CS711: Game Theory and Mechanism Design **Sep-Dec 2020**

Project Report: Decentralised Mechanism design using Blockchains

Group member(s): Abhimanyu Sethia, Atharv Singh Patlan, Rohan Baijal, V Pramodh Gopalan, Yatharth Goswami

Abstract

Most common mechanisms may be manipulated by corrupting a presumed central authority (who conventionally implements the mechanism design) or by revealing the bids/transactions of other agents in the mechanism. In this report, we motivate the need for a decentralised mechanism that preserves privacy; discuss such a mechanism in detail for auctions and simulate auctioneer-less privacy-preserving auctions using existing frameworks. We also model the auction as a game and derive some results to assert that such an auction mechanism is a win-win for both the seller and the bidders.

1 Introduction to the problem

Consider the following examples wherein a mechanism has been manipulated-

1. **Boston School Choice Mechanism**: This mechanism (popularly used to allocate seats in public schools to students) is as follows: (1) students are ranked according to some criteria, (2) students submit a preference order of their most preferred schools, (3) the mechanism tries to assign each student her first choice; in case of over-demand, the seats are allotted to students in decreasing order of ranks (4) the same is done, then for the second choice of the students and so on until either all students are allotted or all the seats are exhausted.
   Now, let there be two colleges C_1 and C_2, each having one vacant seat and three students A,B and C ranked as $A > B > C$ (according to some criteria). If all three of A, B and C fill college C_1 as their first preference and C_2 as their second preference, then according to the mechanism, A will be allotted C_1, B will be allotted C_2 and C will not get either. But now, if C gets to know the choices of B beforehand (either by corrupting the college admission committee or otherwise), then C may strategically change his first preference to C_2 and second preference to C_1. Thererby, C will get an admission in C_2, while B would fail to be admitted in both the colleges.[1]
2. **First Price Auctions:** Assume that the auctioneer acts on behalf of the seller but his interests are not aligned to that of the seller. Now, if the auctioneer is corrupted, he may propose a bid rigging to the highest bidder, at b_1, and allow him to reduce his bid to the second highest bid, b_2 plus some positive number, ϵ. The highest bidder claims the item and pays $b_2 + \epsilon$. The auctioneer and the bidder can share the surplus. The seller loses $b_1 - b_2 - \epsilon$ in potential revenue. Countless such cases of corrupting government officials conducting procurement auctions have been documented [2].
3. **Second price Auctions:** This mechanism may also be manipulated by an auctioneer, whose interests are aligned to that of the seller. After the auctioneer sees the bid, he may ask the second highest bidder to increase his bid to just below the highest bid. The auctioneer can share the gains made and the second-highest bidder does not lose anything. [3]

We observe 2 common reasons in these motivating examples-

1. A corrupt central authority (admission committee in the first case, auctioneer in the second and third)
2. The revelation of a party's bids/choices to the other parties

Corrupting such a trusted centre makes almost all mechanisms susceptible to manipulation. This motivates the need for a decentralised mechanism (wherein we eliminate, to a great extent, the dependence on a central authority), which preserves the privacy of bids/data of all parties from the other parties. In this report, we present such a decentralised auctioneer-less auction mechanism. Most of the ideas presented, however, may be applied to any decentralised mechanism, in general.

1.1 Related work

1. The field of Distributed Algorithmic Mechanism Design came up to tackle the problem of trusting a central party by off-loading computation to the agents. However, the agents could try to manipulate computations (ex. by understating other's bids) in order to gain higher utility. [4] talks about redundant computation by getting multiple players to calculate the same result and getting a central party to step in and punish the agents if results differ. We notice that we can consider BlockChain as a game where, people are incentivized to do honest computation. By combining the idea of Smart Contracts, which is just immutable code, we can run the rules of any mechanism and ensure a consensus of the network without any central party due to the properties of BlockChain.

2. [5] presents an implementation of simple auctions that maintain the privacy of bids, using blockchain, homomorphic encryption schemes and zero knowledge proof. Every bidder is supposed to submit a bid commitment using Pederson Commitment scheme (a homomorphic encryption scheme) to a smart contract in the blockchain. The decrypted bids are however, shared only with a semi-trusted auctioneer. The auctioneer then sorts the bids (decrypted) and anounces the winner. The auctioneer acts as a prover and the the smart contract acts as a verifier. Since smart contract is accessible to all the nodes, Zero knowledge proofs were used to verify the result to the smart contract. However, the propsed solution still needs a trusted auctioneer, who has access to the decrypted bids of the parties.

