Boreas: Fully Anonymous Sealed-Bid Auction

Erjun Zhou, Jing Chen, *Senior Member, IEEE*, Min Shi, Zhengdi Huang, Meng Jia, Kun He, *Member, IEEE*, and Ruiying Du

***Abstract*—With the rise of e-commerce, sealed-bid auctions are widely used in various online scenarios. In auctions, bidders' bids and participants' identities are considered critical private information. However, existing works either only achieve bid privacy or fail to provide complete protection of identity. In this work, we propose the first sealed-bid auction scheme that achieves both bid privacy and identity privacy, called *Boreas*. We propose three fundamental protocols as the building blocks. In particular, *anonymous submission* enables sellers to submit items anonymously, *oblivious bidding* and *locker transaction* enable the seller and the winner to confirm the auction results and complete the transaction without knowing each other's identity. Meanwhile, we formally define the security goal of identity privacy and formalize a new security property called: *fully anonymous*. We prove the security of our scheme in the semi-honest adversary model. We implement Boreas and run experiments comparing its performance against existing schemes. Our experiments show that Boreas improves computation time by 12.6% and reduces communication costs by $10^3\times$ in handling a large-scale auction, while offering stronger security guarantee.**

***Index Terms*—Sealed-bid auction, Tor, ring signature, private information retrieval, oblivious bidding.**

I. Introduction

AUCTION has been a method of trading items or services since ancient Babylon [1]. A widely used type of auction is the sealed-bid auction. In this type, all bidders independently submit their secret bid in a sealed envelope so that no bidder knows the bid of any other bidder. Once all bids are submitted, the auctioneer opens their envelopes and chooses the highest bidder as the winner. Because of its simplicity and efficiency, sealed-bid auctions are widely used in online scenario, such as advertising [2], [3], blockchain [4], [5], and cloud computing [6].

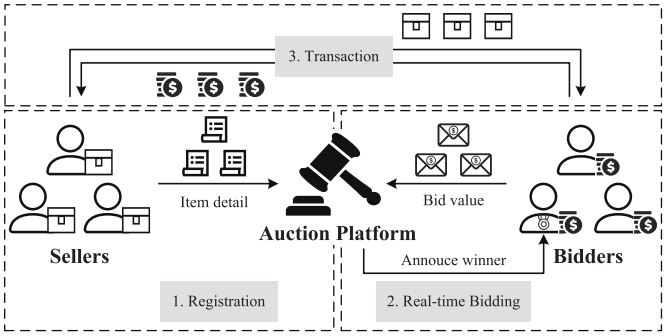

Fig. 1. Modern online sealed-bid auction process. There are three roles involved in the process: the sellers who own items, the bidders who provide bids, and the auction platform that determines the winner. Real-time bidding is the core phase of an auction, where sellers want to know the highest bid and bidders want to know whether they win.

Fig. 1 depicts the modern online sealed-bid auction process, which typically consists of three phases: registration, bidding, and transaction. During the registration phase, sellers register on the auction platform and submit their auction items. During the bidding phase, each bidder submits a bid for the item. The auctioneer determines the winner and announces the auction results including the winner's identity and the sale price, which may be either the highest bid or the second-highest bid (usually called Vickrey auction) [7]. This phase takes place in real-time, with both sellers and bidders expecting to confirm the auction results promptly. During the transaction phase, the seller and the winner complete the exchange of the item and payment based on the results. However, all participants face the risk of sensitive information disclosure in this process. For instance, an honest-but-curious auctioneer may attempt to infer bidders' purchase desire from their bids and identity [8], [9].

To protect the bid privacy in real-time bidding phase, some privacy-preserving auction schemes [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20] use cryptographic primitives to compute the auction result, such as secret sharing, fully homomorphic encryption, and etc. Other works [4], [21], [22], [23] introduce a trusted third party other than the auctioneer to determine the result with its help. However, these schemes

Received 14 October 2024; revised 10 April 2025; accepted 24 July 2025. Date of publication 28 July 2025; date of current version 18 August 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3103300, in part by the National Natural Science Foundation of China under Grant 62172303 and Grant 62472323, in part by the Key Research and Development Program of Hubei Province under Grant 2022BAA039, in part by the Key Research and Development Program of Shandong Province under Grant 2022CXPT055, in part by Wuhan Knowledge Innovation Project under Grant 2023010201010062, and in part by the Wuhan Scientific and Technical Key Project under Grant 2023010302020707. The associate editor coordinating the review of this article and approving it for publication was Dr. Shancang Li. *(Corresponding author: Jing Chen.)*

Erjun Zhou, Min Shi, Zhengdi Huang, Meng Jia, and Kun He are with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: sanfeng@whu.edu.cn; itachi@whu.edu.cn; rechardhuang@whu.edu.cn; jiameng@whu.edu.cn; hekun@whu.edu.cn).

Jing Chen is with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and also with the Rizhao Institute of Information Technology, Wuhan University, Rizhao 276800, China (e-mail: chenjing@whu.edu.cn).

Ruiying Du is with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and also with the Collaborative Innovation Center of Geospatial Technology, Wuhan 430079, China (e-mail: duraying@whu.edu.cn).

Digital Object Identifier 10.1109/TIFS.2025.3593063

fail to address identity privacy. The seller of the item and the winner of the auction are transparent to all participants. The identity leakage becomes a privacy issue in applications where the identity of participant is regarded as key information, such as medical auctions and online advertising [24], [25]. To achieve the identity privacy of bidders, recent works [26], [27], [28] utilized the ring signature on the blockchain to hide bidder's interest. Their schemes only protect the identities of the losing bidders. Chang and Chang [29], [30] proposed an enhanced anonymous auction with freewheeling bids. In their scheme, the anonymity of bidders relies on a certification authority. In MaskAuct [31] researchers proposed a bidder anonymity auction scheme that supports a blocklist mechanism. These schemes do not hide the relationship between the seller and the item. To achieve the identity privacy of sellers, the sealed-bid auction schemes [32], [33], [34], [35], [36] without an auctioneer allows the seller to locally verify the auction results. However, these schemes require bidders to perform extensive computation, and the winner's identity still be leaked. Zhong et al. [2], [3] introduced the oblivious bidding protocol that allows sellers to privately obtain bidders' secret bids. The bidders and sellers do not know each other's identities, but the auctioneer in the middle knows the identities of both parties. In addition, the above works only considers the bidding phase, and does not consider the identity protection of the submission and transaction in the complete process. In general, existing works either only focus on the bid privacy or fail to provide complete protection of identity. There is no formal definition for identity privacy in auction scheme. This lack hampers the development of robust anonymous sealed-bid auction scheme and makes it difficult to design systems that fully protect identity under security proofs.

It is a non-trivial task to achieve identity privacy in complete online sealed-bid auction process. On one hand, both sellers and bidders need to submit their items and bids anonymously. On the other hand, the seller and the winner need to complete the exchange of items and money while hiding their identities. In this work, we propose three fundamental protocols as the building blocks to construct the first sealed-bid auction scheme that achieves both bid and identity privacy, called *Boreas*. We formalize a new security property in the auction framework, called *fully anonymous*, which ensures that the identities of all sellers and bidders remain private throughout the auction process. Informally, only the seller knows who owns the item, and only the winner knows who wins the item.

Contributions: In summary, we make the following contributions in this work:

- We propose a sealed-bid auction framework that considers privacy of both bid and identity. We formalize a new security property for identity privacy called: *fully anonymous*, and we present a sealed-bid auction scheme *Boreas*, which achieves both fully anonymous and bid privacy.
- We propose three fundamental protocols, *anonymous submission*, *oblivious bidding*, and *locker transaction* to build our fully anonymous sealed-bid auction scheme. Each protocol achieves the protection of identity privacy in one phase.
- We formally prove the security of *Boreas* in the semi-honest adversary model using the hybrid argument technique. Our experiments show that compared to state-of-the-art schemes, *Boreas* reduces computation time by 12.6% and communication cost by $10^3\times$, while also providing stronger security.

Technical Overview: We propose three fundamental protocols as the building blocks in Boreas. Importantly, these protocols have broader applications beyond auction systems, with potential applications in a wide range of privacy-preserving systems. Our key technical contributions are detailed as follows:

- **Anonymous submission**. Combining *onion routing* and *ring signature* enables sellers and bidders to anonymously submit messages to the auctioneer, hiding their identities.
- **Oblivious bidding**. Using *pseudorandom permutation* implements a bidding protocol that allows bidders to determine their winning items without knowing the owner, and allows sellers to confirm the sale price without knowing the identity of the winner.
- **Locker transaction**. Utilizing *private information retrieval* implements anonymous exchange between multiple pairs of users, allowing the seller and the winner to trade item and money without revealing their identities.

II. Problem Statement

We introduce the system model, threat model, and design goals of our sealed-bid auction framework. In the design goals, we formalize a new security property called *fully anonymous*.

A. System Model

Our sealed-bid auction framework consists of three roles: N sellers $\{S_i\}_{i\in[N]}$, M bidders $\{B_j\}_{j\in[M]}$, and an auctioneer that is constituted by two non-colluding auction servers A_1 and A_2. For simplicity, we assume that each seller has one item c which includes the item detail cstr and the item key csk. All string cstr have the same bit length. Each bidder has a wallet key wsk for payment. As shown in Fig. 1, the framework consists of three phases: registration, bidding, and transaction. The notations in our auction scheme are shown in Table I.

Definition 1 (Sealed-bid auction): A sealed-bid auction scheme $\Pi_{\mathtt{SBA}}$ consists of three PPT protocols defined as follows.

- `Registration`. This protocol is executed by the seller and the auctioneer. The seller inputs the item detail cstr and the auctioneer inputs nothing. The auctioneer outputs the item detail cstr and the seller outputs nothing.
- `Bidding`. This protocol is executed by the seller, all bidders, and the auctioneer. Each bidder inputs a bid, while the seller and the auctioneer input nothing. The seller and the winner output the auction result, and the auctioneer outputs nothing.
- `Transaction`. This protocol is executed by the seller and the winner. The seller inputs the item key csk and the winner inputs the wallet key wsk. The seller outputs the wallet key wsk and the winner outputs the item key csk.

TABLE I
NOTATIONS

Notation	Description
S, B, A_0, A_1	Denote seller, bidder and two auction servers.
$\mathsf{Apk}, \mathsf{Ask}$	Auction server's public and private keys.
$\mathsf{rvk}, \mathsf{rsk}, \mathsf{mpk}, \mathsf{msk}$	Seller's keys used in the anonymous submission.
$\mathbf{V}, \mathbf{P}$	Verification key list and public key list.
$\mathsf{Spk}, \mathsf{Ssk}$	Seller's public and private keys.
$c = (\mathsf{cstr}, \mathsf{csk})$	Item contains item detail and seller's item key.
$\mathsf{Bpk}, \mathsf{Bsk}, \mathsf{wsk}$	Bidder's public key, private key, and wallet key.
$\mathsf{DB_m}, \mathsf{DB_c}, \mathsf{DB_w}$	The database used to store item details, the ciphertexts of item keys and wallet keys respectively.
$cadd_{j,k}, wadd_{j,k}$	The two indexes of bidder B_j to item c_k indicate the location of item key csk in $\mathsf{DB_c}$ and the location of wallet key wsk in $\mathsf{DB_w}$.
$ncadd_{j,k}, nwadd_{j,k}$	The new indexes corresponding to the original index $cadd_{j,k}$ and $wadd_{j,k}$ in the compressed database.
$b_{j,k}$	The bid of bidder B_j for item c_k.
$\mathbf{b}_k$	Bid vector for item c_k contains all bidder's bid.

B. Threat Model and Design Goals

We assume that our auction framework is in the semi-honest adversary model that all participants execute the auction protocol faithfully while they attempt to infer private information about bids and participants' identities. Unlike existing auction schemes [2], [3], [4], [35] that only focus on privacy protection in the real-time bidding phase, our design goal is to achieve both bid privacy and identity privacy during the whole process.

Informally, bid privacy consists of two aspects: (1) *only the losing bidder knows his/her losing bid*; (2) *only the seller and the winner know the highest bid*. We give the formal definition of bid privacy in the sealed-bid auction framework below.

Definition 2 (Bid Privacy): A sealed-bid auction framework satisfies bid privacy, if for all $\lambda \in \mathbb{N}$, all $M, N = \mathsf{poly}(\lambda)$ and every PPT adversary $\mathcal{A}$ it holds that $\mathcal{A}$ has at most negligible advantage in the following experiment.

$\mathsf{Exp}_{\mathsf{BP}}(\mathcal{A})$:

1) $\mathcal{A}$ samples two bid vectors $\mathbf{b}_0$, $\mathbf{b}_1$ uniformly at random from $\mathbb{Z}_n^M$, subject to the role-specific conditions:
   - If $\mathcal{A}$ acts as a seller, we need $|\mathbf{b}_0|_\infty = |\mathbf{b}_1|_\infty$.
   - If $\mathcal{A}$ acts as a bidder S_j, we need $\mathbf{b}_0[j] = \mathbf{b}_1[j]$.
   - Otherwise, $\mathbf{b}_0$ and $\mathbf{b}_1$ have no additional constraints.
2) The experiment chooses a random bit $t \in \{0, 1\}$. Given an item c, the seller, bidders, and auctioneer execute the protocol `Bidding` where the j-th bidder uses bid $\mathbf{b}_t[j]$.
3) $\mathcal{A}$ outputs a guess t'. If $t' = t$, the experiment outputs 1, otherwise it outputs 0.

The advantage of $\mathcal{A}$ is $\mathsf{Adv}_{\mathsf{BP}}(\mathcal{A}) = |\Pr[\mathsf{Exp}_{\mathsf{BP}}(\mathcal{A}) = 1] - \frac{1}{2}|$.

Informally, identity privacy includes two aspects: (1) *sellers don't know who wins the item*; (2) *bidders don't know who is the owner of the item*. Notably, the losing bidder's privacy is implicit in bid privacy. To better analyze the identity privacy of sellers and bidders, we formalize a new security definition in the sealed-bid auction called *fully anonymous*.

