**Conference Paper for Hybrid Auction**

**Abstract**

*Auctions are one of the most famous market mechanisms for collecting the market value of any good. Auction formats such as English and Dutch, which are variants of open cry auction, where the bid value of each bidder is available to each of the auction participants, suffer from many issues such as bid repudiation, false valuation, and bid privacy concerns so to tackle such problems sealed bid second priced auction have been proposed where the submitted bid values are in encrypted form. Only the auctioneer has the authority to open these bids. Based on comparing these results, the auctioneer declares the highest winning bidder. Still, the constraint with sealed bid auctions is the bid value of each bidder is collected in one round and, based on the submitted bid result, is declared, thus removing the opportunity for the bidders to change their valuation if they think that the value of the auction good can be of higher value for them after a certain period during the auction. In this paper, we propose a blockchain-based hybrid privacy-preserving auction scheme 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. Our privacy-preserving auction scheme achieves the benefits of both open-cry and sealed-bid auctions.*

**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 many commercial websites are open to the general public, thus providing a wide range of auctioned apparel; users can act as buyers or sellers of this product. Auctions are not only limited to the apparel's various other domains, such as electricity trading, spectrum allocation, cloud computing, etc., but have adopted auctions to gather a reasonable valuation of the auctioned product in the market. Classical auctions were open cry auctions where each participant knows the bid value of each bidder in its true form; thus, these auctions suffered from tempered bidding, and in solution to these issues, authors proposed sealed bid auctions where the bid is submitted in encrypted form and are only open to the auctioneer, but this also suffers from problems such as bid repudiation, ….............................. and to tackle such issues, privacy-preserving auctions (PPA) were introduced. 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 issues have been resolved, there is a tradeoff to bidders who may want to change the bid valuation, knowing they are not the highest bidders. Bidders tend to win the auction; if allowed, they change the valuation, knowing they are not the highest bidder. Changing the bid valuation is similar to the classical open-cry auction. Still, to ensure that the scheme does not suffer from bid temptation and to maintain transparency, we propose a blockchain-based hybrid privacy-preserving auction scheme where the bidders can know if they are the highest bidder and can change their bid valuation based on it but will not be able to gather true bid value of highest bidder and will be allowed to raise the bid valuation without bid temptation. 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 scheme ensures that the auctioneer can never gather any true bid valuation of any bidder during or after the auction as well as 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 or not they are getting correct input from the other. Thus, distrust can be seen as a significant advantage, which works as an emotional factor among the colluding agents to never believe in their other colluding agents. We also claim that in our auction scheme, bid anonymity is maintained by each participant during the procedure. After the auction, the privacy of the losing bids is preserved.

Our hybrid auction scheme uses secure multiparty computation (SMPC), homomorphic encryption (HE), and blockchain. SMPC is an excellent tool with a lower overhead than the other PPA techniques (reference). 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 together with lower computational overhead in comparison to other secret sharing techniques and in last blockchain provides us two fundamental properties which are necessary for privacy-preserving auction, i.e., trust over each transaction done during the whole auction procedure and verifiability to the auction results. Another great advantage of the public blockchain platform such as Ethereum[reference] is smart contracts which are self-executing codes running over the blockchain are another useful techniques. Public blockchain ensures trust to the participants[reference], and each publicly stored transaction helps verify the auction results. This blockchain-based hybrid auction scheme is a novel contribution that allows the bidders to change the bid value they previously submitted upon knowing they are not the highest bidder while maintaining all the necessary properties of a privacy-preserving auction.

Our scheme requires four entities: an Auctioneer, a certifier, a losing bidder, and the highest bidder; a discussion of job roles and work is in the methodology section. The paper consists of a literature review in section 2, the preliminaries discussed in section 3, and the methodology discussed in section 4. The experimental evaluation and conclusion are given in sections 5 and 6, respectively.

**2 Literature Review**

PPA has significantly been considered in literature emphasizing issuing public verifiability of the results and avoiding auctioneer's maliciousness. Various trust models have been considered to do so. Table 1 presents the comparison of different models based on the trust model used and the necessary auction properties satisfied by them.

