Practical Verifiable & Privacy-Preserving Double Auctions

Memar Zahedani, Armin; Vos, Jelle; Erkin, Zekeriya

DOI
10.1145/3600160.3600190

Publication date
2023

Document Version
Final published version

Published in
ARES 2023 - 18th International Conference on Availability, Reliability and Security, Proceedings

Practical Verifiable & Privacy-Preserving Double Auctions

Armin Memar Zahedani
Armin1Zahedani@gmail.com
Delft University of Technology
Delft, Netherlands

Jelle Vos
J.V.Vos@tudelft.nl
Delft University of Technology
Delft, Netherlands

Zekeriya Erkin
Z.Erkin@tudelft.nl
Delft University of Technology
Delft, Netherlands

ABSTRACT

Double auctions are procedures to trade commodities such as electricity or parts of the wireless spectrum at optimal prices. Buyers and sellers inform the auctioneer what quantity they want to buy or sell at specific prices. The auctioneer aggregates these offers into demand and supply curves and finds the intersection representing the optimal price. In this way, commodities exchange owners in an economically-efficient manner. Ideally, the auctioneer is a trusted third party that does not abuse the information they gain. However, the offers reveal sensitive information about the traders, which the auctioneer may use for economic gain as insider information. These concerns are not theoretical; investigations against auctioneers in electricity and advertisement auctions for manipulating auctions are ongoing. These concerns call for solutions that conduct double auctions in a privacy-preserving and verifiable way. However, current solutions are impractical: To the best of our knowledge, the only solutions satisfying these properties require full interaction of all participants. In this work, we design a more practical solution. We propose the first privacy-preserving and verifiable double auction scheme that does not require traders to interact actively, tailored to electricity trading on (inter)national exchanges. Our solution relies on homomorphic encryption, commitments, and zero-knowledge proofs. In a simulated auction with 256 traders, we observe that traders take up to 10 seconds to generate their order, the auctioneer takes 10 seconds to verify an order, and the auction result is computed and verified in 30 seconds. We extrapolate these results to larger auctions to show the practical potential.

CCS CONCEPTS

• **Security and privacy** → **Privacy-preserving protocols**; **Pseudonymity, anonymity and untraceability**; **Cryptography**.

KEYWORDS

Periodic Double Auctions, Privacy, Homomorphic Encryption, Zero-knowledge Proofs

ACM Reference Format:
Armin Memar Zahedani, Jelle Vos, and Zekeriya Erkin. 2023. Practical Verifiable & Privacy-Preserving Double Auctions. In *The 18th International Conference on Availability, Reliability and Security (ARES 2023), August 29–September 01, 2023, Benevento, Italy.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3600160.3600190

ARES 2023, August 29–September 01, 2023, Benevento, Italy

ACM ISBN 979-8-4007-0772-8/23/08…$15.00
https://doi.org/10.1145/3600160.3600190

1 INTRODUCTION

Periodic double auctions are a type of auction where traders take the role of buyers and sellers, submitting quantities they want to buy and sell at different prices. The auction aggregates the demand and supply into curves to find the intersection at which demand equals supply. The price at the intersection is called the market-clearing price and is the price at which traders will trade their products. These kinds of auctions are used in different domains such as electricity trading (merit order model) [21] or sugar beets contract trading [4] and are similar to the Walrasian auction [25]. While presenting an efficient way to trade in the market, typical double auction systems rely on a trusted third party, the auctioneer, to compute the auction result correctly. However, in practice, the auctioneer has opportunities to manipulate the auction result and learn sensitive information about buyers and sellers. For example, the auctioneer may declare an auction result different than the auction procedure would dictate, ignore, or change a trader's offer. There is real-life evidence of such manipulation in general auction procedures [7, 16] and double auctions in electricity trading [8]. In these cases, the auctioneer had a financial incentive to act maliciously. This calls for a way to publicly verify the auction procedure to ensure that the auctioneer computes the result correctly.

In current auction systems, the auctioneer also learns sensitive information about traders. The auctioneer may abuse this information to conduct insider trading, predict the behavior of traders or learn about the consumption of certain goods, such as electricity. Indeed, there is evidence that Danish sugar beet farmers, for example, want their offers to be kept confidential from their buyers [4]. We note that the manipulation example in the electricity trading case was only possible because the auctioneer could identify the trader behind offers. Thus we argue that both pseudonymity and confidentiality of offers are needed to ensure that patterns do not identify traders and protect the sensitive information of traders.

Hence, researchers have constructed schemes that allow participants to verify the correctness of the auction result and preserve participants' privacy and confidentiality of their offers. The schemes use various different network topologies and cryptographic primitives, such as homomorphic encryption or secret sharing. However, to the best of our knowledge, no scheme is both privacy-preserving and allows for verification without traders computing the complete auction result themselves (see Table 1).

In this work, we propose a privacy-preserving verifiable double auction that resembles the ones used in electricity trading. We assume access to a public bulletin board to publish public information and a semi-honest third agent to help with the computations to

produce the auction result. Our scheme ensures that traders do not have to participate in the full protocol except to send their order. At the same time, any entity can verify the auction procedure's correctness with public information on the public bulletin board. Hence, our main contribution is a new scheme to conduct double auctions in a privacy-preserving and verifiable manner that allows traders to verify the correctness of the result without being involved in the protocol computations themselves. We apply our scheme to a double auction used in electricity trading and show that it is realistic with typical levels of security.

The rest of the paper is structured as follows: We explain the cryptographic building blocks in Section 2. In Section 3, we explore related works on privacy-preserving and verifiable double auctions. In Section 4, we explain our scheme for privacy-preserving and verifiable periodic double auctions. We evaluate our scheme in Section 5 and conclude in Section 6.

2 PRELIMINARIES

This section discusses periodic double auctions and relevant cryptographic techniques.