Definition 3 (Fully Anonymous): A sealed-bid auction framework satisfies fully anonymous, if for all $\lambda \in \mathbb{N}$, all $M, N = \mathsf{poly}(\lambda)$ and every PPT adversary $\mathcal{A}$ it holds that $\mathcal{A}$ has at most negligible advantage in the following experiment.

$\mathsf{Exp}_{\mathsf{FA}}(\mathcal{A})$:

1) For all $i = 1, \ldots, N$, the seller S_i executes the protocol `Registration` with the auctioneer. Provides item details $(\mathsf{cstr}_1, \ldots, \mathsf{cstr}_N)$ to $\mathcal{A}$.
2) $\mathcal{A}$ chooses an item detail cstr_i. Sellers, bidders, and the auctioneer execute the protocol `Bidding` for this item. We assume that the bidder w_i is the winner. This step can be repeated multiple times.
3) For the auctioned items, sellers and winners execute the protocol `Transaction`.
4) $\mathcal{A}$ outputs a guess $(\mathsf{cstr}_k, b_1, b_2)$. If $b_1 = k$ or $b_2 = w_k$, the experiment outputs 1, otherwise it outputs 0.

The advantage of $\mathcal{A}$ depends on its role:

- Seller: $\mathsf{Adv}_{\mathsf{FA}}(\mathcal{A}) = |\Pr[\mathsf{Exp}_{\mathsf{FA}}(\mathcal{A}) = 1] - \frac{N+M-2}{(N-1)\cdot M}|$
- Bidder: $\mathsf{Adv}_{\mathsf{FA}}(\mathcal{A}) = |\Pr[\mathsf{Exp}_{\mathsf{FA}}(\mathcal{A}) = 1] - \frac{N+M-2}{N\cdot(M-1)}|$
- Others: $\mathsf{Adv}_{\mathsf{FA}}(\mathcal{A}) = |\Pr[\mathsf{Exp}_{\mathsf{FA}}(\mathcal{A}) = 1] - \frac{N+M-1}{N\cdot M}|$

The fully anonymous can be viewed as the identity privacy of sellers and bidders. If we only consider the seller's identity or bidder's identity, we can define the following experiments, which are the same as $\mathsf{Exp}_{\mathsf{FA}}(\mathcal{A})$ except for the output:

- $\mathsf{Exp}_{\mathsf{SI}}(\mathcal{A})$. Outputs (cstr_k, b_1) at step 4. The advantage of $\mathcal{A}$ is $\mathsf{Adv}_{\mathsf{SI}}(\mathcal{A}) = |\Pr[\mathsf{Exp}_{\mathsf{SI}}(\mathcal{A}) = 1] - \frac{1}{N}|$.
- $\mathsf{Exp}_{\mathsf{BI}}(\mathcal{A})$. Outputs (cstr_k, b_2) at step 4. The advantage of $\mathcal{A}$ is $\mathsf{Adv}_{\mathsf{BI}}(\mathcal{A}) = |\Pr[\mathsf{Exp}_{\mathsf{BI}}(\mathcal{A}) = 1] - \frac{1}{M}|$.

Remark. In our defined experiments, we divide the adversary $\mathcal{A}$ into three categories: sellers, bidders, and others. The purpose of this classification distinguishes the initial advantages of $\mathcal{A}$ by its category. For instance, we assume that $\mathcal{A}$ acts as the seller S_i in $\mathsf{Exp}_{\mathsf{FA}}(\mathcal{A})$. The seller's initial advantage is that he/she knows the owner of his/her item. Therefore, $\mathcal{A}$ can always output a correct guess $(\mathsf{cstr}_i, i, \perp)$ in the experiment, and its advantage is not negligible. We need to remove this advantage when $\mathcal{A}$ plays the role of the seller. Notably, we classify the auction server as others because it does not have any initial advantage.

III. PRELIMINARIES

We describe the cryptographic primitives used in Boreas.

A. Ring Signature

Ring signatures [37], [38] allow a signer to sign messages while hiding his/her identity within a group of users, called a ring. Unlike group signatures, there is no tracing authority to de-anonymize signatures in ring signatures.

Definition 4 (Ring Signature): A ring signature scheme Π_{RS} consists of a triple of PPT algorithms defined as follows.

- $\mathsf{KeyGen}(1^\lambda) \to (\mathrm{VK}, \mathrm{SK})$. a randomized algorithm that takes in a security parameter λ, and outputs a pair of verification and signing keys $(\mathrm{VK}, \mathrm{SK})$.
- $\mathsf{Sign}(\mathrm{SK}, m, \mathsf{R}) \to \Sigma$. a randomized algorithm that takes in a signing key SK, a message m and a list of verification keys $\mathsf{R} = (\mathrm{VK}_1, \ldots, \mathrm{VK}_\ell)$, and outputs a signature Σ.
- $\mathsf{Verify}(\mathsf{R}, m, \Sigma) \to 0/1$. a deterministic algorithm that takes in a ring $\mathsf{R} = (\mathrm{VK}_1, \ldots, \mathrm{VK}_\ell)$, a message m and a signature Σ, and outputs either 0 or 1.

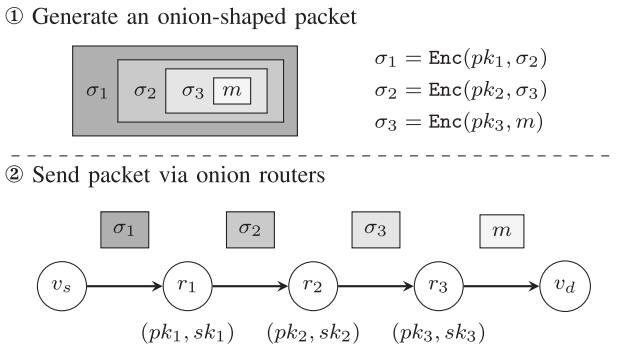

Fig. 2. An example of onion routing.

A ring signature scheme Π_{RS} should satisfy the following properties:

Correctness. A ring signature scheme RS satisfies *correctness*, if for all $\lambda \in \mathbb{N}$, all $\ell = \mathsf{poly}(\lambda)$, all $i \in [\ell]$ and all messages

m that if for $(\mathrm{VK}_i, \mathrm{SK}_i) \leftarrow \mathsf{RS.KeyGen}(1^\lambda)$ and $\Sigma \leftarrow \mathsf{RS.Sign}(\mathrm{SK}_i, m, \mathsf{R})$, where $\mathsf{R} = (\mathrm{VK}_1, \ldots, \mathrm{VK}_\ell)$, such that

$$\Pr[\mathsf{RS.Verify}(\mathsf{R}, m, \Sigma) = 1] \geq 1 - \mathsf{negl}(\lambda)$$

Anonymity. A ring signature scheme RS satisfies *anonymity* if for all $\lambda \in \mathbb{N}$ and all PPT adversary $\mathcal{A}$, such that

$$|\Pr[(\mathsf{R}, m, i_0, i_1) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\cdot)}; b \xleftarrow{\mathrm{r}} \{0,1\}; \Sigma^* \leftarrow \mathsf{RS.Sign}(\mathrm{SK}_{i_b}, m, \mathsf{R}) : \mathcal{A}(\Sigma^*) = b] - \frac{1}{2}| \leq \mathsf{negl}(\lambda)$$

where $\mathrm{VK}_{i_0}, \mathrm{VK}_{i_1} \in \mathsf{R}$ and $\mathsf{KeyGen}(\cdot)$ is an oracle that returns a key pair $(\mathrm{VK}, \mathrm{SK}) \leftarrow \mathsf{RS.KeyGen}(1^\lambda)$ at each query.

Unforgeability. A ring signature scheme RS satisfies *unforgeability*, if for all $\lambda \in \mathbb{N}$ and all PPT adversary $\mathcal{A}$, such that

$$\Pr[(\mathsf{R}, m, \Sigma) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\cdot), \mathsf{Sign}(\cdot), \mathsf{Corrupt}(\cdot)} : \mathsf{RS.Verify}(\mathsf{R}, m, \Sigma) = 1] \leq \mathsf{negl}(\lambda)$$

where $\mathsf{KeyGen}(\cdot)$ is an oracle that generates $(\mathrm{VK}_j, \mathrm{SK}_j) \leftarrow \mathsf{RS.KeyGen}(1^\lambda)$ and returns VK_j; $\mathsf{Sign}(i, m, \mathsf{R})$ is an oracle that returns $\Sigma \leftarrow \mathsf{RS.Sign}(\mathrm{SK}_i, m, \mathsf{R})$ if $(\mathrm{VK}_i, \mathrm{SK}_i)$ was output by $\mathsf{KeyGen}(\cdot)$ and $\perp$ otherwise; $\mathsf{Corrupt}(i)$ is an oracle that returns SK_i if $(\mathrm{VK}_i, \mathrm{SK}_i)$ was output by $\mathsf{KeyGen}(\cdot)$ and $\perp$ otherwise. $\mathcal{A}$ is restricted to output a triple (R, m, Σ) such that: (1) No query of the form $(*, \mathsf{R}, m)$ has been made to $\mathsf{Sign}(\cdot, \cdot, \cdot)$; (2) R only contains public keys VK_i produced by $\mathsf{KeyGen}(\cdot)$ and for which i was never queried to $\mathsf{Corrupt}(\cdot)$.

B. The Onion Routing (Tor)

Onion routing [39], [40] is a privacy-preserving technique for network communications that hides the sender's identity. In a network, the sender is called the source node and the receiver is called the destination node.

The onion routing workflow is shown in Fig. 2. Assume that the source node v_s wants to send a message to the destination node v_d. v_s first encrypts the message m using the public keys of onion routers (r_1, r_2, and r_3) to generate an onion-shaped packet and then sends it to r_1. Only r_1 can decrypt the first layer using the private key corresponding to pk_1 and identify the next onion router r_2. Similarly, r_2 and r_3 can decrypt the corresponding layer. Finally, the last onion router r_3 sends the message to v_d. In this way, each node only knows the previous and next hop of the message. The complete sending path and the sender's identity are hidden.

C. Private Information Retrieval

Private information retrieval (PIR) [41] is a fundamental privacy-preserving cryptographic primitive that allows the client to retrieve a data object from a database server without revealing the index of the obtained object to the server.

Definition 5 (Private Information Retrieval): A private information retrieval scheme Π_{PIR} is a tuple of PPT algorithms defined over a database DB of n data objects as follows, where $\mathsf{DB}[k]$ denotes the k-th data object of the database.

- $\mathsf{Setup}(1^\lambda) \rightarrow \mathsf{pp}$. a randomized algorithm that takes in a security parameter λ, and outputs public parameters pp, which is implicitly provided as input in all algorithms.
- $\mathsf{Query}(i) \rightarrow q_i$. a randomized algorithm that takes in an index $i \in [n]$, and outputs a query q_i.
- $\mathsf{Response}(q_i, \mathsf{DB}) \rightarrow r_i$. a randomized algorithm that takes in a query q_i and a database DB, and outputs a response r_i.
- $\mathsf{Recover}(r_i) \rightarrow d_i$. a deterministic algorithm that takes in a response r_i, and outputs a answer d_i.

A private information retrieval scheme Π_{PIR} should satisfy the following properties:

Correctness. A private information retrieval scheme Π_{PIR} satisfies *correctness*, if for all $\lambda \in \mathbb{N}$, all database DB, all database size $n \in \mathsf{poly}(\lambda)$, and all index $i \in [n]$, such that

$$\Pr\left[d_i = \mathrm{DB}[i] : \begin{array}{c} \mathrm{pp} \leftarrow \mathrm{PIR.Setup}\left(1^\lambda\right) \\ q_i \leftarrow \mathrm{PIR.Query}(i) \\ r_i \leftarrow \mathrm{PIR.Response}\left(q_i, \mathrm{DB}\right) \\ d_i \leftarrow \mathrm{PIR.Recover}\left(r_i\right) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

Privacy. For all $\lambda \in \mathbb{N}$, all database size $n \in \mathsf{poly}(\lambda)$, and all index $i \in [n]$, define the distribution

$$\mathcal{P}(i) := \left\{ q_i : \begin{array}{c} \mathrm{pp} \leftarrow \mathrm{PIR.Setup}\left(1^\lambda\right) \\ q_i \leftarrow \mathrm{PIR.Query}(i) \end{array} \right\}$$

A private information retrieval scheme Π_{PIR} satisfies *privacy* if for all PPT adversaries $\mathcal{A}$, such that

$$\max_{i,j \in [n]} \{\Pr[\mathcal{A}(\mathcal{P}(i)) = 1] - \Pr[\mathcal{A}(\mathcal{P}(j)) = 1]\} \leq \mathsf{negl}(\lambda)$$

IV. Protocols

We propose three fundamental protocols as the building blocks of our sealed-bid auction scheme. Each protocol can achieve the protection of identity privacy in one phase.

A. Anonymous Submission

During the registration phase, sellers register on the auction server and provide their item details. Unfortunately, the direct transmission of item details at the network layer will disclose the seller's identity. One possible method is onion routing, but

it does not achieve identity authentication, allowing any sender to transmit messages without verification. Therefore, we need to explore a method to anonymously submit item details while ensuring that the submissions come from registered sellers. We propose an anonymous submission scheme that allows verified senders to submit messages anonymously. Our idea is that the sender needs to attach the corresponding ring signature when sending a message. Only messages from verified senders can pass the verification process. We give a construction below.

Construction 1: The anonymous submission scheme $\Psi_{\mathtt{AS}}$ make use of a public encryption scheme $\Pi_{\mathtt{PKE}} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$ and a ring signature scheme $\Pi_{\mathtt{RS}} = (\mathtt{KeyGen}, \mathtt{Sign}, \mathtt{Verify})$. The scheme involves N senders and one receiver and consists of the following algorithms. We assume that the sender has an identity id and make them implicit input to all algorithms.

