT data sharing.

The traditional electronic auction system, consisting of a centralized auctioneer and bidders, is an asymmetric structure [3,4,6]. The centralized asymmetric system faces trust issues, as a fully trustworthy auctioneer is difficult to realize [5]. Taking the centralized cloud-based IoT solution as an example, data owners rely on the third-party cloud for storage, access control and business services [7]. In this case, data owners and consumers have to trust the service provider and pay some fee for their services. In addition, the price and amount of data shared between the data owners and consumers can be decided even without the consent of the data owners. As a result of the centralization, these auction mechanisms lack transparency, and data owners have no way to ensure the legitimacy of the sharing [8].

Blockchain, as a trusted P2P symmetric network, is able to facilitate trust and guarantee transparency in a decentralized pattern among various entities [9]. Existing works exploit blockchain technology in data sharing [10,11], task offloading [12] and charging scheduling [13] in IoT. However, the transparency property of blockchain brings about concerns regarding the privacy protection of auction participants [14]. On one hand, an efficient auction tends to stimulate bidders to bid their true valuations to ensure fairness and achieve social welfare maximization. However, this results in the risk of leakage of private information, including the bidder's types of data, geographic location, active time and economic situation. The disclosure of this information would not only bring unfair profit to the informed entities but also cause economic damage to those whose information is disclosed [1]. On the other hand, the widely adopted proof-of-work (PoW) blockchain consensus algorithm is power-consuming and has a long confirmation delay [15,16].

Therefore, it is still an open and critical issue to develop a lightweight consensus mechanism for privacy-preserving auction while making a trade-off between the privacy and social efficiency of the auction mechanism.

In this paper, to cope with the aforementioned issues and realize a distributed reverse auction for IoT data sharing, we develop an effective consensus-based distributed auction scheme for secure data sharing. The main contributions of this work are threefold, as follows.

- A consensus-based distributed reverse auction framework for data sharing is proposed, where symmetric participants are grouped into clusters for privacy, and they reach consensus on the auction result without relying on any fully trusted or semi-trusted auctioneers;
- An incentive compatible privacy-preserving reverse auction mechanism is proposed to prompt data owners to share their data without worrying about privacy leakages. Differential privacy, symmetric encryption and zero-knowledge proofs are incorporated to design the auction mechanism, and a trade-off between privacy preservation and social efficiency of the auction is made;
- An effective hybrid consensus algorithm is constructed, where different kinds of witnesses are selected using anonymous verifiable random functions without peer interactions, and different operations can be conducted in parallel by varying witnesses. In this way, bidders reach consensus on the auction result with low computation costs without performing the auction processes repeatedly.

## 2. Related Works

To maximize the utility of massive IoT data, various solutions have been designed for data owners and consumers to carry out data trading effectively and securely. In this section, we will first review the research on auctions for data sharing and then summarize the research on blockchain for data sharing.

### *2.1. Auction for Data Sharing in IoT*

Some related works designed different auction mechanisms in data trading markets. Cao et al. [17] proposed an iterative auction mechanism to prevent selfish action and preserve private information. It also encourages consumers to bid reasonably during the auction process. Susanto et al. [18] proposed a double auction mechanism for mobile data trading, which achieves Nash equilibrium and truthfulness in a heterogeneous and dynamic environment. Except iterative auctions and double auctions, sealed-bid auctions have been investigated in the data market. Jiao et al. [19] proposed a sealed-bid auction model based on the Bayesian optimal mechanism to achieve profit maximization. Nonetheless, they only considered one round of auctions. An et al. [3] proposed a multi-round auction mechanism, which achieves incentive compatibility and false-name bidding proofness. The data are traded in bundles to prevent false-name bidding attacks.

For bidding privacy, many auction markets have been designed for different applications, such as spectrum [20–22], mobile crowd sensing [23], cloud computing [24],

and electrical vehicle charging markets [25]. Some privacy-preserving auction schemes have been proposed for data trading [5,26].

Brakerski et al. [26] designed a privacy-preserving data auction scheme using Paillier encryption and a one-time pad. It enables only the winner access to the data without trusted auctioneers. They also adopt a signature to verify whether the data has been manipulated. Gao et al. [5] designed a privacy-preserving auction protocol using homomorphic encryption for data auctions in cyber-physical systems. In addition, they also enhanced the auction scheme by adding a signature to further improve security.

### *2.2. Blockchain for Data Sharing in IoT*

Blockchain-based IoT systems are considered an effective technique for establishing secure data sharing. Li et al. [27] implemented a blockchain-based decentralized crowdsourcing system, which allows requesters to directly send tasks to workers without the involvement of traditional centralized trusted platforms. Kang et al. [10] realized secure and efficient data sharing in vehicular edge networks by exploiting permissioned blockchain and smart contracts. To resist collusion in traffic message exchanges, Feng et al. [28] proposed a consortium blockchain and region partition-based scheme to support trusted data management and deal with inside-and-outside collusive attacks. Xiong et al. [29] proposed an anti-collusion data auction mechanism based on smart contracts. Li et al. [30] presented an anonymous ad dissemination framework for advertising in internet of vehicles. The scheme achieves free-rider forbiddance and privacy preservation. The RSUs construct a consortium blockchain and manage the consensus phase. Chen et al. [11] proposed a secure and efficient data trading approach for the internet of vehicles using a consortium blockchain. An iterative double auction is adopted to determine the amount of traded data and the corresponding price. It ensures truthful data trading among vehicles. The edge servers construct the consortium blockchain and manage the auction phase instead of RSUs. Dai et al. [31] introduced a blockchain-based secure data trading ecosystem, where both data brokers and buyers are dishonest and none of them can access the raw data. Data processing algorithms encoded in smart contracts are deployed on and executed by the trusted hardware. Li et al. [32] proposed a blockchain-based data caching scheme and a decentralized data trading model to prevent the tampering of cache data and establish trust among participants. A double auction is used for data trading with maximum social welfare.

Different from the recent work, we take into consideration the performance bottleneck of the blockchain itself in this research. In addition, we achieve privacy preservation to incentivize data owners to share their data without worrying about privacy leakage. In our work, an effective consensus-based distributed reverse auction scheme is constructed, where all participants are in a symmetric structure.

## 3. Preliminaries

This section will sort out the preliminary knowledge, including the traceable ring signature, distributed Laplacian perturbation, Pedersen commitment, zero-knowledge range proof and anonymous verifiable random function.

### *3.1. Traceable Ring Signature*

A traceable ring signature scheme is a tuple of algorithms (Gen, Sig, Ver, Trace) [33].

- Gen: this takes security parameter $k \in N$ and outputs a public/secret-key pair $(PK, SK)$.
- Sig: this takes a secret key, $SK_i$, where $i \in N$, a tag $L = (issue, PK_N)$, and a message $m \in \{0,1\}^*$ and outputs signature $\sigma$.
- Ver: this takes tag $L = (issue, PK_N)$, message $m \in \{0,1\}^*$, and signature $\sigma$ and outputs a bit.
- Trace: this takes tag $L = (issue, PK_N)$ and two message/signature pairs, $(m, \sigma)$, $(m', \sigma')$ and outputs one of the following strings: "indep", "linked", or $PK$, where $PK \in PK_N$.

### *3.2. Distributed Laplacian Perturbation*

The Laplacian distribution is divisible, and it can be constructed as the sum of gamma distributions [34]. The distributed sanitization algorithm is simple: user $i$ calculates value $\widehat{X}_i = X_i + \mathcal{G}_1(N, \lambda) - \mathcal{G}_2(N, \lambda)$, where $\mathcal{G}_1(N, \lambda)$ and $\mathcal{G}_2(N, \lambda)$ denote two random values independently drawn from the same gamma distribution. If all values received from the $N$ users of a cluster are summed up, then $\Sigma_{i=1}^{N} \widehat{X}_i = \Sigma_{i=1}^{N} X_i + \Sigma_{i=1}^{N} [\mathcal{G}_1(N, \lambda) - \mathcal{G}_2(N, \lambda)] = \Sigma_{i=1}^{N} X_i + \mathcal{L}(\lambda)$.

### *3.3. Pedersen Commitment*

The Pedersen commitment [35] consists of the following algorithms:

- $\mathrm{Com}(x, r)$: this commits to a message $x \in \mathbb{Z}_p$ using blinding factor $r \in \mathbb{Z}_p$ and outputs $X = g^x h^r$.
- $\mathrm{Vfy}(X, x, r)$: this verifies whether $X$ commits to $x$ with blinding factor $r$ and outputs $\top$ on success; otherwise, it outputs $\perp$.

### *3.4. Zero-Knowledge Range Proof*

A zero-knowledge range proof allows a prover to convince a verifier that a committed value falls within a given range. In particular, given a commitment $X = g^x h^r \in \mathbb{G}$ for a witness $x \in \mathbb{Z}_p$, bulletproofs [36] allows a prover to generate a non-interactive zero-knowledge (NIZK) argument. We refer to the protocol generating range arguments as $\mathrm{RP} = (\mathrm{Setup}, \mathcal{P}, \mathcal{V})$, which consists of the following probabilistic polynomial-time algorithms:

- $\mathrm{RP.Setup}(1^\lambda, n, m)$: this takes $\lambda$ as the security parameter, $n$ as the range bit-width, and $m$ as the vector cardinality and outputs $\sigma$ as the common reference string (CRS).
- $\mathrm{RP}.\mathcal{P}(\sigma, X, x, r)$: this takes a commitment $X$ along with the opening vectors $x$ and $r$ and generates an argument $\pi$ to prove

$$\{(g, h \in \mathbb{G}, X; x, r) : X = g^x h^r \wedge x \in [0, 2^n - 1]\}.$$

- $\mathrm{RP}.\mathcal{V}(\sigma, X, \pi)$: this returns $\top$ if it accepts $\pi$; otherwise, it returns $\perp$.