2.1 Auction Systems

Periodic double auctions work by traders submitting quantities they are willing to buy and sell at different prices to the auctioneer. For ease of notation, we assign each trader a unique identifier, so $\mathcal{P}_{id}$, denotes the trader with identifier id. Each buying trader $\mathcal{P}_{id}$ makes one final order B_{id}, which is made up of n offers in the form of quantity-price pairs:

$$B_{id} = \{(q_{id,1}, p_{id,1}), (q_{id,2}, p_{id,2}), \ldots, (q_{id,n}, p_{id,n})\} \ . \tag{1}$$

A selling trader does the same, creating a final order S_{id}. For an order to be valid, the quantities and prices must increase or decrease monotonically, depending on if the trader is a buyer or seller. $\mathcal{B}$ and $\mathcal{S}$ denote the identifiers of buyers and sellers, respectively. However, we note that traders can be in both sets simultaneously and that $\mathcal{B}$ and $\mathcal{S}$ can change between auctions.

The auctioneer aggregates the quantities of buyers and sellers at the same price to form a demand and supply curve for all prices. In practice, these prices are inside a fixed range. For prices where traders did not submit a quantity-price pair, the quantity can be inferred by looking at the quantity of the next lowest price when buying and the next highest price when selling. In this way, the auctioneer builds a step-wise demand and supply curve. The auctioneer then aims to find the intersection between demand and supply, the optimal price to trade the products at. This price is called the market-clearing price (MCP). Buyers willing to buy above the MCP get to trade their products with suppliers willing to sell their products below the price. Figure 1 presents the graphical process of intersecting the aggregated demand and supply curve, which consists of traders' individual demand and supply curves.

We note that it is also possible to evaluate the auction without drawing a graph by comparing demand and supply at various prices and finding the highest price for which demand exceeds supply. To find the MCP, it is essential that quantities descend when buying and ascend when selling for increasing prices to ensure the uniqueness of the MCP.

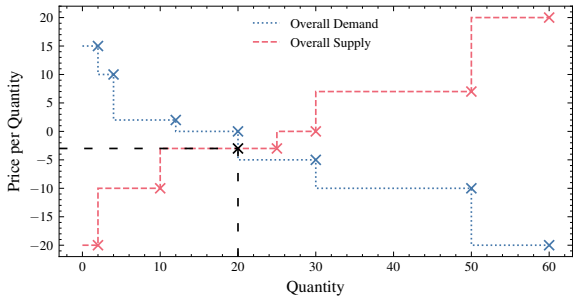

Figure 1: intersection between supply and demand curve

2.2 Cryptographic Building Blocks

We summarize the high-level cryptographic building blocks relevant to our scheme, and mention the instantiations used in our implementation.

2.2.1 Signatures. Our proposed scheme requires digital signatures so only admitted traders can make orders. We also use blind signatures to preserve pseudonymity of the traders. For the former, our implementation relies on Ed25519 [3] as they are computationally-efficient and compact. For the latter, we rely on RSA blind signatures [6], which are conceptually simple.

2.2.2 Additively homomorphic encryption. An additively homomorphic encryption scheme is a public-key cryptosystem for encrypting numbers. It must support an additive homomorphism. In other words, there exists some operation that combines two ciphertexts so that the resulting ciphertext encodes their sum. Our implementation uses two different schemes. It uses the Paillier cryptosystem [12, 17] to encrypt quantities, as it supports a large plaintext space. In places where we only need to check whether an encrypted value is zero or not, the implementation uses elliptic curve-based ElGamal. We use $[m]_{\mathrm{E}}$ to represent encryptions of m.

2.2.3 Additively homomorphic commitments. Apart from encryptions, we also require additively homomorphic commitments. They allow a prover to bind to a value, and convincingly reveal it at a later stage to a verifier. Moreover, the homomorphic property allows commitments to be aggregated in the same way as the encryptions. In our implementation, we use Pedersen commitments [18], a computational binding and information-theoretic hiding commitment scheme. We let $[m]_{\mathrm{C}}$ denote a commitment of value m.

2.2.4 Zero-knowledge proofs. Our scheme requires two different types of arguments that must be proved under zero knowledge. First, traders must prove consistency between the above encryptions and commitments. In other words, such a proof convinces a verifier that an encryption and commitment encode the same message. For this, we use Sigma proofs [20]. We use the Fiat-Shamir heuristic, allowing non-interactive Sigma proofs in the random oracle model that any party can verify.

Secondly, we require range proofs to convince a verifier that a series of committed values is monotonically increasing or decreasing by proving that the difference of consecutive quantities is a positive number in a certain range. Our implementation relies on Bulletproofs [5] for this purpose.

3 RELATED WORK

Several previous works in privacy-preserving auctions already address privacy and correctness concerns of double auctions. In particular continuous double auctions received a great deal of attention as they have the potential to replace the double auctions used in the stock market or other goods. We are concerned with double auctions that are evaluated periodically.

Table 1 provides an overview of solutions for periodic double auctions. We note that for *Anonymity*, a full circle refers to full anonymity, a half circle to pseudonymity, and an empty circle to no anonymity. *Auctioneer* refers to who fulfills the role of the auctioneer in the protocol. Here, `Servers` refers to traders communicating with several servers that fulfill this role. `All` refers to all traders cooperating to fulfill the role of the auctioneer, and `Single` refers to traders communicating with a single server as the auctioneer, and the auctioneer getting help for computations from other servers. *Malicious* refers to whether the protocol considers and protects against malicious behavior of the auctioneer and traders. In the case of the auctioneer, we differentiate whether there are multiple auctioneers (`Servers` and `All`) or a single auctioneer (`Single`). If there are multiple auctioneers, we state how many servers need to be honest (roughly half or not). If there is only one auctioneer, we state whether they can be malicious or not. For inputs of clients, we differentiate based on whether the client input can be malicious or not. Economic rationale refers to the entities acting in their economic interest.

Bogetoft et al. [4] use multi-party computation based on secret sharing to develop a practical double auction. Their scheme uses verifiable secret sharing involving representatives of buyers, sellers, and the research project itself. Traders submit bids and asks representing how much they are willing to buy or sell at all possible prices. The bids and asks are then secret shared among the three servers for aggregation. Each server verifies that their received share is correct by the verification property of verifiable secret sharing. The servers then aggregate the individual shares to construct demand and supply curve shares. The parties compute the market-clearing price using secure comparisons on secret shared values. After traders submit their offers, no interactivity is required (their representatives interact on their behalf), and traders can submit multiple offers. However, the protocol does not allow traders to verify the results independently, and corrupting two out of three parties renders the protocol insecure.