- $(\mathsf{rvk}, \mathsf{rsk}, \mathsf{mpk}, \mathsf{msk}) \leftarrow \mathtt{KeyGen}(1^\lambda)$. Given the security parameter λ, a pair of public and private keys $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathtt{PKE.KeyGen}(1^\lambda)$, and a pair of verification and signing keys $(\mathsf{rvk}, \mathsf{rsk}) \leftarrow \mathtt{RS.KeyGen}(1^\lambda)$. Return $(\mathsf{rvk}, \mathsf{rsk}, \mathsf{mpk}, \mathsf{msk})$.
- $(\mathbf{V}', \mathbf{P}') \leftarrow \mathtt{Auth}(\mathsf{rvk}, \mathsf{mpk}, \mathbf{V}, \mathbf{P})$. Given a verification key and a public key $(\mathsf{rvk}, \mathsf{mpk})$, and a verification keys list $\mathbf{V}$ and a public keys list $\mathbf{P}$, return $(\mathbf{V}', \mathbf{P}')$ where $\mathbf{V}' = \mathbf{V} \cup \{\mathsf{rvk}\}$ and $\mathbf{P}' = \mathbf{P} \cup \{\mathsf{mpk}\}$.
- $M \leftarrow \mathtt{EncPack}(m, \mathsf{rsk}, \mathbf{V}, \mathbf{P})$. Given a message $m \in \mathcal{M}_\lambda$, a signing key rsk, a verification keys list $\mathbf{V} = \{\mathsf{rvk}_1, \ldots, \mathsf{rvk}_N\}$, and a public keys list $\mathbf{P} = \{\mathsf{mpk}_1, \ldots, \mathsf{mpk}_N\}$. Do the following steps.
  1) Generate a ring signature $\sigma \leftarrow \mathtt{RS.Sign}(\mathsf{rsk}, m, \mathbf{V})$;
  2) Choose l random identities $\mathbf{IS} = \{\mathsf{id}_1, \ldots, \mathsf{id}_l\}$ and corresponding public keys $\mathbf{P}' = \{\mathsf{mpk}_{\mathsf{id}_1}, \ldots, \mathsf{mpk}_{\mathsf{id}_l}\}$, where $0 < l \leq N$ and $\mathbf{P}' \subseteq \mathbf{P}$. Set $M = ((m, \sigma), \perp)$;
  3) Check if the set $\mathbf{IS}$ is empty.
     - If $\mathbf{IS} \neq \emptyset$: Choose a random identity $\mathsf{id}_r \xleftarrow{r} \mathbf{IS}$ and encrypt $\mathsf{ct} \leftarrow \mathtt{PKE.Enc}(M, \mathsf{mpk}_{\mathsf{id}_r})$. Update $\mathbf{IS} \leftarrow \mathbf{IS} \setminus \{\mathsf{id}_r\}$ and $M = (\mathsf{ct}, \mathsf{id}_r)$. Repeat this step;
     - Else: Return package result M.
- $(m, \sigma) \leftarrow \mathtt{Routing}(M, \{\mathsf{msk}_{\mathsf{id}_1}, \ldots, \mathsf{msk}_{\mathsf{id}_l}\})$. This protocol is executed between $l + 1$ senders and one receiver. Given a package result M and l private keys $\{\mathsf{msk}_{\mathsf{id}_1}, \ldots, \mathsf{msk}_{\mathsf{id}_l}\}$. Set $k = 1$. Parse $M = (\mathsf{ct}_k, \mathsf{id}_k)$ and send ct_k to the sender id_k. Senders do the following steps recursively until the receiver receives message and signature.
  1) Decrypt ciphertext $M_{\text{next}} \leftarrow \mathtt{PKE.Dec}(\mathsf{ct}_k, \mathsf{msk}_{\mathsf{id}_k})$;
  2) Let $k = k + 1$ and parse $M_{\text{next}} = (\mathsf{ct}_k, \mathsf{id}_k)$;
  3) Checks if the id_k is a valid identity.
     - If $\mathsf{id}_k \neq \perp$: Send ct_k to the sender id_k;
     - Else: Send $\mathsf{ct}_k = (m, \sigma)$ to the receiver.
- $0/1 \leftarrow \mathtt{Verify}(\mathbf{V}, m, \sigma)$. Given a message m, a signature σ and a verification keys list $\mathbf{V}$, return $0/1 \leftarrow \mathtt{RS.Verify}(\mathbf{V}, m, \sigma)$.

Properties. The anonymous submission scheme $\Psi_{\mathtt{AS}}$ provides the following properties.

- *Correctness*. If all senders on the routing path execute the protocol faithfully, the receiver will receive a message and a corresponding signature that can verify it.
- *Anonymity*. For any submitted message, no participant other than the message sender knows who submits it.
- *Unforgeability*. The unauthenticated sender's message will not pass verification.

Remark. In $\Psi_{\mathtt{AS}}$, we actually hide the identity of the sender among a group of senders. The receiver always knows that the verifiable message must come from a set of authorized senders. Furthermore, when the set size is 1, the receiver can explicitly identify the sender. This is an inherent problem of onion routing and is not within the scope of our scheme.

B. Oblivious Bidding

Bidding constitutes the central part of the sealed-bid auction framework. During this phase, sellers and bidders confirm the highest bid for their items and determine the winners with the assistance of the auctioneer. Oblivious bidding enables bidders to hide their identity when submitting bids. The seller and the winner can confirm the auction result without knowing each other's identity. In recent work, Zhong et al. [3] propose the first oblivious bidding scheme based on private information retrieval. However, their scheme still compromises identity privacy because the protocol will reveal the winner's identity. Meanwhile, they do not give a formal definition of oblivious bidding. We formally give the definition below and provide the concrete construction in Section V-B.

Definition 6 (Oblivious Bidding): An oblivious bidding sche-me $\Psi_{\mathtt{OB}}$ consists of the following protocols between N sellers, M bidders, and an auctioneer.

- $\mathbf{b} \leftarrow \mathtt{Submit}(\{b_j\}_{j \in [M]})$. a protocol executed between M bidders and the auctioneer. The input to each bidder is a bid $b_j, j \in [M]$ and the auctioneer does not have any input. At the end of this protocol, each bidder outputs nothing and the auctioneer outputs a bidding vector $\mathbf{b}$.
- $ct \leftarrow \mathtt{Pri\text{-}Compute}(\mathbf{b})$. a deterministic algorithm that takes in a bidding vector $\mathbf{b}$, and outputs the ciphertext ct.
- $\mathsf{AR} \leftarrow \mathtt{Confirm}(ct)$. a protocol executed between N sellers, M bidders and the auctioneer. The input for all participants is ct. At the end of this protocol, the seller (item's owner) and the winning bidder output the auction result AR, and other participants output nothing.

Properties. The oblivious bidding scheme $\Psi_{\mathtt{OB}}$ provides the following properties.

- *Correctness*. If bidders, sellers, and the auctioneer execute the protocol faithfully, sellers can determine the highest bid for their items, and bidders know which items they win.
- *Privacy*. (1) only the losing bidder knows his/her losing bid. (2) only the seller and the winner know the highest bid.
- *Anonymous*. For any item: (1) only the holder (i.e. the seller) knows who owns it. (2) only the winner knows who wins.

C. Locker Transaction

During the transaction phase, the seller transfers the item key to the winning bidder and receives the wallet key in return. This task can be abstracted as a system with n users (including sellers and bidders) and a central server. There are k user pairs in the system (where $k \leq \binom{n}{2}$) that aim to securely exchange specific data (item keys and wallet keys).

We propose the locker transaction, an anonymous message exchange protocol enabling secure and private communication among multiple user pairs within a group setting. We assume that any pair of users who want to exchange messages has a common reference string in advance. We give the definition below and provide the concrete construction in Section V-C.

Definition 7 (Locker Transaction): A locker transaction scheme $\Psi_{\mathtt{LT}}$ consists of the following algorithms between n users and one server. There are k user pairs who want to exchange messages, where $k \leq \binom{n}{2}$.

- $r \leftarrow \mathtt{Submit}(m, \mathsf{crs})$. a randomized algorithm that takes in a message $m \in \mathcal{M}$ and a string crs, and outputs a result r.
- $\mathsf{DB} \leftarrow \mathtt{Lock}(\{r_i\}_{i\in[2k]})$. a deterministic algorithm that takes in a group of results r_i, and outputs a database DB.
- $m \leftarrow \mathtt{Retrieval}(\mathsf{DB}, \mathsf{crs})$. a protocol executed between a user and the server. The input to the server is a database DB and the user is a common reference string crs. At the end of this protocol, the server outputs nothing and the user outputs a message m.

Properties. The locker transaction scheme $\Psi_{\mathtt{LT}}$ provide the following properties:

- *Correctness*. If any two users and the server execute the protocol faithfully, both users will successfully get each other's messages.
- *Privacy*. The sender's result r hides the message m from the server and non-intended receivers.
- *Anonymous*. For any pair of users exchanging messages, it remains unknown to any third party that communication has occurred between them. Additionally, the two users involved in the exchange are unaware of each other's identity.

V. Boreas: A Fully Anonymous Auction Scheme

We introduce a fully anonymous sealed-bid auction scheme, named *Boreas*. The scheme consists of three phases: registration, bidding, and transaction, and involves N sellers denoted by S_i, $i \in [N]$, M bidders denoted by B_j, $j \in [M]$, and two non-colluding auction servers A_0 and A_1. We show Boreas's architecture in Figure 3 and protocol flowchart in Figure 4.

A. The Registration Phase

In the registration phase, the seller interacts with two auction servers to submit a message anonymously. The seller's input is a message m, including item detail cstr and public key Spk. The output of the auction servers is a database DB_{m}, which is publicly accessible and contains N sellers' message. Briefly, the seller first generates keys required for $\Psi_{\mathtt{AS}}$, and the servers add keys to the key lists. It ensures that the messages sent by this seller can be verified on the servers. Then, the seller sends a message with the corresponding signature to the servers via a group of sellers. The servers finally publish the message after verifying it. We use our anonymous submission scheme $\Psi_{\mathtt{AS}}$ and the public encryption scheme $\Pi_{\mathtt{PKE}}$. The servers initially possess an empty database DB_{m}.

Initialization: Public parameters generation.

1) A_0 generates keys $(\mathsf{Apk}_0, \mathsf{Ask}_0) \leftarrow \mathtt{PKE.KeyGen}(1^\lambda)$;
2) A_1 generates keys $(\mathsf{Apk}_1, \mathsf{Ask}_1) \leftarrow \mathtt{PKE.KeyGen}(1^\lambda)$.

Step 1: Seller registration. For each seller S_i, $i \in [N]$:

1) Generates keys $(\mathsf{rvk}_i, \mathsf{rsk}_i, \mathsf{mpk}_i, \mathsf{msk}_i) \leftarrow \mathtt{AS.KeyGen}(1^\lambda)$ in $\Psi_{\mathtt{AS}}$, then sends the verification key and public key $(\mathsf{rvk}_i, \mathsf{mpk}_i)$ to two auction servers;
2) The auction servers add keys $(\mathsf{rvk}_i, \mathsf{mpk}_i)$ to verification and public key list $\mathbf{V}$ and $\mathbf{P}$, and assign an identity id_{S_i} to the seller.

Step 2: Item submission. For each seller S_i, $i \in [N]$:

1) Generates keys $(\mathsf{Spk}_i, \mathsf{Ssk}_i) \leftarrow \mathtt{PKE.KeyGen}(1^\lambda)$;
2) Packages a message $m_i = (\mathsf{cstr}_i, \mathsf{Spk}_i)$;
3) Encrypts and packages the message m_i into the result $M_i \leftarrow \mathtt{AS.EncPack}(m_i, \mathsf{rsk}_i, \mathbf{V}, \mathbf{P})$;
4) Submits M_i to the auction servers via routing protocol together with other l sellers $(m_i, \sigma_i) \leftarrow \mathtt{AS.Routing}(M_i, \{\mathsf{msk}_{\mathsf{id}_1}, \ldots, \mathsf{msk}_{\mathsf{id}_l}\})$;
5) The auction servers receive(m_i, σ_i). Verify the signature$0/1 \leftarrow \mathtt{AS.Verify}(m_i, \sigma_i)$. If the result is 1, insert the message$m_i$into the databaseDB_{m}.

B. The Bidding Phase

During the bidding phase, bidders interact with two auction servers to bid on items. We take the bidding for item c_k as an example and assume that B_j is the winner. The input for each bidder consists of the bid b, the public key Bpk and two indexes $cadd, wadd$ (indicating where the item key and wallet key are placed during the transaction phase). Only the seller and the winner output the string $\mathsf{crs}_k = \mathsf{Spk}_k \| \mathsf{Bpk}_j \| cadd_{j,k} \| wadd_{j,k} \| \mathsf{hbid}_k$that contains the highest bidhbid_k. Briefly, each bidder first hashes the input to get a position, and then encrypts the input with the public keys of auction servers. Each server only decrypts the layer encrypted by its own public key, and uses pseudorandom permutation to obfuscate the order of the results. The servers finally place the bid at the position specified by the bidder in the bid vectors. Through the MPC protocol, the servers obtain the position of the highest bid and encrypts the result with the seller's public key. This ensures that only the seller can recover the highest bid, and the bidder can determine whether he/she is the winner from the position. We use a collision-resistance hash function$\Pi_{\mathtt{H}}$and two pseudorandom permutationsπ_0, π_1.