In one of the early works [[ref](https://people.eecs.berkeley.edu/~tygar/papers/Multi-round_anonymous_auction_protocols/IEIC_multiround_anonymous_auction_protocols.pdf)], a multiple-round-based auction protocol was proposed with the concept of multiple () auctioneers, making the system safe with the threshold of () such that these numbers of auctioneers are not colluding with each other till then the system is secure. In the initial phase, prices are shared for the auction goods, and the bidders generate bid vectors for these prices. For the interested price, they put their , and for the non-interested price, they put 0, together with the random padding generated through the sum of random polynomials. These bid vectors are then input to an SMPC algorithm, and the winning-price number of present at the winning price is revealed. This is made by the auctioneer using the La-Grange scheme to solve the simultaneous equations and obtain the free variable, which gives the sum of the identities of bidders who are willing to bid at that price. If a single bidder is interested in the highest price, then only a single is present. If the sum of multiple is found, then multiple bidders are interested at that price, which is a case of a tie, and again, subsequent rounds are performed with a more refined price list near the highest price to get a single winner at the winning price. If the value of is minimal, then many subsequent rounds of auction will be required, increasing the computation cost, and dependency on the threshold of auctioneers also increases communication cost; this work does not ensure the verifiability of the auction results by each participant.

Further, this work was improved in [[ref](https://ieeexplore.ieee.org/document/884610)] by masking bids in random polynomials, which are generated for each bid by the set of distributed servers to ensure trust in the system. The winner was detected in the same way as earlier, but this time, the are masked by the random polynomials. When the auctioneer declares a winning price, all the distributed servers collaborate and remove the noise from these IDs to know the exact winner ID. No other information is revealed to the auctioneer. This work improved the shortcomings of earlier work in terms of privacy and security, and the second highest bid was kept secret from the winner compared to the previous work through the masking step by the servers. However, in this work, the auctioneer can also know the highest winning price, and verifiability by each auction participant is not ensured. The threat model has been shifted to servers from auctioneer in comparison to previous work. Instead of an auctioneer threshold, here, a server threshold has been used such that servers should be honest to complete the auction safely. Another work that was presented in succession by the same authors is given in[[ref](https://sci-hub.se/https:/link.springer.com/chapter/10.1007/3-540-46088-8_27)]; an auction scheme is proposed where result verifiability has been assured by the side of the auctioneers, which can be considered as partial verifiability. Also, it does not prevent the anonymity of the winning bidder from the auctioneers, and it requires a larger number of auctioneers to fulfill the auction, as mentioned in [[ref](https://sci-hub.se/10.1007/3-540-45664-3_8)].

Authors in [[ref](https://sci-hub.se/10.1007/3-540-45664-3_8)] use ElGamal encrypted bidder-generated bidding vectors consisting of bids encrypted by a public key that is by a set of distributed authorities in a threshold manner. 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: mix and match. Later, the auctioneer declares the winner. In this protocol, the highest bid is revealed to the auctioneer after completion of the auction.

The work in [reference12] is similar to the work that we are proposing. It proposes a first price auction protocol where the concept of distributing the bid value to multiple shares is used; thus, distributing the trust to the bidders itself is considered. In this work, the role of auctioneer is played by the seller, so no auctioneer is involved during the whole process of the auction, thus shifting the threat model from auctioneer to seller; this paper explored bidder's collusion possibility and a threshold of bidders being dishonest is considered to avoid bidders' collusion and resulting in colluding bidders knowing the bid value of other bidders. Another threat is that if the seller becomes malicious and colludes with the bidders, he has to collude with at least bidders to know other bids. This work also needs to give the auction results public verifiability to tell whether or not the bidders' collusion or any corruption in protocol happened in between. If this work is applied in the second-price environment, then it will be impossible to ensure that the seller changes the second-highest bid for his profit as this work doesn't ensure the verifiability of the auction results.

