MEV Hackathon

Conor McMenamin

Bruno Mazorra

September 13, 2022

Abstract

We present a formal framework for analyzing protocol-level MEV, generalizing the loss-versus-balancing phenomenon beyond DEX protocols. We provide ROLVR, an MEV-redistribution protocol which we prove prevents LVR. We conclude by comparing ROLVR to recent attempts at protocol-level MEV redistribution, and demonstrate that ROLVR has best-in-class guarantees.

1 Introduction

The negative externalities of MEV have been well documented [4, 5, 3]. Recent MEV-Boost proposals, summarised in [11], aim to enact proposer-builder separation (PBS). Unfortunately, MEV-Boost has limited benefit to everyday users, providing no MEV protection at the protocol level. We believe protocol-level MEV protection is essential to ensure the long-term viability of the protocols that are needed to encourage the next level of blockchain adoption.

In this paper, we propose a framework for analysing protocol-level MEV opportunities, in line with recent work on loss-versus-rebalancing (LVR) in automated market-makers (AMMs)[8]. We generalise the concept of LVR beyond AMMs, and propose a new term, maximal LVR, which describes the amount of value extractable from a protocol given the only transactions included in a block are rebalancing transactions. With this in hand, we propose the redistribution of LVR (ROLVR, pronounced rollover) protocol which can be programmed to reduce LVR to 0 in any blockchain protocol. This is achieved by running a per-block auction for the right to submit the first transaction (or set of non-composable transactions) to protocols exposed to LVR. This auction is performed off-chain by an protocol-elected auctioneer in exchange for a fee, making the auctioneer process a repeatable game in which correct behaviour is a dominant strategy.

We compare ROLVR to another recent proposal aimed at preventing protocol-level MEV, call McAMMs [6]. Our comparison demonstrates that although both protocols provide comparable levels of MEV protection when MEV-searchers are competing, McAMMs exert a centralizing force which has the potential to reduce this MEV-protection. This is pre

2 A Formal Framework for Protocol-level extractable value

2.1 Maximal extractable value

In general, miner or maximum extractive value (MEV) is a term that refers to any excess profits that a block proposer or searcher can make based on transaction ordering and/or transaction inclusion. MEV was introduced in [2], formally defined in [1], and extended in cross-domain environments in [9]. Another similar definition was given in [10], except MEV was treated independently of a player. The MEV opportunities we consider in this paper are limited to the profits a player can obtain by modifying the blockchain state. In this section, we will formalize this kind of MEV opportunity using the concept of profitable 'bundles'. Then, we will define 'local' MEV as the maximally profitable bundle a specific player can construct. Similar to [9], we start by formally defining the domain and searcher:

Definition 2.1. A *domain* $\mathcal{D}$ is a self-contained system with a globally shared state $\mathtt{st}$. This state is altered by various agents through actions (sending transactions, constructing blocks, slashing, etc.), that execute within a shared execution environment's semantics. Each domain has a predefined consensus protocol that includes a set of valid algorithms to order transactions, denoted by $\mathtt{prt}(\mathcal{D})$.

A blockchain is a domain, however, there are other non-blockchain domains that also have MEV, like centralized exchanges. In our framework, we consider a publicly observable market price for all assets. We now define searchers, a class of players in our model who are able to extract MEV.

Definition 2.2. A *searcher* in a domain $\mathcal{D}$ is a player who assumes that transaction sequencers follow a specific set of rules and take strategic actions (send bundles with specific bids) to maximize their own utility. In general, we will assume that a searcher's utility depends linearly on their token balance values at the current external market prices.

Note, that searcher

In a domain $\mathcal{D}$ with state $\mathtt{st}$, the update of the state $\mathtt{st}$ after executing transactions $\mathtt{tx}$ is given by $\mathtt{st} \circ \mathtt{tx}$. For an ordered set of transactions $B = \{\mathtt{tx}_1, ..., \mathtt{tx}_l\}$, we have the composition $\mathtt{st} \circ B = \mathtt{st} \circ \mathtt{tx}_1 \circ .... \circ \mathtt{tx}_l$.