### *3.5. Anonymous Verifiable Random Function*

An anonymous verifiable random function (AVRF) [37] is the tuple of the algorithms Gen, AVRFprove, AVRFverify, and Update defined as follows, where $H_1$ has range $\{0,1\}^{\ell_\alpha}$, $H_2$ has range $\mathbb{G}$ and $\ell_\alpha$ is the output length of AVRF.

- $\mathrm{Gen}(1^\kappa)$: this chooses a generator $g$ of a group of order $p$ such that $p = \Theta(2^\kappa)$. It then samples a random $k \in \mathbb{Z}_p$ and outputs $(PK_i, SK_i)$, where $SK_i = k, v = g^k, PK_i = (g, v)$.
- $\mathrm{AVRFprove}(PK, x)$: this computes $u = H_2(x)^{SK}, y = H_1(x, u)$ and $\pi' \leftarrow Proof\{(SK) : log_{H_2(x)}(u) = log_g(v)\}$ and outputs $(PK, y, \pi = (u, \pi'))$.
- $\mathrm{AVRFverify}(x, y, \pi)$: this outputs 1 if $y = H_1(x, u)$ and $\pi$ verifies, and 0 otherwise.
- $\mathrm{Update}(PK)$: this chooses a random $r \in \mathbb{Z}_p$, computes $g' = g^r, v' = v^r$, sets $PK' = (g', v')$, and outputs $PK'$.

## 4. System Model and Design Goals

In this section, we introduce the system model, threat model and design goals.

### *4.1. System Model*

In this paper, we introduce a consensus-based distributed reverse auction model, where the sellers compete to sell data to a buyer. Generally speaking, in data sharing markets, the reverse auction mechanism is suitable for the situation where multiple data owners tend to sell data to one data consumer or data collector [1]. In a reverse auction, the buyer submits his data requirements and the maximum price he is willing to pay for each unit of data. Sellers who have the required data can submit their bidding information

including valuation per unit of data and supply volume. Those who bid lower than the buyer's price and the clearing price are the auction winners, and the payment equals the largest bid value lower than the buyer's maximum price. The decentralized system model is illustrated in Figure 1.

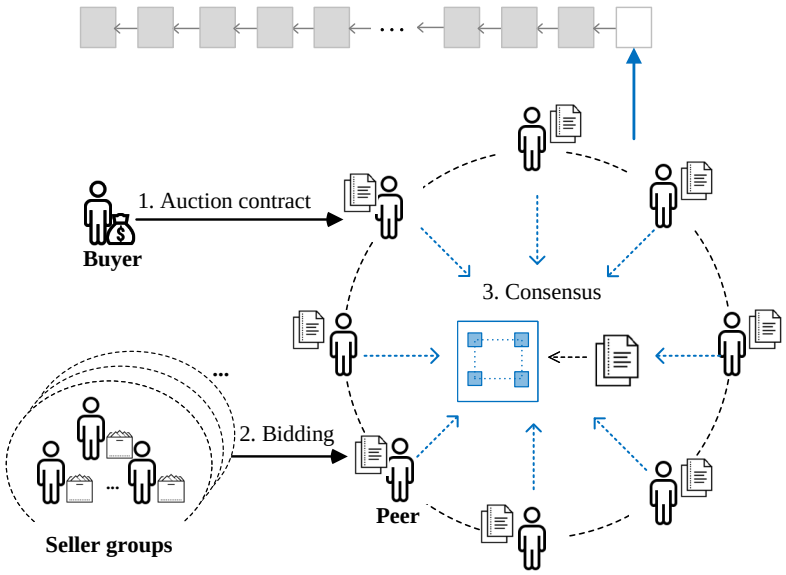



**Figure 1.** The decentralized reverse auction model.

In the distributed system, there is no centralized auctioneer. The reverse auction is realized through off-chain allocation and on-chain verification. Three categories of participants are involved in the proposed auction scheme: sellers $U_i$, buyer $B$ and consensus nodes, where sellers and buyer are data providers and data consumers, respectively, and consensus nodes are selected from bidders. There are two types of consensus nodes, proposer $\mathcal{P}$ and validator, which are in charge of the auction solution and verification, respectively. The validator nodes are further divided into two groups, result verifier $\mathcal{RV}$ and proof verifier $\mathcal{PV}$, which are responsible for verifying the auction result and the winners' proofs. After the verification, the validator submits votes to the contract for on-chain verification of the auction result. Finally, all results and proofs are recorded on the blockchain without manipulation.

The auction model in this paper is a reverse sealed-bid auction. The buyer is responsible for deploying the auction contract and publishing his data requirement. He commits the maximum price he would pay and reveals it after the bidding phase. Data owners can bid their valuation and supply volume during the bidding phase. For privacy preservation, sellers are grouped into clusters of size $N$, and Laplace noise is added to each cluster. $U_i$ randomly selects some nodes from the cluster using a secure pseudo-random function (PRF) [34] such that if $U_i$ selects $U_j$, $U_j$ also selects $U_i$. In particular, $U_i$ selects $U_j$ if $PRF(K_{i,j}, r_1) \leqslant \frac{\omega}{N-1}$. Afterwards, both users generate a common dummy key $dk_{i,j}$ from their pairwise key $K_{i,j}$: $dk_{i,j} = \frac{i-j}{|i-j|} \times PRF(K_{i,j}, r_2)$, where $r_2 \neq r_1$ is a public value. The noised bidding information is then encrypted with the dummy key. In this way, sellers' bid value and supply volume are preserved from other participants and attackers.

After the bidding phase, consensus nodes are selected to determine the winners and their payments. After verification and consensus about the auction result, it is eventually published on the blockchain. Finally, the transactions will be completed. If the winners refuse to conduct the transactions within a limited time, they will lose their deposit as a punishment.

### *4.2. Threat Model and Design Goals*

Before we present the formal development of our solution, we establish the threat model as well as design goals for our system.

This paper considers both malicious participants and outside attackers. All sellers and buyer are selfish and rational, i.e., they aim at maximizing their own profit. For example, sellers may collude together to manipulate the auction result. In addition, bidders are encouraged to submit their truthful bidding, including valuation and the supply volume. Nonetheless, if bidding profiles are obtained by a malicious user or adversary, the bidder privacy is subverted, and the auction fairness is undermined. On obtaining other's private information, attackers can gain unfair benefits and advantages. Blockchain nodes are honest but curious. To protect privacy, on-chain operations should not reveal any private information. The outside attackers may eavesdrop on the communication and try to infer the private information.

With respect to the threat model, we define the security and privacy notions of interest. To support the auditability and traceability of blockchain-based applications, we assume that each user is configured with a private key and obtains the corresponding certificate from the trusted Certification Authority. On this basis, we carry out a distributed data auction with privacy preservation and collusion resistance. The scheme in this paper needs to achieve the following goals.

(1) Distributed Auction.
All participants are in a symmetric structure. Auction allocation and pricing do not depend on trusted third parties. Bidders reach consensus on the auction result in a P2P manner with the assistance of smart contracts.
(2) Privacy Preservation.
First, the real identities of the data providers and data consumers participating in the auction are hidden and cannot be inferred from the user account address, public key, signature and other information. Second, bidders submit their own bids without knowing others' valuation, and all bids remain private throughout the auction process.
(3) Incentive Compatibility.
The bidders can obtain highest utility if and only if they submit their bids truthfully.
(4) Collusion Resistance.
Bidders can not collude together to manipulate the auction results for illicit profit. Peer nodes are prevented from colluding to announce false auction results for unfair profits.
(5) Efficiency.
The overhead realizing the above goals should be acceptable from the perspective of system users. The consensus process should minimize communication and computation overhead instead of repeating the costly auction assignment and verification calculation by all miners.

## 5. Effective Consensus-Based Distributed Reverse Auction

The proposed scheme consists of three stages: the preparation stage, auction stage and consensus stage. In the first stage, a new auction smart contract is deployed by the buyer, and bidders register themselves to join in the auction. Bidders cluster together to establish dummy keys for bidding. During the auction stage, bidders submit their bid value and supply volume in private using differential privacy and modulo addition-based encryption [34]. The private bidding information is gathered and handled to generate the auction result. Then, the winning bidders prove themselves privately using a zero-knowledge range proof without revealing their bid value. In the last stage, consensus nodes are selected, and they reach consensus on the auction result. The first witnesses are selected including proposer and verifier. Then, they respectively perform the auction allocation and verification off-chain in an efficient form. Finally, the contract confirms the auction result by collecting verifiers' votes.

### *5.1. Preparation Stage*

A buyer deploys a new smart contract on the blockchain in which the data demand is announced. Potential bidders monitor the blockchain for new auctions and register themselves to join the auction. To participate in the auction, each bidder $U_i$ needs to commit his bid $BidComm_i = g^{bid_i}h^{r_i}$ and supply volume $VolComm_i = g^{vol_i}h^{ran_i}$ and make a deposit which has a prescribed minimum $dpt_\Delta$.