The works of [[ref](https://link.springer.com/chapter/10.1007/978-3-642-03549-4_20)] 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 ask, representing how much they are willing to buy or sell at all possible prices. The bids and asks are then secretly 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. No interactivity is required after traders submit their offers (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.

In the works [[ref](https://dl.acm.org/doi/abs/10.1145/3600160.3600190)], a double auction is proposed based on homomorphic encryption and zero-knowledge proof of consistencies, and it satisfies major auction properties such as pseudonymity, unforgeability, traceability, and non-repudiation. The scheme is fulfilled by the assumption of a non-colluding third agent which helps compute the auction results. This protocol ensures privacy-preserving as well as public verifiability. However, the trusted third party can collude with the auctioneer and disclose the key, resulting in the disclosure of the confidential data of the bidders.

Privacy-preserving auctions are important in auction applications such as spectrum allocation, energy trading, data trading, etc.

One such work of [[ref]](https://ieeexplore.ieee.org/document/8384016), an application of privacy-preserving auction, has been proposed through which big data trading is completed. In the suggested protocol, an intermediate platform is considered, and a single auctioneer is considered with an assumption of both parties being independent of each other. Bidders bid their bidding price and forward it to the intermediate platform by encrypting it with the auctioneer's public key. The intermediate platform further adds a padding of common random number homomorphic to all gained bids and transfers these padded bids to the auctioneer, and based on these padded bids, the winner is declared. Even though this model fulfilled the auction, the assumption of independence between the auctioneer and intermediate platform doesn't hold in real life; if they both collude, then the auctioneer can easily gather all bids in the original form, and public verifiability of the auction results is not ensured.

Another recent work in the domain of energy trading is [[ref](https://www.sciencedirect.com/science/article/abs/pii/S0306261923000284)] where a privacy-preserving model has been used to fulfill energy trading without disclosing any confidential information to the auctioneer about bidders. This proposed architecture is based on blockchain and it requires a certifier who certifies the participant and creates unique IDs and paillier homomorphic key pairs for the bidders and the auctioneer. Further, each bidder prepares padded information containing its original bid and a random number in the form of a product, further, this information is encrypted with the auctioneer's public key. This information is not directly sent to the auctioneer. Instead, it is propagated to each bidder present in the auction, who multiplies his random number to this information homomorphically and then forwards it to the auctioneer. The auctioneer collects all such padded bids and compares these padded bids with the help of a secure two-party comparison protocol (explained in section 3) and declares the winner. This model of PPA lacks public verifiability and fails when two bidders collude with each other. Even a single bidder can make the auction results incorrect, which is not detectable.

**3 Preliminaries**

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

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

RSA [reference] 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 private key. Digital signature is a very famous example of an application based on such cryptosystems.

**3.1 Digital Signature**

A digital signature[reference] 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 encrypts the message with the private key and the receiver can verify the legitimacy of the message by decrypting that message with the sender's public key. Let's say Ram sends a message to Shyam to know the genuineness of the message. Ram attaches a digital signature with the original message by encrypting the message with its private key, and Shyam decrypts this encrypted text with Ram's public key. If the message and decrypted message are the same, then Shyam can be assured that the message is authentic and comes from Ram.

**3.1 Paillier Homomorphic Encryption**

It is a probabilistic asymmetric algorithm for public key cryptography [reference]. The main advantage of Paillier homomorphic encryption is taking 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 nth 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 (keyPub)* and a *Private key (keyPri)*. For creating this pair of keys, two large random prime numbers *(p, q)* 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 so to ensure that *n* divides the order of *g* 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 *r* 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)

**3.2 Secure Multi-Party Computation**

Secure multi-party computation (SMPC) [reference] is an efficient way to know whose bid is the highest without disclosing the actual value of the bid. SMPC will help sort the ciphertexts of the bid values, and based on the result of sorting, the auctioneer will decide who is the winner.

