**Conference Paper for Hybrid Auction**

**Abstract**

Auctions are one of the most famous market mechanisms for collecting the market value of any good. In open-cry auctions, each bidder's bid is known to all auction participants, causing various issues, such as bid repudiation, false valuation, and bid privacy. In contrast, bids are sealed and submitted to the auctioneer in sealed-bid auctions. Only the auctioneer opens the bid and declares the highest bid as the winner. Unlike open-cry auctions, sealed-bid auctions do not allow bidders to change bids. Furthermore, auction platforms are generally centralized and managed by a third party, raising various issues such as no transparency, the possibility of tampering, no data integrity, no guarantee of privacy, etc. In this paper, we propose a blockchain-based hybrid privacy-preserving auction that preserves the privacy of the bid values and allows a bidder to change the bid valuation when finding that someone else is the highest bidder. While providing the opportunity to raise the bid, the proposed auction never discloses the highest bid to other bidders, giving security against bid temptation. The proposed auction achieves the benefits of both open-cry and sealed-bid auctions. Blockchain makes the proposed auction transparent, secure, and reliable. Theoretical analysis proves that in the proposed auction, the auctioneer cannot know the bid value, and bidders' collusion leads to a distrust situation, i.e., bidders cannot verify the truthfulness of the shared information. The behaviour of the proposed auction is tested in the Ethereum platform, and a cost analysis is given, demonstrating the practical feasibility of the proposed auction.

**1 Introduction**

Auctions are the best way to gather the actual market value of the product, and rich literature proves it. E-auctions (electronic auctions) have brought traditional auctions to a broader level, and various other domains, such as electricity trading, spectrum allocation, cloud computing, etc., have adopted auctions to gather a reasonable valuation of the auctioned product in the market. Auctions are majorly categorized as open-cry auctions and sealed-bid auctions. In open-cry auctions, each participant knows the bid value of each bidder in its true form. Open-cry auctions provide an open market where a losing bidder knows the highest bid value and can change its bid to win the auction. Open-cry auction leads bidders to tempted bidding to beat the winner. Sealed-bid auctions are proposed to prevent the temptation to raise the bid value. In sealed-bid auctions, bidders submit the sealed bids, i.e., encrypted bids, to the auctioneer, and only the auctioneer opens the bid and declares the winner after comparing the bid values.

E-auction platforms [1], acting as auctioneers, provide services to conduct online auctions and allow the auction of anything, providing ease to sellers and buyers. Various leading companies, such as, … Amazon EC2, are reported to use auctions for resource purchasing and selling. No physical presence of bidders and geographical restrictions make e-auction convenient for everyone. E-auction platforms reduce time and cost for sellers and buyers compared to traditional auctions. It is believed that these platforms ensure the required security and privacy and do not misuse the data of bidders. However, incidents have been reported where the e-auction platform managed by a third party maliciously used the collected data related to the auction to get some financial incentive [2]. The e-auction platform may sell data related to bids for a particular good to sellers. Sellers may set a high reserved price for the good in the auction or fix the price in the market based on the data to earn more. No guarantee of privacy of bids is a major challenge with e-auction platforms.

In literature, various privacy-preserving auctions (PPA) [2] are reported where the privacy of the bids was secured in sealed-bid auctions. Sealed bids and privacy-preserving auctions require the submission of all the bid values in one round, and based on these values, the auctioneer declares the results. Even though the earlier privacy and tempted bidding issues have been resolved, these auction mechanisms do not allow bidders to change their bid valuation after knowing they are not the highest bidders. A bidder may win the auction if allowed to change the bid, knowing that the bidder is not the highest bidder. A major disadvantage of e-auction platforms is the use of a centralized system where information related to bidders is stored and managed, and centralized management systems are vulnerable to failure, data loss, and manipulation [2]. Furthermore, the platform may also influence the auction results.

From the above discussion auction, we identified four issues in the auction: (i) Sealed-bid auctions do not allow a bidder to modify its submitted bid, unlike open-cry auctions, (ii) Open-cry auctions encourage tempted bidding, (iii) E-auction platforms may misuse the data related to the auction if privacy is not preserved, and (iv) Centralized e-auction platforms lead to single-point failure and may change the allocation results. To address these four issues, we propose a blockchain-based hybrid privacy-preserving auction in this work. The proposed auction scheme uses secure multiparty computation (SMPC), homomorphic encryption (HE), and blockchain. SMPC is an excellent tool with a lower overhead than other privacy-preserving techniques, such as garbled circuits [3]. HE, on the other hand, can perform mathematical operations on ciphertext, which again gives us the advantage of sharing the secure information among the unsecured channel and getting the required operational output on that data without disclosing the real information. Besides this, HE requires lower computational overhead than other secret-sharing techniques.

Blockchain provides two fundamental properties necessary for a privacy-preserving auction, i.e., trust over each transaction done during the whole auction procedure and verifiability of the auction results. Another great advantage of blockchain platforms such as Ethereum [4] is smart contracts, self-executing codes based on predetermined conditions. If conditions are met, they are self-executed. These smart contracts run over the blockchain and are useful in designing blockchain-based auction schemes. We have used two such smart contracts to complete our proposed hybrid auction scheme. This blockchain-based hybrid auction scheme is a novel contribution that allows the bidders to change the bid value they previously submitted after knowing they are not the highest bidder while maintaining all the necessary properties of a privacy-preserving auction.

In the proposed auction, bidders who are not the highest bidder can change their bid valuation. However, no bidder can gather the true bid value of the highest bidder, which prevents tempted bidding. During the whole process of the auction, no actual bid value is revealed to any of the auction participants, and the auction preserves the privacy of the losing bids. Our proposed hybrid privacy-preserving auction ensures that the auctioneer can never gather any true bid valuation of any bidder during or after the auction except the highest and second highest bidder. After the end of the auction, for transparency and verification of the highest bid value, the highest and second-highest bidder have to reveal their bid values. The collusion among bidders does not impact the auction results, and such collusion always results in a distrustful situation. We provide the concept of distrust among the colluding bidders so that one colluding bidder can never be sure whether it is getting correct input from the other. Thus, distrust can be seen as a significant advantage, which is an emotional factor among the colluding agents to never believe in their other colluding partner. We also claim that in our auction scheme, bid anonymity, non-repudiation, and bid privacy are maintained by each participant during the procedure. After the auction, the privacy of the losing bids is preserved. The major contributions of the work are summarized as follows:

1. The auction allows a bidder to revise its bid if it is not the highest bidder.
2. The auction does not reveal the highest bid value, preventing tempted bidding.
3. The auction preserves the privacy of losing bids.
4. The auction does not require centralized management, preventing single-point failure.
5. The auction implements auction rules in smart-contracts, preventing the change in auction results.
6. The auction is tested by implementing it using Ethereum blockchain platform.

The remaining paper is organized as follows. A literature review is given in section 2, and the preliminaries are discussed in section 3. The proposed auction is discussed in section 4. Threat model and security analysis are given in section 5. Section 6 discusses the experimental findings of our proposed hybrid auction scheme. Section 7 concludes our work.

**2 Literature Review**

Auctions are the most used method to gather the true market value of any product, and various variants of auctions have been proposed as a solution to various issues related to auctions. Open cry auction is the oldest method, which suffers from false valuation, bid repudiation, bid privacy, tempted bidding, etc. Sealed bid auctions address the associated issues with open-cry auctions. In sealed bid auctions, closed bids are submitted to the auctioneers, and the auctioneer opens the bid and makes the highest bidder the winner. Sealed bid auctions have two variants, first-pricing and second-pricing auctions. In the first pricing auction, the highest bidder pays its bid value to the seller. In the second pricing, the highest bidder pays the second highest bid value to the seller. William Vickrey [6] proposed the second-pricing auction, also known as the Vickrey auction. A comparison study between open cry and sealed bid auctions based on the U.S. Forest Service timber auctions has been presented in [5]. In sealed bid auctions, bids are open to the auctioneers. Therefore, auctioneers must be trusted to behave honestly to prevent the misuse of losing bids. However, this assumption does not hold in real life. Auctioneers may behave maliciously and try to use losing bidders’ data for their own profit. Therefore, sealed bid auctions suffer from the privacy issue of losing bids. To address privacy issues associated with sealed bid auctions, privacy-preserving auctions have been proposed.