When sellers join the auction, they are grouped into clusters of size $N$ to realize differential private bidding in the next stage. The detailed process of bidder grouping and key establishment is shown in Algorithm 1. First, bidders are grouped in to clusters of $N$ users according to their joining time. Then pairwise keys $K_{i,j}$ between each pair of users inside a cluster are established, which is done by using the traditional Diffie–Hellman key exchange protocol. Finally, each user, $U_i$, selects $l$ other users of the cluster randomly to generate dummy keys. $U_i$ selects $U_j$ if $PRF(K_{i,j}, r_1) \leqslant \frac{\omega}{N-1}$, where $r_1$ is a public value. It should be noted that if $U_i$ selects $U_j$, $U_j$ also selects $U_i$ because of the secure pseudo-random function (PRF) [34]. Afterwards, both users generate a common dummy key $dk_{i,j}$ from their pairwise key $K_{i,j}$: $dk_{i,j} = \frac{i-j}{|i-j|} \times PRF(K_{i,j}, r_2)$. We denote by $l$ the number of selected participating users, and $ind_i[j](for\ j = 1, \cdots, l)$ the index of the $l$ users selected by node $U_i$. Note that $dk_{i,j} = -dk_{j,i}$.

**Algorithm 1** Bidder Grouping and Key Establishment

**Input:** Seller registration information, pseudo-random function (PRF), $\omega, n = 0, N, r_1, r_2$;
**Output:** $K_{i,j}, ind_i, dk_{i,ind_i[j]}$, seller groups;

**for** $n < N$ **do**
  Add user to the same group;
  $n$ ++;
**end for**
**for** $1 < i, j < N$ **do**
  Generate $K_{i,j} = K_{j,i}$;
**end for**
**for** $1 < i < N$ **do**
  Randomly choose $l$ other members:
  **for** $1 < j \neq i < N$ **do**
    **if** $PRF(K_{i,j}, r_1) \leqslant \frac{\omega}{N-1}$ **then**
      $j$ is added to $ind_i$;
    **end if**
  **end for**
  **for** $1 < j < l$ **do**
    Generate $dk_{i,j} = \frac{i-j}{|i-j|} \times PRF(K_{i,j}, r_2)$;
  **end for**
**end for**
**return** $K_{i,j}, ind_i, dk_{i,ind_i[j]}$;

The grouped bidders can join the current auction. The dummy key is used to encrypt the bidding information in the auction stage to preserve bidding information privacy, including bid privacy and supply volume privacy. As a result, the consensus nodes cannot decrypt the individual ciphertexts because they do not know $dk$. However, by adding up the ciphertexts of a given cluster, they can cancel out all $dk$s and retrieve the cluster bidding, which is described in detail in next subsection.

### *5.2. Auction Stage*

In the auction stage, data suppliers report their bid value and supply volume. The proposer selected from the bidders computes the auction solution according to Algorithm 2. Then winning bidders prove themselves in time. The auction result is verified and chained to the block after consensus.

**Algorithm 2** Bidding Information Processing and Valid Price Determination

**Input:** Bidding information $Enc(\widehat{bid}_i)$, $Enc(vol_i)$, the number of clusters $ClusterNum$, the acceptable price of buyer $value_B$

**Output:** Group average bid $\widehat{bid}_G$, group supply volume $vol_G$, valid price $vp$

**function** BIDPROCESSING($Enc(\widehat{bid}_i), Enc(vol_i)$)
  **while** $j \leqslant ClusterNum$ **do**
    **while** $i \leqslant N$ **and** $i \in Cluster_j$ **do**
      $\widehat{Gbid}_j = \widehat{Gbid}_j + Enc(\widehat{bid}_i)$
      $vol_{G_j} = vol_{G_j} + Enc(vol_i)$
    **end while**
    $\widehat{bid}_{G_j} = \frac{\widehat{Gbid}_j}{N}$
    $\widehat{bid}_G[j] = \widehat{bid}_{G_j}$
    $vol_G[j] = vol_{G_j}$
  **end while**
  **return** $\widehat{bid}_G[], vol_G[]$
**end function**
**function** VALIDPRICEDETERMINE($\widehat{bid}_G[], vol_G[]$)
  Resort $\widehat{bid}_G[]$ in ascending order
  Resort $vol_G[]$ according to the new $\widehat{bid}_G[]$
  **for** $j = 1; j++; j \leqslant ClusterNum$ **do**
    **if** $\widehat{bid}_G[j] \geqslant value_B$ **then**
      $vp = \widehat{bid}_G[j-1]$
      Break
    **end if**
  **end for**
  **return** $vp$
**end function**

5.2.1. Bidding

Grouped bidders have committed their bid value and make a deposit in the previous stage. They will bid in private in this stage through obfuscating the true bid value and encrypting the bid and volume information. Each user bids as follows:

**Step 1 (Bid obfuscation)**: Calculate the obfuscated bid value

$$\widehat{bid}_i = bid_i + \mathcal{G}_1(N, \lambda) - \mathcal{G}_2(N, \lambda) \tag{1}$$

where $\mathcal{G}_1(N, \lambda)$ and $\mathcal{G}_2(N, \lambda)$ are random values, independently drawn from the same gamma distribution. $N$ is the cluster size.

**Step 2 (Bidding information encryption)**: Each noisy bid value $\widehat{bid}_i$ is then encrypted using the modulo addition-based encryption:

$$Enc(\widehat{bid}_i) = \widehat{bid}_i + \Sigma_{j=1}^{l} dk_{i,ind_i[j]} (mod\ m) \tag{2}$$

where $m$ is a large integer. The supply volume is also encrypted:

$$Enc(vol_i) = vol_i + \Sigma_{j=1}^{l} dk_{i,ind_i[j]} (mod\ m) \tag{3}$$

The dummy keys are needed to prevent the peer node from retrieving $\widehat{bid}_i$ and the volume $vol_i$ information.

**Step 3 (Broadcasting the bidding information)**: Sign the bidding information with the ring signature scheme and broadcast $(Enc(\widehat{bid}_i), Enc(vol_i), sig)$.

5.2.2. Auction Allocation

After the bidding phase, any peer node can compute the valid price using all the bidding information. First, the bidding information is gathered and decrypted to get the

group bidding information; then, the valid price is determined. Finally, winning bidders prove themselves in private.

**Step 1 (Bidding information processing)**: The bidding information from $N$ cluster members are collected and decrypted to obtain the group bid $\widehat{Gbid} = \Sigma_{i=1}^{N} Enc(\widehat{bid}_i)$ and the group supply volume $vol_G = \Sigma_{i=1}^{N} Enc(vol_i)$, respectively (Line 1–12).

$$\begin{aligned} \widehat{Gbid} &= \Sigma_{i=1}^{N} Enc(\widehat{bid}_i) \\ &= \Sigma_{i=1}^{N} (\widehat{bid}_i + \Sigma_{j=1}^{l} dk_{i,ind_i[j]}) \\ &= \Sigma_{i=1}^{N} (bid_i + \mathcal{G}_1(N,\lambda) - \mathcal{G}_2(N,\lambda)) + \Sigma_{i=1}^{N} \Sigma_{j=1}^{l} (dk_{i,ind_i[j]}) \end{aligned} \tag{4}$$

As $dk_{i,j} = -dk_{j,i}$, there is $\Sigma_{i=1}^{N} \Sigma_{j=1}^{l} dk_{i,ind_i[j]} = 0$. Thus, the Laplacian noise is generated in a fully distributed way, as is illustrated in the bidding phase.

$$\begin{aligned} \widehat{Gbid} &= \Sigma_{i=1}^{N} bid_i + \Sigma_{i=1}^{N} (\mathcal{G}_1(N,\lambda) - \mathcal{G}_2(N,\lambda)) \\ &= \Sigma_{i=1}^{N} bid_i + \mathcal{L}(\lambda) \end{aligned} \tag{5}$$

In this way, no peer node can decrypt and obtain the individual bid without the dummy keys. By adding all the encrypted contributions of a cluster, $dk$ cancels one another and the noised bid sum reveals. The added noise ensures the differential privacy of each bid [34].

The group average bid is computed as $\widehat{bid}_G = \frac{\widehat{Gbid}}{N}$. The group volume is the addition of all menbers' volume $vol_G = \Sigma_{i=1}^{N} Enc(vol_i) = \Sigma_{i=1}^{N} vol_i$.

$$\begin{aligned} vol_G &= \Sigma_{i=1}^{N} Enc(vol_i) \\ &= \Sigma_{i=1}^{N} (vol_i + \Sigma_{j=1}^{l} dk_{i,ind_i[j]}) \\ &= \Sigma_{i=1}^{N} (vol_i) + \Sigma_{i=1}^{N} \Sigma_{j=1}^{l} (dk_{i,ind_i[j]}) \\ &= \Sigma_{i=1}^{N} vol_i \end{aligned} \tag{6}$$

**Step 2 (Valid Price Determination)**: The valid price is computed using the group average bids and group volumes. To ensure truthfulness, a good pricing mechanism should ensure that the payment of winners is not determined by their bid value. In doing so, the winners cannot manipulate their bid by misreporting their valuations. The valid price determination procedure is as shown in Algorithm 2 from Line 13 to Line 22.

(i) The valuation and supply of a seller group is recorded as $\widehat{bid}_{G_j}, vol_{G_j}$, respectively. Then, the seller groups' valuations $\widehat{bid}_G[]$ are arranged in ascending order.