**Secure Two-Party Comparison (STPC)**

In ARPAN, we have used a secure two-party comparison protocol to compare the bid values of two bidders. This protocol is performed between every bidder. Let's say bidder and bidder generate bid values

and . Bidders and don't want either the auctioneer or other bidders to know their bid value. For this purpose, a mathematical formulation is performed, where both bidders generate a random valuecollaboratively, and then it is padded to both and , thus causing the expressions and . Here, the auctioneer can easily compare both expressions, and no secret information is revealed to him or other bidders. The same mathematical formulation will be performed for each bidder, thus preserving the privacy of the bid and yet making it possible to compare the bid values. The notions that we have used throughout the paper have been described in Table 2.

|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
|  | Personal unique ID of bidder |
|  | Id of bidder |
|  | Bidder i |
|  | Highest Bidder |
|  | The random number of the bidders |
|  | The bid value of the bidder |
|  | Highest bid |
|  | The auctioneer |
|  | The public key of the auctioneer |
|  | The private key of the auctioneer |
|  | The public key of bidder |
|  | The private key of bidder |
|  | Message is homomorphically encrypted with the public key of the auctioneer. |
|  | Message is homomorphically encrypted with the public key of bidder |
|  | The decryption of the ciphertext by the public key of bidder |
|  | Message encrypted with RSA public key of the bidder |
|  | The decryption of ciphertext with the RSA private key of bidder |

Table 2: Notations used in HARAN.

**4 The proposed hybrid auction scheme**

This section introduces the entities involved in the proposed hybrid privacy-preserving auction scheme. Then it discusses the job roles of each entity in various phases to complete the whole auction in a privacy-preserving manner. Phase-by-phase architecture is used to make the auction scheme efficient and further achieve the information flow of each phase. To avoid maliciousness, every phase has its timeout, before which each participant has to ensure the completion of the phase from its side. Those unable to complete their job roles 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 a total of four entities to fulfill the PPA. All four entities are described below.

* Auctioneer: The auctioneer is a single party that provides the platform for conducting the auction. The auctioneer is also an active participant and calculates the auction result based on the received ciphertexts through the auction phases. The auctioneer is interested to know the actual bid values of the bidders for its profit and is considered a semi-honest entity.
* Bidders: Bidders are the interested buyers of the seller's product. Bidders are interested in winning the auction, and as a result of this, they may try to collude with other bidders to declare themselves as the winner and are considered semi-honest entities. We still need to prove the bidder's collusion fully, but we present a condition of distrust among bidders, making them reluctant to collude with each other.
* Seller: The seller is the entity here to sell an item to interested buyers and is considered an honest agent. The seller’s interest is only in selling the product at the best price that it can get from the auction.
* Certifier: It is an entity that has been considered to generate the for Auction as well as the pallier homomorphic and RSA key-value pairs for all the parties involved in the auction. We have considered certifier for the auction scheme to work. The proposed hybrid auction scheme can work without the certifier too but we will have to shift the key generation to the participants themselves.

**4.2 High-level overview of the proposed auction**

The following subsection consists of an overview of the working of HARAN; different phases, and the working of those phases have been covered in this subsection**.** Each of the bullets represents the phases in the HARAN and discusses what are the job roles performed by the entities in brief during the HARAN.