Step 1: Bid submission. For a bidderB_j, $j \in [M]$:

1) Generates keys $(\mathsf{Bpk}_j, \mathsf{Bsk}_j) \leftarrow \mathtt{PKE.KeyGen}(1^\lambda)$ and two additive shares Bpk_j^0 and Bpk_j^1 such that $\mathsf{Bpk}_j = \mathsf{Bpk}_j^0 + \mathsf{Bpk}_j^1$;
2) Sets bid $b_{j,k} \in \mathbb{Z}_p$ and chooses two indexes $cadd_{j,k}$, $wadd_{j,k}$. Generates two additive shares $b_{j,k}^0$ and $b_{j,k}^1$ such

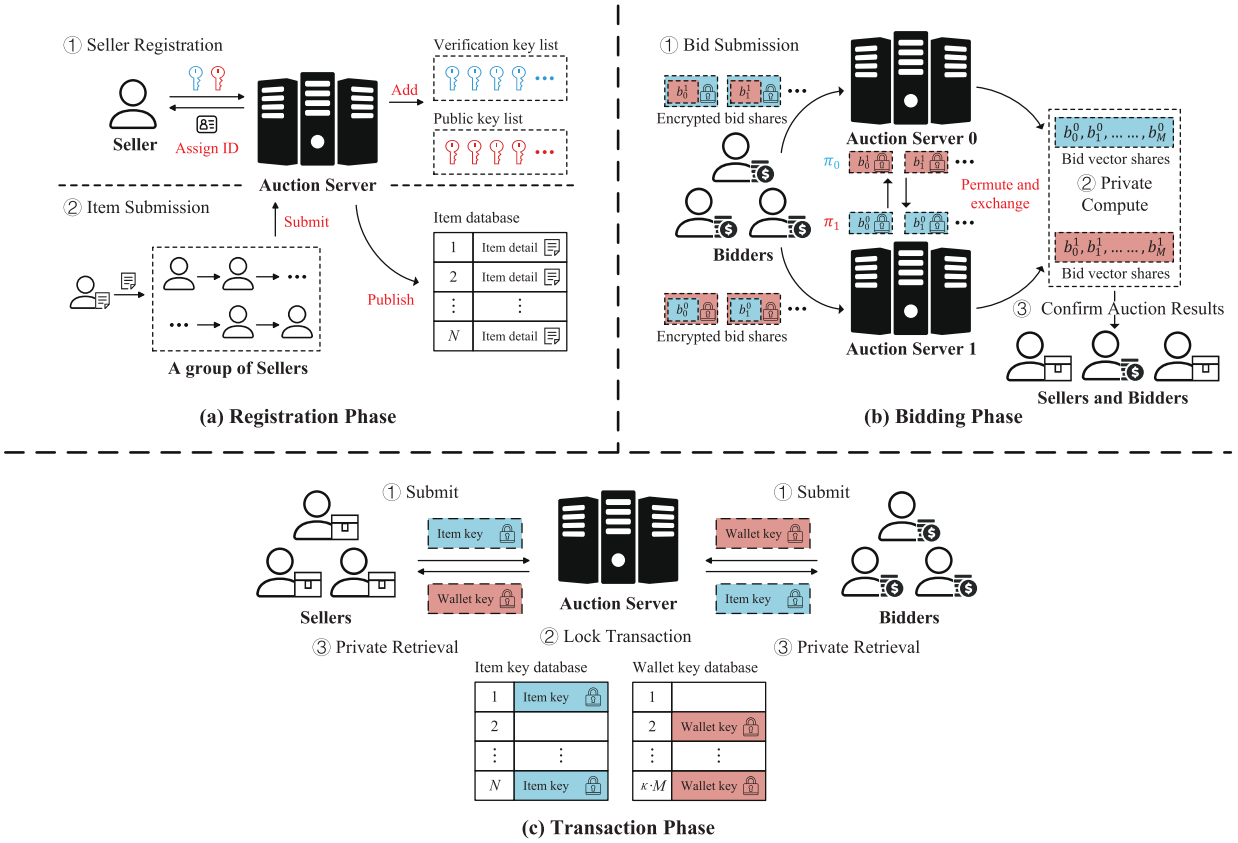

Fig. 3. Overview of Boreas's architecture. It consists of three phases: registration, bidding, and transaction.

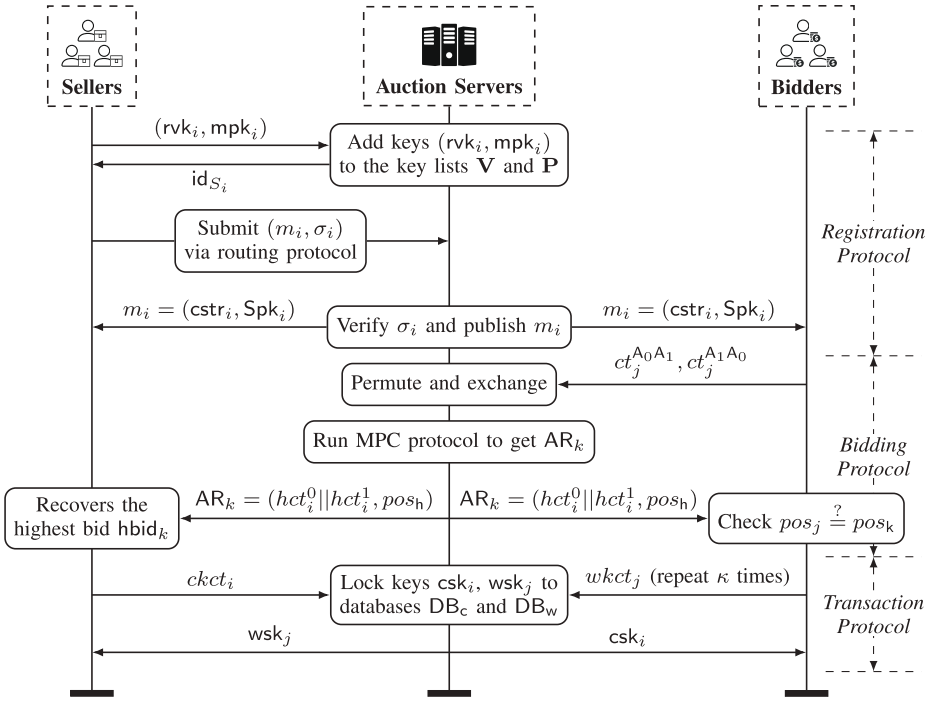

Fig. 4. Protocol flowchart of Boreas.

that $b_{j,k} = b^0_{j,k} + b^1_{j,k}$. Lets $badd_{j,k} = cadd_{j,k} \| wadd_{j,k}$ and generates two additive shares $badd^0_{j,k}$ and $badd^1_{j,k}$ such that $badd_{j,k} = badd^0_{j,k} + badd^1_{j,k}$;

3) Chooses a random element $r \xleftarrow{r} \mathbb{Z}_p$ and computes the position $pos_j \leftarrow \mathtt{Hash}(r \| b^0_{j,k} \| b^1_{j,k} \| badd^0_{j,k} \| badd^1_{j,k})$. Packages the position with the shares and

obtains $m^0_{j,k} = (pos_j, \mathsf{Bpk}^0_j, b^0_{j,k}, badd^0_{j,k})$ and $m^1_{j,k} = (pos_j, \mathsf{Bpk}^1_j, b^1_{j,k}, badd^1_{j,k})$;

4) Encrypts the message $m^i_{j,k}, i \in \{0,1\}$ by using the public keys Apk_i and Apk_{1-i} of the auction servers in sequence. That is, $ct^{\mathsf{A}_1\mathsf{A}_0}_{j,k} \leftarrow \mathtt{PKE.Enc}(\mathtt{PKE.Enc}(m^0_{j,k}, \mathsf{Apk}_0), \mathsf{Apk}_1)$, and $ct^{\mathsf{A}_0\mathsf{A}_1}_{j,k} \leftarrow \mathtt{PKE.Enc}(\mathtt{PKE.Enc}(m^1_{j,k}, \mathsf{Apk}_1), \mathsf{Apk}_0)$;
5) Sends $ct^{\mathsf{A}_0\mathsf{A}_1}_{j,k}$ to A_0 and $ct^{\mathsf{A}_1\mathsf{A}_0}_{j,k}$ to A_1.

For the auction server $A_i, i \in \{0,1\}$, upon receiving ciphertexts $\{ct^{\mathsf{A}_{1-i}\mathsf{A}_i}_{j,k}\}_{j\in[M]}$ from all bidders:

1) Generates a empty bidding vector $\mathbf{b}^i_k$ of length $\lambda \cdot M$;
2) Decrypts $ct^{\mathsf{A}_{1-i}}_{j,k} \leftarrow \mathtt{PKE.Dec}(ct^{\mathsf{A}_{1-i}\mathsf{A}_i}_{j,k}, \mathsf{Ask}_i), j \in [M]$. Sets $\mathbf{ct}_{1-i} = \{ct^{\mathsf{A}_{1-i}}_{1,k}, \ldots, ct^{\mathsf{A}_{1-i}}_{M,k}\}$ and sends $\pi_i(\mathbf{ct}_{1-i})$ to the auction server A_{1-i};
3) Upon receiving $\pi_{1-i}(\mathbf{ct}_i)$ from the auction server A_{1-i}, decrypts $m^i_{j,k} \leftarrow \mathtt{PKE.Dec}(ct^{\mathsf{A}_i}_{j,k}, \mathsf{Ask}_i), j \in [M]$;
4) Parses $m^i_{j,k}$ and sets $\mathbf{b}^i_k[pos_j] = (\mathsf{Bpk}^i_j, b^i_{j,k}, badd^i_{j,k})$.

At the end of this step, A_0 has bid vector share $\mathbf{b}^0_k$ and A_1 has bid vector share $\mathbf{b}^1_k$. The values at the same position in two bid vector shares are two additive shares of one bidder's bid for item c_k. Each server performs a permutation π_i before sending ciphertexts to another server. The permutation disrupts the order of bidders' ciphertexts. Position pos_j ensures that bid is placed correctly. For any bid in the bid vectors, the servers cannot learn which bidder submitted it.

Step 2: Private compute.

1) A_0 and A_1 run a MPC protocol in which the input is two bid vector shares $\mathbf{b}^0_k, \mathbf{b}^1_k$ and the output is pos_{h} which is the position of the highest value in the vector shares;
2) A_0 outputs $hct^0_k \leftarrow \mathtt{PKE.Enc}(\mathbf{b}^0_k[pos_{\mathsf{h}}], \mathsf{Spk}_k)$;
3) A_1 outputs $hct^1_k \leftarrow \mathtt{PKE.Enc}(\mathbf{b}^1_k[pos_{\mathsf{h}}], \mathsf{Spk}_k)$;
4) Sets the auction result $\mathsf{AR}_k = (hct^0_k || hct^1_k, pos_{\mathsf{h}})$.

Step 3: Confirm. All sellers and bidders download the auction result $\mathsf{AR}_k = (hct^0_k || hct^1_k, pos_{\mathsf{h}})$. We assume that bidder B_j is the winner. For a bidder B_j, $j \in [M]$:

1) If $pos_j = pos_{\mathsf{h}}$, bidder B_j is the winner of item c_k.

For the seller S_k (the owner of item c_k):

1) Decrypts $(\mathsf{hbid}^0_k, badd^0_{j,k}, \mathsf{Bpk}^0_j) \leftarrow \mathtt{PKE.Dec}(hct^0_k, \mathsf{Ssk}_k)$ and $(\mathsf{hbid}^1_k, badd^1_{j,k}, \mathsf{Bpk}^1_j) \leftarrow \mathtt{PKE.Dec}(hct^1_k, \mathsf{Ssk}_k)$;
2) Recovers the highest bid $\mathsf{hbid}_k \leftarrow \mathsf{hbid}^0 + \mathsf{hbid}^1$, two indexes $cadd_{j,k} || wadd_{j,k} = badd_{j,k} \leftarrow badd^0_{j,k} + badd^1_{j,k}$, and winner's public key $\mathsf{Bpk}_j \leftarrow \mathsf{Bpk}^0_j + \mathsf{Bpk}^1_j$.

Remark. This phase is the concrete construction of oblivious bidding and each step corresponds to a protocol in definition 6. Specifically, step 1 corresponds to the protocol `OB.Submit`, the step 2 corresponds to the algorithm `OB.Pri-Compute`, and the step 3 corresponds to the protocol `OB.Confirm`.

C. The Transaction Phase

In the transaction phase, the seller and the winner exchange item key and wallet key through the auction server. The seller's input is the item key csk, and the winner's input is the wallet key wsk. The seller outputs wsk, while the winner outputs csk. Briefly, the seller and the winner lock their keys in the server's databases and then retrieve the target keys from the databases privately via the PIR protocol. The result of the bidding phase indicates where the keys are stored in the databases.

For an item c_k, we assume that bidder B_j is the winner. In this phase, we take the exchange of item key csk_k and wallet key wsk_j as an example. Seller S_k and bidder B_j possess the string $\mathsf{crs}_k = \mathsf{Spk}_k || \mathsf{Bpk}_j || cadd_{j,k} || wadd_{j,k} || \mathsf{hbid}_k$, where the first parameter can be directly obtained from DB_{m} and the remaining parameters can be recovered from the auction result AR_k. We use a public encryption scheme $\Pi_{\mathtt{PKE}}$ and a private information retrieval $\Pi_{\mathtt{PIR}}$.

Step 1: Lock transaction. For the seller S_k:

1) Encrypts item key $ckct_k \leftarrow \mathtt{PKE.Enc}(\mathsf{csk}_k, \mathsf{Bpk}_j)$;
2) Packages ciphertext $ckct_k$ and index $cadd_{j,k}$ for item key into a message $sm_k = (ckct_k, cadd_{j,k})$;
3) Encrypts and sends $sct_k \leftarrow \mathtt{PKE.Enc}(sm_k, \mathsf{Apk}_0)$ to A_0.

For the bidder B_j (repeat κ times, where κ is a predetermined value):

1) Encrypts wallet key $wkct_j \leftarrow \mathtt{PKE.Enc}(\mathsf{wsk}_j, \mathsf{Spk}_k)$;
2) Packages ciphertext $wkct_j$ and index $wadd_{j,k}$ for wallet key into a message $bm_j = (wkct_j, wadd_{j,k})$;
3) Encrypts and sends $bct_j \leftarrow \mathtt{PKE.Enc}(bm_j, \mathsf{Apk}_0)$ to A_0.

We assumes that each seller only holds one item. Therefore, each seller only sends one ciphertext. However, a bidder may win more than one item and need to send multiple ciphertexts. To hide the number of items won by bidders, we predetermine the number of ciphertexts submitted by each bidder. In addition to the ciphertext of their wallet key, each bidder also generates a certain amount of dummy ciphertext. For the auction server A_0, upon receiving ciphertexts from all sellers and bidders:

1) Decrypts $sm_k \leftarrow \mathtt{PKE.Dec}(sct_k, \mathsf{Ask}_0)$. Parses $sm_k = (ckct_k, cadd_{j,k})$ and appends $ckct_k$ at $\mathsf{DB}_{\mathsf{c}}[cadd_{j,k}]$;
2) Decrypts $bm_j \leftarrow \mathtt{PKE.Dec}(bct_j, \mathsf{Ask}_0)$. Parses $bm_j = (wkct_j, wadd_{j,k})$ and appends $wkct_j$ at $\mathsf{DB}_{\mathsf{w}}[wadd_{j,k}]$.