(ii) Plot the supply $vol_G$ of seller groups versus the valuation $\widehat{bid}_G$ of seller groups in ascending order. Similarly, plot the revealed acceptable price of buyer $value_B$ in the same figure. The intersection shows the valid price $vp$. It is defined as the key price of winning sellers. Finally, a bidder $U_i$ wins if, and only if, his bid value is smaller than the key price: $bid_i < vp$.

**Step 3 (Privacy-preserving Washing Out)**: The valid price is determined in Step 2. Those bidders whose valuations are lower than the valid price $bid_i < vp$ are the winners. In this subsection, bidders prove themselves that they are the winners without revealing their bid value to wash out those with higher bids. The winners are saved to the winners set $W_S$. The procedure of the washing-out protocol is illustrated in Figure 2.

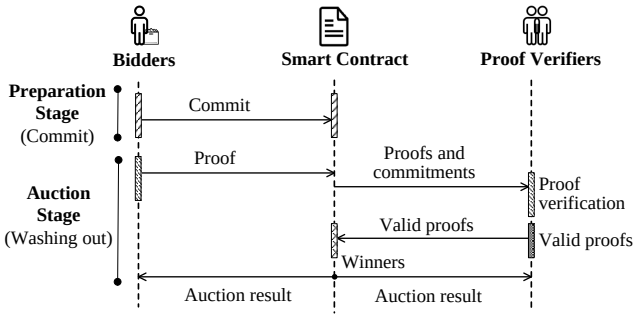



**Figure 2.** The washing-out protocol.

(i) Winning sellers prove that their committed bid values are indeed smaller than the valid price, i.e., $bid_i < vp$.

$$\pi_i \leftarrow \text{RP}.\mathcal{P}(\sigma, BidComm_i, bid_i, r_i) \tag{7}$$

(ii) The proofs are verified according to the on-chain commitment. Bidders who can provide valid proofs are the final winners. The winners are saved to the set $W_S$.

$$\text{RP}.\mathcal{V}(\sigma, BidComm_i, \pi_i) = 1 \tag{8}$$

The buyer pays each winning seller by $P$ per unit, which is determined by the valid price in the former step $P = vp$. Each winning seller receives $payment = vp \times vol_i$ by providing $vol_i$ units of data to the buyer.

### *5.3. Consensus Stage*

In this part, we devise an effective hybrid consensus protocol to efficiently reach consensus on the auction result and enhance the security of the distributed data auction. The detailed design of the consensus protocol consists of the following three steps: witness selection, block generation by proposer and verification by validators. Figure 3 elaborates the optimized hybrid consensus procedure, where the efficiency comes from efficient witness selection by applying AVRF and parallel operations conducted by different kinds of witnesses.

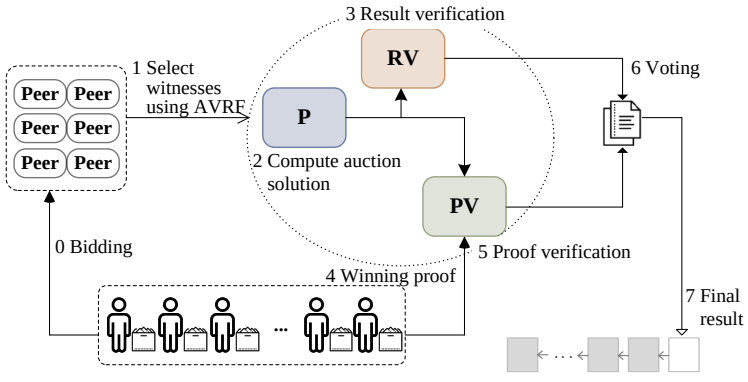



**Figure 3.** The consensus procedure.

#### 5.3.1. Witness Selection

There are two categories of witnesses in our proposed consensus protocol: the proposer $\mathcal{P}$, the block producer responsible for computing the auction allocation and payment, and the validator, responsible for the block verification and commitment. The validators are further divided into two categories: the result verifier $\mathcal{RV}$ and proof verifier $\mathcal{PV}$. The witnesses are decided using AVRF without peer interaction, which guarantees fast and efficient auction allocation and verification.

The witness selection works as shown in Algorithm 3. The probability $p_i$ that a bidder $U_i$ is selected as a witness in this auction is independent of others. It depends only on $U_i$'s relative deposit $\alpha_i = dpt_i/DPT$, where $DPT$ is the total stake in the system. $p_i$ is precisely given by $p_i = \varnothing_f(\alpha_i) = 1 - (1-f)^{\alpha_i}$ (Line 2), where $f$ is the difficulty parameter.

**Algorithm 3** Witness selection.

**Input:** $dpt_i, DPT, j = 1, Num_{\mathcal{P}}, Num_{\mathcal{RV}}, Num_{\mathcal{PV}}$;
**Output:** $\mathcal{P}, \mathcal{RV}, \mathcal{PV}$;
1: Compute $\alpha_i = dpt_i/DPT$
2: $p_i = 1 - (1-f)^{\alpha_i}$
3: **procedure** $\mathcal{P}$DETERMINE($dpt_i, DPT, Num_{\mathcal{P}}$)
4: **for** $j \leqslant Num_{\mathcal{P}}$ **do**
5: $x = (PK_i, Proposer, j)$
6: $u = H_2(x)^{SK_i}, y = H_1(x,u)$
7: **if** $y < 2^{\ell_\alpha} \times p_i$ **then**
8: $U_i$ is the $j$-th proposer
9: Generate $\pi' \leftarrow Proof\{(SK_i) : log_{H_2(x)}(u) = log_g(v)\}$
10: **end if**
11: **end for**
12: **return** $(\mathcal{P} : PK, y, \pi = (u, \pi'))$
13: **end procedure**
14: **procedure** $\mathcal{RV}$DETERMINE($dpt_i, DPT, Num_{\mathcal{RV}}$)
15: **for** $j \leqslant Num_{\mathcal{RV}}$ **do**
16: $x = (PK_i, Verifier_R, j)$
17: $u = H_2(x)^{SK_i}, y = H_1(x,u)$
18: **if** $y < 2^{\ell_\alpha} \times p_i$ **then**
19: $U_i$ is the $j$-th result verifier
20: Generate $\pi' \leftarrow Proof\{(SK_i) : log_{H_2(x)}(u) = log_g(v)\}$
21: **end if**
22: **end for**
23: **return** $(\mathcal{RV} : PK, y, \pi = (u, \pi'))$
24: **end procedure**
25: **procedure** $\mathcal{PV}$DETERMINE($dpt_i, DPT, Num_{\mathcal{PV}}$)
26: **for** $j \leqslant Num_{\mathcal{PV}}$ **do**
27: $x = (PK_i, Verifier_P, j)$
28: $u = H_2(x)^{SK_i}, y = H_1(x,u)$
29: **if** $y < 2^{\ell_\alpha} \times p_i$ **then**
30: $U_i$ is the $j$-th proof verifier
31: Generate $\pi' \leftarrow Proof\{(SK_i) : log_{H_2(x)}(u) = log_g(v)\}$
32: **end if**
33: **end for**
34: **return** $(\mathcal{PV} : PK, y, \pi = (u, \pi'))$
35: **end procedure**

$U_i$ evaluates AVRF using $SK_i = k$ and $(y, \pi) = \text{AVRF}(PK_i, role, j)$. Here, $role \in \{Proposer, Verifier_{\mathcal{R}}, Verifier_{\mathcal{P}}\}$, and the number of each kind of witnesses is $Num_{\mathcal{P}}$, $Num_{\mathcal{RV}}$,$Num_{\mathcal{PV}}$, respectively. To check if $U_i$ is a witness, the stakeholder computes his threshold $T = 2^{\ell_\alpha} \times p_i$, where $\ell_\alpha$ is the output length of the AVRF. $U_i$ is selected as a witness if $y < T$. The proof $\pi$ allows others to verify $U_i$'s claim using $U_i$'s verification key. The decision function for witness determination is given by:

$$PD(dpt_i; y) = \begin{cases} 1, & if\ y < 2^{\ell_\alpha} \times (1 - (1-f)^{\frac{dpt_i}{DPT}}); \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The proposer selected, donated as $\mathcal{P}$, can prove his validity by AVRFprove$(PK_{\mathcal{P}}, x)$, where $PK_{\mathcal{P}} = (g, v)$, $x = (PK_{\mathcal{P}}, Proposer, j)$. $\mathcal{P}$ first computes $u = H_2(x)^{SK_{\mathcal{P}}}, y = H_1(x,u)$ using his secret key, then secondly generates proof that he is the proposer $\pi' \leftarrow$

$Proof\{(SK_{\mathcal{P}}) : log_{H_2(x)}(u) = log_g(v)\}$ and finally outputs $(PK_{\mathcal{P}}, y, \pi = (u, \pi'))$. Other peer nodes can verify that $\mathcal{P}$ is indeed the proposer responsible for auction solution.

The stakeholder $U_i$ is selected as the $j-$th result verifier $\mathcal{RV}$ if there is $y < T$ for $(y, \pi) = \text{AVRF}(PK_i, Verifier_R, j)$. $\mathcal{RV}$ proves his correctness by generating proof $\pi' \leftarrow Proof\{(SK_{\mathcal{RV}}) : log_{H_2(x)}(u) = log_g(v)\}$ and outputting $(PK_{\mathcal{RV}}, y, \pi = (u, \pi'))$. Other peer nodes verify $\mathcal{RV}$'s validity using AVRFverify.

The proof verifier $\mathcal{PV}$ is selected in the same way with the node type $Verifier_P$ (Line 25–35).