To ensure bid privacy, many works used the concept of decentralization. For example, in [7], the authors proposed a multiple-round-based auction protocol with the concept of multiple auctioneers. The work defined a threshold, and if the number of auctioneers colluding is less than the threshold, then the system is secure. The work also used secure multiparty computation. The winning price and winning bidders are revealed after the auction. The authors in [8] used ElGamal encrypted bidder-generated bidding vectors consisting of bids encrypted by a public key that is generated by a trusted third party. For each bid, bidders submit a differential of their bid values as proof of their casted bid. The auctioneer publicly computes the integrals of these differential bids submitted by the bidders to verify the bids. Later, the auctioneer declares the winner. In this protocol, only the highest bid is revealed to the auctioneer after completion of the auction. Both works, [7] and [8], require that authorities do not collude. Both works do not allow bidders to revise their bids.

In previous works, [7] [8], cryptography was used to gain trust and ensure the privacy of losing bids, and for decentralization, trust has been distributed among multiple authorities. Later, many works considered blockchain in auctions to ensure trust, bid privacy and transparency. In [9], [10], the authors proposed the privacy-preserving auction scheme over blockchain, which ensures privacy through zero-knowledge proofs and uses these proofs to verify the auction results off-chain. Authors in [11] proposed SBRAC, a blockchain-based decentralized reverse auction using zero-knowledge proofs, providing verifiability to the auction. The auction is collusion-resistant if at least one of the bidders remains honest. The proposed work ensures that a malicious bidder cannot claim itself the winner because of Pederson scheme [reference]. This work requires bidders to submit data multiple times, increasing communication overhead. The smart contract replaces the auctioneer in [11], considering that smart contracts will ensure trust. However, the buyer deploys the smart contract; thus, the trust shifts to the buyer. If the buyer is dishonest, then the buyer can maliciously use the losing bidders’ data. In [12], the authors proposed a hybrid blockchain-based privacy-enabled accountable auction. It encompasses a mixture of public and private blockchain, and a set of voters are considered to vote for the participation of the bidder, and they vote if a breach is suspected. The auctioneer is the owner of the private chain, and a bidder can join the private blockchain to participate in the auction if the bidders get sufficient votes. The work [12] assumes that the auctioneer and highest bidder will not collude. However, if they collude, no cheating from the auctioneer will be detected, and the privacy and truthfulness of the auction will be compromised.

In [13], the authors proposed a framework for fair auctions and gave solutions to exchange dilemmas and exchange fairness. An exchange dilemma refers to the unwillingness of the winning buyer and seller to exchange money/goods upon completion of the auction due to a lack of mutual trust. This paper also addresses the issue of the bidders abruptly withdrawing from the auction, which invalidates the auction results and wastes the computation power. This issue has been raised as exchange fairness. The auctioneer is equipped with SGX (trusted processor), which generates the keys for participants and is responsible for auction handling. Bidders must verify if the software on SGX is loaded trustfully, and then the auction starts. The framework also includes blockchain to achieve trust and fairness, but this auction doesn’t provide the ability to revise the bid values. Authors in [14] proposed an auctioneer-free sealed bid auction protocol with a linear time computation and communication complexity; bidders are responsible for computing the maximum bid and preserving the privacy of the losing bids. The work used a modified anonymous veto protocol to complete the auction. However, a lot of zero-knowledge-proof operations are required. Our work doesn’t require zero knowledge proofs and satisfies the necessary properties of auction together with the ability to revise the earlier submitted bid in privacy-preserving manner.

Authors in [15] modelled the power demand response problem as a reverse auction, minimizing social cost. To efficiently schedule diverse energy in the energy system, authors in [16] designed two auction schemes for energy trading, one auction works for real time environment and other works for day ahead auction. In one of the recent works in energy trading [17], authors proposed a blockchain-based privacy-preserving double auction for energy trading using secure multiparty computation and homomorphic encryption. The underlying auction model uses secure two-party comparison to declare the auction winner, but the proposed model fails if any of the bidder or auctioneer itself becomes malicious, and privacy of losing bids is threatened. The payment to the seller is not guaranteed to be equal to the winning bid value as no method is given to verify it.

Our work proposed a blockchain-based privacy preserving auction which ensures privacy of losing bids, maintains trust, provides verifiability to auction result, and allows bidders to change their bid if they are not highest bidder. A verifiable market clearing strategy is also given which enables the seller to verify that paid price is equal to the winning bid value. Table 1 compares the discussed related works with our proposed work.

**Table 1: Comparative Study**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Schemes** | **Decentralization using** | **Privacy of losing bids** | **Public verifiability** | **Bid Revision** |
| [7] | Multiple Auctioneers |  |  |  |
| [8] | Distributed Authorities |  |  |  |
| [11] | Blockchain |  |  |  |
| [13] | Blockchain |  |  |  |
| [14] | Multiple Bidders |  |  |  |
| [17] | Blockchain |  |  |  |
| The proposed work | Blockchain |  |  |  |

**3 Preliminaries**

The section briefly introduces the tools considered in the proposed privacy-preserving auction scheme.

**3.1 RSA(Rivest–Shamir–Adleman) cryptosystems**

RSA [18] is a type of asymmetric encryption that is based on the idea of the computational complexity of factorizing the product of large random prime numbers. It consists of two keys, one public and the other private. The public key is used to encrypt the data, and the private key is used to recover that data. One who holds the private key can decrypt any message encrypted with the respective public key. Digital signature is a very famous example of an application based on such cryptosystems.

**3.2 Digital Signature**

A digital signature [18] is a type of mathematical scheme where a recipient of the message can be sure that a particular message comes from a genuine source. For this purpose, a sender generates the digital signature before sharing of information by encrypting the cryptographic hash of the original message with the private key. The sender shares both the message and digital signature, and the receiver can verify the legitimacy of the source of the message by decrypting this signature with the sender's public key.

**3.1 Paillier Homomorphic Encryption**

It is a probabilistic asymmetric algorithm for public key cryptography [19]. The main advantage of Paillier homomorphic encryption is that it takes advantage of the mathematical operation that can be easily performed over the ciphertext without decrypting it. The encryption scheme is secure and is based on the residue problem, which is a computationally hard problem. The main functions that are available in this scheme of cryptography are *KeyGeneration()*, *Encryption()*, and *Decryption()*.

*KeyGeneration()*: This function creates a pair of keys, that is, the public key and a private key. For creating this pair of keys, two large random prime numbers are chosen such that their Greatest Common Divisor, i.e., is 1, where is . We find a . Now we select a random number , such that it belongs to . We do this to ensure that divides the order of by checking the expression, where *L* is the function such that

*Encryption()/E()*: For encryption of any text, we use the Public key, which is *)*. Let the plain text denoted by , which belongs in . For encrypting this plain text, we select a random number that belongs in and is 1. Then, we compute ciphertext *c* as.

*Decryption()/D()*: For decryption of the ciphertext, we use the Private key, which is . Let the ciphertext be decrypted and compute the plaintext , and then we do it with the help of the following expression .

The following equations indicate the algebraic operation that can be performed over the ciphertext generated through this cryptosystem; the result after decrypting it will be the same as if we have done these operations on the plain text directly.

(1)

(2)

(3)

**3.3 Secure Multi-Party Computation**

Secure multi-party computation (SMPC) [20] is an efficient way to know whose bid is the highest without disclosing the actual value of the bid. In the proposed auction, we have used a secure two-party comparison protocol to compare the bid values of two bidders. Each bidder submits padded bid in place of original bid value. For padding, each bidder adds its random number in the bid value. The auctioneer also adds random number in the padded bid of bidders. Here, the auctioneer can easily compare padded bids of two bidders, and no secret information is revealed to the auctioneer or other bidders.