* *Registration phase:* In this phase, the auctioneer and bidders register for the auction. As a first step, shares the auction details, rules, and protocols with the certifier and gets registered; these details are listed as a transaction on the blockchain and are made public to everyone. The willing bidders register for the auction by depositing the fee to the certifier, and, in return, the certifier provides the public and private keys to the participants; all these transactions are registered on the blockchain. The certifier generates the keys and stores UID in an off-chain manner. The certifier uses the smart contract for the registration phase. Figure 2 shows the registration phase.
* *Bid submission phase:* This phase works in two halves with the help of the auctioneer’s smart contract . In the first half, the bidder submits its bid value to in the form of the addition of random number and its bid value (random numbers are generated by bidders and auctioneers beforehand at their node and are always kept secret from others), which is further encrypted with the public key of the bidder , i.e., destination bidder. In the next half, homomorphically adds a fraction of its random number, i.e., , to the submitted bid value and forwards it to . In the second half, decrypts the message, adds its random number, , and resends this information to the for STPC. The algorithm and working of this phase are discussed in detail in upcoming sections.
* *Bid comparison phase*: The comparison of bid values takes place off-chain on the auctioneer's node, and the comparison is performed using the STPC protocol. Upon comparison, A declares the highest bidder, and publishes it’s among all other bidders by publishing it to the blockchain. The comparison results are made public on the blockchain. The bidders who want to change their valuation after knowing they are not the highest can again perform the bid submission phase with and try to do a higher valuation in a privacy-preserving manner to win the auction. The algorithm and working of this phase are discussed in detail in upcoming sections.
* *Market clearance**phase*: When no bidder is willing to change the valuation, the auctioneer declares the by publishing the as the final winner, based on the results of the last bid comparison phase to the blockchain network, and the connects to the seller to purchase the auctioned good. Figure 3 demonstrates this phase.

|  |
| --- |
| **Protocol:**   1. Interested entities submit their to . 2. Certifier generates and for bidders and auctioneers, respectively.   , *, }*  = { , *, }*   1. The certifier publishes the public keys of bidders and auctioneers to the blockchain.   **Smart contract ():**   1. For every request for new registration   call ***userReg***() function:  **If** is not in the database**:**  **call** ***deposit()*** function**;**  generate **;**  update the to the database**;**  return **;**   1. ***deposit ()*** function is to deposit the registration fees from each bidder. |

|  |
| --- |
| **Market Clearance Protocol:**   1. propagates to the network. 2. declares himself the winner and contacts the seller to purchase the goods. |

**Figure 2**: Registration protocol & smart contract .

**Figure 3**: Market clearance phase protocol.

In the following two subsections, we elaborate on the working of the bid submission and bid comparison phase, along with the algorithms followed during these two phases.

**4.3 Bid submission phase**

This phase works with the help of smart contract Bidder submits its current valuation of the auctioned product, padded with its own generated large random number and the digital signature to . The submitted padded bid value of bidder is encrypted with the targeted bidder and further encrypted with the . decrypts the encrypted information and recovers encrypted upon which homomorphically adds a random fraction of its own large random number while keeping the other fraction secret to itself. then forwards this updated to by updating the private variable corresponding to . After receiving through, decrypts this information with and adds its random . After addition, forwards back the new updated to encrypted with by . Algorithm 1 demonstrates the working of the bid submission phase. The key consideration is that generates different random fractions of its random number for each Smart contract helps to verify all information submitted to the auctioneer. The computation that the auctioneer and the bidders perform during this phase is performed in an off-chain manner, and the transfer of the data from one entity to another is done via smart contract . Off-chain computation saves the extra burden of gas costs for encryption and decryption for the participants, and on-chain data transfer ensures trust and verifiability.

|  |  |  |
| --- | --- | --- |
| **Algorithm 1: Bid Submission Phase** | | |
|  | **Input:** | |
|  | **Output**: STPC Pairs | |
| **1** | ***if*** *current time!* ***=*** *phase1 timeout****:*** | |
| **2** |  | *submits to* |
| **3** |  | *decrypts and adds .* |
| **4** |  | *forwards to* |
| **5** |  | *decrypts and pads with* |
| **6** |  | *forwards updated to* |
| **7** |  | *fter decryption, it adds and keeps the information within itself.* |
| **8** |  | ***end*** |
| **9** | ***else:*** | |
| **10** |  | *Don’t accept any .* |
| **11** |  | ***end*** |