Similar to [1], we use $\mathtt{Addr}$ to denote the set of all possible accounts and $\mathbf{T}$ to denote the set of all tokens. We define $b : A \times \mathbf{T} \to \mathbb{Z}$ as the function that maps a pair of, an account and a token, to its current balance. More precisely, for $a \in \mathtt{Addr}$ we let $b(a, \cdot)$ denote the balance of all tokens held in a and $b(a, T)$ denote the account balance of token T. Abusing notation, we will denote by $b(a)$ the value of $b(a, \cdot)$ priced by a numéraire E. That is, if there is a pricing vector $p = (p_{T \to E})_{T \in \mathbf{T}}$, then $b(a) = p \cdot b(a, \cdot) = \sum_{T \in \mathbf{T}} p_{T \to E} b(a, T)$.

Definition 2.3. Let $\mathcal{D}$ be a domain with state $\mathtt{st}$, a player P with local mempool view $\mathcal{T}_P^M$ and a set of transactions $\mathcal{T}_P$ that the player P can construct. We denote by $\mathcal{C}_P = \mathcal{T}_P^M \cup \mathcal{T}_P$ to be the set of reachable transactions. We define the *local* MEV *of P with state $\mathtt{st}$* ($\mathrm{MEV}_P(\mathtt{st}, \mathcal{C}_P)$) as the solution to the following optimization problem

$$\begin{aligned} \max_{B} \quad & \Delta b(P; \mathtt{st} \circ B, \mathtt{st}) \\ \text{s.t.} \quad & B \subseteq \mathcal{C}_P, \\ & \mathtt{st} \to \mathtt{st} \circ B \text{ is a valid state transition in } \mathcal{D} \end{aligned}$$

Definition 2.4. Let $\mathcal{D}$ be a domain with state $\mathtt{st}$, $\mathcal{P}$ a set of players with local mempool $\mathcal{T}^M$ and a family of set of transactions $\mathcal{C} = \{\mathcal{T}_P\}_{P \in \mathcal{P}}$. We define the observable extractable value as

$$\mathrm{OEV}(\mathtt{st}, \mathcal{P}, \mathcal{C}) = \underset{P \in \mathcal{P}}{\mathrm{argmax}_2} \{\mathrm{MEV}_P(\mathtt{st})\}$$

2.2 Protocol extractable value

Protocol extractable value refers to the cross domain maximal extractable value that can be extracted through interacting with the transactions that just interact with the protocol. For a protocol $\mathfrak{P}$, we denote by $\mathcal{T}(\mathfrak{P})$ the set of transactions that the *access lists*[1] is included in the address and storage keys specified by the protocol $\mathfrak{P}$. In other words, $\mathcal{T}(\mathfrak{P})$ is the set of transactions that only interact with $\mathfrak{P}$. We say that two protocols are concurrent if they do not share access lists. In the following, we will define the MEV restricted to a specific protocol $\mathfrak{P}$, that is, the maximal value that a player can extract by just interacting with $\mathfrak{P}$.

Definition 2.5. Let $\mathcal{D}$ be a domain with state $\mathtt{st}$, $\mathcal{T}_P^M(\mathfrak{P}) \coloneqq \mathcal{T}_P^M \cap \mathcal{T}(\mathfrak{P})$ and $\mathcal{T}_P(\mathfrak{P}) = \mathcal{T}(\mathfrak{P}) \cap \mathcal{T}_P$. We define the local PEV of a player P as:

$$\mathrm{PEV}_P(\mathtt{st}, \mathfrak{P}) \coloneqq \mathrm{MEV}_P(\mathtt{st}, \mathcal{T}_P^M(\mathfrak{P}) \cap \mathcal{T}_P(\mathfrak{P}))$$

In general, the MEV is super additive over the composition of protocols. That is, for a pair of protocols $\mathfrak{P}_1$ and $\mathfrak{P}_2$ and a state $\mathtt{st}$, we have that:

$$\mathrm{PEV}_P(\mathtt{st}, \mathfrak{P}_1 \cup \mathfrak{P}_2) \geq \mathrm{PEV}_P(\mathtt{st}, \mathfrak{P}_1) + \mathrm{PEV}_P(\mathtt{st}, \mathfrak{P}_2).$$

[1] https://eips.ethereum.org/EIPS/eip-2930

We now define the concept of MEV-separable protocols.