Wallrabenstein et al. [25] propose a privacy-preserving Walrasian auction, closely resembling periodic double auctions. Their protocol uses the Paillier cryptosystem. The basic protocol works by a seller initiating an initial price. The first buyer initializes the demand they are willing to buy at that price and sends it to the subsequent buyers, who add on their demand homomorphically. The final buyer checks the current round and determines whether another round is needed. If another round is needed, the final buyer computes the excess demand and the price update and sends the new price to the first buyer to restart the procedure for a new price. If the protocol reaches the last round, the final buyer sends the demand to the seller. The seller finalizes the protocol by decrypting the price and demand. The authors extend their scheme to consider the malicious behavior of traders by buyers committing to their utility function. However, their scheme only considers a single seller for an item and not multiple sellers selling the same homogeneous item. Furthermore, communication between buyers is required as the coalition of buyers effectively emulates the role of the auctioneer. The interactivity may be expensive regarding communication costs as it may take multiple rounds to find the equilibrium price.

Liu et al. [13] propose BFSDA, a blockchain-based secure double auction protocol. Using secret sharing and Pedersen commitments, they construct an interactive round-based protocol where participants submit the quantity they want to buy or sell at the current price, resembling a Walrasian mechanism. These offers are then secret shared amongst all participants, and each participant verifies that their received share is correct. Each participant then aggregates the received shares and broadcasts this to reconstruct overall demand and supply at the current price. Using the Pedersen commitment, the participants check the consistency of shares. The traders find the market-clearing price using a binary search. However, their scheme requires the interactivity of all participants to resolve the protocol. Furthermore, participants cannot bid on all prices at once but only on the current price in each round, requiring interactivity in all rounds.

Sarenche et al. [19] propose a smart grid electricity trading scheme with low communication overhead and round complexity. Traders get tokens from a trusted control center for different categories of double auctions. In each category, traders with similar demand and supply intend to trade electricity. Traders place offers in a category from which they hold a token. In the first phase, traders send a commitment to the auctioneer and bulletin board, with which they get a fresh pseudonym. Using this pseudonym, traders generate their actual offers, encrypt them with the public key of the auctioneer and send them to the bulletin board and auctioneer, who decrypts the offers. The auctioneer uses the decrypted offers to compute the market-clearing price according to a defined algorithm. While there is a low overhead in communication, the scheme does not protect the confidentiality of offers against the auctioneer nor against traders in the same category (albeit traders are pseudonymous). Furthermore, all traders need to submit offers in each round.

Galal et al. [9] propose a verifiable periodic double auction for dark pools (a stock exchange with a non-public order book), which prevents malicious behavior of traders and the auctioneer. Traders first commit to their bid and later send the same offer, encrypted with the El-Gamal public key of the auctioneer, to the auctioneer. At the same time, traders prove to a smart contract that the commitment hides the same information as the encryption. The auctioneer decrypts all offers, sorts the bids and asks by price, and finds the price that allows traders to trade the highest quantity. The auctioneer then creates new orders that fit into the correctly sorted list of buy and sell orders representing the overall buying and selling that occurs and proves the correctness of these new orders to the smart contract. Hence, their procedure allows for verifying the correctness of the market-clearing price. However, there is no confidentiality of offers towards a central auctioneer, which is arguably especially important in dark pools. Furthermore, their scheme is expensive in terms of the transaction fees required to send transactions to the blockchain due to the cryptographic

Table 1: Existing Double Auction schemes, *Assuming economic rationality.

Work	Properties			Auctioneer	Malicious	
	Confidentiality	Public Verifiability	Anonymous		Auctioneer	Traders
Bogetoft et al. [4]	●	○	○	Servers	◐	●*
Abidin et al. [1]	●	○	○	Servers	◐	○
Wallrabenstein et al. [25]	●	●	○	All	◐	◐
Liu et al. [13]	●	●	○	All	◐	◐
Sarenche et al. [19]	○	●	◐	Single	●	●
Galal et al. [9]	○	●	○	Single	●	●
This work	●	●	◐	Single	●*	●*

operations required for verification (in the order of 10^6 Gwei per auction, around €1-2).

In the context of electricity trading, researchers propose several schemes such as PEM [27] or a novel way to compute inner products [10]. However, these schemes do not tackle the lack of privacy and verification of (inter)national electricity exchanges. The only schemes that tackle both the correctness and verification properties require full interaction in the auction procedure. Moreover, we note that these schemes do not consider how to register traders and how to retain their anonymity.

We note that publicly verifiable or auditable privacy-preserving auctions can also be achieved using the generic scheme from Baum et al. [2] that applies to any multi-party computation protocol. In such a protocol, the role of the auctioneer would be fulfilled by multiple (possibly malicious) servers that perform multiple interactions. However, to the best of our knowledge, the concrete efficiency of such an auction system has not been studied before.

4 OUR SCHEME

This section presents our scheme for privacy-preserving and verifiable double auctions. Apart from the traders and the auctioneer, we consider a non-colluding third agent and a public bulletin board. The auctioneer is still the primary entity that computes the auction result and performs most of the computations. The bulletin board allows any entity to verify the auction result and that the auctioneer included all offers, while the third agent ensures the confidentiality of offers. We assume that both the third agent and the bulletin board are semi-honest and do not collude with the auctioneer. An example of a real-life instantiation for the third agent would be the European Union. Figure 2 provides a high-level overview of an auction. The first half represents the registration procedure, while the second half pertains to an auction.

As explained in Section 2.1, the goal of the auctioneer and the ideal functionality of our scheme is to compute the market clearing price p_j such that:

$$\sum_{\mathcal{P}_b \in \mathcal{B}} q_{b,j} \geq \sum_{\mathcal{P}_s \in \mathcal{S}} q_{s,j} \quad \text{and} \quad \sum_{\mathcal{P}_b \in \mathcal{B}} q_{b,(j+1)} < \sum_{\mathcal{P}_s \in \mathcal{S}} q_{s,(j+1)} \,. \quad (2)$$

All parties receive output p_j, whereas the other inputs stay hidden.