#### 5.3.2. Auction Allocation by Proposer

Instead of performing costly assignment calculations on-chain, the smart contract allows a dedicated node to perform the allocation off-chain according to the on-chain information. The off-chain calculation of the auction solution does not suffer from the overhead associated with distributed and privacy-preserving solutions such as secure multi-party computation [38]. The auction solution provided by the proposer declares the valid price.

The proposer performs Algorithm 2 to deal with the bidding information and determine the valid price, which is then verified by the result verifier. Bidders who bid lower than the valid price are the winners. Winning bidders prove themselves using zero-knowledge range proof according to the washing-out protocol.

In our scheme, the number of proposers is set to 1 for efficiency. As there is only one proposer, we face a problem if the proposer is offline and fails to submit the auction result in time. To overcome this issue, we have result verifiers. They are responsible for verifying and voting for the proposer's auction resolution. If the proposer fails to submit the auction solution in time, the result verifier can act as the leader and give the solution. This is reasonable because that the result verifiers can also calculate the auction solution while their verification.

#### 5.3.3. Verification by Validators

The auction solution, once submitted, can be contested within a specified amount of time by any user, while the selected result verifiers start their computation and verification in parallel with the proposer. Once the solution is submitted, verifiers can vote for the solution on the beat. The smart contract can efficiently test the validity of the solution by calculating the votes. Alternatively, if enough validators verify the auction result, it is received as the final solution. Bidders can perform the data trading.

The validators first check the validity of the proposer: $\text{AVRFverify}(x, y, \pi)$. The validators output 1 if $y = H_1(x, u)$ and $\pi$ verifies, and 0 otherwise. Then the validators conduct two rounds of verifications: first, the result verifiers verify the group bids sorting; second, the proof verifiers verify the winning sellers' proofs. The procedure is given in Algorithm 4. The winners' proofs are verified if $\text{RP}.\mathcal{V}(\sigma, BidComm_i, \pi_i) = 1$. Bidders who can provide valid proofs are the final winners, who are saved to the winners set $W_S$.

The 2/3 voting solution is adopted, and the auction solution generated by the proposer can be immediately confirmed if more than $\frac{2 \times Num_{\mathcal{RV}}}{3}$ verification information is received. The verification of the auction solution can be conducted in the meantime with the proposer to save verifying and committing time, thereby greatly reducing the consensus delay. As we can see in Figure 3, the improved hybrid consensus procedure is presented based on an anonymous verifiable random function.

**Algorithm 4** Validator verification.

**Input:** $Enc(\widehat{bid}_i), value_B, \pi_i, BidComm_i$;
**Output:** $W_S$, $Block$;

**procedure** RESULTVERIFICATION($Enc(\widehat{bid}_i), value_B$)
  AVRFverify$(x, y, \pi)$:
  **if** $y = H_1(x, u)$ **and** $\pi$ verifies **then**
    $\mathcal{P}$ is valid
  **end if**
  Verify the group bids sorting $\widehat{bid}_G$
  Verify the valid price $vp$
  **if** $vp$ verifies **then**
    **return** Auction result verified
  **else**
    **return** Wrong result
  **end if**
**end procedure**
**procedure** PROOFVERIFICATION($\pi_i, BidComm_i$)
  **while** $\pi_i$ **do**
    **if** RP.$\mathcal{V}(\sigma, BidComm_i, \pi_i) = 1$ **then**
      Winner $U_i$ is verified
      $U_i$ is added to $W_S$
    **end if**
  **end while**
  **return** $W_S$
**end procedure**

## 6. Theoretical Analysis

We now provide the detailed theoretical analysis of the proposed scheme, including privacy preservation, incentive compatibility and collusion resistance.

### *6.1. Privacy Preservation*

The proposed scheme achieves privacy preservation, including bidder anonymity, bid privacy and supply volume privacy. Bidder anonymity, i.e., the fact that the bidder's real identity cannot be inferred from any information during the entire auction process, is guaranteed through the ring signature and an anonymous verifiable random function. The ring signature adopted allows a bidder to leak messages anonymously, without the risk of identity escrow [33]. The anonymous verifiable random function preserves the witnesses' identities during the consensus process. Additionally, the bidder's public key $PK$ can be updated using Update ($PK$) to break the linkability [37].

Both bid privacy and supply volume privacy are preserved by differential privacy and symmetric encryption. In our privacy-preserving auction, two random processes, perturbation and clustering, are used to ensure the privacy protection. For bid privacy, each bid is first obfuscated through distributed Laplacian perturbation, then symmetrically encrypted using the dummy keys in the bidder cluster. Because of the divisibility of the Laplacian distribution and the $\frac{1}{\lambda}-$differential privacy property [34,39], all bidders in a cluster maintain a private bid value. Supply volume is also encrypted using the dummy keys. The peer nodes can only decrypt the cluster volume and are prevented from retrieving the volume information $vol_i$ of a bidder. For the washing out in the auction stage, bid privacy is implemented through the zero-knowledge property [36]. The volume privacy during the trading can be preserved by using a secure communication channel between the buyer and bidder, where the precise trading process is out of our scope.

### *6.2. Incentive Compatibility*

According to our system model described in Section 4.1, we consider that some bidders would misreport their bidding information to gain a higher utility. The bidding information that the bidders can misreport as false is their valuation. We now explain that the bidders

cannot improve their utility by misreporting their values, i.e., our scheme is incentive-compatible. For any bidder $U_i$, the utility obtained by misreporting is not larger than the utility acquired by truthful bidding. The seller's utility is denoted as the difference between the payment by the buyer and the true valuation. For simplicity of explanation, we consider that there is no seller whose valuation is larger than the buyer's reserved price, since the seller whose valuation is larger than the reserved price will not affect the auction results. To prove the incentive compatibility, we consider untruthful bidding.

**Theorem 1.** *The proposed scheme satisfies incentive compatibility, meaning that a seller cannot improve his utility by untruthful bidding in the system.*

**Proof of Theorem 1.** In our privacy-preserving auction scheme, perturbation and clustering are used to ensure the privacy protection. During the bidding phase, the bid value is obfuscated by $\widehat{bid}_i = bid_i + \mathcal{G}_1(N, \lambda) - \mathcal{G}_2(N, \lambda)$. All the obfuscated bid values are added according to cluster to decrypt and obtain the cluster average bid $\widehat{bid}_G = \Sigma_{i=1}^{N} Enc(\widehat{bid}_i)/N$. These random processes lower the impact of bid changing on the auction result. According to our allocation mechanism, the winner's payment is fixed lower than the buyer's reserved price. Thus, a larger evaluation of the seller will only reduce the probability of winning, and a smaller evaluation will not improve the payment. The utility will not be increased as a result. In conclusion, a seller cannot improve his utility by untruthful bidding within our auction scheme. Above all, we can conclude that the proposed scheme satisfies incentive compatibility. □

### *6.3. Collusion Resistance*

Our proposed mechanism guarantees that no single bidder has incentive to lie for extra benefit and prevents bidder coalitions from colluding to improve their utilities. This comes from the fortunate property of differential privacy that it degrades smoothly with the number of changes in the data set.

**Theorem 2** ([40]). *For the proposed auction mechanism $\mathcal{M}$ giving $\epsilon-$differential privacy and the non-negative utility function $\mu$ of its range, for any $D_1$ and $D_2$ differing on, at most, $t$ inputs*

$$E[\mu(\mathcal{M}(D_1))] \leqslant \exp(\epsilon t) \times E[\mu(\mathcal{M}(D_2))] \tag{10}$$

This applies to the notable case that $\mu$ is the sum of the utility functions of $t$ bidders, ensuring that their collective utility does not increase by much. Apparently, $\exp(\epsilon t)$ is larger than $\exp(\epsilon)$, and the resistance to collusion declines as the size of the coalition increases. For collusion coalitions smaller than $\frac{1}{\epsilon} = \lambda$, the gain is essentially linear in size. Empirical studies on commercial non-collusion-resistant auctions reveal that most collusion coalitions are small (<6 bidders per coalition) [41].

Note that, to achieve the above security properties, differential privacy and clustering are adopted. This will influence the auction social efficiency, which is further discussed in the next section.

## 7. Performance Analysis

In this section, we conduct extensive theoretical and experimental analysis to evaluate the performance of the proposed scheme.

### *7.1. Computation and Communication Cost*

First, we discuss the computation overhead of our proposed scheme, which is as shown in Table 1. During the preparation stage, each bidder generates commitments to his bid price and supply volume. Each commitment requires two exponentiations and one multiplication. Bidders also need to form clusters for private bidding, during which they are required to establish two shared keys $K_{i,j}, dk_{i,j}$ as in Algorithm 1. The buyer commits his

preserved accessible price. In the auction stage, a bidder first computes the perturbated bid value, then encrypts the obfuscated bid and supply volume, and finally signs the bidding information. After the valid price is determined, a winning bidder proves his bid is smaller than the valid price. After bidders make a deposit to join the auction, they can perform AVRF locally to verify if they are selected as a witness: $\mathcal{P}$, $\mathcal{RV}$ or $\mathcal{PV}$. As witnesses are divided into different categories responsible for different operations, they can conduct the consensus process in parallel with the auction allocation.