The auction servers compress two databases by deleting all entries that do not store ciphertext and use a mapping table to record the index relationship between the compressed database and the original database. After compression, the size of DB_{c} is approximately N, and the size of DB_{w} is approximately $\kappa \cdot M$. We use $ncadd_{j,k}$ and $nwadd_{j,k}$ to represent the new indexes corresponding to $cadd_{j,k}$ and $wadd_{j,k}$.

Step 2: Private retrieve. For the seller S_k:

1) Executes the PIR protocol with the auction server A_0 for index $nwadd_{j,k}$ in database DB_{w} to obtain $wkct_j$;
2) Decrypts the wallet key $\mathsf{wsk}_j \leftarrow \mathtt{PKE.Dec}(wkct_j, \mathsf{Ssk}_k)$.

For the bidder B_j:

1) Executes the PIR protocol with the auction server A_0 for index $ncadd_{j,k}$ in database DB_{c} to obtain $ckct_k$;
2) Decrypts the item key $\mathsf{csk}_k \leftarrow \mathtt{PKE.Dec}(ckct_k, \mathsf{Bsk}_j)$.

Similar to step 1, each bidder needs to perform the PIR protocol κ times on database DB_{c} to hide the number of winning items. Since the PIR protocol requires large computational and communication overhead. If there are fewer sellers, bidder may consider downloading the entire database DB_{c} directly.

Remark. This phase constructs the locker transaction scheme, with each step corresponding to a protocol in definition 7. In step 1, the operation performed by a bidder or seller is algorithm `LT.Submit`, and the operation performed by the

auction server is algorithm $\mathtt{LT.Lock}$. The step 2 corresponds to the protocol $\mathtt{LT.Retrieval}$.

VI. Analysis of the Boreas Scheme

We present security proofs to demonstrate correctness, bid privacy, and fully anonymous aspects of Boreas.

A. Correctness

We say each protocol in sealed-bid auction framework satisfies correctness if the following conditions hold with an overwhelming probability.

1) Registration: at the end of this protocol, the auction servers output all item details.
2) Bidding: at the end of this protocol, the seller and the winner output the auction result.
3) Transaction: at the end of this protocol, the seller outputs the wallet key and the winner outputs the item key.

We say that the sealed-bid auction scheme satisfies correctness iff all protocols simultaneously satisfy correctness.

Lemma 1: In Boreas, the registration protocol satisfies correctness if the public encryption scheme $\Pi_{\mathtt{PKE}}$ and the ring signature $\Pi_{\mathtt{RS}}$ satisfies correctness.

Proof: The seller submits the result M to the servers, which is the ciphertext of message $m = (\mathsf{cstr}, \mathsf{Spk})$ with the signature σ encrypted sequentially using a set of keys $\{\mathsf{mpk}_1, \ldots, \mathsf{mpk}_l\}$. According to the correctness of $\Pi_{\mathtt{PKE}}$ and $\Pi_{\mathtt{RS}}$, the servers obtain (m, σ) and verify the signature $1 \leftarrow \mathtt{RS.Verify}(m, \sigma, \mathbf{V})$. Finally, the servers output the item detail cstr and the public key Spk. In summary, we complete this proof. The proof also implies the correctness of $\Psi_{\mathtt{AS}}$ in construction 1.

Lemma 2: In Boreas, the bidding protocol satisfies correctness if the public encryption scheme $\Pi_{\mathtt{PKE}}$ satisfies correctness.

Proof: Given an item c_t, each bidder B_j submits two shares $ct_j^{\mathsf{A}_{1-i}\mathsf{A}_i} \leftarrow \mathtt{PKE.Enc}(\mathtt{PKE.Enc}(m_j^i, \mathsf{Apk}_i), \mathsf{Apk}_{1-i}), i \in \{0,1\}$ to the corresponding server. The server $A_i, i \in \{0,1\}$ decrypts $m_j^i \leftarrow \mathtt{PKE.Dec}(\mathtt{PKE.Dec}(ct_j^{\mathsf{A}_{1-i}\mathsf{A}_i}, \mathsf{Ask}_{1-i}), \mathsf{Ask}_i)$. Since m_j^i contains the bid share b_j^i, each server can recover a bid vector share $\mathbf{b}_t^0$ or $\mathbf{b}_t^1$ from bidders' ciphertexts.

The servers then execute the MPC protocol, where the input is two bid vector shares $\mathbf{b}_t^0$ and $\mathbf{b}_t^1$, and the output is pos_{h}. The servers output the auction result $\mathsf{AR}_t = (hct_t^0 || hct_t^1, pos_{\mathsf{h}})$, where $hct_t^i \leftarrow \mathtt{PKE.Enc}(\mathbf{b}_t^i[pos_{\mathsf{h}}], \mathsf{Spk}_t), i \in \{0,1\}$.

All sellers and bidders can download the result AR, but only the seller S_t can recover the highest bid $\mathsf{hbid}_t = \mathbf{b}_t^0[pos_{\mathsf{h}}] + \mathbf{b}_t^1[pos_{\mathsf{h}}]$, where $\mathbf{b}_t^i[pos_{\mathsf{h}}] \leftarrow \mathtt{PKE.Dec}(hct_t^i, \mathsf{Ssk}_t), i \in \{0,1\}$. If $pos_j = pos_{\mathsf{h}}$, the bidder B_j knows he/she is the winner. In summary, we complete this proof. □

Lemma 3: In Boreas, the transaction protocol satisfies correctness if the public encryption scheme $\Pi_{\mathtt{PKE}}$ and the private information retrieval $\Pi_{\mathtt{PIR}}$ satisfies correctness.

Proof: Assume that the seller S_i and the winner B_j exchange the item key ck_i and the wallet key wk_j. They both have $\mathsf{crs}_i = \mathsf{Bpk}_j || \mathsf{Spk}_i || cadd_{j,i} || wadd_{j,i} || \mathsf{hbid}_i$ when the bidding phase is finished. The seller S_i sends $sct_i \leftarrow \mathtt{PKE.Enc}((ckct_i, cadd_{j,i}), \mathsf{Apk}_0)$ to A_0, where $ckct_i \leftarrow \mathtt{PKE.Enc}(\mathsf{ck}_i, \mathsf{Bpk}_j)$. According to the correctness of $\Pi_{\mathtt{PKE}}$ and $\Pi_{\mathtt{PIR}}$, A_0 decrypts $sct_i \leftarrow \mathtt{PKE.Dec}(ckct_i, \mathsf{Ask}_0)$ and sets $\mathsf{DB_c}[cadd_{j,i}] = ckct_i$. The winner B_j gets $ckct_i$ from $\mathsf{DB_c}$ through the PIR protocol with an overwhelming probability and decrypts $\mathsf{ck}_i \leftarrow \mathtt{PKE.Dec}(ckct_i, \mathsf{Bsk}_j)$. The proof for the seller S_i gets wk_j is the same. In summary, we complete this proof. This proof also implies the correctness proof of $\Psi_{\mathtt{LT}}$ in construction V-C. □

Theorem 1: Boreas satisfies correctness.

Proof: According to the Fact 2, Lemma 1, Lemma 2 and Lemma 3. We complete this proof. □

B. Bid Privacy

Bid privacy in the sealed-bid auction framework is equivalent to bid privacy in the bidding phase if the wallet key is private to both the bidder and the seller.

Proof: The registration phase does not include the submission of bids. In the transaction phase, the wallet key's leakage will reveal the highest bid (because the adversary can use the wallet key to get money). In summary, we complete this proof.

Lemma 4: The locker transaction scheme $\Psi_{\mathtt{LT}}$ in Construction V-C satisfies privacy if the public encryption scheme $\Pi_{\mathtt{PKE}}$ satisfies semantic security.

Proof: In construction V-C, the winner sends the ciphertext of the wallet key to the server. According to the semantic security of $\Pi_{\mathtt{PKE}}$, no adversary can infer useful information from the ciphertext. In summary, we complete this proof. □

Theorem 2: Boreas satisfies bid privacy.

Proof: We assume that the PPT adversary $\mathcal{A}$ acts as the seller. $\mathcal{A}$ samples two bid vectors $\mathbf{b}_0$ and $\mathbf{b}_1$ uniformly at random from $\mathbb{Z}_n^M$ subject to $|\mathbf{b}_0|_\infty = |\mathbf{b}_1|_\infty$. The experiment chooses a random bit $t \in \{0,1\}$. The experiment is executed as follows.

(1) Each bidder B_j splits the bid $\mathbf{b}_t[j]$ into two pieces $\mathbf{b}_t^0[j]$ and $\mathbf{b}_t^1[j]$, encrypts them and sends ciphertexts $ct_t^0[j]$ and $ct_t^1[j]$ to the servers. Each server decrypts all ciphertexts and obtains a bid vector share $\mathbf{b}_t^0$ or $\mathbf{b}_t^1$.
In the public channel, the adversary $\mathcal{A}$ has $2M$ ciphertexts $\{ct_t^i[j]\}_{i\in\{0,1\}, j\in[M]}$, where $ct_t^i[j] \leftarrow \mathsf{PKE.Enc}(\mathbf{b}_t^i[j], \mathsf{Apk}_i)$. According to the semantic security of $\Pi_{\mathtt{PKE}}$, we have

$$\Pr[\mathcal{A}(1^\lambda, \{ct_t^i[j]\}_{i\in\{0,1\}, j\in[M]}) = t] \le \frac{1}{2} + \mathsf{negl}(\lambda) \quad (1)$$

Intuitively, this step ensures that even if the adversary $\mathcal{A}$ controls one auction server, the probability that $\mathcal{A}$ correctly infers bit t is no better than random guessing (i.e. 1/2).

(2) The servers run a MPC protocol and output index pos. Encrypt $hct_t^i \leftarrow \mathsf{PKE.Enc}(\mathbf{b}_t^i[pos], \mathsf{Ssk}), i \in \{0,1\}$ and output the auction result $\mathsf{AR} = (hct_t^0 || hct_t^1, pos)$.
We use the MPC protocol in a black-box manner, where the adversary knows nothing about the inputs. According to the semantic security of $\Pi_{\mathtt{PKE}}$, we have

$$\Pr[\mathcal{A}(1^\lambda, hct_t^0, hct_t^1) = t] \le \frac{1}{2} + \mathsf{negl}(\lambda) \quad (2)$$

Intuitively, this step ensures that even if the adversary $\mathcal{A}$ knows the auction result AR, the probability that $\mathcal{A}$ correctly infers bit t is no better than random guessing (i.e. 1/2).

(3) The seller who owns the item and the winner can determine the highest bid locally. The adversary $\mathcal{A}$ gets the highest bid $|\mathbf{b}_t|_\infty$.
Since the maximum values in the two bid vectors are equal, the adversary $\mathcal{A}$ cannot determine which bid vector was chosen in the experiment based on the highest bid $|\mathbf{b}_t|_\infty$.

(4) The adversary $\mathcal{A}$ outputs a guess t'.

The probability that the adversary $\mathcal{A}$ wins the experiment is the maximum of equations 1 and 2 and it has at most negligible advantage in this experiment.

$$\mathsf{Adv}_{\mathsf{BP}}(\mathcal{A}) = |\Pr[\mathsf{Exp}_{\mathsf{BP}}(\mathcal{A}) = 1] - \frac{1}{2}| \leq \mathsf{negl}(\lambda)$$

This means that $\mathcal{A}$ does not know which bid vector was chosen in the experiment, and therefore cannot obtain bidders' bids.

The same proof holds when the adversary $\mathcal{A}$ plays the role of the bidder or other entity. The difference is that the two bid vectors require different constraints when $\mathcal{A}$ plays a different role. We combine all situations and complete this proof. □

C. Fully Anonymous

To understand the proof of fully anonymous in Boreas, we begin with the identity privacy of sellers and bidders, and then demonstrate their combined security.

Lemma 5: Boreas satisfies the identity privacy of sellers.

Proof: In the registration protocol, each seller S_i submits item details cstr_i and public key Spk_i to the servers using anonymous submission scheme $\Psi_{\mathtt{AS}}$. The identity of the submitted seller is hidden in a group of sellers. The servers receive a signature indicating that the message sender is an authenticated seller. According the anonymity of $\Pi_{\mathtt{RS}}$, we have

$$|\Pr[(\mathsf{V}, m, i_0, i_1) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\cdot)}; b \xleftarrow{\mathrm{r}} \{0,1\}; \Sigma^* \leftarrow \mathsf{RS.Sign}(\mathsf{msk}_{i_b}, m, \mathsf{V}) : \mathcal{A}(\Sigma^*) = b] | \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

It means that $\mathcal{A}$ does not know which seller signs this message. Therefore, we have

$$\Pr[\mathcal{A}(1^\lambda, \mathsf{cstr}_i, \mathsf{Spk}_i, \sigma_i) = i] \leq \frac{1}{N} + \mathsf{negl}(\lambda) \tag{3}$$

In the bidding protocol, the seller S_i downloads the auction result AR_i. Then S_i recovers the highest bid hbid_i, two indexes $cadd_{j,i}$ and $wadd_{j,i}$ locally. This means that $\mathcal{A}$ does not know which seller correctly recovers the result, and therefore cannot associate item c_i with its seller.