Definition 2.6. Let $\mathcal{D}$ be a domain with two protocols P_1 and P_2. We say that P_1 and P_2 are $(\mathcal{T}^M(\mathfrak{P}_1 \cup \mathfrak{P}_2), \texttt{st})$–weak MEV-separable if for every player

$$\mathrm{PEV}_P(\texttt{st}, \mathfrak{P}_1 \cup \mathfrak{P}_2) = \mathrm{PEV}_P(\texttt{st}, \mathfrak{P}_1) + \mathrm{PEV}_P(\texttt{st}, \mathfrak{P}_2).$$

We say that P_1 and P_2 are strong separable if they are weak separable for every state.

We can then define the analogue of observable extractable value at a protocol level.

Definition 2.7. Let $\mathcal{D}$ be a domain with state $\texttt{st}$, $\mathcal{T}_P^M(\mathfrak{P}) \coloneqq \mathcal{T}_P^M \cap \mathcal{T}(\mathfrak{P})$ and $\mathcal{T}_P(\mathfrak{P}) = \mathcal{T}(\mathfrak{P}) \cap \mathcal{T}_P$. We define the observable PEV as:

$$\mathrm{OPEV}(\texttt{st}, \mathcal{P}, \mathfrak{P}) = \underset{P \in \mathcal{P}}{\mathrm{argmax}_2}\{\mathrm{PEV}_P(\texttt{st}, \mathfrak{P})\}$$

To describe how bundles should be ordered in our framework, we introduce the concept of effective gas price.

Definition 2.8. Following the Flashbots documentation [11], the effective gas price of a bundle B is defined as

$$sc(\texttt{st} \circ B) \coloneqq \frac{\Delta_{coinbase}(\texttt{st}) + \sum_{tx \in B \setminus \mathcal{TX}} g(tx) m(tx)}{\sum_{tx \in B} g(tx)}$$

where $\Delta_{coinbase}$ denotes the direct payment to the miner, $\mathcal{TX}$ is the set of mempool transactions, $g(tx)$ is the gas used by tx and $m(tx)$ is the gas price of tx.

2.3 Loss-versus-rebalancing

We now translate the definition of loss-versus-rebalancing (LVR) as introduced in [8], in line with the terminology of our paper. We consider a set of protocols $\mathfrak{P} = \{\mathfrak{P}_1, ..., \mathfrak{P}_n\}$ and a set of assets $A = \{a_1, ..., a_m\}$, where each asset $a_i \in A$ has a value $V(a_i)$ which can be traded with frictionlessly by a set of arbitrageurs. The value of any $\mathfrak{P}_i \in \mathfrak{P}$, denoted $V(\mathfrak{P}_i)$ can be represented as a linear combination of $\Lambda_i = \{\lambda_1, ..., \lambda_m\}$ of $V(A) = \{V(a_1), ..., V(a_m)\}$. Each protocol defines how this linear combination should change with respect to $V(A)$ (example: constant function market maker). Another way of interpreting these values is for a given block B_t, $V(\mathfrak{P}_i)$ describes the amount (Λ_i) of each asset (in A) controlled by $\mathfrak{P}_i$ at the market price ($V(A)$) when B_t was created .

Arbitrageurs are able to interact with each $\mathfrak{P}_i \in \mathfrak{P}$ to adjust $V(\mathfrak{P}_i)$ to reflect the changes in $V(A)$. For any given pair of adjacent blocks B_t, B_{t+1} and a protocol $\mathfrak{P}_i \in \mathfrak{P}$, the amount of value that can be extracted by an arbitrageur who frictionlessly readjusts $V(\mathfrak{P}_i)$ vs. trading at the prices described by $V(A)$ is denoted $R_{t+1}(\mathfrak{P}_i)$. In [8], $R_t(\mathfrak{P}_i)$ is the LVR of protocol $\mathfrak{P}_i$ in block B_t, for any $t > 0$.

We introduce *maximal LVR*, denoted $MR_t(\mathfrak{P}_i)$, which describes the maximal LVR of a protocol $\mathfrak{P}_i$ in block B_t. This maximal LVR is achieved given only rebalancing transactions are allowed to interact with $\mathfrak{P}_i$ in block B_t.

Observation 2.9. If the only transactions in a blockchain are rebalancing transactions, for any protocol $\mathfrak{P}_\mathrm{i} \in \mathfrak{P}$, competitive arbitrageurs bid $MR_t(\mathfrak{P}_i)$ for the right to rebalance $\mathfrak{P}_\mathrm{i}$ in block B_t.