The high-level insight behind our auction procedure is as follows. Traders send homomorphic encryptions of the quantities they want to buy or sell to the auctioneer, along with commitments. Zero-knowledge proofs allow the trader to prove to the auctioneer that the committed quantities are well-formed, and that the encryptions and commitments are equivalent. By collaborating with the third agent, the auctioneer determines the market clearing price without decrypting the quantities but through the use of secure comparisons. We explain the procedure in Figure 3, and provide more details of the sub-routines in the proceeding subsections. We assume that all parties have access to the public key of the third agent $pk_{\mathcal{T}}$, which belongs to an additively homomorphic cryptosystem.

Auction execution from start to finish

(1) A trader $\mathcal{P}_{id}$ signs up with the third agent $\mathcal{T}$ and finishes registration with the auctioneer $\mathcal{A}$ using the protocol by Wang et al. [26]. This yields a token t_{id} signed by $\mathcal{T}$ and $\mathcal{A}$, and a pseudonymous key pair (sk_{id}, vk_{id}).

(2) A trader $\mathcal{P}_{id}$ submits its order(s) $[B_{id}]$ and/or $[S_{id}]$, encrypted using $pk_{\mathcal{T}}$ to $\mathcal{A}$ following (3). It also submits corresponding commitments to the bulletin board following (4). The trader includes signatures using its signing key sk_{id} and zero-knowledge proofs to show that the commitments match the encryptions (5) and the bids are well-formed (6), (7).

(3) The auctioneer $\mathcal{A}$ verifies the orders and aggregates them using Alg. 1. Buyers increase the total demand, while sellers increase the total supply.

(4) Given the encrypted aggregated supply and demand, $\mathcal{A}$ and $\mathcal{T}$ compute the MCP p_j using $\lceil \log_2 M \rceil$ secure comparisons. The auctioneer reveals the encryptions and consistency proofs at the MCP.

(5) The third agent $\mathcal{T}$ verifies the consistency proofs, decrypts the quantities at the MCP $[q_{id,j}]$ using its secret key, and sends the decrypted winning quantities $q_{id,j}$ to the auctioneer $\mathcal{A}$.

Figure 3: One auction execution from registration to revealing the MCP.

Our scheme provides the following properties: *confidentiality*, *public verifiability*, *pseudonymity*, *unforgeability*, *traceability*, and *non-repudiation* [19].

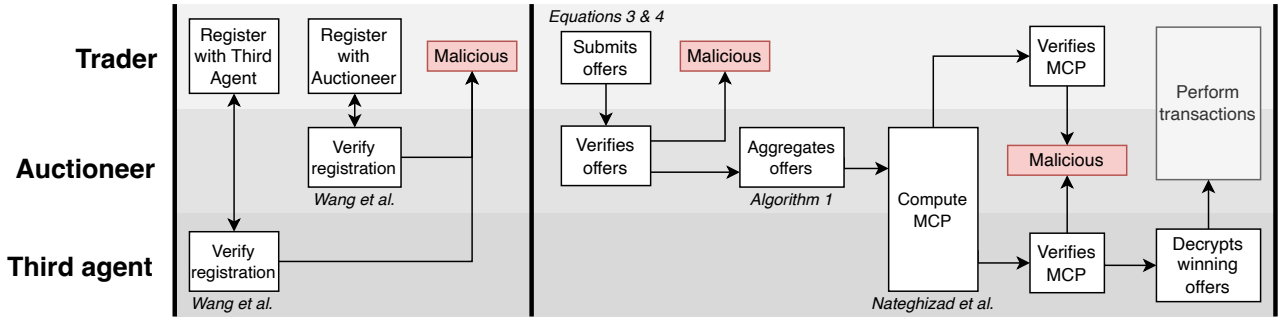

Figure 2: High-level overview of the registration & auction procedure. T is a trader, A is the auctioneer, and TA is the third agent. Note that the registration procedure only has to be executed once for each trader.

4.1 Registration

In the registration procedure, traders interact with the third agent and the auctioneer to receive a token containing a new pseudonym to use in upcoming auctions. We note that the registration procedure only needs to be run once for each trader but can also be run before every auction period to achieve full anonymity until winning the auction. The registration procedure consists of the trader first interacting with the third agent and then with the auctioneer and is based on the registration procedure of Wang et al. [26]. We note that the registration procedure can be replaced with a registration procedure with the same properties but different approach, for example using zero-knowledge proofs. In particular, the registration procedure needs to provide a token to achieve pseudonymity, enforceability, traceability and non-repudiation. In our case, this is achieved by the third agent blindly signing on a pseudonym but having a copy of the real-identity of the trader, and the auctioneer signing on the pseudonym of the trader.

New traders interact with the third agent to register in the first registration step. We use the cut-and-choose technique to protect against malicious traders. First, a trader $\mathcal{P}_{id}$ creates λ pseudonyms and public-private keypairs. Only one of those pseudonyms and keypairs will be used in the end. The third agent $\mathcal{T}$ asks to unblind $\lambda - 1$ of them to verify the correctness of the structure. If all of them are of the correct structure, $\mathcal{T}$ will blindly sign the remaining one to create a signature of the third agent on the pseudonym and public key. In particular, we use Ed25519 signature keys in this part of the protocol for the keys traders generate due to fast key generation to support large λ. In the second step of the registration, new traders interact with the auctioneer. The protocol remains unchanged from the original solution by Wang et al. After the registration step, traders have a pseudonym and a public-private key pair that they will use to submit their offers. In particular, each trader $\mathcal{P}_{id}$ has a pseudonym, a public-private key pair, and the signatures of the third agent and auctioneer on these, which represent a token t_{id} to use in the auctions. The trader can only cheat with probability λ^{-1}.