**Table 1.** Computation overhead for different entities.

| Entity | | Preparation Stage | | Auction Stage | | | Consensus Stage |
|---|---|---|---|---|---|---|---|
| | | Key Establishment | Commitment | Perturbation | Cryptogram | Bulletproofs | |
| Buyer | | - | $2Exp + Mul$ | - | - | - | - |
| Bidders | | $K_{i,j}, dk_{i,j}$ | $4Exp + 2Mul$ | 1 | $2Enc + Sig$ | RP.$\mathcal{P}$ | - |
| Witnesses | $\mathcal{P}$ | - | $AVRFprove + 2Dec + Div + Sort$ | | | | |
| | $\mathcal{RV}$ | - | - | $AVRFprove + AVRFverify + Sort$ | | | |
| | $\mathcal{PV}$ | - | - | $AVRFprove + AVRFverify +$ RP.$\mathcal{V}$ | | | |

Table 2 shows the communication overhead of the proposed scheme, where the user's address $Addr$ is 20B in length, the signature $\sigma$ is $|G| + 2|\mathbb{Z}_p|$, the symmetric encryption result is in $\mathbb{Z}_p$, and the verifier's vote $vote$ after his verification is 0 or 1. During the preparation stage, each bidder submits his bid commitment and volume commitment. Each commitment is an element in $\mathbb{G}$. Bidders' clustering can be conducted off-chain and here ignores the corresponding communication overhead. The buyer publishes the auction contract, in which $data$ represents the auction code. In the auction stage, a bidder broadcasts his bidding information $Enc(\widehat{bid}_i), Enc(vol_i)$ with his signature, while the buyer reveals his preserved price $value_B$ after the bidding phase. All peers perform AVRF locally to check if they are selected as a $\mathcal{P}$, $\mathcal{RV}$ or $\mathcal{PV}$, so no communication is needed. After the verification, 2/3 positive votes need to be collected by the smart contract to finish the consensus. The votes from affirmative $\mathcal{RV}$ and $\mathcal{PV}$ are represented by 1.

**Table 2.** Communication overhead for different entities.

| Entity | | Preparation Stage | | Auction Stage | | Consensus Stage | |
|---|---|---|---|---|---|---|---|
| | | Overhead | Content | Overhead | Content | Overhead | Content |
| Buyer | | $\|Addr\| + \|G\| + 2\|\mathbb{Z}_p\| + \|data\|$ | $(Addr, data, \sigma)$ | $\|Addr\| + \|G\| + 3\|\mathbb{Z}_p\|$ | $(Addr, value_B, \sigma)$ | - | - |
| Bidders | | $\|Addr\| + 3\|G\| + 2\|\mathbb{Z}_p\|$ | $(Addr, BidComm, VolComm, \sigma)$ | $\|Addr\| + \|G\| + 4\|\mathbb{Z}_p\|$ | $(Addr, Enc(\widehat{bid}_i), Enc(vol_i), \sigma)$ | - | - |
| Witnesses | $\mathcal{P}$ | - | - | - | - | $\|\mathbb{Z}_p\|$ | $(vp)$ |
| | $\mathcal{RV}$ | - | - | - | - | $\|G\| + 3\|\mathbb{Z}_p\|$ | $(vote, \sigma)$ |
| | $\mathcal{PV}$ | - | - | - | - | $\|G\| + 3\|\mathbb{Z}_p\|$ | $(vote, \sigma)$ |

### *7.2. Trade-Off between Privacy and Social Efficiency*

In our evaluations, we used a dataset containing the real bids of electronic auctions from eBay online auctions [42] to model the auction settings. We adopted the statistics of auction ID 8214355679 to mould the reverse auction with 75 bidders. We also shuffled the bidding information to disturb the bids order, which simulates the bidding randomness and forms different bidder clusters. The shuffle process can give the performance of our scheme in the worst case. The design goal of our scheme is to provide a decentralized reverse auction with privacy preservation and incentive compatibility. One on hand, the differential privacy and clustering incentivize bidders to give their true valuation, while on the other hand, we lose some social efficiency as a result of privacy preservation. To evaluate this

property, we adopted the following two customized performance metrics: social welfare and revenue. Social welfare is the difference between the sum of winning bids and the cost:

$$welfare = \Sigma_{i \in W_S}(bid_i \times vol_i) - c \times |W_S| \tag{11}$$

Revenue is the difference between the payment and the cost:

$$revenue = P \times \Sigma_{i \in W_S}(vol_i) - c \times |W_S| \tag{12}$$

where $P$ is the payment per data unit for the winners, and $|W_S|$ is the number of bids higher than or equal to the valid price $vp$. The marginal cost $c$, i.e., the average increment in cost for allocating one additional user, and the supply volume $vol_i$ were set to be 1 for simplicity in the experiment. We used the ratio of the social welfare and revenue to the one without privacy preservation as an evaluation metric. We carried out the experiment on the same dataset hundreds of times to test the average performance.

Figure 4 shows the social welfare ratio and revenue ratio changing with different privacy parameters $\epsilon$, i.e., $\frac{1}{\lambda}$ and the number of bidders in a cluster $N$. Figure 4a,c shows the welfare ratio with different numbers of cluster members $N$. It can be observed that a bigger $N$ will reduce the welfare while providing better privacy. Similarly, a bigger $\epsilon$ provides an improved welfare efficiency and lowers the privacy level. We can see that with increasing $\epsilon$, the social welfare grows with fluctuation (Figure 4a). This is because the welfare is not only influenced by the differential parameter but also the bidder clustering. The randomness from the different bidder clustering results in the variation of the social welfare. Figure 4b,d gives the revenue ratio with different $N$. The revenue ratio keeps the same relationship with the parameters $\epsilon$ and $N$ as the welfare ratio. The difference is that our scheme achieves a better revenue than social welfare, and the revenue ratio is more than 1 in many cases. The reason is that our noised bid and cluster average bid give a much more appropriate valid price and payment.

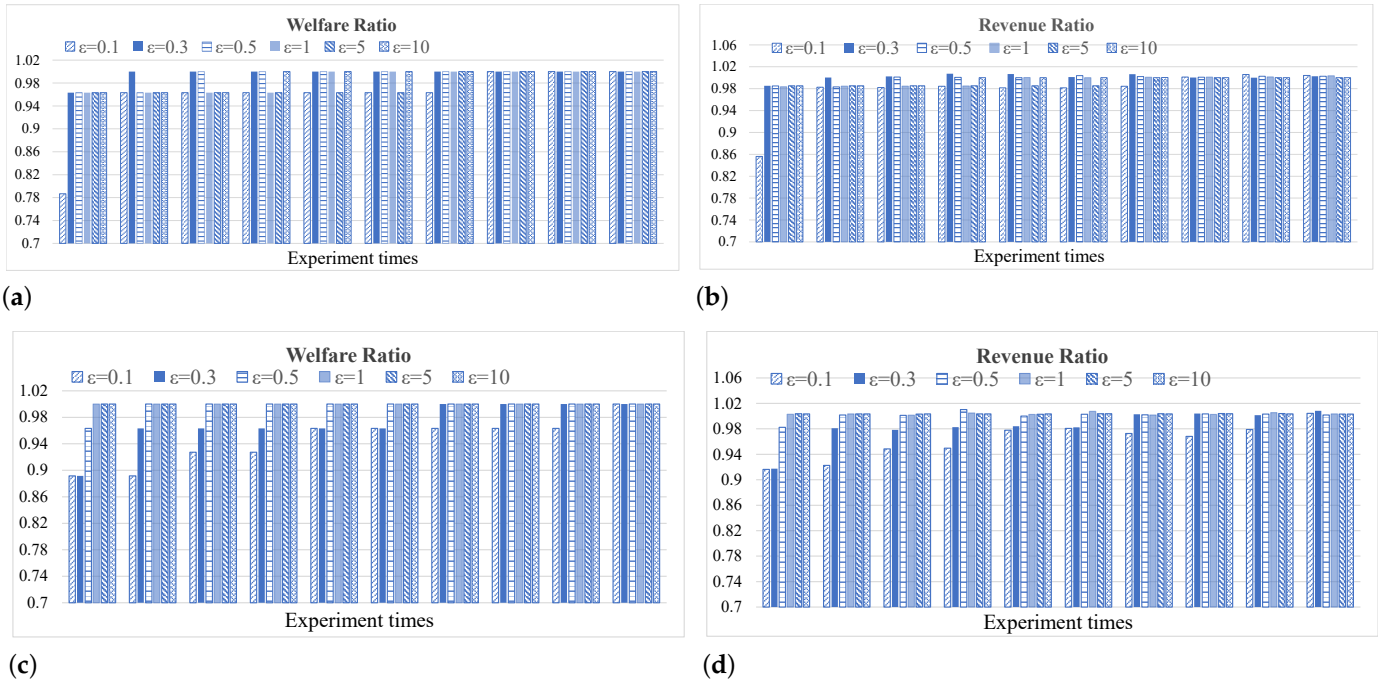



**Figure 4.** Welfare and revenue ratio. (**a**) Welfare ratio with $N = 5$. (**b**) Revenue ratio with $N = 5$. (**c**) Welfare ratio with $N = 3$. (**d**) Revenue ratio with $N = 3$.

Figure 5 illustrates the average welfare and revenue ratio with different $\epsilon$. It gives an obvious explanation how the welfare ratio and revenue ratio change with $\epsilon$. It can be

observed that the revenue ratio keeps the same tendency as the welfare ratio with a bit raise. In the majority of instances, the welfare ratio is above 0.95. In some cases, our social welfare with differential privacy is equal to the optimal ones without privacy, i.e., the ratio is 1. Different from the welfare ratio, our scheme enhances the revenue with a ratio of 1.000877474, 1.001214056, 1.002415409, 1.003091384 or 1.003669422 when $N = 3$. This is because the processes of noise addition and bidder clustering give a better valid price and a better payment as a result.


