

Auction Biding based **On Blockchain**

A report submitted in partial fulfillment of the
requirements for the degree of
Bachelor of Technology
in
Information Technology

by

Anurag Upadhyay(20198017)
Aman Singh (20198056)
Anshuman Bhargava (20198110)
Dikshita Bhatia (20198019)

under the guidance of

Dr. Dinesh Kumar



to the

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
Motilal Nehru National Institute of Technology, Allahabad,
Prayagraj, Uttar Pradesh (India)

Acknowledgements

The completion of this project required a lot of effort, guidance and support from many people. We feel privileged and honoured to have got this all along the development of the project. We would like to express our gratitude to our mentor Dr. Dinesh Kumar for their perennial support, guidance and monitoring throughout the making of this project. We would also like to acknowledge and thank our professors, colleagues and seniors for supporting us and enabling us to successfully complete our project.

Contents

Preface

Acknowledgements

1 Introduction

1.1 Motivation

1.2 Blockchain e-Auction

2 Related Studies

3 Analysis of problem statement

4 Proposed Work

4.1 Blockchain

4.2 Smart contract

4.3 Ethereum

4.4 Some Common Auction

4.5 Selead Bid Auction

4.6 Metamask

5 Implementation

5.1 IDE Setup

5.2 Code

6 Conclusion and Future Work

6.1 Conclusion

6.2 Future Work

Reference

Chapter 1

Introduction

we proposed a sealed-bid e-auction scheme based on blockchains with commitment algorithm, smart contracts, and zero-knowledge proof to protect the bid information from leakage and verify the auction result with all bidders anonymously, which successfully implemented the secure and fair auction without the third-party auctioneer.

1.1 Motivation

Online e-auctions are one of that have made it more convenient for customers to buy cheap products and have made it easy for sellers to be sellers and also third party payment options for secure payments. It is not a physical location, so you can make offers on your perfect home while relaxing at home or while on your work. Using e-auction will save you both time and money. By doing your shopping online, you won't waste time traveling to a bunch of properties. This is mainly because e-auction removes the physical limitations of traditional auctions such as space, time geography, and a small target audience.

1.2 Blockchain e-Auction

e-Auction improves the efficiency of bid transaction. However, the protection of bidders' privacy, transaction fairness and verifiability, transaction data security, high cost of third-party auction center, and other issues have attracted more attention. According to the transaction process and basic principles of the sealed auction, we explored the problems existing in the current sealed-bid e-auction schemes. Based on the blockchain technology, we proposed a

sealed-bid e-auction scheme with smart contract technology . We conducted the experiment to show that the proposed scheme protected the bid information from leakage well and successfully verified the winning bid price and the related bidder by all transaction participants without the third-party auctioneer.

Chapter 2

Related Studies

At present, there have existed a lot of related studies on sealed-bid e-auctions.

Franklin and Reiter (1995) presented a distributed service for performing sealed-bid auctions. This service can issue secret bids to the service for an advertised auction. Once the bid period ends, the auction service opens the bids, which determines the winning bid. Using novel cryptographic techniques, the service is constructed to provide strong protection for both the auctioneer and trusted bidders.

Brandt et al. proposed an approach that does not rely on trusted third parties, for example, auctioneers. The proposed technique is based on ElGamal encryption to protect the bidders' privacy. However, all computation in the auction was only performed by the auctioneer and time-consuming .

Peng et al. designed a blockchain-based sealed-bid auction with concurrent signature. In their scheme, the fuzziness of concurrent signature is used to protect the privacy of bidders and hide the bid price

Chapter 3

Analysis of Problem Statement

In blind e-Auction all bidders submit their bid within a certain time period but they are not visible to other until the bidding period not ended . Initially no one knows how much others are bidding and only final result is revealed once the bidding period is finished.

A bidder send hashed value of a bid instead of bid, i.e bidders commits to its bid with its hash value. After bidding period , bidders have to reveal their bids so they send their values encrypted form and contract checks that hash value is same as on provided during bidding period.

Chapter 4

Proposed work

4.1 Blockchain

Blockchain as a decentralized distributed ledger technology has become a hot research topic in recent years. Blockchains protect the data with encryption technology and consensus mechanism. Different from the traditional file systems, blockchains use chained file storages, put data records into blocks, and organize blocks into chain. It has several significant characteristics.