In the transaction protocol, each bidder sends κ ciphertexts of the wallet keys (some are dummy ciphertexts) to the servers. Each seller executes the PIR protocol to obtain the wallet key that wins his/her item. According to the privacy of $\Pi_{\mathtt{PIR}}$, the adversary $\mathcal{A}$ does not know which seller obtains which key. Assuming that $wadd_{j,i}$ is the query index of seller S_i, we have

$$\Pr[\mathcal{A}(\mathcal{P}(wadd_{j,i})) = 1] \leq \frac{1}{N} + \mathsf{negl}(\lambda) \tag{4}$$

The probability that $\mathcal{A}$ wins the experiment $\mathsf{Exp}_{\mathsf{SI}}(\mathcal{A})$ is the maximum of equations 3 and 4 and it has at most negligible advantage in $\mathsf{Exp}_{\mathsf{SI}}(\mathcal{A})$.

$$\mathsf{Adv}_{\mathsf{SI}}(\mathcal{A}) = |\Pr[\mathsf{Exp}_{\mathsf{SI}}(\mathcal{A}) = 1] - \frac{1}{N}| \leq \mathsf{negl}(\lambda)$$

It means that the probability that $\mathcal{A}$ outputs a valid seller and item pair $(\mathsf{cstr}_i, i), i \in [N]$ (S_i is the owner of the item c_i) is no better than random guessing (i.e. $1/N$). In summary, we complete this proof. □

Lemma 6: Boreas satisfies the identity privacy of bidders.

Proof: In the bidding protocol, bidders encrypt their bid shares and send them to the corresponding servers. Servers A_0 and A_1 permute and exchange the ciphertexts. It means that even if the adversary $\mathcal{A}$ controls one auction server, it cannot infer which bidder submitted the bid shares since it does not know the permutation of the other server.

In the transaction protocol, each seller submits a ciphertext of the item key to the servers. Each bidder executes the PIR protocol to obtain the item key he/she wins. According to the privacy of $\Pi_{\mathtt{PIR}}$, the adversary $\mathcal{A}$ does not know which bidder obtains which key. Assuming that $cadd_{j,i}$ is the query index of bidder B_j, we have

$$\Pr[\mathcal{A}(\mathcal{P}(cadd_{j,i})) = 1] \leq \frac{1}{M} + \mathsf{negl}(\lambda) \tag{5}$$

The probability that $\mathcal{A}$ wins the experiment $\mathsf{Exp}_{\mathsf{BI}}(\mathcal{A})$ is the equation 5 and it has at most negligible advantage in $\mathsf{Exp}_{\mathsf{BI}}(\mathcal{A})$.

$$\mathsf{Adv}_{\mathsf{BI}}(\mathcal{A}) = |\Pr[\mathsf{Exp}_{\mathsf{BI}}(\mathcal{A}) = 1] - \frac{1}{M}| \leq \mathsf{negl}(\lambda)$$

It means that the probability that $\mathcal{A}$ outputs a valid bidder and item pair $(\mathsf{cstr}_i, j), j \in [M]$ (B_j is the winner of the item c_i) is no better than random guessing (i.e. $1/M$). In summary, we complete this proof. □

Theorem 3: Boreas satisfies fully anonymous.

Proof: From Lemma 5 and Lemma 6, we know that the advantage of the adversary $\mathcal{A}$ in the experiments $\mathsf{Exp}_{\mathsf{SI}}(\mathcal{A})$ and $\mathsf{Exp}_{\mathsf{BI}}(\mathcal{A})$ is negligible. We assume that the adversary $\mathcal{A}$ plays the role of the seller S_i who knows a seller and item pair (cstr_i, i). In this case, we have

$$\Pr[\mathsf{Exp}_{\mathsf{SI}}(\mathcal{A}) = 1] \leq \frac{1}{N-1} + \mathsf{negl}(\lambda)$$

$$\text{and} \quad \Pr[\mathsf{Exp}_{\mathsf{BI}}(\mathcal{A}) = 1] \leq \frac{1}{M} + \mathsf{negl}(\lambda)$$

For item c_j, we assume S_j is the owner and B_k is the winner. The adversary $\mathcal{A}$ outputs a guess $(\mathsf{cstr}_j, b_1, b_2)$ in experiment $\mathsf{Exp}_{\mathsf{FA}}(\mathcal{A})$. When $b_1 = j$, the value of b_2 can be anything, resulting in M valid guesses. When $b_2 = k$, the value of b_1 can be anything except i, resulting in $N-1$ valid guesses. By removing the repeated result (cstr_j, j, k), we have

$$\Pr[\mathsf{Exp}_{\mathsf{FA}}(\mathcal{A}) = 1] \leq \frac{N + M - 2}{(N-1) \cdot M} + \mathsf{negl}(\lambda)$$

It means that $\mathcal{A}$ wins the experiment $\mathsf{Exp}_{\mathsf{FA}}(\mathcal{A})$ is no better than random guessing and it has at most negligible advantage in $\mathsf{Exp}_{\mathsf{FA}}(\mathcal{A})$. The same proof holds when $\mathcal{A}$ plays the role of bidder or other entity. We combine all situations and complete this proof. □

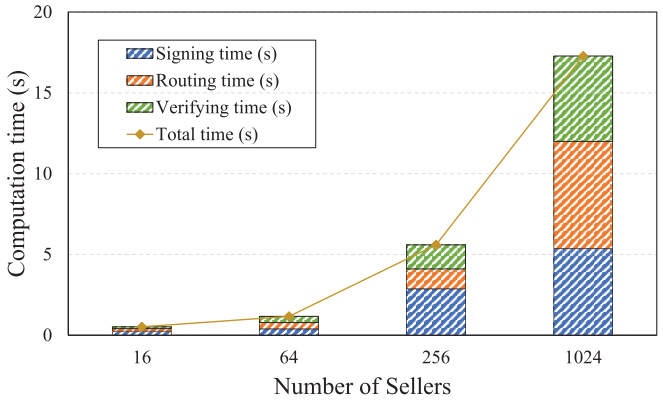

Fig. 5. The computation time for each step of submitting an item detail under different numbers of sellers in Boreas.

TABLE II
THE COMPARISON OF ITEM SUBMISSION PERFORMANCE BETWEEN BOREAS AND EXISTING AUCTION SCHEMES

Number of Sellers	16	64	256	1024
Boreas's submission				
Avg. Comm. Cost (KB)	71.68	96.77	157.2	430.34
Tot. Comm. Cost (MB)	0.28	0.76	2.46	13.45
Comp. Time (s)	0.52	1.17	5.59	17.28
Direct submission in [2–4, 35]				
Comm. Cost (KB)	2	2	2	2
Comp. Time (s)	0.93	0.87	1.1	0.96

VII. PERFORMANCE EVALUATION

We implement Boreas using C++ on a server with Ubuntu 20.04.6 LTS operating system, Intel Xeon Gold 5218 CPU @2.30GHz and 32GB DDR4 RAM. We use Falafl [38] as the ring signature scheme and OpenSSL 3.1.0 [42] to implement the public key encryption scheme. The semi-honest two-party private computation protocol is implemented with the sh2pc protocol [43] in EMP toolkit [44]. We adopt the open-sourced implementation [45] of SealPIR [46] as the PIR scheme.

A. Registration Phase

In this phase, we compare the item submission performance between Boreas and existing auction schemes [2], [3], [4], [35]. Compare to existing schemes where sellers submit item details directly to the auctioneer, we use an anonymous submission protocol to hide the seller's identity. We evaluate performance when the number of sellers is 2^4, 2^6, 2^8 and 2^{10}. We set the size of the item detail to 1 KB.

The experiment results are shown in Table II and Fig 5. We take the number of sellers 2^8 as an example. In Boreas's submission, the computation time is 5.59 s, where the signing time is 2.9 s, the routing time is 1.2 s, and the verification time is 1.5 s. The communication cost is 2.46 MB and the average cost per forwarding seller is 157.2 KB. In direct submission, the computation time is 1.1 s and the communication cost is 2 KB. In our solution, the seller's item needs to be forwarded and submitted to the servers by $O(\sqrt{N})$ sellers, which incurs more computational and communication overhead. But our solution hides the seller's identity and provides enhanced security. In addition, this phase is independent of real-time bidding phase, so the resulting moderate overhead is acceptable.

B. Bidding Phase

Real-time bidding is the core phase in the auction scheme. However, there are no existing works that achieve the identity privacy of sellers and bidders during the bidding phase. For a comprehensive performance evaluation, we compare this phase with an advanced scheme Ibex [3], which only achieves seller's privacy. Despite the limitations of this comparison, the experimental results still demonstrate the advantages of our solution. We set the number of bidders to 2500, 5000, 7500 and 10000 to demonstrate the capability of Boreas in handling a large-scale auction. We compare the performance with Ibex in four aspects including total computation time and communication cost, and each bidder's computation time and communication cost. The experimental results are depicted in Fig. 6. In particular, when the number of bidders is 2500, the total time of Boreas is 1.96 s, while the total time of Ibex is 2.24 s, achieving a 12.6% performance improvement. Meanwhile, our solution realizes a $10^5\times$ performance improvement in each bidder's computation time. Our result is 61 us, while Ibex's result is 936 ms. Our total communication cost is 1.8 MB and each bidder's cost is 0.5 KB, while Ibex's results are 442.1 MB and 271.8 KB. Our scheme realizes a $10^3\times$ performance improvement.

The results show that, Boreas has a significant performance improvement compared to Ibex in the real-time bidding phase. This improvement stems from a different way that the servers obtain secret bids from bidders. In Ibex, bidders bid on all items and store the encrypted bids in their database. The seller retrieves the secret bids from each bidder's database through the PIR protocol and submits them to the auction servers. The bidder's computation time increases as the number of bidders increases. In Boreas, each bidder submits secret bids directly to the auction server with two encryption operations. The bidder's computation time is independent of the number of bidders. Our solution avoids the expensive PIR protocol, making it more suitable for auctions with a large number of bidders. We also enhance the flexibility of the auction, as bidders do not need to prepare secret bids in advance for each item.

C. Transaction Phase

In this phase, the main costs for sellers and bidders come from executing the PIR protocol. Each seller executes the PIR protocol once on a wallet key database of size $\kappa \cdot M$ to obtain a wallet key and each bidder executes the PIR protocol κ times on the item key database of size N to obtain several item keys they win. With a fixed number of sellers at 32 and bidders at 256, we evaluate the computation time and communication cost of each seller and each bidder when the predefined value κ is taken as 1, 2, 4, and 8. The experiment results are shown in Fig 7. When the predefined value is 4, the computation

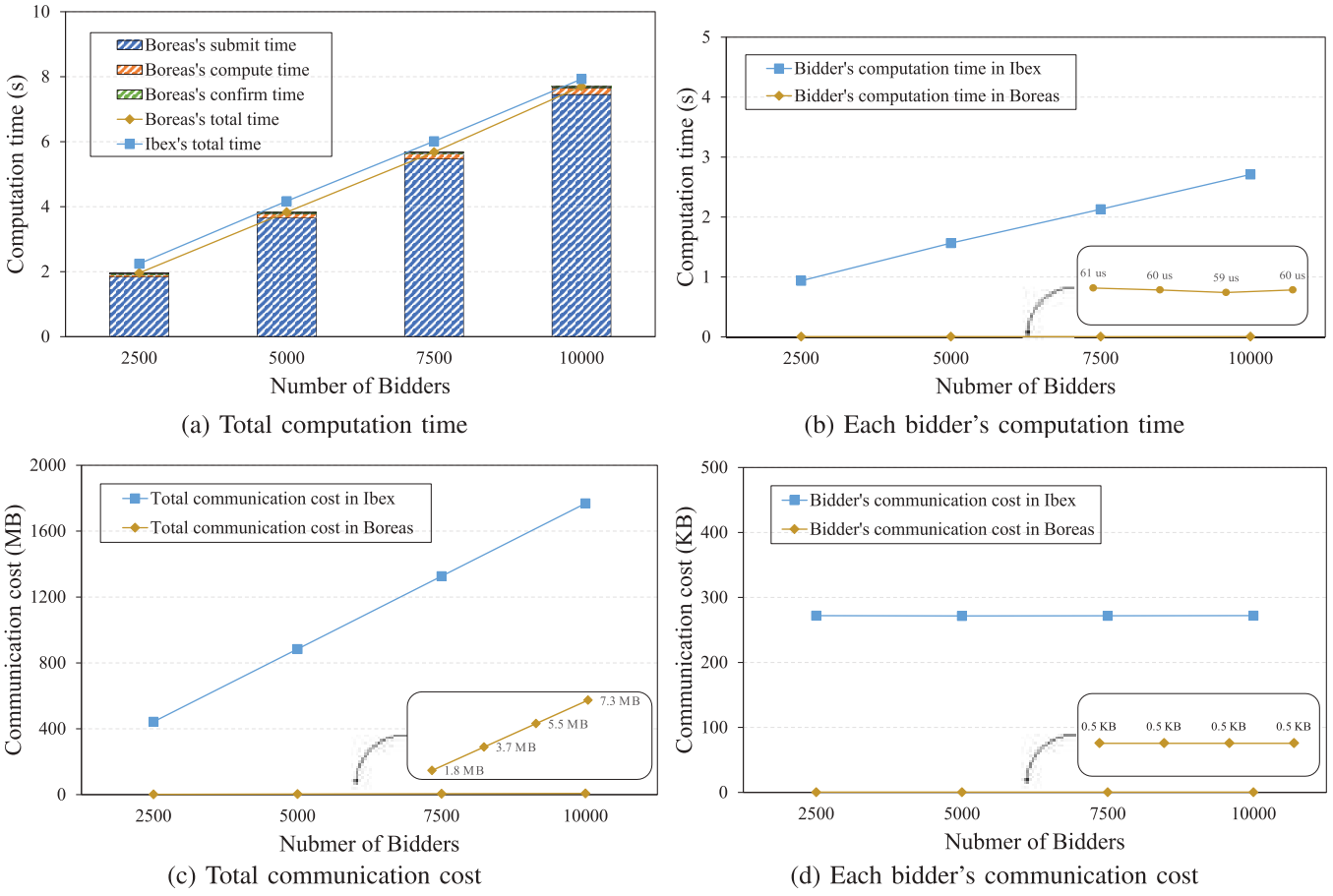

Fig. 6. Performance comparisons of real-time bidding in Boreas and Ibex with different numbers of bidders. (a) shows the total computation time, including the detailed time for each step in Boreas. (b) shows each bidder's computation time. (c) shows the total communication cost. (d) shows each bidder's communication cost.