This can be seen as any strategy bidding $MR_t(\mathfrak{P}_i) - \delta_i$, for some $\delta_i > 0$, is dominated by bidding $MR_t(\mathfrak{P}_i) - \delta_i + \epsilon_i$ for any ϵ_i with $0 < \epsilon_i < \delta_i$.

Observation 2.10. Given multiple transactions with arbitrary semantic meaning can be proposed for addition to a blockchain, arbitrageurs bid at least $MR_t(\mathfrak{P}_i)$ for exclusive access to submit transactions to $\mathfrak{P}_i$ in block B_t.

This follows from Observation 2.9, as the value of the ability to only submit rebalancing transactions must lower-bound the bid.

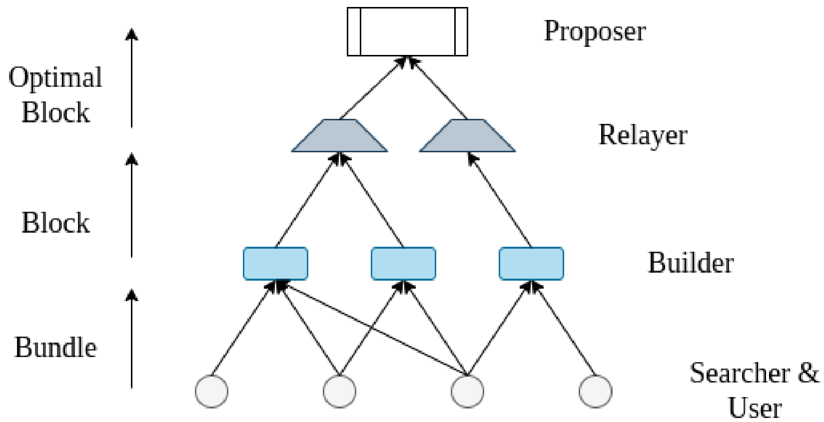

Figure 1: MEV Boost

3 ROLVR architecture

In this section, we introduce the ROLVR architecture. The ROLVR protocol uses the MEV-boost/PBS for a proper execution. In the following we introduce the participating parties of the MEV-boost protocol.

Definition 3.1. A *proposer* is a validator selected to choose the current block to be voted on by the set of validators, and potentially added to the chain. In PBS, the proposer signs the block header to be voted without seeing the block contents, and as such must trust the player(s) submitting the block header.

Definition 3.2. A *builder* is a player between searchers/users and validators who builds the most profitable valid blocks using searcher bundles and normal user transactions, and submits them to the validators via a relayer.

Definition 3.3. A *relayer*[2] is a player between builders and validators who runs auctions between builders for the right to publish blocks to the block proposer. The relayer collects prospective builder blocks/bids from builders, verifies that the bids correspond to the blocks, and at the end of the auction, sends the bid to the proposer. The proposer then signs the header for the block corresponding to the highest bid (from all relayers). The relayer then publishes the block body, and as the header is signed, the block is then added to the blockchain. Relayers are trusted.

Every round, builders send to relayer(s) a block, including a bid for the value the builder places on the block, with this bid to be paid to the validator if the auction is won. The relayer verifies this block matches the header, and sends to the proposer. Afterwords, the proposer of that slot chooses the header block with the largest payoff (this should be the largest bid) and commits to execute it by signing it. Upon receiving the signed block header from the proposer, the relayer sends the block and signed header to the validator network[3]. The only constraints that builders have is to build valid blocks with the consensus protocol semantics. Since the builders are trusted identities, to build trust among users and searchers, builders commit to ordering mechanisms. In ROLVR, this is considered to be effective gas price.

The ROLVR protocol introduces a new player, the auctioneer. The auctioneer is responsible for creating a set of pools, e.g. Uniswap-Fork of USD/ETH, creating mega-bundles (sub-blocks) and relaying them to a set of trusted builders.

Definition 3.4. An *Auctioneer* is a player in the ROLVR protocol between searchers & users and builders that construct mega-bundles with a specific Auction mechanism $(\mathbf{x}, \mathbf{pr})$, a distribution mechanism $\mathbf{D}$ and a bidding algorithm $\mathbf{B}$. In the section, 4 we will give a specific auction mechanism. The auctioneers compete against each other to maximize LP provider's and users revenue.