**3.4 Blockchain**

A blockchain is a distributed ledger with a growing list of records (blocks) joined through the cryptographic hash. This ledger is immutable, and multiple nodes in the blockchain, which consist of their own copy of this ledger, provide trust in the information stored on the blockchain. Each block is added via a consensus mechanism where the majority of nodes need to approve the addition of a block to the distributed ledger. Blockchain transactions are irreversible because, once recorded, the data in any given block cannot be altered retroactively without altering all subsequent blocks; thus, this enables trust over past transactions that they are incorruptible. Satoshi Nakamoto proposed the cryptocurrency Bitcoin and related white paper [21] in 2008.

**3.5 Smart Contracts**

Smart contracts, first proposed in the Ethereum white paper [22] by Vitalik Buterin, are agreements or contracts' terms automatically executed, controlled, or documented by a computer program or transaction protocol. These reduce the need for trusted intermediaries, arbitration fees, fraud losses, and unintentional and malicious exceptions. Cryptocurrencies are often linked to smart contracts, and Ethereum's smart contracts are widely regarded as an essential component of decentralized finance (DeFi) and NFT applications. Each smart contract has its own address after being deployed on the blockchain, and once deployed, this action cannot be reverted. The methods written in the smart contract can be accessed from the deployed smart contract address, and respective operations can be performed.

**4 The proposed hybrid auction scheme**

This section first introduces the entities involved in the proposed hybrid privacy-preserving auction scheme. Then, it provides an overview of each phase of the auction. The auction has four phases: registration, bid submission, bid comparison, and market clearance. Later, each phase is discussed in detail. The phase-by-phase architecture of the auction makes the auction efficient and helps to understand the flow of information. Each phase has its timeout to avoid unnecessary delays in the auction. In each phase, a participant must complete the required action within the deadline of the phase. Participants who cannot complete their jobs within the given time limit will be discarded from the auction, and a penalty will be levied on them.

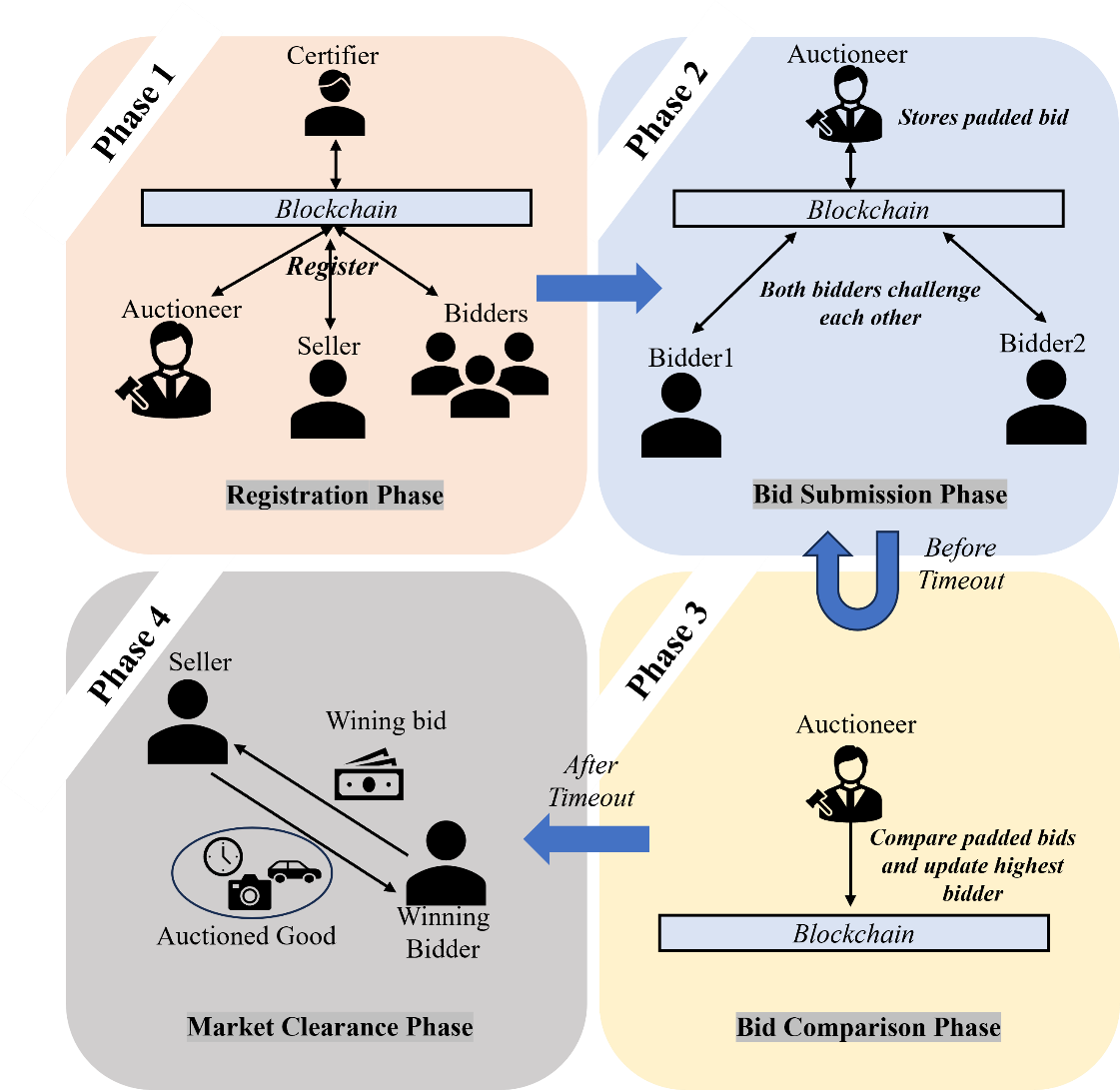
**4.1 Entities involved in the proposed auction.**

Our proposed hybrid auction scheme requires four entities to fulfil the PPA. All four entities are described below.

* Auctioneer: The auctioneer is the entity that provides the blockchain platform for conducting the auction. The auctioneer is an active participant and calculates the auction result based on the received ciphertexts in various auction phases. The auctioneer is considered a semi-honest entity that follows the defined auction rules; however, the auctioneer may be interested in knowing the actual bid values of the bidders to get some financial incentive by revealing the bid values to interested market participants.
* Seller: The seller is the entity that wants to sell an item to interested buyers. Since the seller’s interest is only in selling the product at the best price that it can get from the auction, the seller is considered an honest agent.
* Bidders: Bidders are the buyers interested in buying the seller's product. Since a bidder is interested in winning the auction, the bidder may collude with other bidders to declare itself a winner. Hence, bidders are considered semi-honest entities.
* Certifier: It is an entity that has been considered that generates a unique ID to auction and other entities. The certifier also generates the pallier and RSA key-value pairs for entities involved in the auction, depending on their needs and roles.

**4.2 Various phases of the proposed auction**

The proposed auction is like an open-cry auction; however, a bidder does not reveal the bid value; instead, a padded bid under a secure two-party comparison protocol is shared with the auctioneer, and the auctioneer can decide the highest bid by comparing padded bids. Losing bidders get the opportunity to bid again by changing their bid values. Like a sealed-bid auction, the bid value is not revealed to the auction participants; it prevents bidders from being tempted to bid. Like an open-cry auction, a bidder challenges the highest bidder, and if its bid value is higher than the highest bidder, the bidder becomes the new highest bidder. Our scheme requires four entities: an auctioneer, a certifier, and two bidders at a time, challenging each other to become the highest bidder. There are four phases in the proposed auction, and each phase is discussed individually in brief to provide an overview of the system, as given in Figure 1. Later, each phase is discussed in detail.



**Figure 1**: Phases in the proposed auction

Phase 1: Registration

In this phase, interested auction participants are required to register first. Registration provides all the necessary information, making them capable of participating in the auction. For our proposed auction, each participant registers as a certifier via a unique ID and gets the necessary information to participate in the auction.

Phase 2: Bid Submission

In this phase, two bidders submit the encrypted padded bid value to the auctioneer to challenge each other. Bid values submitted to the auctioneer are in encrypted and padded form; thus, they are prevented from any type of hampering and malicious acts that can be done by the auctioneer.

Phase 3: Bid Comparison

Auctioneer compares the padded bid values submitted by the bidder in the bid submission phase and declares the current highest bidder. If some other bidder wants to challenge the current highest bidder, the bid submission phase starts for the highest and challenging bidder.