4.2 Submitting offers

In a new auction period, traders may decide to submit offers to participate in the auction. This stage allows traders to submit an order, as well as providing confidentiality and necessary structures for the public verifiability. Let M be the maximum price a trader can offer: Traders will first decide on multiple quantity-price pairs $(q_1, p_1), \ldots, (q_n, p_n)$ they want to submit to the auction. They then take the separate quantities and encrypt them with public key $pk_{\mathcal{T}}$. At the same time, traders create a similar-looking order but do not encrypt quantities $q_1, \ldots, q_n$ but commit to them using an additively homomorphic commitment scheme, which are sent to the bulletin board. Hence a buying order for the auctioneer $[B_{id}]$ and bulletin board $[B_{id}]$ look as follows:

$$[B_{id}]_{\mathrm{E}} \leftarrow \left(([q_{id,1}]_{\mathrm{E}}, p_{id,1}), \ldots, ([q_{id,n}]_{\mathrm{E}}, p_{id,n})\right) , \tag{3}$$

$$[B_{id}]_{\mathrm{C}} \leftarrow \left(([q_{id,1}]_{\mathrm{C}}, p_{id,1}), \ldots, ([q_{id,n}]_{\mathrm{C}}, p_{id,n})\right) . \tag{4}$$

A selling order looks similar, but we write S_{id}.

Traders also prove that their commitment hides the same value as their encryption. In our implementation we do so with a Zero-knowledge proof (ZKP) of consistency by Jurik [11]. In essence, traders use a Sigma protocol and apply the Fiat-Shamir heuristic to create a proof for the auctioneer that the encryption hides the same value as the commitment. The final proof of trader $\mathcal{P}_{id}$ is a collection of individual proofs:

$$\mathrm{ZKP}([q_{id,1}]_{\mathrm{E}}, [q_{id,1}]_{\mathrm{C}}), \ldots, \mathrm{ZKP}([q_{id,n}]_{\mathrm{E}}, [q_{id,n}]_{\mathrm{C}}) . \tag{5}$$

The auctioneer verifies the proof and then decides to accept or deny the order. Of course, there is nothing preventing the auctioneer to deny a correct order. However, we believe this is not an issue since traders are behind a pseudonym, and their quantities are encrypted. Hence, the auctioneer cannot ignore the orders of specific traders. Moreover, it is not in the economic interest of the auctioneer to lower the trade volume.

The last step is for traders to create range proofs (RP) on their committed quantities. In our case, we use bulletproofs [5]. The range proof on the commitments combined with the zero-knowledge proof of consistency together prove that the encrypted quantities are inside a specific range, which the auctioneer uses to ensure that the auction result is unique. Traders not only prove that each of their offers is inside a specific range (e.g., 32-bit unsigned integer) but also prove that their offers are increasing for ascending prices when selling and descending when buying, which is essential for the uniqueness of the market-clearing price. We use the same technique as Galal et al. [9], by creating range proofs on the difference of successive quantities with the corresponding difference of

randomness used. The commitments of the differences are created using the homomorphic property of the commitments. Each trader $\mathcal{P}_{id}$ submits the following collection of individual proofs:

$$\mathrm{RP}([q_{id,1}]_{\mathrm{C}}), \ldots, \mathrm{RP}([q_{id,n}]_{\mathrm{C}}) \text{ and} \tag{6}$$

$$\mathrm{RP}([q_{id,2} - q_{id,1}]_{\mathrm{C}}), \ldots, \mathrm{RP}([q_{id,n} - q_{id,n-1}]_{\mathrm{C}}) \,. \tag{7}$$

Traders also provide their token T. They sign all information using their signing key sk_{id}.

The hiding property of the encryptions and commitments along with the zero-knowledge property provides *confidentiality* of all offers and *non-repudiation*. The pseudonymous signing guarantees both *pseudonymity* and *unforgeability*. The zero-knowledge proofs that are shared to a public bulletin board ensure that the auction is *publicly verifiable* and they enable *traceability*.

4.3 Aggregation

The auctioneer $\mathcal{A}$ receives the orders and queries the public bulletin board for the corresponding commitments. As mentioned before, $\mathcal{A}$ then verifies the signatures, consistency proofs, and range proofs to ensure the correctness of the auction procedure. After that, it homomorphically adds the quantities for each possible price. For prices where the trader provided no quantity, the auctioneer infers the quantity as described in Section 2 and presented in Algorithm 1. The result is a curve that resembles a step-function.

Algorithm 1 Aggregates the offers of $\mathcal{P}_{id}$, where a is the current demand or supply and $\{[q_{id,j}]_{\mathrm{E}}, p_{id,j} \mid \forall j\}$ are quantity-price pairs

```
1: procedure AGG(([a_0]_E, ..., [a_n]_E), M, {[q_{id,j}]_E, p_{id,j} | ∀j})
2:     j ← 1
3:     for P = 0, ..., M do
4:         if P > p_{id,j} then
5:             j ← j + 1
6:         [a_P]_E ← [a_P + q_{id,j}]_E
```

Since all quantities are encrypted with the public key of the third agent, the auctioneer cannot infer any of the quantities in an order during the aggregation process.

4.4 Compute market-clearing price

After the auction period closes, the auctioneer must compute the market-clearing price with the encrypted quantities. Here, the auctioneer and the third agent cooperate in running a binary search. Given encrypted quantities at all prices, the auctioneer compares the encrypted demand and supply at a given price to decide whether to increase the price or decrease it. This is repeated in a binary search until the auctioneer finds the highest price for which demand $\geq$ supply. This is a variant of the millionaire's problem where the auctioneer holds values a, b encrypted with the third agent's public key, and the agent holds the corresponding private key. The auctioneer wants to compute $a > b$ without either party learning the individual values. We use a protocol by Nateghizad et al. [15] to realize this behavior. Instead of DGK, we use elliptic curve-based ElGamal for fast public-key operations and key generation. Hence, after the binary search, the auctioneer finds the highest price for which demand $\geq$ supply and knows which quantities traders will trade.

We note that while the result of the comparisons is exposed to the auctioneer, this does not reveal any additional information that could not be deduced from the market clearing price: Since the supply and demand curves are monotonic, being the inflection point, the market clearing price already reveals the result of all other comparisons.