(i) Decentralized:- It does not require a trusted central point to maintain the ledger, by leveraging the consensus mechanism.

(ii) Transparent:- All data stored by blockchain are public and can be accessed by users.

(iii) Immutable:- Each block formed by some verified transactions is linked to the previous block via a secure hashing process. This approach makes the data on the chain impossible to be manipulated or deleted after it has been validated.

4.2 The Smart Contract:- smart contracts as a set of promises, specified in a digital form, including protocols within which the parties perform on these promises. From common law, economic theory, and contractual conditions often found in practice, four basic objectives of smart contracts are designed. The first of these is observability, the ability of the principals to observe the performance of the contract of each other or to prove their performance to other principals. A second objective, verifiability, is the ability of a principal to prove to an arbitrator that a contract has been performed or breached, or the ability of the arbitrator to find this out by other means. The third objective of contract design is privity, the principle that knowledge and control over the contents and performance of a contract should be distributed among parties only as much as necessary for the performance of that contract. The fourth objective is enforceability and at the same time minimizing the need for enforcement

4.3 Ethereum:- Ethereum is a blockchain-based decentralized platform featuring smart contract functionality. Smart contracts in Ethereum can facilitate and verify the process of a contract, base on the powerful Ethereum virtual machine (EVM). The mechanism supports for Turing-complete scripting, and thus makes it feasible for us to design various applications. Meanwhile, smart contracts also inherit the blockchain's properties on decentralization, immutability, and verifiability. For example, after deployed to Ethereum, the smart contract code cannot be modified by any user even for its creator.

4.4 Some Common Auction Types

- **The English Auction:-** The most common of the auction formats, goods are sold to the highest bidder with bids taking place in ascending order. Frequently, a reserve price must be met. The reserve price is the lowest price at which the auctioneer will sell the goods. This price is sometimes disclosed to the bidders and sometimes not.
- **The Dutch Auction:-** In a Dutch auction, bidding starts at an extremely high price and is progressively lowered until a buyer claims an item by calling "mine," or by pressing a button that stops an automatic clock.

When multiple units are auctioned, normally more takers press the button as price declines. In other words, the first winner takes his prize and pays his price and later winners pay less. When the goods are exhausted, the bidding is over.

- **The Vickrey Auction:-** It is also known as the second-price auction. Bids are sealed and the item is awarded to the highest bidder but at a price equal to the second highest bidder's price. If, for example, three bids are received, one for \$100, one for \$90 and one for \$75, the winner will be the \$100 bidder. However, the winner will only have to pay \$90, the second highest bidder's price.
- **The Sealed Bid Auction:-** As the name implies, this auction uses a sealed bid, where each bidder is allowed to bid only once. Generally, there are two steps to the process. First, the requirements are established by the buyer and, second, the sealed bids are opened. The highest qualified bidder receives the goods or, in the case of a service, the lowest qualified bid wins.
- **The Reverse Auction:-** In a Reverse Auction, the seller provides bids for a seller's requirements. At the end of an allotted period of time, the bid is awarded to the lowest priced, qualified supplier.

4.5 Sealed Bid Auction

sealed-bid auction is also referred to as a blind-auction. This is because the bidders in a sealed bid auction have no knowledge of what the other bidders are submitting. Traditionally, each of the participants will place their bid in a sealed envelope to be opened by the auctioneer. Then the best bid wins.

In a sealed-bid auction, bidders can only submit one sealed bid and therefore cannot adjust their bids based on competing bids. This sets it apart from the more common English auction, also known as the open ascending price auction, where

participants can make multiple bids and bid against each other. A sealed-bid auction process may also not be as transparent as an English auction. The seller retains a tremendous amount of control in a sealed-bid auction because they can see how each bidder values the property up for sale. Sealed-bid auctions are generally used in bidding for government contracts.

KEY TAKEAWAYS

- A sealed-bid auction is a type of auction in which bids are not viewed until the auction date.
- The bids are sealed, often physically in an envelope, and are all opened at once.
- Sealed-bid auctions are generally used in bidding for government contracts.
- Unlike an open bid, where buyers can make multiple bids and compete against each other actively, in a sealed-bid auction, they only get once chance.

4.6 Metamask

Metamask is therefore the only Ethereum blockchain-only wallet. you will do something with the blockchain you will need and a crypto wallet. Your personal key to communicating with the private world is your wallet in the blockchain. Allows you to purchase and transfer products. MetaMask is the world's most unique ethereum blockchain wallet.