**4.4 Bid comparison phase**

This phase contains the steps through which the bidder , is declared to other bidders, and the opportunity is given to the bidders to increase their bid valuation and compete with the . To declare the highest bidder compares all the collected and by STPC protocol. Upon declaration of , the call is open to all remaining bidder if any of them want to change their bid valuation and compete with If some bidder wants to do so, then has to raise a request through the smart contract and submit the specific gas cost from its own deposit, upon successful submission of a request from bidder again Algorithm 1 is followed for . This phase continues until no bidder is willing to change their bid valuation, and the final highest bidder is the auction winner, which is declared publicly and is announced with the help of the market clearing phase. Algorithm 2 demonstrates the bid comparison phase. This phase executes in similar manner as the bid submission phase i.e. computations are performed off-chain, and data transfer is performed on-chain.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm 2: Bid Submission Phase** | | | | |
|  | **Input:** *, , ,* | | | |
|  | **output**:Highest Bidder | | | |
| **1** | ***if*** *current time!* ***=*** *phase 2 timeout****:*** | | | |
| **2** |  | *compares &* | | |
| **3** |  | *declares current and waits for a call to raise valuation.* | | |
| **4** |  | ***while*** *(willing to change bid == true)****:*** | | |
| **5** |  |  | |  |
| **6** |  |  | | ***end*** |
| **7** |  | *Declare the last as the winner.* | | |
| **8** |  | ***end*** | | |
| **9** | ***else****:* | | | |
| **10** |  | | *don't perform a bid comparison* | |
| **11** |  | | ***end*** | |

**5 Experimental Evaluation**

**6 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 that has been collected during the bid submission phase*.

Proof: The data that collects concerning is , and these both can be represented as

(4)

(5)

As equations 4 and 5 contain a total of 4 unknown variables, and the number of equations is only two, 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.

Another threat that can arise in our scheme is the concern of collusion between the certifier and .

Theorem 2: *and on collusion can never get .*

Proof: , in collusion with , can get the private keys of and ; thus, can open the encryption of the bidders. This encryption can be opened at bid submission phase and on decryption auctioneer can get the information sent by . The information will be

(7)

(6)

In both equations, the bids are padded by the random number of the bidders, and they are secret to them only, so we can claim that the auctioneer can never disclose the bid values are safe till the bid submission phase and cannot be disclosed from collusion A and C.

Discuss with sir about shifting key generation to the bidders and certifier being the bulletin board.

Another threat that is crucial from the perspective of the proposes auction scheme is the safety from the collusion of &. In this aspect, we show that our scheme cannot avoid this threat, but we prove the condition of distrust in such a situation, where a participant will be reluctant to believe blindly the information provided by the other participants and solely rely on the information of others may lead them to lose the auction.

Theorem 2*: Collusion of to will always result in a distrust situation, and no one of them will be able to verify the truthfulness of the shared information.*

Proof: On collusion, *and*  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. On collusion, shares its and *.* can put this information in the data that it receives from in the bid submission phase, and it can be represented as

= (8)

recovers from this information upon putting the values of and *.* Upon the completion of the auction result, gets the public information

(9)

(10)

If it tries to validate the information shared by , it can verify this information using equations 9 and 10. From equation 9, it can know the auctioneer random number by adding , and further from equation 10, if it needs to verify the data of , it has to rely again on to share the correct value of , which can be easily changed because, for any sum such as , there always exits and both expression yield equal results so can always fake by and will have no way to verify the correctness of the shared information.

**6 Security Evaluation**

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 this hybrid 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 UIDs 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, or another approach to providing bid privacy is that bidders share the hash of their , and based on the hash value, the certifier issues them keys and .

*2) Bid Privacy:* Our proposed hybrid auction scheme achieves another security parameter, which is bid privacy; for each bid of bidder , we claim that bid privacy will be maintained from as well as . Theorem 1 and 2 prove that bid privacy is always maintained during and after the auction.

*3) Non-repudiation:* In the proposed hybrid auction scheme, the bidders are not allowed to change their bid submitted in a single iteration of the bid submission phase; is only allowed to change its bid valuation upon declaration of . changes its as a challenge to . Each bid submission is registered as a transaction on the blockchain; thus, our proposed hybrid auction scheme satisfies the non-repudiation of the bids.

**7 Conclusion**