Phase 4: Market Clearance

After the expiring of the timeout for bid submission, the auctioneer makes all the bid submissions public and declares the final highest bidder as the winner of the auctioneer, and the winning bidder is asked to purchase the auctioned goods from the seller by paying the winning bid price.

The notions we have used throughout the paper are given in Table 2.

Table 2: Notations

|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
|  | Personal unique ID of participant |
|  | Auction of participant |
|  | Bidder |
|  | Highest Bidder |
|  | The random number of the bidders |
|  | The bid value of the bidder |
|  | The auctioneer |
|  | The paillier public key of participant |
|  | The paillier private key of participant |
|  | The RSA public key of participant |
|  | The RSA private key of participant |
|  | Message is RSA encrypted with the public key of participant |
|  | Message is homomorphically encrypted with the public key of participant |

**4.3 First Phase: Registration**:

Sellers, buyers, and auctioneers must register with the certifier before participating in any auction. The certifier issues a set of Paillier public and private keys, a set of RSA public and private keys and a unique auction ID to each participant, depending on the need. The auction ID of each participant is a unique ID known to all entities involved in the auction for better communications. A participant can use their keys and auction ID to participate in any auction. The certifier deploys a smart contract, i.e., , for registration of participants.

Three participants in the auction needs to register first, namely, auctioneer, i.e., , a seller, i.e., and a bidder (buyer), i.e., . Since the considered auction is a forward auction, there will be one auctioneer, one seller and number of buyers or bidders, where . To register, a participant , where , sends its unique real identity encrypted with the certifier’s public key, i.e., , to the certifier using the smart contract . The certifier decrypts and verifies the . If is legitimate, the certifier issues a set of Paillier public and private keys, i.e., and , a set of RSA public and private keys, i.e., and , and a unique auction ID, i.e., , to the participant, depending on need. Private keys are shared from the certifier to the participant using Diffie–Hellman key exchange protocol. The certifier sends the public keys of participant, i.e., and as a message to the participant. Only the public keys of the participant are stored in the blockchain and can be accessed using . The participant submits the following message to certifier to register for the auction:

,

where, represents encrypted with certifier’s public key, i.e., , and is digital signature of the participant on submitted information. The certifier sends the following message to the participant :

, if the participant is an auctioneer or seller.

, if the participant is a bidder,

where, is digital signature of the certifier on submitted information.

Figure 2 illustrates the registration process coded in .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pseudo code for registration:** | | | | |
|  | A participant prepares . | | | |
|  | submits to certifier using . | | | |
|  | The certifier fetches the from . | | | |
|  | The certifier decrypts and gets the . | | | |
|  | The certifier checks: | | | |
|  | ***if*** is valid: | | | |
|  |  | ***if*** is not in the database: | | |
|  |  |  | ***if*** *participant is an auctioneer or a seller* | |
|  |  |  |  | The certifier generates and RSA public and private keys. |
|  |  |  | ***if*** *participant is a bidder* | |
|  |  |  |  | The certifier generates and paillier public and private keys. |
|  |  | The certifier shares private keys with using Diffie–Hellman key exchange protocol. | | |
|  |  | The certifier adds in database by mapping it to the . | | |
|  |  | ***else*** | | |
|  |  |  | discard registration and sends a message that the participant is already registered. | |
|  | ***else*** | | | |
|  |  | discard registration and sends a message that is not a valid ID. | | |

**Figure 2**: Procedure of the registration phase

Here, it is worth mentioning that all messages are recorded as transactions in the blockchain. Data privacy is a challenge since the proposed work can be deployed in the public blockchain. However, in the aforementioned transactions, the encrypted information, such as encrypted , are shared in blockchain platform, and private keys are not shared in the blockchain; no real and confidential information is openly shared.

**4.4 Second Phase: Bid submission**

A seller , who wants to sell a product using auction, contacts an auctioneer and sends the auctioneer the details related to the product and other necessary information, i.e., , encrypted with the private key of the auctioneer in the form of the following message:

,

where, is a digital signature of the seller on the submitted information.

To start an auction, the auctioneer prepares a smart contract considering all the requirements of auction. The auctioneer shares the auction details, time when the auction starts and ends, , rules, protocols, smart contract addresses and products to be auctioned and other necessary details with the certifier, and these details can be stored in a decentralized manner on a decentralized file storage systems such as IPFS [23] or on a central server of the certifier. These file storage systems work as the bulletin board for the interested participants to know the auction related details. Based on this information, interested bidders decide to participate in an auction.

The auctioneer starts the bid submission phase, and this phase works with the help of the auctioneer’s smart contract We assume that the bidder is the first bidder. registers to auctioneer through the smart contract by submitting its and pays registration fee to the auctioneer. fetches the public key of the auctioneer, i.e., using through the smart contract . Since is the first bidder and if later no other bidders participate, secure two-party comparison cannot be followed. To ensure privacy in the case of a single bidder when SMPC cannot be implemented to ensure privacy, the first bidder submits the hash value of its bid , i.e., . The auctioneer’s smart contract declares as highest bidder by updating byand informs other bidders. represents the auction ID of the highest bidder. If no other bidders are ready to compete within the submission time or there is only one bidder, is declared as the winner. The auctioneer will share with the seller, mentioning that is the winner. When the buyer contacts the seller to purchase the auctioned item, it shares with the seller, and the seller accepts the trading if the hash value of the bid given by the is matching with .

If a bidder wants to challenge the within the submission time, it first registers with the auctioneer by submitting its and pays the registration fee and participation fee. fetches the public key of the auctioneer, i.e., using through the smart contract . Then, the following steps are executed. Bidder first fetches the public key of from the as in this case is, the key fetched from is . To challenge , prepares a padded bid, which is the sum of its current bid valuation of the auctioned product and its own generated large random number , i.e., . encrypts its padded bid with . Further, encrypts the same ciphertext with of auctioneer. sends this encrypted information with a digital signature to the auctioneer. declares as challenger and shares with . The message sent by to the auctioneer is as follows:

.

After receiving , decrypts the encrypted information and recovers . homomorphically adds a random fraction of its own large random number with while keeping the other fraction secret to itself. then forwards this updated information to with its digital signature. The message sent by to is as follows:

.

After receiving , decrypts the encrypted information in and recovers and adds its random number . sends this updated information to . The message sent by to the auctioneer is as follows:

.

After receiving , the auctioneer decrypts the message and recovers and adds . keeps , i.e., final padded bid of , i.e., .

If bidder challenges the to become the winner, must prove that its bid value is high compared to bid value of . To do this, considers an opponent and follows the same procedure that followed to challenge . The bidder first fetches the public key of from the , the key fetched from is . Messages exchanged among , and are similar to messages exchanged when challenged the , only jobs of and are swapped. The messages exchanged among , and in a sequence are as follows:

,

,

.

After receiving , the auctioneer decrypts the message and recovers , and adds . keeps , i.e., final padded bid of , i.e., .

The key consideration is that generates different random fractions of its random number when challenges and when challenges , i.e., , such that and . The computations that the auctioneer and the bidders perform during this phase, such as encryption, decryption, homomorphic addition, normal addition, etc., are performed in an off-chain manner, and the transfer of a message from one entity to another is done via smart contract . Off-chain computations reduce computational effort on blockchain, which in turn reduces the gas costs in blockchain like Ethereum, and on-chain data transfer ensures trust and verifiability.

In the proposed auction, a bidder is allowed to change its bid value if the bidder is not the highest bidder. The bidder modifies its original bid and challenges the highest bidder. We can say there is a challenging bidder and a highest bidder. To preserve the privacy of the bid values, padded bids are used in place of original bids. The challenging bidder cannot know the highest bid. After challenging the highest bidder, it only knows whether it is the highest bidder. A bidder may change its bid value multiple times by challenging the highest bidder. The challenging bidder may increase its bid value very slowly and challenge the highest bidder iteratively to prevent itself from paying very high compared to the highest bidder. If all bidders behave in such a way, the seller will get the payment near the first highest bid if any other challenging bidder wins. To encourage bidders to bid true value and to limit bidders challenging highest bidder frequently, a participation fee is included that a bidder must pay when it challenges the highest bidder. This challenge fee for a bidder increases exponentially as the bidder's attempts to challenge the highestincreases. A part of the participation fee paid by a challenging bidder is paid to the highest bidder for its computational effort such as gas cost in Ethereum, and the remaining part is paid to the auctioneer for conducting the auction. Let be the number of times the bidder challenged the highest bidder and participated in the auction. The participation fee, i.e., paid by when it challenges time is as given in equation (1).