3. [6] proposed a novel and efficient solution to comparing two numbers which are private to two parties with the help of an oblivious third party who learns nothing about the two numbers (not even which one is bigger). This involves a novel construction, based on homomorphic encryption, with Φ-hiding assumption. The paper also, talks briefly about a protocol for conduction of cryptographically-secure sealed bid-auctions with only two auction servers, wherein the bidders would require only one round of interaction with auction service and the auction servers would require $O(n)$ rounds of interaction. However, the auction can be easily manipulated, if the third party colludes with at least one bidder.

4. **Enigma Protocol** [7] is the most recent solution to privacy in decentralized systems. The paper discusses an inherent trade-off in privacy by dividing BlockChain systems into two : fully decentralized with all data public and a somewhat centralized network where some nodes are trusted to ensure privacy. By developing efficient implementations of secret-sharing and having an off-chain secure database in which data and computational responsibilities are distributed in a manner similar to Ethereum, the Enigma Protocol ensures a greater degree of privacy while maintaining decentralization and scalability.
   The most recent application making use of these concepts is the Secret Network which is a new BlockChain network which essentially functions like Ethereum but also has a functionality for privacy.

1.2 Preliminaries

1. **BlockChains** : BlockChain was proposed as a decentralized tamper-proof ledger which first records transactions and then, they are verified by a decentralized consensus process. Some features of BlockChain are:

   (a) Decentralized Network : All nodes must use computational power and each transaction must achieve agreement among all the nodes.

   (b) Tamper-proof ledger : The cryptographic primitives used ensure that the log of transactions cannot be tampered with, without spending all of the computational power all over again.

(c) Trustless but Secure trading : Digital asymmetric key signatures are used which ensure only the pair which possess the pair of keys can execute the transaction without any third party.

2. **Smart Contracts** : Smart Contracts are code which are stored on the BlockChain. The code is immutable and requires user to submit some fees in order to run it. The function calls are also mined in a block, which ensures that consensus is achieved on the state of the Smart Contract as well.

3. **Secret Network** : Secret Network combines aspects of both Blockchains and Smart Contracts to provide tamper-proof ledgers and get rid of a central designer simultaneously. To provide privacy of bids, it uses TEE's(Trusted Execution Environments) which provide strong privacy guarantees for programs that are run in them. Hence, inputs which are encrypted can be passed as arguments, which are then decrypted inside the TEE to produce correct output. Therefore, Secret Network has aspects of privacy, decentralization, and eliminates a possibly corrupt mechanism designer. The correctness of the output is verified by validators(akin to miners). The state of the transaction is committed to the secret network if and only if a consensus is reached among two-thirds of the validators.

1.3 Brief overview of the report

The model has We introduce the model we have built in Section 2. We present the major theorems derived, to ascertain the significance of an auctioneer-less privacy-preserving auctions in Section 3.1. Thereafter, we discuss an alternative algorithmic (and not hardware based) solution for conducting such an auction in Section 3.2. In section 3.3, a shortcoming of the current Enigma protocol has been described and a solution has been proposed thereafter. Section 4 provides the details of the implementations and simulations of auctions we conducted. The summary of the report and some ideas on future work have been provided in Section 5.

2 Formal model of the problem

Here, we model a game to determine in which cases, a person would choose to participate in a mechanism involving blockchains. We enumerate some positive (+) and negative (-) factors, which could be important for a user in making this decision-
(+) **Decentralised in nature:** Eliminates the risk of a corrupt central authority.
(+) **Security:** Blockchain ledger cannot be tampered easily by any malicious node.
(-) **Lack of Privacy:** Every transaction made by a node in the network is visible to all the nodes in the network.
Now, based on these factors, we model the situation as a game and derive some results using game theory perspectives.
We define a Normal Form Game, with:
- The set of players, $\mathcal{N} = \{Seller, Bidder\}$
- The set of actions available for the seller, $\mathcal{A}_s = \{Blockchain, Centralised\}$
- The set of actions available for the bidder, $\mathcal{A}_b = \{Participate, NotParticipate\}$