The main features:-

- **Buy:** Put simply, you can buy Ether or other ERC-20 tokens and deposit them into your MetaMask wallet. There are normally a couple of different payment gateways you can use, so you will be able to select the most suitable for your transaction and location.
- **Send:-** As you would expect, you can send Ether with MetaMask too. By typing in the public key for

the wallet of the person you wish to send to, you can then type in the amount you wish to send.

- **Swap:-** Aside from buying and sending, you can also search through various exchanges to find tokens to swap, such as Ether and other ERC-20 tokens.

Chapter 5

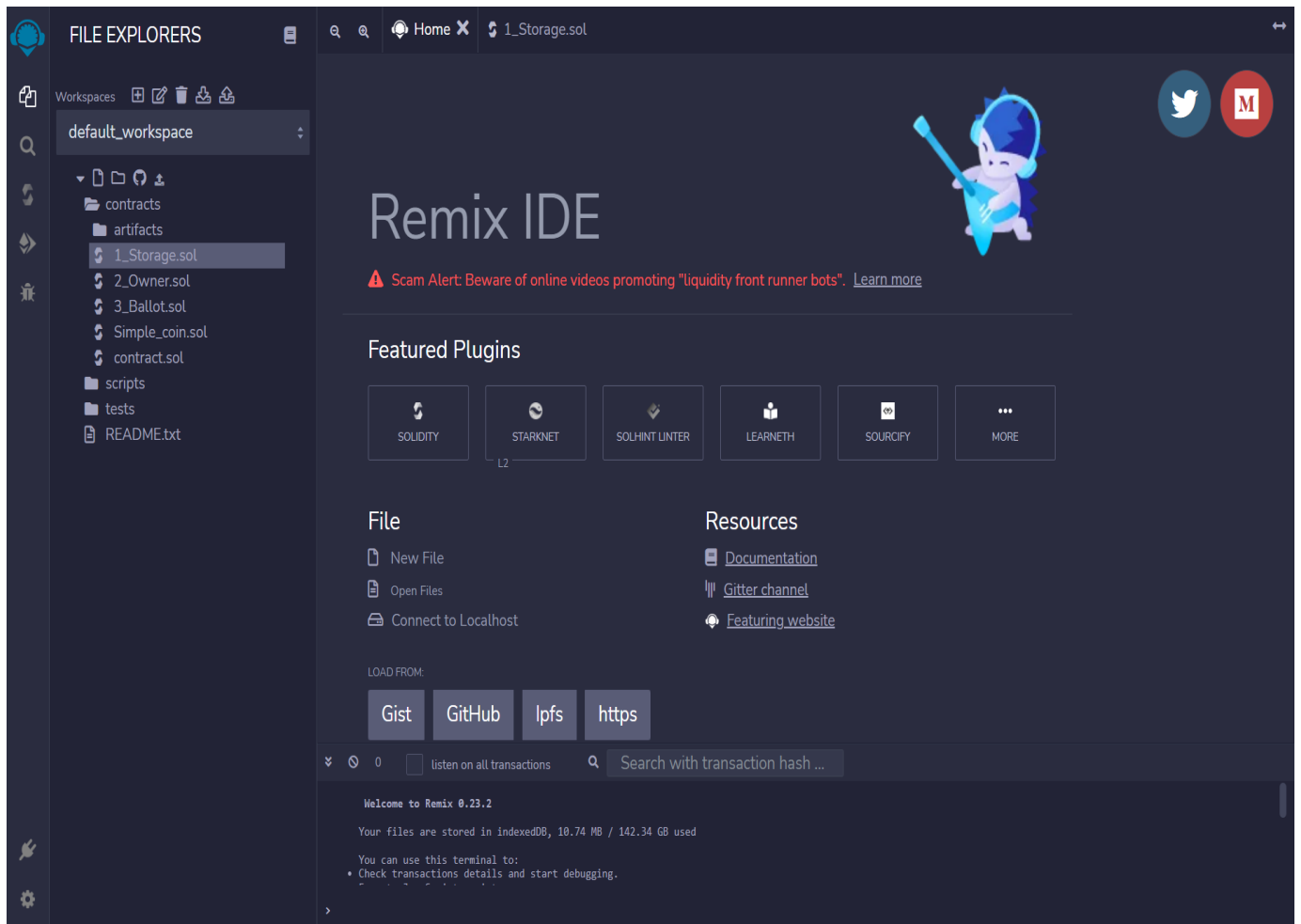
Implementation

5.1 IDE Setup:-

Remix IDE:- Remix is most commonly referred to as Remix IDE (Integrated Development Environment), although this is somewhat of a simplification. It is an open-source web and desktop application, a development environment, if you will. It packs a rich set of plugins and fosters a fast development cycle via intuitive GUI. Moreover, Remix IDE is primarily used for the entire process of smart contract development. In addition, it serves as a playground for teaching and learning how to use the Ethereum network.

Remix IDE Layout

- **Icon Panel** – An area where you click to change which plugin appears in the “Side Panel”.
- **Side Panel** – An area where most (but not all) plugins will have their GUI shown.
- **Main Panel** – It now offers tabs where plugins or files for the IDE to compile can be.
- **Terminal** – An area where you’ll see the results of your interactions with the GUI’s. You can also run your scripts here.



Remix Ide

5.2 Code:-

Functions:-

1:-placeBid():- It make sure transfer is done , bid is actually recorded inside our objects and we will map.

2. withdraw():- Once auction is ended , each person will come in and call this function to withdraw the funds they initially bid and didn't win.

3. reveal():-Once auction is ended then we pay out whoever needs to be paid but before that we are goinf to reveal all the bids and reveal what was the highest bid.

4. bid():- It is used to record the bids that will be placed during the auction .

5. auctionEnd() :- This function is called when we auction get to end and then whoever is the beneficiary gets the money after the end of the reveal time.