(1),

where, is the initial participation fee set for the first round, is a longevity parameter, and . and help to control the degree of discouragement for challenging bidder to challenge the highest bidder frequently. Since the highest bid value is not known to the challenger and participation fee increases exponentially with the number of participations of a bidder, it encourages the bidders to bid true value as early as possible without many attempts to challenge the highest bidder to save the participation fee.

There is a possibility that more than one bidder can challenge the highest bidder, which will cause incorrect results and irregularities in the system. To prevent multiple bidders from entering the bid submission phase, the auctioneer sets a flag *lock* to True whenever a bidder challenges the highest bidder. Other bidders cannot challenge the highest bidder if *lock* is set to True. Whenever the auctioneer declares the highest bidder after the bid comparison phase, it sets *lock* to False, which means a bidder can challenge the highest bidder. We use one more flag *comphase*, which is set to True by the auctioneer when the auctioneer has two padded bids to compare, i.e., two bidders challenging each other exist. The bid comparison phase executes when the *comphase* is True. The procedure of the bid submission phase is given in Figure 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Bid Submission Phase*** | | | | |
| ***1.*** | ***if*** *current time <= submission phase timeout* | | | |
| ***2.*** |  | *Bidder registers at auction by submitting the using .* | | |
| ***3.*** |  | *pays the registration fee to the auctioneer.* | | |
| ***4.*** |  | *fetches .* | | |
|  |  | ***if*** *is interested in bidding* | | |
|  |  |  | *deposits the fee for participation in the auction.* | |
| ***5.*** |  | ***if*** *== NULL // is the first bidder* | | |
| ***6.*** |  |  | ***if*** *deposited fee == participation fee* | |
| ***7.*** |  |  |  | *submits the hash of the bid value, i.e., using .* |
| ***8.*** |  |  |  | *sets .* |
| ***9.*** |  |  | ***else*** |  |
| ***10.*** |  |  |  | *Reject registration of* |
| ***11.*** |  |  |  | *Exit;* |
| ***12.*** |  | ***elseif*** *deposited fee == participation fee* | | |
| ***13.*** |  |  | *The auctioneer sets lock to True.* | |
| ***14.*** |  |  | *fetches public key of , i.e., .* | |
| ***15.*** |  |  | *creates the padded bid value, i.e., .* | |
| ***16.*** |  |  | *first encrypts the padded bid using and then using .* | |
| ***17.*** |  |  | *prepares and sends to the auctioneer using .* | |
| ***18.*** |  | ***else:*** | | |
| ***19.*** |  |  | *rejects the registration of* | |
| ***20.*** |  |  | *Exit;* | |
| ***21.*** |  | *The auctioneer retrieves homorphically encrypted padded bid, i.e., from .* | | |
| ***22.*** |  | *The auctioneer adds partial part of its random number () homorphicaly with the retrieved ciphertext in previous step.* | | |
| ***23.*** |  | *The auctioneer encrypts obtained ciphertext in previous step using and sends it to using by creating the message .* | | |
| ***24.*** |  | *decrypts the cipher text in and recovers .* | | |
| ***25.*** |  | *adds its random number in decrypted padded bid, i.e., .* | | |
| ***26.*** |  | *encrypts the modified padded bid in previous step using .* | | |
| ***27.*** |  | *prepares the message and sends it to the auctioneer using .* | | |
| ***28.*** |  | *The auctioneer decrypts ciphertext in and recovers .* | | |
| ***29.*** |  | *The auctioneer adds remaining partial part of its random number, i.e., , with the recovered value in previous steps and stores .* | | |
| ***30.*** |  | *fetches public key of , i.e., .* | | |
| ***31.*** |  | *creates the padded bid value, i.e., .* | | |
| ***32.*** |  | *first encrypts the padded bid using and then using .* | | |
| ***33.*** |  | *prepares and sends to the auctioneer using .* | | |
| ***34.*** |  | *The auctioneer retrieves homorphically encrypted padded bid, i.e., from .* | | |
| ***35.*** |  | *The auctioneer adds partial part of its random number () homorphicaly with the retrieved ciphertext in the previous step.* | | |
| ***36.*** |  | *The auctioneer encrypts obtained ciphertext in previous step using and sends it to using by creating the message .* | | |
| ***37.*** |  | *decrypts the cipher text in and recovers .* | | |
| ***38.*** |  | *adds its random number in the decrypted padded bid, i.e., .* | | |
| ***39.*** |  | *encrypts the modified padded bid in the previous step using .* | | |
| ***40.*** |  | *prepares the message and sends it to the auctioneer using .* | | |
| ***41.*** |  | *The auctioneer decrypts ciphertext in and recovers .* | | |
| ***42.*** |  | *The auctioneer adds remaining partial part of its random number () with recovered value in previous steps and stores .* | | |
| ***43.*** |  | *The auctioneer sets comphase to True.* | | |
| ***44.*** | ***else:*** | | | |
| ***45.*** |  | *does not accept any bid submission.* | | |
| ***46.*** |  | *Exit;* | | |

**Figure 3**: Procedure of the bid submission phase

**4.5 Third Phase: Bid Comparison**

In the previous phase, when and are opponents, the auctioneer has final padded bids of and , i.e., and , respectively. The auctioneer compares and , and the bidder with the highest padded bid becomes the new highest bidder, i.e., . The auctioneer declares the highest bidder by updating the with winner’s . If the submission phase time expires, the becomes the winner, and if a bidder challenges the before the expiry of the submission phase time, enter into the bid submission phase. The procedure of bid comparison phase is demonstrated in the Figure 4.

|  |  |  |  |
| --- | --- | --- | --- |
| **Bid Comparison Phase** | | | |
| **1.** | **if** *current time <=**submission phase timeout**&& comphase == True* | | |
| ***2.*** |  | *The auctioneer compares and .* | |
| ***3.*** |  | ***if*** | |
| ***4.*** |  |  | *.* |
| ***5.*** |  | *The auctioneer sets Lock to False.* | |
| ***6.*** | ***else****:* | | |
| ***7.*** |  | *The auctioneer checks .* | |
| ***8.*** |  | *if ()* | |
| ***9.*** |  |  | *The auctioneer declares last as the final winner.* |
| ***10.*** |  | *else* | |
| ***11.*** |  |  | *No bidders participated in the auction.* |

**Figure 4**: Procedure of the bid comparison phase

**4.6 Fourth Phase: Market clearance:**

When submission time expires, the auctioneer declares the highest bidder. The last two bidders are required to disclose their bid values and random numbers to the auctioneer. This information about the last two bidders is required to be disclosed to ensure that remaining bidders and the seller hold the power to verify the winning price of the auctioned good being the highest bid price. It also ensures that the seller does not get the wrong payment for the auctioned goods. Auctioneer declares as the winner by publishing the and makes the final STPC pairs public on the blockchain together with the bid values and random numbers. Next, connects to the seller to purchase the auctioned goods and pays its bid value as the winning price. The seller verifies the paid bid as the highest bid and provides the auctioned goods to . Figure 5 demonstrates the procedure of the market clearance phase.

|  |  |
| --- | --- |
| ***Market Clearance Phase:*** | |
| ***1.*** | *When the submission phase time expires* |
| ***2.*** | *declares as the winner and makes the STPC pairs public.* |
| ***3.*** | *The last two bidders disclose their bid values and random numbers to the auctioneer* |
| ***4.*** | *The auctioneer makes the information obtained in the previous step public.* |
| ***3.*** | *, as the winner, pays to the seller.* |
| ***4.*** | *The seller verifies the paid bid and provides the goods to .* |

**Figure 5**: The procedure of market clearance phase.

**5 Threat Model and Security Analysis**

**5.1 Threat Model**

In this section, we analyze possible threats that our proposed hybrid auction can face, and we also provide proof that our proposed model will be safe against such types of threats.