4.5 Identify Winners

Given the market clearing price, the auctioneer releases the orders made at that price and the subsequent price to the public. The public verifies that all orders on the bulletin board have a corresponding zero-knowledge proof of consistency that the auctioneer releases. For all commitments made at the MCP p_j, the auctioneer presents the encrypted offers that won $[q_{id,j}]_{\mathrm{E}}$ for all id, as well as the zero-knowledge proofs. This ensures that the auctioneer did not tamper with orders after having found the correct market-clearing price, since the commitments on the bulletin board are immutable. The third agent then decrypts all the offers made at the market clearing price, and traders step forward to engage in the transfer of goods. If traders do not step forward, the auctioneer and the third agent cooperate to reveal the identity behind a pseudonym [26]. Winners then engage in the transfer of goods. We argue that the leakage of the subsequent quantities to verify the MCP is permissible as these quantities are extremely close to the actual quantities in practical scenarios.

5 EVALUATION

Next, we evaluate our scheme's performance, communication cost, and security.

5.1 Communicational Complexity

The communicational complexity is the size of the messages that entities send each other. Let the number of offers in an order be denoted by n and the cut-and-choose parameter be denoted by λ. The message size in the registration procedure for traders with the third agent is $0.544\lambda - 0.096$ KB, and with the auctioneer, 0.928 KB. For traders sending orders, the communication size is $4.324n - 0.610$ KB. Hence, for parameters such as $n = 256$, $\lambda = 4096$, we have communication costs of $\approx$ 2228KB, 0.928KB and $\approx$ 1106 KB, respectively for the trader. These correspond to registering with the third agent, registering with the auctioneer, and sending an order to the auctioneer.

5.2 Runtime

We implement our scheme in Rust, using various libraries [14, 22–24]. In our proof-of-concept, each entity is assigned a single thread and communicates over channels without delays. We use security parameters equivalent to AES-128 security. Hence, we use a Paillier and RSA modulus of 3072 bits. Ed25519 and elliptic curve-based ElGamal provide AES-128 security by default. We evaluate our implementation on a machine with an Intel Core i7-7700HQ CPU and 16GB RAM. For each step of the scheme, we present the evaluation times. The largest setting we test consists of 256 traders submitting 256 offers in their order and 350,000 possible prices. We choose

these parameters as they are close to realistic settings [21]. Since we evaluate on a single machine, the runtime increases linearly for the number of traders generating offers; hence these results can be extrapolated to larger numbers of traders. We note that our implementation is not guaranteed to run in constant time, and the results may slow down when considering constant time code. Experiments showed that RSA signatures especially suffer from this, where we observed a 2x increase in run time for RSA blind signatures.

5.2.1 Asymptotic. Asymptotically, we note that the registration between traders and the third agent takes $O(\lambda)$ time, where λ is the cut-and-choose parameter chosen. For the trader and the auctioneer, it is $O(1)$ time. For creating an order, traders spend $O(n)$ time depending on the number of offers n. The auctioneer also spends $O(n)$ time verifying a single order. Hence, when m traders send orders, the auctioneer spends $O(mn)$ time to verify all orders. Finally, computing the market-clearing price takes $O(1)$, and verifying the consistency with the bulletin board takes $O(m)$ time.

5.2.2 Preparation. Table 2 presents the mean key generation time for all keypairs needed. Even though both elliptic curve-based ElGamal and Ed25519 use the same curve, ECEG is slightly slower due to creating a multiplication table for faster encryption. The public keys of the third agent need to be distributed to all parties.

Table 2: Mean key generation time for keys with 128-bit security over 10 runs.

Entity	Key	Time
Trader	Ed25519	**0.05** ± 0.03 ms
Auctioneer	Ed25519	**0.05** ± 0.03 ms
Third Agent	RSA	**35.44** ± 23.24 s
	Paillier	**29.52** ± 11.93 s
	ECEG	**1.05** ± 0.04 ms

5.2.3 Registration. We present the times for each entity to run the registration procedure in Figure 4. In particular, we test the registration for an increasing security parameter λ that controls the number of rounds in the cut-and-choose step. We observe that the time for registration depends on the parameter λ. For a low λ, registration takes less than 100ms, where a significant amount of time is spent between the trader and the third agent. This is because of the cut-and-choose involved. For a high λ, traders can take up to 500ms to run the cryptographic operations and verifications. We note that the interaction between the trader and auctioneer is unaffected by λ and is below 1ms.

5.2.4 Preparing offers. Traders prepare and send their offers to the auctioneer and bulletin board in the bidding stage. Here, the runtime for each trader is affected by the number of offers in an order. We test for several different offers in an order for traders, reporting on the time for the individual cryptographic structures. We present the runtime in Figure 5. We see that as the number of offers in an order increases, the runtime also increases. For a single offer, traders need a few milliseconds to generate the structures related to range proofs. However, we need 50ms for the Paillier

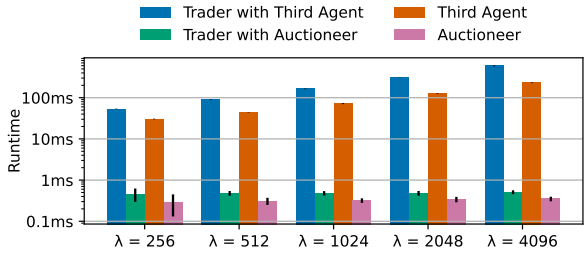

Figure 4: Runtime of registration for traders, third agent, and auctioneer for increasing rounds in cut-and-choose. Vertical black lines represent the standard deviation.

encryption and consistency proof. The runtime increases when a trader submits 256 offers, where it takes roughly a second to generate the range-proof structures but 10 seconds to generate the structures around Paillier ciphertexts.

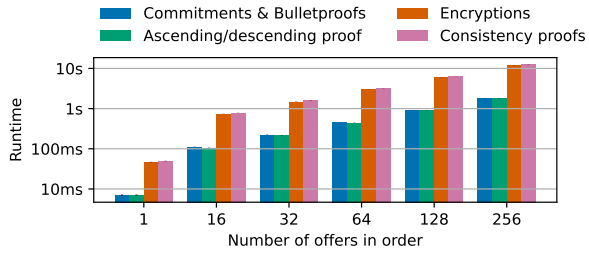

Figure 5: Runtime for traders generating cryptographic structures for an increasing number of offers in an order. Vertical black lines represent the standard deviation.