For concreteness, we restrict ourselves to the simplest and, arguably, most natural choices that is, we consider only privacy, security and benefit(in terms of money) to model the utilities. However, there might be other significant choices in other situations. So our choice here is certainly, with a loss of generality.
Let us define some utilities for the above discussed factors,

$$\alpha \rightarrow \text{Utility related to the privacy factor for the bidder}$$
$$\beta_b \rightarrow \text{Utility related to the data security factor for bidder}, \quad \beta_s \rightarrow \text{Utility related to the data security factor for seller}$$
$$\delta_b \rightarrow \text{Monetary utility for bidder}, \quad \delta_s \rightarrow \text{Monetary utility for seller}$$

3 Main results/findings

3.1 Theorems

Theorem 1 *For an auction, it is always beneficial for the seller to conduct an auction using blockchain, over conducting an auction without blockchain.*

Theorem 2 *If an auction is modelled as a game, then the game always has either Pure Strategy Nash Equilibrium or Weakly Dominant Strategy Equilibrium.*

Theorem 3 *For auctions over blockchain, if the blockchain mechanism retains privacy of the bids, then participating in the auction is always the dominant strategy for the bidder, over not participating in the auction.*

Proof: Proofs for the above three theorems are in appendix. ■

3.2 Secret Auctions using Blockchains (Algorithmic Solution)

Here, we will try to provide an algorithmic solution to building Decentralised and Private Auctions using Blockchains. Our algorithm works as follows:

- The smart contract maps randomly, the bidders' address to a bidder ID in the start.
- We will work with a simple VCG auction setting where bidders are nodes in the Blockchain and they are going to publish their encrypted bids in the smart contract, during the bidding time. This is done to verify the top 2 bids at the end, and provide bidders the chance to raise concerns if they feel the results have been unfair.
- The task at hand is to compute the winner among n bidders without revealing their bids, while also keeping the auction decentralized. The underlying solution here requires the solution of Yao's Millionaire problem (see third protocol from [8]) to be applied multiple times. The solution discusses encryption schemes and protocols via which the 2 parties can compute the result privately, albeit in a computationally expensive manner.
- We first run the protocol for p_1 and p_2, and find the higher bidder (p_M). Then we ask p_M to compare with p_3. Similarly, at each round we find a winner and get him to compare with the next player until all n players have compared the value once. This is an $O(n)$ scheme. Similarly, we can find the second highest bidder in $O(n)$ again. Finally, we could get the top two bidders to reveal their actual bids to the smart contract. If any other player feels there has been some cheating, he can publicly reveal his bid to the Smart Contract and the players would be appropriately punished. So, this check brings an incentive to use the correct values in the protocol.

3.3 Extension of Enigma Protocol

We observed that the current implementation of enigma is not very profitable for the worker nodes.

- In enigma, tasks are allotted to each node once per epoch (a set number of computation tasks) and if a node gets allotted a particular task, then it has to perform the computation no matter what the computation fee is. This can lead to losses for a node which is offered to compute a very resource intensive task, but with a comparatively low transaction fee offered.
- It has been proposed in [9] that a node will be given the option to reject computing an assigned task. However, a node is allotted only one task in an epoch which means that if a node rejects a task then it will have to wait through the whole epoch till it gets allotted a task again. Hence rejecting a computation will be a waste of resources and time.

We aim to resolve this problem by introducing a 2^{nd} lowest price auction, providing nodes the ability to select which task to execute. As and when the enigma contracts receives a task for computation, the nodes can choose to bid on the task and how much reward they would like to receive in order to perform the requested calculation. We ensure that once a node wins an auction for a particular task, it can not bid for another task in the same epoch, which removes any advantage for fast nodes and nodes in places with lesser computation costs (as they might otherwise win multiple auctions in an epoch). The proposed method follows the standard VCG auction scheme. All the players are incentivized to bid their actual valuations, as deviation might lead either losing the auction, or getting less payoffs than they would have otherwise received. We provide steps in the appendix to implement this mechanism in a similar way as provided in [7] in it's Worker Selection section.

4 Experiments/Simulations

Auctions using Smart Contracts: We ran a simulation of VCG auction on ethereum blockchain. We made a smart contract for the auction and deployed it on a private blockchain using **Ganache** and **Remix-Ethereum**. We found that while running the auction, all nodes can access the data stored inside the contract and as a result all the nodes on the network were able to see the highest bid and the highest bidder anytime during the auction. The scripts we used and the implementation details of the simulation may be found on *this github repository*.