**Figure 5.** Average welfare and revenue ratio. (**a**) $N = 5$. (**b**) $N = 3$.

In Figure 6, we compare the proposed scheme to the optimal single price omniscient (OPT) [43] against the privacy parameter $\epsilon$. We can observe that our scheme gains high approximate welfare and revenue to OPT. Our revenue even exceeds OPTs when $\epsilon$ is bigger than 0.5. The reason is that differential privacy and clustering output a better valid price. Not surprisingly, the revenue is larger than the welfare for both OPT and our scheme because the payment is larger than the winner's bid.

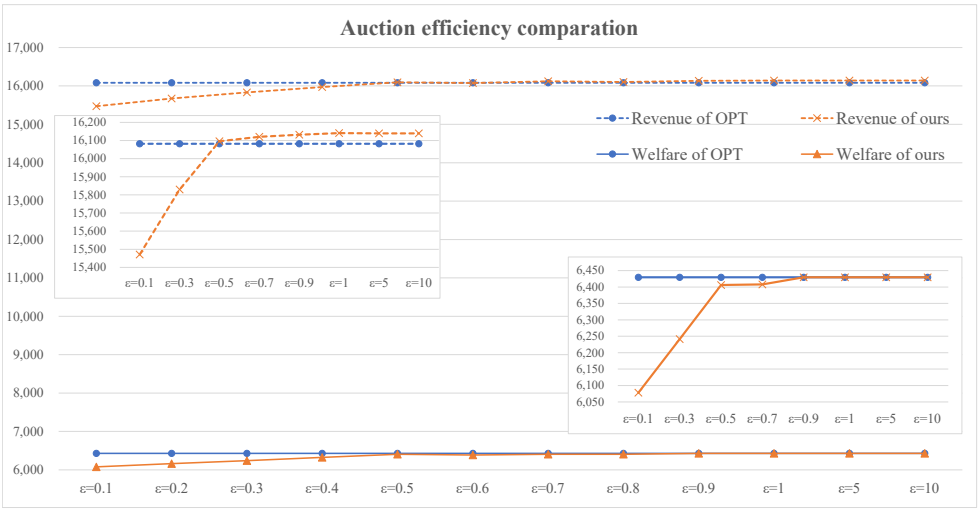
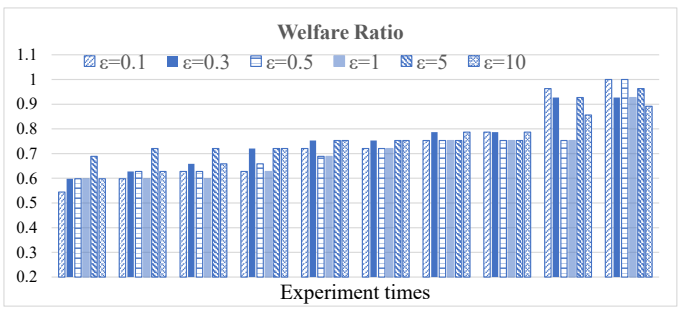


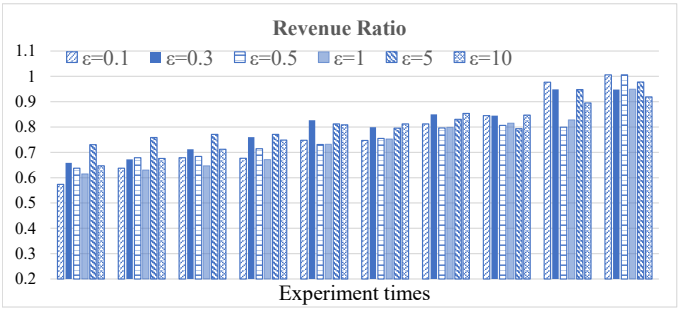

**Figure 6.** Auction efficiency comparison with $N = 3$.

Figure 7 clarifies the auction performance with bidder random bidding and clustering. It is obvious that the welfare and revenue ratio grows in a fluctuating manner as $\epsilon$ increases when $N = 3$ (Figure 7a,b). The shuffle results in a bigger fluctuation range. It can be observed that the lowest welfare and revenue ratio is about 0.6 in the worst case that the bidder cluster average bid results in a bad valid price. It is apparent that our scheme retains a better revenue than welfare in the worst case (Figure 7c). With varying $\epsilon$ and $N = 3$, the average revenue ratio is greater than 0.6, which reveals that our scheme gives a sensible valid price and payment. For the social welfare, the worst case is about 55% of

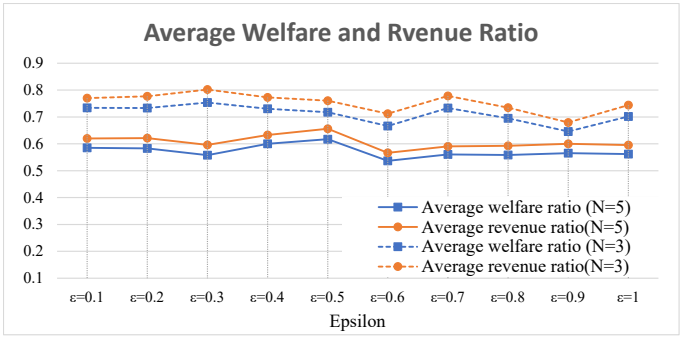
the optimized one, which is acceptable. The relationship between the social efficiency and the privacy parameter shown in all the above figures provides a visual illustration that our scheme achieves an acceptable trade-off between privacy and social efficiency.




(**a**)




(**b**)




(**c**)

**Figure 7.** Randomly shuffled bids. (**a**) Welfare ratio with $N = 3$. (**b**) Revenue ratio with $N = 3$. (**c**) Average ratios.

### *7.3. Deployment and Execution Cost*

We implemented the logic procedure of the proposed scheme based on Ethereum. Our contract was deployed on a locally simulated network Ganache. We used a laptop with 1.6 GHz Dual-Core Intel Core i5 CPU and 8 GB of memory. The smart contract was written in Solidity 0.4.17. The interactions between participants and the involved functions were realized using Truffle based on Javascript and Web3.js. Since a small number of transactions and blocks were generated during the auction, we did not consider the impact of the world state on the invocation of the smart contracts.

In our framework, the buyer publishes the auction contract, which is responsible for bidder committing and depositing in the preparation stage and vote counting during consensus agreement. Note that the auction allocation, result verification and proof verification are conducted off-chain to save on-chain computation cost.

In Table 3, we show the cost of setting up and executing the contract on the simulated Ethereum network. The cost is in the amount of gas consumed by each function, which can be converted to the monetary value in dollar according to the gas price. The gas cost is roughly related to the computational and storage complexity of the function. As we can see, the financial cost of using the smart contract on the Ethereum network is low. For example, the contract initialization (to store the contract on the blockchain) and the bidder registration (to cluster the bidders) cost more than the bidder committing (requiring commitment storage). For the auction smart contract, the total cost for the buyer is about 1.3 million gas and about 4 million gas for each bidder. Compared with the on-chain collusion-resistant scheme [41], our scheme can save on-chain cost and support auctions with many more bidders. Overall, our scheme improves the efficiency of data sharing and has certain scalability in aspect of both on-chain and off-chain costs.

**Table 3.** The gas costs for buyer, seller and witness using the smart contract.

| Entity | Function | | Transaction Cost | Execution Cost | Total Cost |
|---|---|---|---|---|---|
| Buyer | Publish contract | | 619,778 | 619,778 | 1,239,556 |
| | Start auction | | 46,998 | 46,998 | 93,996 |
| Bidder | Register | | 110,034 | 110,034 | 220,068 |
| | Commit | | 98,366 | 98,366 | 196,732 |
| Witness | $\mathcal{P}$ | Valid price | 43,544 | 43,544 | 87,088 |
| | $\mathcal{RV}$ | Start voting | 73,791 | 73,791 | 147,582 |
| | | Vote | 72,331 | 72,331 | 144,662 |
| | $\mathcal{PV}$ | Start voting | 53,891 | 53,891 | 107,782 |
| | | Vote | 72,331 | 72,331 | 144,662 |

## 8. Conclusions

To meet the need for symmetric data sharing and address the security and efficiency issues in the decentralized IoT system, an effective consensus-based distributed auction model is proposed. For privacy leakages derived from the transparency of the blockchain, distributed Laplacian perturbation and symmetric encryption are adopted to realize lightweight privacy preservation. To overcome the performance bottleneck of the blockchain, an optimized hybrid consensus algorithm is constructed. Bidders can reach consensus on the auction results with low computation and communication costs. The analyses show that the proposed scheme ensures the property of privacy preservation, incentive compatibility and collusion resistance with a bit loss of social efficiency. Regarding both on-chain and off-chain costs, it reduces the computation overhead and has a certain amount of scalability to support large-scale auctions.

**Author Contributions:** X.J. designed the distributed framework and realized and verified the proposed scheme. X.S. was responsible for the organizational structure and edited the manuscript. M.S. reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Key R&D Program of China (No. 2020YFB1005500), the Leading-edge Technology Program of Jiangsu Natural Science Foundation (No. BK20202001), and the Postdoctoral Science Foundation of Jiangsu Province (No. 2021K596C).