**Theorem 1**: A cannot gather any true bid from the data collected during the bid submission phase.

**Proof**: When challenges the highest bidder , sends to the auctioneer. In , the auctioneer cannot recover any original information. After adding the random number homomorphically, the auctioneer sends to . In , only is known to the auctioneer. adds its random number in and sends it to the auctioneer, i.e., . In , the auctioneer knows . We assume that

(2)

Since and are known to the auctioneer, we can simplify equation (a) and can be rewritten as follows:

(3)

(4)

Where . In equation (4), only is known to the auctioneer.

Similarly, when challenges the bidder , the messages are exchanged among , and the auction. At the end the auctioneer has . We assume that

(5)

Since and are known to the auctioneer, we can simplify equation (5) and can be rewritten as follows:

(6)

(7)

Where . In equation (7), only is known to the auctioneer.

The auctioneer knows that random numbers of both bidders and are present in equations (4) and (7). Therefore, the equations (4) and (7) can be simplified as follows:

(8)

(9)

where .

In equations (8) and (9), the auctioneer is aware of and . But the auctioneer is not aware of , and . Since there are three unknown variables and two equations, then no exact value of any variable can be calculated from this much information.

The above theorem proves that bidders’ data is safe with , and no real bid value can be revealed with the help of STPC pairs collected.

**Theorem 2***:* Collusion of the challenging bidder and the highest bidder will always result in a distrust situation, and no one of them will be able to verify the correctness of the shared information until one of them is declared the highest bidder.

**Proof**: If and collude, they can share their information with each other. We will see the proof from the side of to , and from the other side, it will be the same, so there is no need to prove it. shares its and with so that keeps its bid value in such a way that . receives the data from the auctioneer, which is . Using this received information from the auctioneer and and from , cannot verify whether is truthful. If is not truthful and decides its considering , may lose the auction. We term this situation as distrust as and are colluding but can never be sure of the correctness of the shared data from as the information possessed is not enough to verify that is honest and the values that has shared are correct. will be able to verify the correctness of shared data upon the declaration of the highest bidder in the bid comparison phase, but it will be too late as the highest bidder has already been declared.

**5.2 Security Analysis**

The proposed auction scheme is secure, and its safety is evaluated in three aspects.

*1)Anonymity*: Here, anonymity refers to the bidders whose is entirely anonymous and cannot be traced by the other participants except the certifier. Other entities participating in the proposed auction never recover the original of the bidders. In the registration phase, & submit their in encrypted form with the help of the certifier’s public key; thus, the certifier can only open the encryption. As the are always secret only to the certifier, we can claim that the bidder’s anonymity will be maintained even after the auction. For more safety, certifiers can be considered in a distributed manner [[reference](https://dl.acm.org/doi/abs/10.1145/3423211.3425669)].

*2) Bid Privacy:* Our proposed hybrid auction scheme achieves another security parameter, bid privacy, for each bid of bidder . We claim that bid privacy will be maintained from the challenging bidder as well as the auctioneer. Theorems 1 and 2 prove that bid privacy is always maintained during and after the auction. Here, it is worth mentioning that in the proposed auction, bid values of the final winner and the bidder who challenged the final winner are made public by the auctioneer. The revelation of these bid values is a must to ensure correct payment to the seller by the final winner and to provide the opportunity for losing bidders to verify their bid values are less than the bid value of the final winner.

*3) Non-repudiation:* In the proposed auction, in the submission phase, a bidder challenges the highest bidder . Once the challenging bidder and the highest bidder submit their padded bid in the submission phase, they are not allowed change their bid in the submission phase. After the end of the bid submission phase, they will enter the bid comparison phase, and the auctioneer declares the highest bidder. Now, a bidder can change its bid value and challenge the highest bidder. In summary, a bidder cannot change its bid value once it enters the bid submission phase after submitting the bid. The bidder can change its bid value after the end of the bid comparison phase, i.e., upon declaration of the highest bidder. Furthermore, each bid submission is registered as a transaction on the blockchain; therefore, malicious modification of bids is also not possible. Thus, the proposed auction scheme satisfies the non-repudiation of the bids.

**6 Experimental Evaluation**

There are numerous blockchain platforms; the most commonly used are Ethereum and Hyperledger. We have used the Ethereum blockchain over Hyperledger fabric because we propose a decentralized privacy-preserving auction based on a public blockchain. Ethereum allows us to simulate both public and private blockchains. In contrast, Hyperledger fabric [Reference] is a private and permissioned network. Ethereum platforms also enable us to study the real-life gas cost of the proposed hybrid privacy-preserving auction scheme. Tools such as truffle and web3.py are helpful in real life analysis of blockchain based applications.

Each operation in Ethereum requires a specific amount of gas, which is to be paid by the user for the change of state of the deployed smart contract. The amount of computing work needed for any Ethereum network operation is measured in terms of gas. Users must pay in Ether (ETH) for the amount of gas used for each transaction and smart contract execution. The network is kept safe, effective, and impervious to denial-of-service and spam attacks by paying this gas price. The user pays the gas amount in two steps. First, the user pays the gas for contract deployment, i.e., execution cost, which is the cost for the internal storage and changes in the contract; the user makes the second type of gas payment is the transaction cost, which is paid upon the change in the state of the contract, other than this no gas payment is needed by the user to the blockchain, accessing the value of a particular variable from the deployed contract cost nothing. So, in our simulation setup, the user only pays gas for value submission and contract deployment. Fund transfer is similar to transaction type, but in our analysis, we have shown it separately to understand better the applicability of the proposed hybrid auction scheme in real life.

The implementation and testing of the proposed hybrid privacy-preserving auction scheme has been implemented and tested over the intel i7-12700 12-gen processor with 32GB RAM and a max clock speed of 4900MHz. The software suite that we have used is Ganache [24] for simulating the Ethereum blockchain on a local node together with the web3.py [25] library to call and run smart contracts inside the Python Jupiter notebook. Ganache and Python help simulate the on-chain and off-chain environments on the local node. Smart contracts are implemented in the Solidity programming language version 0.8.2. Various functions are defined in the smart contracts and to achieve the various functionalities required in the proposed auction, which are summarized in Table 3 and Table 4, respectively. Table 5 shows the Fund Transfer function we used to transfer Ether from one account to another. This function is not written in any contract but has been implemented directly using web3.py. The function transfers the funds from one Ganache account to another Ganache account. A sequence diagram of the proposed auction considering functions discussed in Table 3 and Table 4 is given in Figure 6.

Table 3: Functions defined in

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Input** | **Returned Value** | **Descriptions** |
| *store()* | *Encrypted* | *NA* | *Stores encrypted on blockchain* |
| *retrieve()* | *NA* | *Encrypted UID* | *Fetches encrypted* |
| *submitKeysAndBID()* | *Public keys and of bidders* | *NA* | *Maps public key to respective of bidder* |
| *mapEncUIDToBID()* | *Encrypted and of bidders* | *NA* | *Maps encrypted to of bidder* |
| *fetchBID()* | *Encrypted* | *AID of respective bidder* | *Fetches of bidder using encrypted* |
| *fetchEntityDetails()* | *of bidder* | *Public key* | *Fetches public key of bidder using of the bidder* |

Table 4: Functions defined in

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Input** | **Returned Value** | **Descriptions** |
| *register()* |  | *NA* | *Registers the bidder to auctioneer for the auction* |
| *bidderEncryptedBidSubmission()* | *Target , sender & padded bid* | *NA* | *Submits the encrypted padded bid together with of sending bidder and receiving bidder* |
| *retrieveBidderEncrypted -BidSubmission()* | *NA* | *padded bid* | *Retrieves the padded bid* |
| *updateDataForOtherBidder()* | *padded bid from auctioneer* | *NA* | *Updates the padded bid from auctioneer respective to the bidder* |
| *fetchData()* | *NA* | *padded bid from auctioneer* | *Fetches the padded bid from auctioneer* |
| *updateHighestBidder()* | *of highest bidder* | *NA* | *Updates the highest bidder* |
| *updateLock()* | *Flag value 0/1* | *NA* | *Changes the flag to prevent the case of more than two bidders bid submission* |
|  |  |  |  |
| *fetchLock()* | *NA* | *Flag value 0/1* | *Fetches the flag* |
| *fetchCurrentHighestBidder()* | *NA* | *of highest*  *bidder* | *Fetches the current highest bidder* |