Auctions using Secret Network: We simulated a VCG auction on the Secret Network on the Testnet. We created the tokens and the auctions as per the protocol and then used a script 'auction.sh' for a better interface. While running, we tried to decrypt the data of transactions (bids) made by other players and found that we couldn't access the data, since we didn't possess their keyring pair. Through this we could demonstrate a higher degree of security and privacy of the Secret Network. The scripts we used and the implementation details of the simulation may be found on *this github repository*.

5 Summary and Discussions

In this report, we motivated the need for decentralised privacy-preserving mechanisms in general and implemented a auctioneer-less privacy-preserving auction in particular. We also modelled the auction mechanism in a Normal Form Game and inferred that blockchain-based privacy-preserving auctions are the logical choice for implementing auctions, for both the bidders and the seller. Additionally, we have suggested an alternative algorithmic solution for privacy-preserving auctions and proposed an improvement in the current enigma framework.

This project may be further bettered by modelling the utilities in the blockchain-based auction model, based on some experimentation and statistical methods. Moreover, while we implemented only the VCG auctions in the Secret Network, but it may be extended to several other mechanisms.

References

[1] Tayfun Sönmez, Atila Abdulkadiroglu, and Tayfun Sonmez. School choice: A mechanism design approach. *American Economic Review*, 93:729–747, 02 2003.

[2] Dmitry Ivanov and Alexander Nesterov. Identifying bid leakage in procurement auctions: Machine learning approach. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC '19, page 69–70, New York, NY, USA, 2019. Association for Computing Machinery.

[3] Mohammad Akbarpour and Shengwu Li. Credible mechanisms. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, EC '18, page 371, New York, NY, USA, 2018. Association for Computing Machinery.

[4] David C. Parkes and Jeffrey Shneidman. Distributed implementations of vickrey-clarke-groves mechanisms. page 261–268, 2004.

[5] Hisham S. Galal and Amr M. Youssef. Verifiable sealed-bid auction on the ethereum blockchain. In *Financial Cryptography and Data Security - FC 2018 International Workshops, BITCOIN, VOTING, and WTSC, Nieuwpoort, Curaçao, March 2, 2018, Revised Selected Papers*, volume 10958 of *Lecture Notes in Computer Science*, pages 265–278. Springer, 2018.

[6] C. Cachin. Efficient private bidding and auctions with an oblivious third party. In *CCS '99*, 1999.

[7] Enigma Co. *Enigma Docs, Subsytem Architecture*, 2018 (accessed November 15, 2020). `https://www.enigma.co/protocol/SubsystemArchitecture.html`.

[8] Ashish Kumar and Anupam Gupta. Some efficient solutions to yao's millionaire problem. 10 2013.

[9] Guy Zyskind, O. Nathan, and A. Pentland. Enigma: Decentralized computation platform with guaranteed privacy. *ArXiv*, abs/1506.03471, 2015.

Appendices

A Proof of Theorem 1

Using the above, utilities we make the following game matrix

	Blockchain	Centralised
Participate	$(-\alpha + \beta_b + \delta_b\ , \beta_s + \delta s)$	$(\alpha - \beta_b + \delta_b, \delta_s - \beta_s)$
Don't Participate	$(\alpha, 0)$	$(\alpha, 0)$

Considering the seller's strategies, it can be easily observed that choosing Blockchains is a Weakly Dominant Strategy for him.

B Proof of Theorem 2

Using the above utilities as defined in the formal model, we make the following game matrix

	Blockchain	Centralised
Participate	$(-\alpha + \beta_b + \delta_b\ , \beta_s + \delta s)$	$(\alpha - \beta_b + \delta_b, \delta_s - \beta_s)$
Don't Participate	$(\alpha, 0)$	$(\alpha, 0)$

Now, let's try to infer some things from the above game matrix. In the case of bidder, since the values of the parameters defined above will vary with respect to applications, therefore we will have different cases with us.

Case 1: $-\alpha + \beta_b + \delta_b \geq \alpha$

There are further two cases two explore in this

- $\delta_b \geq \beta_b$

  This implies that

  $$-\beta_b + \delta_b \geq 0$$

  or

  $$\alpha - \beta_b + \delta_b \geq \alpha$$

. In this case, the bidder will have a Weakly Dominant Strategy of participating in a blockchain mechanism and hence the action set (Blockchain, Participate) will become a WDSE of this game in this case. Also, solving the two inequalities in this case simultaneously, will lead to the inequality $\delta_b \geq \alpha$. So, we can infer that if a person values the privacy less than benefit than we should end up in this WDSE.