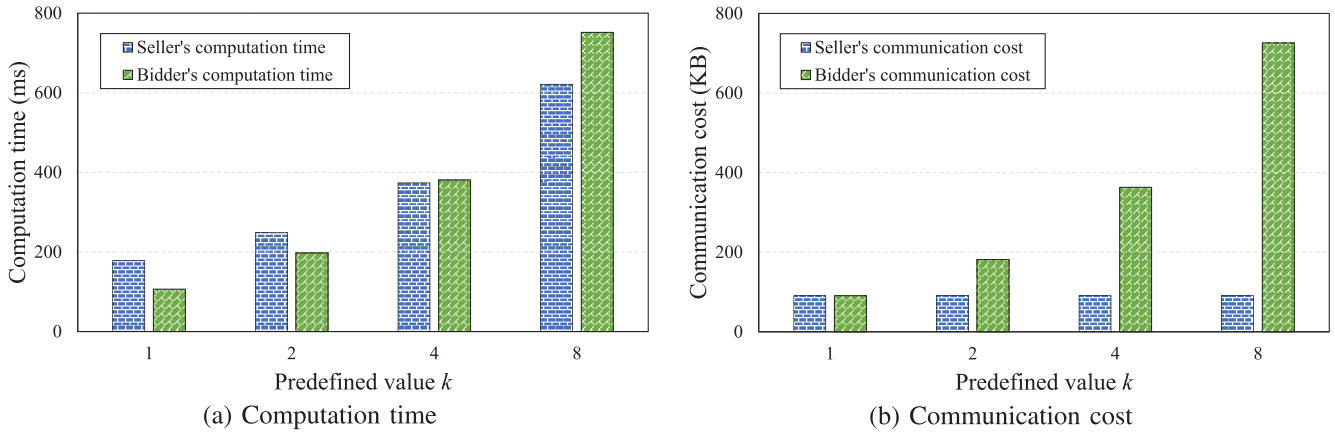

Fig. 7. The computation time and communication cost of executing the PIR protocol for each seller and each bidder.

time and communication cost for each seller is 374 ms and 91 KB. The computation time and communication cost for each bidder is 381 ms and 363 KB. The experimental results show that after bidding, both sellers and bidders can quickly obtain wallet keys and item keys with low communication costs.

VIII. Related Work

We introduce the related work on privacy-preserving sealed-bid auctions. We categorize existing works according to the type of privacy protection: *bid privacy* and *identity privacy*.

A. Bid Privacy

In the sealed-bid auctions [52], the bid privacy requires that a bidder's bid remains confidential, which is the basic property.

In 1996, Franklin and Reiter [10] proposed the first privacy-preserving sealed-bid auction scheme. In their scheme, bidders share their bids with multiple auctioneers via secret sharing. Following their work, many researchers [11], [12], [13], [14], [15] used MPC techniques to distribute the trust into multiple auctioneers. Cheng et al. [11], [12] proposed a double auction that performed operations on secret-shared data through

additive sharing and garbled circuit. In PROST [15], the auction results are obtained by the auctioneer and the agent executing a secure interaction protocol. In these schemes, the bid privacy is compromised if a certain number of auctioneers collude with each other.

Other works [4], [21], [22], [23] introduce a trusted third party other than the auctioneer into the auction. Montenegro et al. [21] proposed a sealed-bid online scheme, which includes a random server and an auctioneer. The random server needs to honestly provide randomness to bidders. In [23], Galal and Youssef determined the winner through a trusted execution environment SGX. Chen et al. [4] combined the trusted processor with smart contract on blockchain to construct a sealed-bid auction framework. In these works, the bid privacy relies on a trusted third party.

To prevent auctioneers from colluding with each other to compromise the bid privacy, some researchers [16], [17], [18], [19], [20] have proposed auction schemes to protect bid through homomorphic techniques. Zhang et al. [16] constructed a HE-based bid comparison circuit that allows all bidders to directly compute the highest bid. Blass and Kerschbaum [17] designed a verifiable bit comparison circuit. Each bidder encrypts his/her bid with others' public keys and compares it with the received encrypted bids. In FACT [18], each bidder partially decrypts the ciphertext of auction result through threshold homomorphic encryption, and the seller recovers the highest bid. However, these schemes incur significant computational overhead.

A common problem of above works is that they ignore the identity privacy. The seller of the item and the winner of the auction are transparent to all participants.

B. Identity Privacy

In the sealed-bid auctions [52], the bidder's identity privacy requires that the winner cannot be associated with a specific participant, and the seller's identity privacy requires that items cannot be linked with a specific participant.

In terms of bidder, Chang and Chang [29], [30] proposed an enhanced anonymous auction with freewheeling bids. Following their works, Jiang et al. [47] proposed an improved scheme that resist man-in-the-middle attack. These schemes ensure the anonymity for bidders but rely on a certification authority. Li et al. [48] used zero-knowledge proof to ensure that no one can identify any bidder from the bids. In essence, their method is a commitment scheme where the winner needs to publish his/her bid. Additionally, these schemes [29], [30], [47], [48] only focus on computing the highest bid and do not involve a seller.

The schemes proposed by Xiong et al. [49], [50] achieve the non-repudiation of bidders while preserving their anonymity through ring signatures and encrypted key chains. Similarly, recent works [26], [27], [28] used the ring signature on the blockchain to hide bidder's interest but reveal the winner's identity. Their overhead scales linearly with the number of bidders.

In other works, Huang et.al [51] introduced a trusted agent in auction to perform a secret permutation to anonymize bidders. MaskAuct [31] presented a blocklistable group signature to achieve bidder anonymity. Their blocklist mechanism can prevent specific bidders from winning the auction. The above schemes do not hide the relationship between sellers and items. All participants know which item belongs to which seller. This oversight becomes an issue in auctions where the items involve sensitive information about the seller.

In terms of seller, Shi [36] proposed an auction scheme in which sellers utilized private set intersection to determine the auction results. However, bidders can only bid within a given set of prices. In SEAL [35], Bag et.al provided a decentralize auction scheme via an anonymous veto protocol. Each bidder publishes the zero-knowledge proofs for the bits of bid and securely computes the logical-OR of binary inputs. This type of scheme [32], [33], [34] do not involve an auctioneer and is self-resolved by bidders. Since the results are publicly verifiable, sellers can confirm the results locally without revealing their identity. The computation works are all done by the bidders themselves, since the bidders need to have high computing power. Meanwhile, sellers can obtain the winner's identity.

Zhong et al. [3] proposed an oblivious bidding scheme to prevent bidders from knowing the seller to which the item belongs. The sellers secretly retrieve bids for their items from bidders' databases through the PIR protocol. Due to the high overhead of PIR, their scheme is not suitable for large-scale auction, and it lacks flexibility because bidder is required to bid on all items in advance. In Addax [2], they proposed the ad exchange, that hide seller's identity in a group and bidders bid on the group identifiers. Their schemes make sellers and bidders unaware of each other's identities, but the auctioneer in the middle can obtain the identities of both parties.

As shown in Table III, we compare the computational and communication overhead of Boreas with existing works that achieve at least partial identity privacy. These schemes only focus on the privacy in the bidding phase. If we consider the complete auction process, sellers usually submit items directly to the auctioneer in existing works. The seller and the winner usually directly exchange items and payment. This only incurs $O(1)$ computational and communication overhead. In contrast, each seller incurs $O(\sqrt{N})$ computational and communication overhead when submitting an item in Boreas. Each seller needs to execute the PIR protocol once, and each bidder needs to execute the protocol κ times during the transaction phase.

IX. Discussion

Integration into online auction platforms. A practical idea is how to integrate Boreas into online auction platforms such as eBay, Sotheby's, etc. We discuss the deployment requirements for each phase separately. In the registration phase, the seller's identity is essentially hidden in a group of sellers. The platform needs to determine the number of sellers in a group, and sellers join an unfilled group to submit items. In the bidding phase, the platform controls at most one auction server, since our scheme is based on the assumption of two non-colluding servers. To do this, we need to introduce a semi-honest third party as another auction server. In the transaction phase, the platform serves as the PIR server, and sellers and bidders act as PIR clients. For blockchain-based

TABLE III

THE COMPARISON OF COMPUTATIONAL AND COMMUNICATION OVERHEADS OF EXISTING SCHEMES AND BOREAS.WE USE BID, ID(S), AND ID(B) TO DENOTE THE BID PRIVACY, THE IDENTITY PRIVACY OF SELLER, AND THE IDENTITY PRIVACY OF BIDDER RESPECTIVELY.WE MEASURE THE OVERHEAD BY THE COSTLIEST OPERATION IN THESE SCHEMES.TAKE $O(1)|\mathtt{E}$ AS AN EXAMPLE, E REPRESENTS THE COSTLIEST OPERATION, AND $O(1)$ REPRESENTS THE NUMBER OF THE OPERATION.SPECIFICALLY, E REPRESENTS PUBLIC KEY ENCRYPTION, ZK REPRESENTS ZERO-KNOWLEDGE PROOF, RS REPRESENTS RING SIGNATURE, C REPRESENTS COMMITMENT, HE REPRESENTS THE HOMOMORPHIC ENCRYPTION, H REPRESENTS HASH FUNCTION, SS REPRESENTS THE PRIVATE INFORMATION RETRIEVAL, AND P REPRESENTS THE PRIVATE INFORMATION RETRIEVAL.IN TERMS OF PARAMETERS, M REPRESENTS THE NUMBER OF BIDDERS, c REPRESENTS THE BIT LENGTH OF THE HIGHEST BID, AND t IS THE DEADLINE.SYMBOL "-" INDICATES THAT THE ENTITY IS NOT PRESENT IN THE SCHEME.SYMBOL "●" INDICATES THAT THE PRIVACY IS ACHIEVED, AND SYMBOL "◐" INDICATES THAT THE PRIVACY IS NOT ACHIEVED.SYMBOL "○" INDICATES THAT ONLY PART OF THIS PRIVACY IS ACHIEVED

Scheme	Computation cost			Communication cost			Privacy		
	Seller	Bidder	Auctioneer	Seller	Bidder	Auctioneer	Bid	ID(S)	ID(B)
[29, 30, 47]	–	$O(1)\|\mathtt{E}$	$O(M)\|\mathtt{E}$	–	$O(1)\|\mathtt{E}$	$O(M)\|\mathtt{E}$	●	–	◐
[48]	–	$O(1)\|\mathtt{E}+\mathtt{ZK}$	$O(M)\|\mathtt{E}$	–	$O(1)\|\mathtt{E}+\mathtt{ZK}$	$O(M)\|\mathtt{E}+\mathtt{ZK}$	●	–	◐
[49, 50]	0	$O(1)\|\mathtt{ZK}+\mathtt{RS}$	$O(M)\|\mathtt{ZK}+\mathtt{RS}$	$O(1)$	$O(1)\|\mathtt{ZK}+\mathtt{RS}$	$O(M)\|\mathtt{ZK}+\mathtt{RS}$	●	○	◐
[26–28]	0	$O(1)\|\mathtt{RS}+\mathtt{C}+\mathtt{E}$	$O(M)\|\mathtt{RS}+\mathtt{C}+\mathtt{E}$	$O(1)$	$O(1)\|\mathtt{RS}+\mathtt{C}+\mathtt{E}$	$O(M)\|\mathtt{RS}+\mathtt{C}+\mathtt{E}$	●	○	◐
[51]	0	$O(1)\|\mathtt{E}$	$O(M)\|\mathtt{E}$	$O(1)$	$O(1)\|\mathtt{E}$	$O(M)\|\mathtt{E}$	●	○	◐
[31]	$O(1)\|\mathtt{HE}$	$O(1)\|\mathtt{HE}$	$O(\log M)\|\mathtt{HE}$	$O(1)\|\mathtt{HE}$	$O(1)\|\mathtt{HE}$	$O(M)\|\mathtt{HE}$	●	○	●
[36]	$O(tM)\|\mathtt{H}$	$O(t)\|\mathtt{HE}+\mathtt{H}$	–	$O(tM)\|\mathtt{HE}+\mathtt{H}$	$O(t)\|\mathtt{HE}+\mathtt{H}$	–	●	●	◐
[35]	$O(cM)\|\mathtt{ZK}+\mathtt{C}$	$O(c)\|\mathtt{ZK}+\mathtt{C}$	–	$O(cM)\|\mathtt{ZK}+\mathtt{C}$	$O(c)\|\mathtt{ZK}+\mathtt{C}$	–	●	●	◐
[32, 33]	$O(M)\|\mathtt{C}$	$O(2^c)\|\mathtt{SS}$	–	$O(M)\|\mathtt{C}$	$O(2^c)\|\mathtt{SS}$	–	●	●	◐
[3]	$O(M)\|\mathtt{E}$	$O(1)\|\mathtt{P}$	$O(M)\|\mathtt{E}$	$O(M)\|\mathtt{E}$	$O(1)\|\mathtt{P}$	$O(M)\|\mathtt{E}$	●	◐	◐
[2]	0	$O(1)\|\mathtt{E}$	$O(M)\|\mathtt{E}$	$O(1)$	$O(1)\|\mathtt{E}$	$O(M)\|\mathtt{E}$	●	◐	◐
Boreas	$O(1)\|\mathtt{E}$	$O(1)\|\mathtt{E}$	$O(M)\|\mathtt{E}$	$O(1)\|\mathtt{E}$	$O(1)\|\mathtt{E}$	$O(M)\|\mathtt{E}$	●	●	●

auction, the seller's item detail and item key and the bidder's wallet key are stored on the blockchain. The auction servers can be implemented by smart contracts.

In the real-world deployment, we focus on the hardware requirements of the auction servers. Since the servers need to handle a large number of requests from sellers and bidders, we use multi-core processors (such as Intel Xeon or AMD EPYC) and more than 32GB of RAM to achieve high concurrent processing. The server's storage needs to be above 500GB and the network bandwidth needs to be above 1GBps to ensure low latency. The major bottleneck in deployment is the assumption that the two servers do not collude. In addition, since the time required for sellers to submit item details increases with the size of the seller group, we only choose a smaller number of sellers per group for seller devices with less computing power.