6. generateBlindedBidByte32() :- when the bidder puts the bid , it returns the hashed value of the bid so that no one can know who made the bid until the reveal period .

Final code:-

```
// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.7.0 <0.9.0;

contract BlindAuction{
    //variables
    struct Bid{
        bytes32 blindedBid;
        uint deposit;
    }

    address payable public beneficiary; // final amount goes to this address
    uint public biddingEnd;
    uint public revealEnd;
    bool public ended;

    mapping(address=>Bid[]) public bids;
    address public highestBidder;
    uint public highestBid;

    mapping(address=> uint) pendingReturns; // shouldn't be public as it will show the
bids.

    //events

    event auctionEnded(address winner, uint highestBid);

    //modifiers
    modifier onlyBefore(uint _time){ require(block.timestamp < _time); _; }
    modifier onlyAfter(uint _time){ require(block.timestamp > _time); _; }

    //functions

    constructor(uint _biddingTime, uint _revealTime, address payable _beneficiary){
        beneficiary=_beneficiary;
        biddingEnd = block.timestamp + _biddingTime;
        revealEnd = biddingEnd + _revealTime;
    }
}
```



```

    function generateBlindedBidByte32(uint value, bool fake) public pure returns
(bytes32) {
    //for hashed value of bids
    //using this function we will call our bid() function
    return keccak256(abi.encodePacked(value,fake));

}

function bid(bytes32 _blindedBid) public payable onlyBefore(biddingEnd) {
    bids[msg.sender].push(Bid({
        blindedBid: _blindedBid,
        deposit: msg.value }));
}

function reveal(
    uint[] memory _values,
    bool[] memory _fake
)
    public
    onlyAfter(biddingEnd)
    onlyBefore(revealEnd)
{
    uint length = bids[msg.sender].length;
    require(_values.length == length);
    require(_fake.length == length);

    for (uint i=0; i<length; i++) {
        Bid storage bidToCheck = bids[msg.sender][i];
        (uint value, bool fake) = (_values [i], _fake[i]);
        if (bidToCheck.blindedBid != keccak256(abi.encodePacked(value, fake))) {
            continue;
        }
        if(!fake && bidToCheck.deposit >= value) {
            if (!placeBid(msg.sender, value)) {
                payable(msg.sender).transfer(bidToCheck.deposit * (1 ether));
            }
        }
        bidToCheck.blindedBid = bytes32(0);
    }
}