- $\delta_b < \beta_b$
  This simply implies that the strategy sets (Blockchain, Participate) and (Centralised, Don't Participate) are the two PSNEs of the game. Solving the two inequalities also leads to $\beta_b \geq \alpha$. Therefore, we can infer that if a person values security more than privacy than the game ends up with these two actions as PSNEs.

Case 2: $-\alpha + \beta_b + \delta_b < \alpha$

- $\delta_b \geq \beta_b$
  This implies that

  $$-\beta_b + \delta_b \geq 0$$

  or

  $$\alpha - \beta_b + \delta_b \geq \alpha$$

  . This simply implies that the strategy sets (Blockchain, Don't Participate) and (Centralised, Participate) are the two PSNEs of the game. Solving the two inequalities also leads to $\beta_b < \alpha$. Therefore, we can infer that if a person values security less than privacy than the game ends up with these two actions as PSNEs.
- $\delta_b < \beta_b$
  In this case, the bidder will have a Weakly Dominant Strategy of participating in a blockchain mechanism and hence the action set (Blockchain, Don't Participate) will become a WDSE of this game in this case. Also, solving the two inequalities in this case simultaneously, will lead to the inequality $\delta_b < \alpha$. So, we can infer that if a person values the privacy more than benefit than we should end up in this WDSE.

C Proof of Theorem 3

If the blockchains do not have the problem of making the bids public, then the above game matrix gets modifies as

	Blockchain	Centralised
Participate	$(\alpha + \beta_b + \delta_b\ , \beta_s + \delta s)$	$(\alpha - \beta_b + \delta_b, \delta_s - \beta_s)$
Don't Participate	$(\alpha, 0)$	$(\alpha, 0)$

We implicitly assume here that $\beta_b < \delta_b$ as the bidder is participating in the auction to get the item. Considering the seller's strategies, it can be easily observed that choosing Blockchains is a Weakly Dominant Strategy for him and considering the biddder's strategy we can see that Participating in a Blockchain mechanism is a Weakly Dominant Strategy for him. Hence, we can say that in such a case, the strategy tuple (Blockchain, Participate) will turn out to be a Weakly Dominant Strategy Equilibrium for the game.

D Implementation of worker allotment mechanism

1. All the tasks to be computed in an epoch are submitted by Ethereum nodes to the Enigma Nodes. (no transaction fee is proposed by the Ethereum nodes)

2. The principal node uses SGX's true random (i.e. *sgx_read_rand*) to generate a fresh random value (256-bit), which will later be used by the nodes as a seed.
3. It passes this value to the untrusted peer app running on the principal's host. The untrusted peer then commits it to the Enigma Contract. (MPC will be used in future to generate the random seed, but the method of generating the seed is not relevant here).
4. The Enigma Contract stores a mapping of: 1) the current block number; 2) the seed; 3) an ordered list representing a snapshot of all active workers.
5. The Enigma Contract emits a WorkersParameterized event. Every node in the network can observe this value, as they are all watching the chain. This tells the nodes about the nodes competing with them at the beginning of each epoch.
6. When the contract receives a compute request, it publishes a few details about the task such as the type of contract, number of inputs, values of the unencrypted inputs and other such details which do not reveal the actual computation.
7. Using the available details, the nodes need to guess the estimated gas they would spend on computing the task (using any type of analysis of their choice), and submit bids of the transaction fee they would like as a reward.
8. The Enigma Contract evaluates these bids, finds the lowest bidder, and assigns the task to that bidder, giving it a transaction fee demanded by the 2nd lowest bidder. The contract ensures that the person requesting the computation pays this fee. Ties are broken by arranging the nodes in an order determined by the random seed and choosing the first node with the lowest bid.
9. Thus, the enigma contract generates a taskId unique to that task and node. Then, it emits a ComputeTask event (see Computation). This ensures that the same node can not bid for two tasks in the same epoch.
10. The address of the winning node is revealed, and can be verified by any node by the results of the auction.
11. Now, all nodes in the network know the address of the worker selected for the task. Only the selected worker executes the computation task.
12. The selected worker commits the results on-chain including the block number that originated the task.
13. The Enigma Contract retrieves the worker selection parameters corresponding to the block number submitted.
14. The Enigma Contract verifies that the worker who submitted the block was the one assigned, by rerunning the auction winner determining algorithm. A greedy worker trying to compute more than its share of tasks would simply waste gas, as the unauthorized submissions get rejected by this verification method.