Table 5: Fund transfer function implemented in web3.py library

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Input** | **Returns** | **Description** |
| *bidderPayGas(etherValue, bidderAccount)* | *Amount of ether that is to paid for participation and bidder’s address* | *NA* | *Transfer a given amount of ether from bidder’s account to auctioneer’s account* |

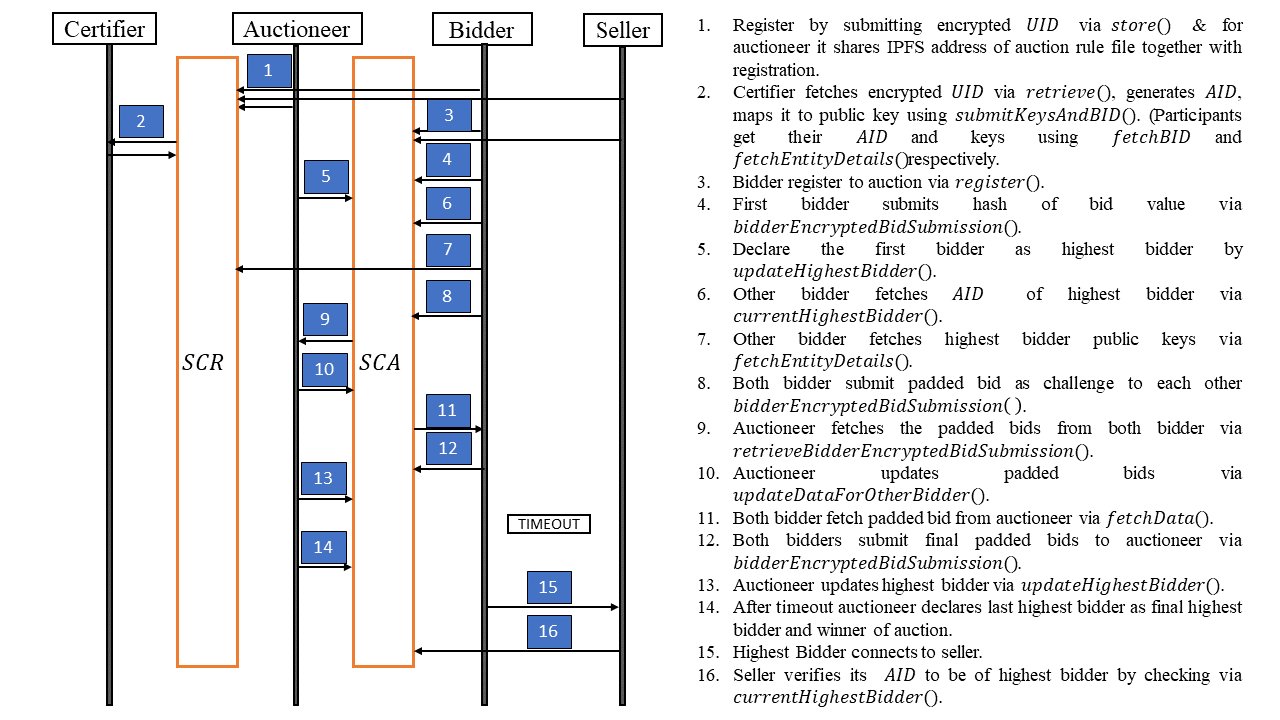
****

Figure 6: Sequence diagram of the proposed auction considering functions defines in the smart contract.

**5.1 Cost Analysis**

Ethereum uses the gas unit concept to quantify the computation needed to carry out the intended function over the deployed contract. The speed at which the network completes a transaction is determined mainly by the gas price. A user who wants some computation has to specify the gas price, and the gas cost is calculated based on the gas units used for computation. Gas cost equals the multiplication of gas price and number of gas units. Transactions with higher gas prices will frequently be processed faster than those with lower gas prices since miners are more likely to prioritize greater reward transactions. So, in our analysis, we chose the default gas price and performed the analysis. It is important to mention that gas prices fluctuate based on supply and demand and are not permanently fixed. When demand is high, gas prices may increase, and when demand is low, they may decrease. When submitting a transaction, users can indicate the gas price they will pay. In Solidity, every operation has a gas cost. The total gas cost of all the actions that make up a transaction or function execution is the sum of those gas costs. When a user submits a transaction or calls a function, they specify the maximum amount of gas they are willing to spend on the operation. If the gas runs out before the procedure is completed, it will be terminated, and all changes made will be reversed. This ensures the stability and security of the Ethereum network and wards off the dangers of resource exhaustion.

**Experiment 1: Percentage of gas costs used by function type**

In the current scenario, the conversion rate for the gas price is, on average, 10 Gwei. Gwei is a denomination of Ether (ETH). 10 Gwei is nearly 0.0001824 ETH, nearly equal to 0.5822 USD. We have calculated the associated gas cost based on this conversion rate and analyzed the feasibility of the proposed hybrid privacy-preserving auction scheme in real-life settings. Table 6 shows the percentage of gas utilized by different functions compared to the total gas cost in the proposed privacy-preserving hybrid auction scheme. The calculations are calculated considering the RSA key size of 5500 bits and the Homomorphic encryption key size of 1024 bits. The number of bidders is considered ten, and the number of times the highest bidder is challenged is five. From Table 6, we can see that a significant part of the gas cost is due to the transactions over the blockchain. The deployment of the smart cost takes much less gas than the transaction gas cost. Because when we deploy the smart contract, a specific gas cost is paid one time based on the storage required and other computations that will be performed in the smart contract. However, in the case of the transactions, the gas cost is charged each time the state of the variable is changed. In the proposed auctions, such transactions are performed multiple times throughout the auction, thus contributing to a significant percentage of the total gas cost. Fund transfer transactions are carried out only during registration fee deposition and participation fee deposition whenever a bidder challenges the highest bidder; thus, they contribute minimally to the total gas cost.

**Table 6: Percentage of gas costs used by the functions type.**

|  |  |
| --- | --- |
| **Function Type** | **Percentage** |
| *Deployment Function* | *2.7521019 %* |
| *Transaction Functions* | *96.722316%* |
| *Fund Transfer Functions* | *0.5255812 %* |

**Experiment 2: Real-life costs used by function types**

In this experiment, we calculate the real-life cost of the proposed hybrid auction scheme under the same settings as earlier. The conversion of gas cost to ETH and then further to USD has been presented, and during the conversion, 1 ETH was nearly 2873.13 USD. Table 7 shows the real-life cost of various function types. Table 7 shows that the proposed hybrid auction is feasible for real-life auctions, which need privacy and the chance of bid increment as a challenge to the winning bidder.  The overall expenditure of the proposed hybrid auction scheme will be 123 USD for 10 participants, with the number of bids raised to five. If we calculate the cost that a particular bidder will pay on average (Discuss).

**Table 7: Real-life costs used by the functions type.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Type** | **Gas** | **ETH** | **USD** |
| *Deployment Function* | *1319548* | *0.0013195480* | *4.19* |
| *Transaction Functions* | *39400401* | *0.039400401* | *117.01* |
| *Fund Transfer Functions* | *252000* | *0.0002520* | *0.80* |

**Experiment 3: Gas cost analysis with respect to the number of bidders and the number of bid raises.**

In this experiment, we analyze gas cost by changing the number of bidders and bid raises. Next in Figure 9 we show the variation of gas cost with respect to increasing number of bidder and increasing number of bid raise. During the experiment we have considered the number of bid raise to be half of the total number of bidder and the figure shows that the total gas cost increases as the number of bid raise and total number of bidder increases. The total gas cost increase because higher the number of bidders more the number of transactions is done over the blockchain and each transaction costs a specific gas amount as discussed earlier thus the increases the total gas cost. As on high number of bidders the gas cost is high which can be reduced if the number of bid raises are lower.