5.2.5 Receiving offers. Figure 6 presents the runtime for the auctioneer verifying the orders for an increasing number of offers in the order. We observe that verifying a single consistency proof takes 54ms while verifying the signatures and range proofs only takes 1-2ms. For 16 offers, the verification takes around 1s, while it takes more than 10s for 256 offers. Most time is spent verifying the consistency proofs, which takes 10s for 256 offers, while the other structures take less than 1s. The subsequent homomorphic aggregation takes less than 50ms for 256 orders.

5.2.6 Compute winners. The search for the market-clearing price depends on the number of possible prices M. In our case, this means $\lceil \log_2(350000) \rceil = 19$ comparisons. A single comparison takes roughly 500ms; hence, finding the market-clearing price takes approximately 10s, irrespective of the number of orders traders submit. The auctioneer then forwards the offers made at the market-clearing price to the third agent to verify correct behavior. The third agent checks the consistency between the bulletin board and the offers and decrypts the offers made. Figure 7 presents the time for the third agent to decrypt the offers and calculate the consistencies. For 256 orders, it takes roughly 10s to verify and decrypt the

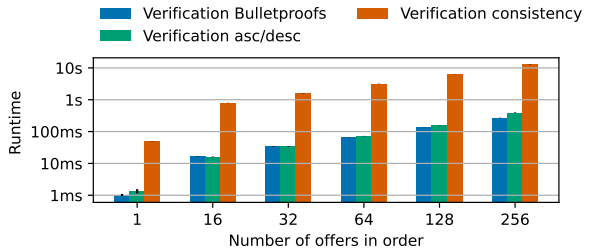

Figure 6: Runtime for auctioneer verifying a single order with an increasing number of offers. Vertical black lines represent the standard deviation.

offers. Since the evaluation occurred on a single machine, we did not experiment with more traders, as that would require more registrations, preparation of offers etc. However, we extrapolated the results to see that for the third agent, it would take less than 60 seconds to decrypt and verify the results, for over 1024 orders/traders participating in the auction.

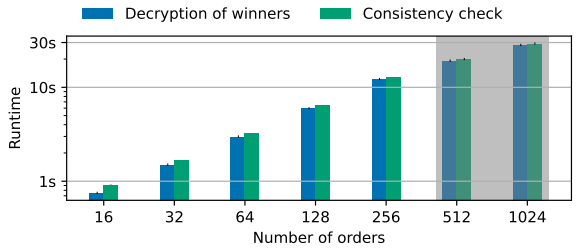

Figure 7: Runtime for the third agent to verify the auction result. Grayed-out bars represent interpolations of the run time.

5.3 Security

The proposed scheme inherits its properties directly from established subroutines, so we refrain from a formal security proof. In particular, the registration procedure from Wang et al. [26] provides pseudonymity and traceability. Traders are behind a pseudonym created by themselves, but the auctioneer and third agent cooperating can trace their identity. The registration procedure also ensures that no one can frame traders to be behind a pseudonym due to the pre-image resistance of cryptographic hash functions. The confidentiality of quantities is implied by the semantic security of the Paillier cryptosystem, as we assume no collusion between the auctioneer and the third agent. Moreover, Pedersen commitments and zero-knowledge proofs provide no additional knowledge of these secrets. We note that only the quantities are encrypted, and the prices are in plaintext, which leaks information about the prices. If the price points that traders bid on are considered sensitive, the trader might insert dummy offers, for example, zero at low/high prices or any quantity present on traders' demand/supply curve.

The verification property ensures that no offers were excluded after being accepted on the bulletin board by the auctioneer. After an order is on the bulletin board, the auctioneer has to provide zero-knowledge proofs of consistency that are consistent with encryptions of the same order, which cannot be forged without knowledge of the quantities, due to the soundness property of the Zero-knowledge proofs. The final price is verified by verifying the consistency at the market-clearing price and the subsequent price to ensure that no better price maximizes the quantity to trade. Finally, we achieve unforgeability and non-repudiation with the Ed25519 signatures attached to the orders.

6 CONCLUSION

In conclusion, we propose to the best of our knowledge, the first periodic double auction protocol that is both privacy-preserving and verifiable without traders fulfilling the role of the auctioneer. We design a solution using various cryptographic techniques, such as homomorphic encryption, zero-knowledge proofs, and commitment schemes and introduce new entities, such as a third agent and a bulletin board. Depending on the security parameter λ, registration of traders takes around 500ms with the highest parameter tested $\lambda = 4096$. When traders submit offers, they generate Paillier ciphertexts, Pedersen commitments, bulletproof range proofs, and zero-knowledge proof of consistencies. The commitments are sent explicitly to the bulletin board, while traders sends the other structures to the auctioneer. The bulletin board allows anyone to verify the correctness of the offers considered. The auctioneer and third agent then cooperate to find the market-clearing price and the winners, and the third agent verifies the correctness of the winning offers. We show that traders require less than 10 seconds to create the structures on consumer hardware, and the auctioneer requires the same amount of time to verify these structures. Once all offers have been submitted and verified, the market-clearing price and winners are computed in less than 30 seconds for 256 traders. Using our scheme, we provide the properties of confidentiality, verifiability, pseudonymity, unforgeability, traceability, and non-repudiation, with traders not being involved in the computation of the auction. The main disadvantages of our scheme are the assumptions we made, such as the economic rationality of the auctioneer and that the entities are non-colluding. Another assumption comes from leaking the two prices around the market-clearing price, which are used to verify it.