Malicious adversary. The security of Boreas is proved under the semi-honest adversary model. We can extend our scheme to malicious adversary model through cryptographic primitives such as commitment [53], zero-knowledge proof [54], [55], etc. A malicious adversary can violate the correct execution of the protocol, such as refusing to pay or using malicious bids to disrupt the auction. We require the bidder to generate a key commitment and share the open value with two servers in advance. In the bidding phase, the bidder generates two zero-knowledge proofs: (1) a proof that his/her wallet key has been submitted to the servers, and (2) a proof that his/her wallet key is sufficient to pay the secret bid. This method eliminates the risk of the malicious bidder refusing to pay or overbidding. The seller also generates a key commitment and a zero-knowledge proof to prove that the commitment contains the item key for the submitted item. For malicious servers, we can make the auction results publicly verifiable through fully homomorphic encryption [56], [57].

X. CONCLUSION

In this paper, we propose the first sealed-bid auction scheme Boreas that both achieves bid privacy and identity privacy. In this scheme, the seller and the winner can exchange payment and item while completely hiding their identities. Bidders' bids are also kept secret from others (except the seller knows the highest bid). We propose three fundamental privacy-preserving protocols as the building blocks. Additionally, we define the security goal of identity privacy and formalize a new security property called fully anonymous within the auction framework. We provide formal security proofs showing that all protocols in

Boreas are secure under the semi-honest adversary model. Our extensive experiments show that Boreas achieves significantly better performance than existing schemes, while achieving enhanced security guarantee.

REFERENCES

[1] V. Krishna, *Auction Theory*. New York, NY, USA: Academic, 2009.

[2] K. Zhong, Y. Ma, Y. Mao, and S. Angel, "Addax: A fast, private, and accountable ad exchange infrastructure," in *Proc. USENIX NSDI*, 2023, pp. 1–21.

[3] K. Zhong, Y. Ma, and S. Angel, "Ibex: Privacy-preserving ad conversion tracking and bidding," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2022, pp. 3223–3237.

[4] Y. Chen, X. Tian, Q. Wang, J. Jiang, M. Li, and Q. Zhang, "SAFE: A general secure and fair auction framework for wireless markets with privacy preservation," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 3, pp. 2038–2053, May 2022.

[5] J. Li, X. Ni, Y. Yuan, and F.-Y. Wang, "A novel GSP auction mechanism for dynamic confirmation games on Bitcoin transactions," *IEEE Trans. Services Comput.*, vol. 15, no. 3, pp. 1436–1447, May 2022.

[6] L. Zhang, Z. Li, and C. Wu, "Dynamic resource provisioning in cloud computing: A randomized auction approach," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2014, pp. 433–441.

[7] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *J. Finance*, vol. 16, no. 1, p. 8, Mar. 1961.

[8] O. Samorodnitzky, E. Tromer, and A. Wool, "Analyzing unique-bid auction sites for fun and profit," in *Proc. NDSS*, 2013, pp. 1–18.

[9] L. Olejnik, T. Minh-Dung, and C. Castelluccia, "Selling off user privacy at auction," in *Proc. NDSS*, 2014, pp. 1–15.

[10] M. K. Franklin and M. K. Reiter, "The design and implementation of a secure auction service," *IEEE Trans. Softw. Eng.*, vol. 22, no. 5, pp. 302–312, May 1996.

[11] K. Cheng, Y. Shen, Y. Zhang, X. Zhu, L. Wang, and H. Zhong, "Towards efficient privacy-preserving auction mechanism for two-sided cloud markets," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[12] K. Cheng, L. Wang, Y. Shen, Y. Liu, Y. Wang, and L. Zheng, "A lightweight auction framework for spectrum allocation with strong security guarantees," in *Proc. IEEE INFOCOM*, Feb. 2020, pp. 1708–1717.

[13] F. Brandt and T. Sandholm, "On the existence of unconditionally privacy-preserving auction protocols," *ACM Trans. Inf. Syst. Secur.*, vol. 11, no. 2, pp. 1–21, Mar. 2008.

[14] M. Pan, J. Sun, and Y. Fang, "Purging the back-room dealing: Secure spectrum auction leveraging Paillier cryptosystem," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 866–876, Apr. 2011.

[15] Q. Wang, J. Huang, Y. Chen, C. Wang, F. Xiao, and X. Luo, "PROST: Privacy-Preserving and truthful online double auction for spectrum allocation," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 374–386, Feb. 2019.

[16] Z. Zhang et al., "A blockchain-based privacy-preserving scheme for sealed-bid auction," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 5, pp. 4668–4683, Sep. 2024.

[17] E.-O. Blaß and F. Kerschbaum, "Strain: A secure auction for blockchains," in *Proc. ESORICS*, 2018, pp. 87–110.

[18] E. Zhou et al., "FACT: Sealed-bid auction with full privacy via threshold fully homomorphic encryption," *IEEE Trans. Services Comput.*, vol. 17, no. 6, pp. 3627–3639, Nov. 2024.

[19] E.-O. Blass and F. Kerschbaum, "BOREALIS: Building block for sealed bid auctions on blockchains," in *Proc. 15th ACM Asia Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 558–571.

[20] B. David, L. Gentile, and M. Pourpouneh, "FAST: Fair auctions via secret transactions," in *Proc. ACNS*, 2022, pp. 727–747.

[21] J. A. Montenegro, M. J. Fischer, J. Lopez, and R. Peralta, "Secure sealed-bid online auctions using discreet cryptographic proofs," *Math. Comput. Model.*, vol. 57, nos. 11–12, pp. 2583–2595, Jun. 2013.

[22] F. Tramer, F. Zhang, H. Lin, J.-P. Hubaux, A. Juels, and E. Shi, "Sealed-glass proofs: Using transparent enclaves to prove and sell knowledge," in *Proc. IEEE Eur. Symp. Secur. Privacy*, Apr. 2017, pp. 19–34.

[23] H. S. Galal and A. Youssef, "Verifiable sealed-bid auction on the Ethereum blockchain," in *Proc. FC workshops*, 2019, pp. 265–278.

[24] W. Ou et al., "Optimal real-time bidding strategy for position auctions in online advertising," in *Proc. 32nd ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2023, pp. 4766–4772.

[25] Y. Liu et al., "Boosting advertising space: Designing ad auctions for augment advertising," in *Proc. 16th ACM Int. Conf. Web Search Data Mining*, Feb. 2023, pp. 1066–1074.

[26] G. Sharma, D. Verstraeten, V. Saraswat, J.-M. Dricot, and O. Markowitch, "Anonymous sealed-bid auction on Ethereum," *Electronics*, vol. 10, no. 19, p. 2340, Sep. 2021.

[27] G. Sharma, D. Verstraeten, V. Saraswat, J.-M. Dricot, and O. Markowitch, "Anonymous fair auction on blockchain," in *Proc. 11th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Apr. 2021, pp. 1–5.

[28] Z. Ye, C.-L. Chen, W. Weng, H. Sun, W.-J. Tsaur, and Y.-Y. Deng, "An anonymous and fair auction system based on blockchain," *J. Supercomput.*, vol. 79, no. 13, pp. 13909–13951, Sep. 2023.

[29] C.-C. Chang and Y.-F. Chang, "Efficient anonymous auction protocols with freewheeling bids," *Comput. Secur.*, vol. 22, no. 8, pp. 728–734, Dec. 2003.

[30] Y.-F. Chang and C.-C. Chang, "Enhanced anonymous auction protocols with freewheeling bids," in *Proc. 20th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, 2006, pp. 6 pp.–358.

[31] S. Li et al., "MaskAuct: Seller-autonomous auction with bidder anonymity and bidding confidentiality," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 7853–7865, 2024.

[32] F. Brandt, "*Secure and private auctions without auctioneers*," Technishce Universitat Munchen, Munich, Germany, Technical FKI-245–02, 2002.

[33] F. Brandt, "How to obtain full privacy in auctions," *Int. J. Inf. Secur.*, vol. 5, no. 4, pp. 201–216, Oct. 2006.

[34] F. Brandt and T. Sandholm, "Efficient privacy-preserving protocols for multi-unit auctions," in *Proc. FC*, 2005, pp. 298–312.

[35] S. Bag, F. Hao, S. F. Shahandashti, and I. G. Ray, "SEAL: Sealed-bid auction without auctioneers," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2042–2052, 2020.

[36] W. Shi, "A sealed-bid multi-attribute auction protocol with strong bid privacy and bidder privacy," *Secur. Commun. Netw.*, vol. 6, no. 10, pp. 1281–1289, Oct. 2013.

[37] M. Backes, N. Döttling, L. Hanzlik, K. Kluczniak, and J. Schneider, "Ring signatures: Logarithmic-size, no Setup–from standard assumptions," in *Proc. EUROCRYPT*, 2019, p. 196.

[38] W. Beullens, S. Katsumata, and F. Pintore, "Calamari and Falafl: Logarithmic (Linkable) ring signatures from isogenies and lattices," in *Proc. ASIACRYPT*, 2020, pp. 464–492.

[39] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–90, Feb. 1981.

[40] R. Dingledine, N. Mathewson, and P. F. Syverson, "Tor: The second-generation onion router," in *Proc. USENIX Secur.*, 2004, pp. 303–320.

[41] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," in *Proc. IEEE FOCS*, May 1998, pp. 965–981.

[42] (2023). *Openssl*. [Online]. Available: https://www.openssl.org

[43] (2022). *Emp Sh2pc*. [Online]. Available: https://github.com/emp-toolkit/emp-sh2pc

[44] J. K. Xiao Wang, Alex J. Malozemoff,. (2016). *Emp-toolkit: Efficient Multiparty Computation Toolkit*. [Online]. Available: https://github.com/emp-toolkit

[45] (2022). *Sealpir: A Computational Pir Library That Achieves Low Communication Costs and High Performance*. [Online]. Available: https://github.com/microsoft/SealPIR

[46] S. Angel, H. Chen, K. Laine, and S. Setty, "PIR with compressed queries and amortized query processing," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 962–979.

[47] R. Jiang, L. Pan, and J.-H. Li, "An improvement on efficient anonymous auction protocols," *Comput. Secur.*, vol. 24, no. 2, pp. 169–174, Mar. 2005.

[48] M.-J. Li, J. S.-T. Juan, and J. H.-C. Tsai, "Practical electronic auction scheme with strong anonymity and bidding privacy," *Inf. Sci.*, vol. 181, no. 12, pp. 2576–2586, Jun. 2011.

[49] H. Xiong, Z. Qin, and F. Li, "An anonymous sealed-bid electronic auction based on ring signature," *Int. J. Netw. Secur.*, vol. 8, pp. 235–242, Apr. 2009.

[50] H. Xiong, Z. Chen, and F. Li, "Bidder-anonymous English auction protocol based on revocable ring signature," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7062–7066, Jun. 2012.

[51] Q. Huang, Y. Gui, F. Wu, G. Chen, and Q. Zhang, "A general privacy-preserving auction mechanism for secondary spectrum markets," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1881–1893, Jun. 2016.

[52] R. Alvarez and M. Nojoumian, "Comprehensive survey on privacy-preserving protocols for sealed-bid auctions," *Comput. Secur.*, vol. 88, Jan. 2020, Art. no. 101502.

[53] A. Kate, G. M. Zaverucha, and I. Goldberg, "Constant-size commitments to polynomials and their applications," in *Proc. ASIACRYPT*, 2010, pp. 177–194.
[54] S. Setty, "Spartan: Efficient and general-purpose zkSNARKs without trusted setup," in *Proc. CRYPTO*, 2020, pp. 704–737.
[55] H. Duan, L. Xiang, X. Wang, P. Chu, and C. Zhou, "A new zero knowledge argument for general circuits and its application," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 3906–3920, 2023.
[56] J. H. Cheon, D. Kim, D. Kim, H. H. Lee, and K. Lee, "Numerical method for comparison on homomorphically encrypted numbers," in *Proc. ASIACRYPT*, 2019, pp. 415–445.
[57] J. H. Cheon, D. Kim, and D. Kim, "Efficient homomorphic comparison methods with optimal complexity," in *Proc. ASIACRYPT*, 2020, pp. 221–256.

Erjun Zhou received the B.S. degree in information security from Central South University, Changsha, China, in 2019. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Wuhan University, Wuhan, China. His research interests include applied cryptography and privacy computing.

Jing Chen (Senior Member, IEEE) received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, Wuhan. He is currently a Full Professor with the School of Cyber Science and Engineering, Wuhan University. He has published more than 150 research papers in many international journals and conferences, including USENIX Security, ACM CCS, INFOCOM, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and IEEE TRANSACTIONS ON SERVICES COMPUTING. His research interests include the areas of network security, cloud security, and mobile security. He was twice runner-up for the Best Paper at INFOCOM 2018 and INFOCOM 2021. He served as the Vice Chair for the ACM Turing Award Celebration Conference (TURC) 2023.

Min Shi received the B.S. degree in information security from Jiangsu University, Zhenjiang, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Wuhan University, Wuhan, China. His research interests include cryptography and formal analysis.

Zhengdi Huang received the B.Eng. degree in cyberspace security from Wuhan University, China, in 2023, where he is currently pursuing the M.Sc. degree in cyberspace security with the Data Security and Privacy Lab. His research interests include secure multiparty computation and homomorphic encryption.

Meng Jia received the Ph.D. degree from the School of Cyber Science and Engineering, Wuhan University, in 2024. Her research interests include blockchain and applied cryptography.

Kun He (Member, IEEE) received the Ph.D. degree from Wuhan University, Wuhan, China. He is currently an Associate Professor with Wuhan University. He has published more than 30 research papers in various journals and conferences, such as IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON MOBILE COMPUTING, USENIX Security, CCS, and INFOCOM. His research interests include cryptography and data security.

Ruiying Du received the B.S., M.S., and Ph.D. degrees in computer science from Wuhan University, Wuhan, China, in 1987, 1994, and 2008, respectively. She is currently a Professor at the School of Cyber Science and Engineering, Wuhan University. She has published more than 80 research papers in many international journals and conferences, such as IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, USENIX Security, CCS, INFOCOM, SECON, TrustCom, and NSS. Her research interests include network security, wireless networks, cloud computing, and mobile computing.