The auctioneer will pay for the gas that the execution of the auction will require, and each bidder contributes to this gas cost. The auctioneer can adjust the bid raise parameter so that the gas cost of the total execution of the auction can be paid. As seen in the experiment, gas cost increases with a higher number of bid raise attempts; the bidders can adjust the number of rounds they will be participating in bid raises to avoid suffering colossal losses and have the chance to win the auction.

A graph with a line graph

Description automatically generated with medium confidence

Figure 9: Gas cost vs number of bidder vs number of bid raise

**Experiment 4: Gas cost analysis with respect to the key sizes of RSA encryption and Paillier homomorphic encryption.**

Figure 10 shows the variation of gas cost on increasing the key sizes of RSA encryption and Paillier homomorphic encryption. As the key size increases the length of ciphertext also increases, and each ciphertext is to be passed through the blockchain thus the storage required also increases thus the overall gas cost also increases and same has been shown in Figure 10.

The key size of the encryption is directly proportional to the size of the ciphertext generated upon encryption. The larger the size of the ciphertext, the larger the space required on the blockchain, and it will increase the gas cost. This experimental study shows that the key size can be adjusted optimally based on the security requirement. The key size must be selected in a way that does not compromise the security of the submitted information and that the gas cost required to store these ciphertexts on the blockchain is reasonable.

A graph of a line

Description automatically generated

Figure 10: Gas cost vs Paillier key size vs RSA key size.

**Experiment 5: Effect of change in participation fee**

**Paricpation fee can bee seen as the degree**

**5.2 Discussion**

Final opinion or verdict

**8 Conclusion**

Our proposed hybrid auction scheme gives an extra edge to the bidders of the privacy-preserving auction, i.e., raising their bid valuation as a challenge to the highest bidder while satisfying all the necessary properties of the privacy-preserving sealed bid auction. A change in bid valuation due to not being the highest bidder does not suffer from the bid temptation as it was the biggest flaw with the open-cry auctions. The proposed hybrid auction gives the advantage of raising the bid valuation by revealing the highest bid rather than declaring the highest bidder.

As the proposed hybrid auction scheme is implemented, the public blockchain will be used to ensure trust in the participants, and each publicly stored transaction will help to verify the auction results. The practicability of the proposed hybrid auction scheme has been discussed in the experiment evaluation, which shows that the proposed scheme is practical in real life. [More discussion of experiment results after experiment evaluation]. The mixture of off-chain and on-chain data handling and data comparison ensures that no extra financial burden is raised upon the participants when using smart contracts and public blockchain while maintaining the necessary properties of the privacy-preserving auction.

The proposed privacy-preserving hybrid auction scheme is capable of withstanding the threats that may arise and make the auction faulty, and respective proofs of the threats have been presented in threat analysis. The scheme also provides key features such as bid anonymity, nonrepudiation, and bid privacy.

Thus, our proposed hybrid auction scheme achieves the benefits of both privacy-preserving and open-cry auctions. The scheme is practicable and can be helpful in real-life applications.

**References**

[1] “https://eauction.gov.in/eauction/#/”. Accessed on 01/05/2024

[2] R. Alvarez and M. Nojoumian, “Comprehensive survey on privacy-preserving protocols for sealed-bid auctions,” Comput Secur, vol. 88, p. 101502, Jan. 2020, doi: 10.1016/J.COSE.2019.03.023.

[3] A. C. -C. Yao, "How to generate and exchange secrets," 27th Annual Symposium on Foundations of Computer Science (sfcs 1986), Toronto, ON, Canada, 1986, pp. 162-167, doi: 10.1109/SFCS.1986.25.

[4] “https://ethereum.org/en/learn/”. Accessed on 01/05/2024

[5] S. Athey, J. Levin, and E. Seira, “Comparing open and Sealed Bid Auctions: Evidence from Timber Auctions\*,” *The Quarterly Journal of Economics*, vol. 126, no. 1, pp. 207–257, Feb. 2011, doi: https://doi.org/10.1093/qje/qjq001.

[6] “Vickrey, William. ‘Counterspeculation, Auctions, and Competitive Sealed Tenders.’ The Journal of Finance, vol. 16, no. 1, 1961, pp. 8–37. JSTOR, https://doi.org/10.2307/2977633. Accessed 3 May 2024.”.

[7] H. Kikuchi, M. Harkavy, and J. Tygar, “Multi-round Anonymous Auction Protocols.” Available: https://tygar.net/papers/Multi-round\_anonymous\_auction\_protocols/DARE.pdf

[8] M. Abe and K. Suzuki, “M + 1-st price auction using homomorphic encryption,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Verlag, 2002, pp. 115–124. doi: 10.1007/3-540-45664-3\_8.

[9] H. Galal and A. Youssef, “Verifiable Sealed-Bid Auction on the Ethereum Blockchain.” Available: https://eprint.iacr.org/2018/704.pdf?ref=hackernoon.com

[10] H. S. Galal and A. M. Youssef, “Succinctly Verifiable Sealed-Bid Auction Smart Contract,” Lecture Notes in Computer Science, pp. 3–19, Sep. 2018, doi: https://doi.org/10.1007/978-3-030-00305-0\_1.

[11] B. Chen, X. Li, T. Xiang, and P. Wang, “SBRAC: Blockchain-based sealed-bid auction with bidding price privacy and public verifiability,” Journal of Information Security and Applications, vol. 65, Mar. 2022, doi: 10.1016/j.jisa.2021.103082.

[12] H. Desai, M. Kantarcioglu and L. Kagal, "A Hybrid Blockchain Architecture for Privacy-Enabled and Accountable Auctions," 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 2019, pp. 34-43, doi: 10.1109/Blockchain.2019.00014.

[13] 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, 2022, doi: 10.1109/TDSC.2020.3045449.

[14] S. Bag, F. Hao, S. F. Shahandashti, and I. G. Ray, “SEAL: Sealed-Bid Auction without Auctioneers,” IEEE Transactions on Information Forensics and Security, vol. 15, pp. 2042–2052, 2020, doi: 10.1109/TIFS.2019.2955793.

[15] R. Zhou, Z. Li and C. Wu, "An online procurement auction for power demand response in storage-assisted smart grids," 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 2015, pp. 2641-2649, doi: 10.1109/INFOCOM.2015.7218655.

[16] W. Zhong, K. Xie, Y. Liu, C. Yang and S. Xie, "Auction Mechanisms for Energy Trading in Multi-Energy Systems," in IEEE Transactions on Industrial Informatics, vol. 14, no. 4, pp. 1511-1521, April 2018, doi: 10.1109/TII.2017.2787751.

[17] B. Wang, L. Xu, and J. Wang, “A privacy-preserving trading strategy for blockchain-based P2P electricity transactions,” Appl Energy, vol. 335, Apr. 2023, doi: 10.1016/j.apenergy.2023.120664.

[18] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978, doi: https://doi.org/10.1145/359340.359342.

[19] P. Paillier, “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes,” *Advances in Cryptology — EUROCRYPT ’99*, pp. 223–238, 2021, doi: https://doi.org/10.1007/3-540-48910-x\_16.

[20] C. Zhao et al., “Secure Multi-Party Computation: Theory, practice and applications,” Inf Sci (N Y), vol. 476, pp. 357–372, Feb. 2019, doi: 10.1016/J.INS.2018.10.024.

[21] S. Nakamoto, “Bitcoin: a Peer-to-Peer Electronic Cash System,” Oct. 2008. Available: https://bitcoin.org/bitcoin.pdf

[22] V. Buterin, “A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM,” 2014. Available: https://finpedia.vn/wp-content/uploads/2022/02/Ethereum\_white\_paper-a\_next\_generation\_smart\_contract\_and\_decentralized\_application\_platform-vitalik-buterin.pdf

[23] “https://docs.ipfs.tech/”. Accessed on 01/05/2024

[24] “https://archive.trufflesuite.com/docs/ganache/”. Accessed on 01/05/2024

[25] “https://web3py.readthedocs.io/en/stable/”. Accessed on 01/05/2024