**Conflicts of Interest:** The authors declared no potential conflict of interest with respect to the research, authorship, or publication of this article.

## References

1. Liang, F.; Yu, W.; An, D.; Yang, Q.; Fu, X.; Zhao, W. A survey on big data market: Pricing, trading and protection. *IEEE Access* **2018**, *6*, 15132–15154. [CrossRef]
2. Byabazaire, J.; O'Hare, G.; Delaney, D. Data quality and trust: Review of challenges and opportunities for data sharing in iot. *Electronics* **2020**, *9*, 2083. [CrossRef]
3. An, D.; Yang, Q.; Yu, W.; Li, D.; Zhang, Y.; Zhao, W. Towards truthful auction for big data trading. In Proceedings of the 2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC), San Diego, CA, USA, 10–12 December 2017; pp. 1–7.
4. Cao, X.; Chen, Y.; Liu, K.R. Data trading with multiple owners, collectors, and users: An iterative auction mechanism. *IEEE Trans. Signal Inf. Process. Netw.* **2017**, *3*, 268–281. [CrossRef]
5. Gao, W.; Yu, W.; Liang, F.; Hatcher, W.G.; Lu, C. Privacy-preserving auction for big data trading using homomorphic encryption. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 776–791. [CrossRef]
6. Feng, Z.; Chen, J.; Liu, T. An online truthful auction for iot data trading with dynamic data owners. In Proceedings of the International Conference on Collaborative Computing: Networking, Applications and Worksharing, Virtual Event, 16–18 October 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 554–571.
7. Hou, L.; Zhao, S.; Xiong, X.; Zheng, K.; Chatzimisios, P.; Hossain, M.S.; Xiang, W. Internet of things cloud: Architecture and implementation. *IEEE Commun. Mag.* **2016**, *54*, 32–39. [CrossRef]

8. Manzoor, A.; Braeken, A.; Kanhere, S.S.; Ylianttila, M.; Liyanage, M. Proxy re-encryption enabled secure and anonymous iot data sharing platform based on blockchain. *J. Netw. Comput. Appl.* **2021**, *176*, 1–17. [CrossRef]
9. Jiang, T.; Fang, H.; Wang, H. Blockchain-based internet of vehicles: Distributed network architecture and performance analysis. *IEEE Internet Things J.* **2018**, *6*, 4640–4649. [CrossRef]
10. Kang, J.; Yu, R.; Huang, X.; Wu, M.; Maharjan, S.; Xie, S.; Zhang, Y. Blockchain for secure and efficient data sharing in vehicular edge computing and networks. *IEEE Internet Things J.* **2018**, *6*, 4660–4670. [CrossRef]
11. Chen, C.; Wu, J.; Lin, H.; Chen, W.; Zheng, Z. A secure and efficient blockchain-based data trading approach for internet of vehicles. *IEEE Trans. Veh. Technol.* **2019**, *68*, 9110–9121. [CrossRef]
12. Guo, J.; Ding, X.; Wang, T.; Jia, W. Combinatorial resources auction in decentralized edge-thing systems using blockchain and differential privacy. *Inf. Sci.* **2022**, *607*, 211–229. [CrossRef]
13. Guo, J.; Ding, X.; Wu, W. A double auction for charging scheduling among vehicles using dag-blockchains. *arXiv* **2020**, arXiv:2010.01436.
14. Kosba, A.; Miller, A.; Shi, E.; Wen, Z.; Papamanthou, C. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 839–858.
15. Su, Z.; Wang, Y.; Xu, Q.; Zhang, N. Lvbs: Lightweight vehicular blockchain for secure data sharing in disaster rescue. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 19–32. [CrossRef]
16. Wang, Y.; Su, Z.; Zhang, N. Bsis: Blockchain-based secure incentive scheme for energy delivery in vehicular energy network. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3620–3631. [CrossRef]
17. Cao, X.; Chen, Y.; Liu, K.R. An iterative auction mechanism for data trading. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 5850–5854.
18. Susanto, H.; Zhang, H.; Ho, S.-Y.; Liu, B. Effective mobile data trading in secondary ad-hoc market with heterogeneous and dynamic environment. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 645–655.
19. Jiao, Y.; Wang, P.; Niyato, D.; Alsheikh, M.A.; Feng, S. Profit maximization auction and data management in big data markets. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6.
20. Huang, Q.; Gui, Y.; Wu, F.; Chen, G.; Zhang, Q. A general privacy-preserving auction mechanism for secondary spectrum markets. *IEEE/ACM Trans. Netw.* **2015**, *24*, 1881–1893. [CrossRef]
21. Wang, Q.; Huang, J.; Chen, Y.; Tian, X.; Zhang, Q. Privacy-preserving and truthful double auction for heterogeneous spectrum. *IEEE/ACM Trans. Netw.* **2019**, *27*, 848–861. [CrossRef]
22. Cheng, K.; Wang, L.; Shen, Y.; Liu, Y.; Wang, Y.; Zheng, L. A lightweight auction framework for spectrum allocation with strong security guarantees. In Proceedings of the IEEE Conference on Computer Communications (INFOCOM), Toronto, ON, Canada, 6–9 July 2020; pp. 1708–1717.
23. Wang, Z.; Li, J.; Hu, J.; Ren, J.; Li, Z.; Li, Y. Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 2053–2061.
24. Cheng, K.; Tong, W.; Zheng, L.; Fu, J.; Mu, X.; Shen, Y. A secure and fair double auction framework for cloud virtual machines. *IEEE Access* **2021**, *9*, 87982–87994. [CrossRef]
25. Hassan, M.U.; Rehmani, M.H.; Chen, J. Deal: Differentially private auction for blockchain-based microgrids energy trading. *IEEE Trans. Serv. Comput.* **2019**, *13*, 263–275. [CrossRef]
26. Brakerski, Z.; Vaikuntanathan, V. Efficient fully homomorphic encryption from (standard) lwe. *SIAM J. Comput.* **2014**, *43*, 831–871. [CrossRef]
27. Li, M.; Weng, J.; Yang, A.; Lu, W.; Zhang, Y.; Hou, L.; Liu, J.-N.; Xiang, Y.; Deng, R.H. Crowdbc: A blockchain-based decentralized framework for crowdsourcing. *IEEE Trans. Parallel Distrib. Syst.* **2018**, *30*, 1251–1266. [CrossRef]
28. Feng, J.; Liu, N.; Cao, J.; Zhang, Y.; Lu, G. Securing traffic-related messages exchange against inside-and-outside collusive attack in vehicular networks. *IEEE Internet Things J.* **2019**, *6*, 9979–9992. [CrossRef]
29. Xiong, W.; Xiong, L. Anti-collusion data auction mechanism based on smart contract. *Inf. Sci.* **2021**, *555*, 386–409. [CrossRef]
30. Li, M.; Weng, J.; Yang, A.; Liu, J.-N.; Lin, X. Toward blockchain-based fair and anonymous ad dissemination in vehicular networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 11248–11259. [CrossRef]
31. Dai, W.; Dai, C.; Choo, K.-K.R.; Cui, C.; Zou, D.; Jin, H. Sdte: A secure blockchain-based data trading ecosystem. *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 725–737. [CrossRef]
32. Li, C.; Liang, S.; Zhang, J.; Wang, Q.-E.; Luo, Y. Blockchain-based data trading in edge-cloud computing environment. *Inf. Process. Manag.* **2022**, *59*, 102786. [CrossRef]
33. Fujisaki, E.; Suzuki, K. Traceable ring signature. In Proceedings of the International Workshop on Public Key Cryptography, Beijing, China, 16–20 April 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 181–200.
34. Ács, G.; Castelluccia, C. I have a dream!(differentially private smart metering). In Proceedings of the International Workshop on Information Hiding, Prague, Czech Republic, 18–20 May 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 118–132.

35. Pedersen, T.P. Non-interactive and information-theoretic secure verifiable secret sharing. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 11–15 August 1991; Springer: Berlin/Heidelberg, Germany, 1991; pp. 129–140.
36. Bünz, B.; Bootle, J.; Boneh, D.; Poelstra, A.; Wuille, P.; Maxwell, G. Bulletproofs: Short proofs for confidential transactions and more. In Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–24 May 2018; pp. 315–334.
37. Ganesh, C.; Orlandi, C.; Tschudi, D. Proof-of-stake protocols for privacy-aware blockchains. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, 19–23 May 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 690–719.
38. Blass, E.O.; Kerschbaum, F. Strain: A secure auction for blockchains. In Proceedings of the European Symposium on Research in Computer Security, Barcelona, Spain, 3–7 September 2018; pp. 87–110.
39. Kotz, S.; Kozubowski, T.; Podgórski, K. *The Laplace Distribution and Generalizations: A Revisit with Applications to Communications, Economics, Engineering, and Finance*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2001.
40. McSherry, F.; Talwar, K. Mechanism design via differential privacy. In Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07), Providence, RI, USA, 21–23 October 2007; pp. 94–103.
41. Wu, S.; Chen, Y.; Wang, Q.; Li, M.; Wang, C.; Luo, X. Cream: A smart contract enabled collusion-resistant e-auction. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1687–1701. [CrossRef]
42. Kaggle, Online Auctions Dataset: Modeling Online Auctions Dataset from Ebay. 2010. Available online: https://www.kaggle.com/datasets/onlineauctions/online-auctions-dataset (accessed on 1 July 2022).
43. Goldberg, A. Competitiveness via consensus. In Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, Baltimore, MD, USA, 12–14 January 2003; pp. 215–222.