```

```

}














function auctionEnd() public payable onlyAfter(revealEnd){
    require(!ended);
    emit auctionEnded(highestBidder, highestBid);
    ended=true;
    beneficiary.transfer(highestBid * (1 ether));
}

function withdraw() public {
    uint amount = pendingReturns[msg.sender];
    if(amount>0){
        pendingReturns[msg.sender] = 0;
        // returning amount of non winners
        payable(msg.sender).transfer(amount * (1 ether));
    }
}

function placeBid(address bidder, uint value) public returns(bool success) {
    if (value <= highestBid) {
        return false;
    }
    if (highestBidder != address(0)) {
        pendingReturns[highestBidder] += highestBid;
    }
    highestBid = value;
    highestBidder = bidder;
    return true;
}
}

```

Code Images:



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT

JavaScript VM (London)

ACCOUNT

0x5B3...eddC4 (99.9999999999999933076 ether)

GAS LIMIT

3000000

VALUE

0

Ether

CONTRACT

BlindAuction - miniProject.sol

DEPLOY

_BIDDINGTIME

300

_REVEALTIME

240

_BENEFICIARY

0x5B38D6a701c568545dCf803FcB875f56beddC4














transact

☐ Publish to IPFS

OR

At Address

Load contract from Address



DEPLOY & RUN TRANSACTIONS

▼ BLINDAUCTION AT 0XD91...39138 (MEMORY)

auctionEnd

bid

reveal

withdraw

beneficiary

biddingEnd

bids

ended

generateBlind...

highestBid

highestBidder

revealEnd

0x01afee42731aec5ddf3c9dd438ab81358424ca97b1c4ffa51a8db130e0ef2d61

uint256[] _values, bool[] _fake

0: address: 0x5B38D6a701c568545dCf803FcB875f56beddC4

0: bytes32: blindedBid 0x01afee42731aec5ddf3c9dd438ab81358424ca97b1c4ffa51a8db130e0ef2d61

1: uint256: deposit 1000000000000000000

0: bytes32: 0x01afee42731aec5ddf3c9dd438ab81358424ca97b1c4ffa51a8db130e0ef2d61

10,false

19

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Compared with the related sealed-bid auction schemes, the proposed scheme in this paper used the features of blockchain technologies to realize decentralized auctions. The risks from the third-party auctioneers were well eliminated.

The proposed sealed-bid e-auction scheme showed the following features:

(1) *Sealability*. All information in the auction transaction was encrypted by the public keys of the smart contract, owner, and bidders, which prevented the information from leakage. The bid price was only passed to the smart contract other than published on the blockchain. In addition, the bid price was mapped to a commitment price by the Pedersen commitment function. For all users getting the commitment price, they can never get the real bid price of bidders.

(2) *Fairness*. All bidders were equally treated by the smart contract. They got all commitment prices and the winning commitment price C . Then, they verified the auction result autonomously. If bidders tried to tamper with the auction result, their auction security would be confiscated and their permissions to the auction transaction were also frozen or canceled. The punishment was conducted automatically by the smart contract.

(3) *Validity*. The smart contract selected the winning bid price and the related bidder as the auction result, which obeyed the basic rule of the first-price sealed auction.

(4) *Nonrepudiation*. All information in the auction transaction was saved in the blockchain and can never be denied under the consensus mechanism of blockchains.

(5)Decentralized Verification. All bidders can verify and prove the auction result with zero-knowledge proof protocol (Bulletproofs). None of them can deny the bid price.

(6)Cost-Effective. The scheme was free of the cost of the third-party auctioneer, which made the biggest benefits of all auction parties.

6.2 Future Work

In this project, accuracy is a great point of concern. There are many areas where we can improve our system. Few of them are listed below.

- We can implement other different types of auction using solidity language and its oops concepts.
- More security can be provided in terms of auction bidding to ensure smooth and fair bidding.
- Frontend part can be added to take input from it and show output on it and not all functionalities need to run by us.

References

- [1] Verifiable Sealed-Bid Auction on the Ethereum Blockchain by Hisham S. Galal and Amr M. Youssef Concordia from Institute for Information Systems Engineering, Concordia University, Montr´eal, Queb´ec, Canada
- [2] Designing smart-contract based auctions by Chiara Braghin¹ , Stelvio Cimoto¹ , Ernesto Damiani^{1,2} , and Michael Baronchelli . Centre on Cyber-Physical Systems, Khalifa University, Abu Dhabi, UAE
- [3] <https://cloudname.com/en/what-is-metamask/>
- [4] <https://moralis.io/remix-explained-what-is-remix/>