REFERENCES

[1] Aysajan Abidin, Abdelrahaman Aly, Sara Cleemput, and Mustafa A. Mustafa. 2016. An MPC-Based Privacy-Preserving Protocol for a Local Electricity Trading Market. In *Cryptology and Network Security*, Sara Foresti and Giuseppe Persiano (Eds.). Springer International Publishing, Cham, 615–625. https://doi.org/10.1007/978-3-319-48965-0_40

[2] Carsten Baum, Ivan Damgård, and Claudio Orlandi. 2014. Publicly Auditable Secure Multi-Party Computation. In *Security and Cryptography for Networks*, Michel Abdalla and Roberto De Prisco (Eds.). Springer International Publishing, Cham, 175–196.

[3] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. 2011. High-Speed High-Security Signatures. In *Cryptographic Hardware and Embedded Systems – CHES 2011*, Bart Preneel and Tsuyoshi Takagi (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 124–142.

[4] Peter Bogetoft, Dan Lund Christensen, Ivan Damgård, Martin Geisler, Thomas Jakobsen, Mikkel Krøigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter, Michael Schwartzbach, and Tomas Toft. 2009. Secure Multiparty Computation Goes Live. In *Financial Cryptography and Data Security*,

Roger Dingledine and Philippe Golle (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 325–343.

[5] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. 2018. Bulletproofs: Short Proofs for Confidential Transactions and More. In *2018 IEEE Symposium on Security and Privacy (SP)*. 315–334. https://doi.org/10.1109/SP.2018.00020

[6] David Chaum. 1983. Blind Signatures for Untraceable Payments. In *Advances in Cryptology*, David Chaum, Ronald L. Rivest, and Alan T. Sherman (Eds.). Springer US, Boston, MA, 199–203.

[7] Gilad Edelman. 2021. Google's Alleged Scheme to Corner the Online Ad Market. https://www.wired.com/story/google-antitrust-ad-market-lawsuit/

[8] European Commision. 2021. Antitrust: Commission opens investigation into possible anticompetitive behaviour by the power exchange EPEX Spot. https://ec.europa.eu/commission/presscorner/detail/en/ip_21_1523

[9] Hisham S. Galal and Amr M. Youssef. 2021. Publicly Verifiable and Secrecy Preserving Periodic Auctions. In *Financial Cryptography and Data Security. FC 2021 International Workshops*, Matthew Bernhard, Andrea Bracciali, Lewis Gudgeon, Thomas Haines, Ariah Klages-Mundt, Shin'ichiro Matsuo, Daniel Perez, Massimiliano Sala, and Sam Werner (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 348–363. https://doi.org/10.1007/978-3-662-63958-0_29

[10] Turabek Gaybullaev, Hee-Yong Kwon, Taesic Kim, and Mun-Kyu Lee. 2021. Efficient and Privacy-Preserving Energy Trading on Blockchain Using Dual Binary Encoding for Inner Product Encryption. *Sensors* 21, 6 (2021). https://doi.org/10.3390/s21062024

[11] Mads J. Jurik. 2003. *Extensions to the Paillier Cryptosystem with Applications to Cryptological Protocols.* Ph. D. Dissertation. University of Aarhus.

[12] Jonathan Katz and Yehuda Lindell. 2014. *Introduction to Modern Cryptography, Second Edition* (2nd ed.). Chapman & Hall/CRC.

[13] Lietong Liu, Mingxiao Du, and Xiaofeng Ma. 2020. Blockchain-Based Fair and Secure Electronic Double Auction Protocol. *IEEE Intelligent Systems* 35, 3 (May 2020), 31–40. https://doi.org/10.1109/MIS.2020.2977896

[14] Isis Lovecruft. 2020. ed25519-dalek. https://crates.io/crates/ed25519-dalek

[15] Majid Nateghizad, Zekeriya Erkin, and Reginald L. Lagendijk. 2016. An efficient privacy-preserving comparison protocol in smart metering systems. *EURASIP Journal on Information Security* 2016, 1 (2016), 11. https://doi.org/10.1186/s13635-016-0033-4

[16] OECD. 2016. Preventing Corruption in Public Procurement. https://www.oecd.org/gov/ethics/Corruption-Public-Procurement-Brochure.pdf

[17] Pascal Paillier. 1999. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology — EUROCRYPT '99*, Jacques Stern (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 223–238. https://doi.org/10.1007/3-540-48910-X_16

[18] Torben Pryds Pedersen. 1992. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In *Advances in Cryptology — CRYPTO '91*, Joan Feigenbaum (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 129–140.

[19] Roozbeh Sarenche, Mahmoud Salmasizadeh, Mohammad Hassan Ameri, and Mohammad Reza Aref. 2021. A secure and privacy-preserving protocol for holding double auctions in smart grid. *Information Sciences* 557 (2021), 108–129. https://doi.org/10.1016/j.ins.2020.12.038

[20] Berry Schoenmakers. 2022. Lecture Notes Cryptographic Protocols. https://www.win.tue.nl/~berry/CryptographicProtocols/LectureNotes.pdf

[21] Devnath Shah and Saibal Chatterjee. 2020. A comprehensive review on day-ahead electricity market and important features of world's major electric power exchanges. *International Transactions on Electrical Energy Systems* 30, 7 (2020), e12360. https://doi.org/10.1002/2050-7038.12360

[22] Henry de Valence. 2021. Bulletproofs. https://crates.io/crates/bulletproofs

[23] Henry de Valence. 2021. curve25519-dalek-ng. https://crates.io/crates/curve25519-dalek-ng

[24] Jelle Vos. 2022. Scicrypt. https://crates.io/crates/scicrypt

[25] John Ross Wallrabenstein and Chris Clifton. 2014. Privacy Preserving Tâtonnement. In *Financial Cryptography and Data Security*, Nicolas Christin and Reihaneh Safavi-Naini (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 399–416. https://doi.org/10.1007/978-3-662-45472-5_26

[26] Changjie Wang and Ho-fung Leung. 2004. Anonymity and security in continuous double auctions for Internet retails market. In *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the.* 10 pp.–. https://doi.org/10.1109/HICSS.2004.1265431

[27] S. Xie, H. Wang, Y. Hong, and M. Thai. 2020. Privacy Preserving Distributed Energy Trading. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE Computer Society, Los Alamitos, CA, USA, 322–332. https://doi.org/10.1109/ICDCS47774.2020.00078