[2] https://writings.flashbots.net/writings/understanding-mev-boost-liveness-risks/
[3] More architecture details can be seen in https://github.com/flashbots/mev-boost

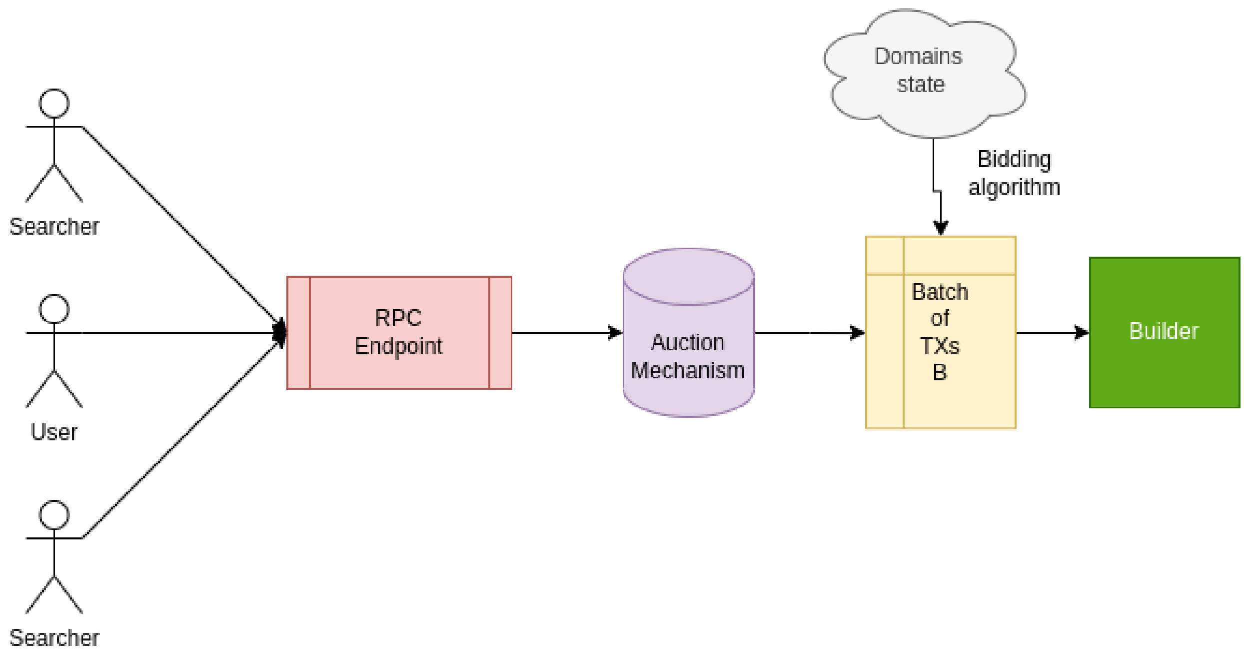

Figure 2: ROLVR architecture

The Auction mechanism $(\mathbf{x}, \mathbf{p})$ is the algorithm responsible for allocating the block-space to the users/searchers and recollect the payments. In general, as mentioned in [7] is a dominant strategy to not allow for reverted transactions and construct the block with higher payment. However, as mentioned in [7], this problem is a NP-problem known as the Knapsack auction problem with a conflicting graph. Nevertheless, there are greedy approximation algorithms that provide a sufficiently optimal solutions.

Given this concept of an auction mechanism, we consider the following property a basic requirement of auction mechanisms.

Property 3.5. An auctioneer algorithm $\mathbf{A} = ((\mathbf{x,p}),\mathbf{D},\mathbf{B})$ is *off-chain agreement proof (OFCA proof)* if using MEV-boost/PBS and the auction mechanism are incentive compatible. In other words, the auctioneer does not have incentives to provide a non-maximal bid to the builder/proposal, and pay another part of the bid through side-payment channels.

The distribution mechanism $\mathbf{D}$ is the algorithm responsible for distributing the MEV collected (in the case the payments induced by the auction) to the Liquidity provider and the traders (in general, the auctioneer will provide a private pool to minimize sandwich type of strategies). There are two important properties that a distribution mechanism should have, *strong against off chain agreements* and *strong against Fake MEV strategies* (see section 3.1)

The Bidding algorithm $\mathbf{B}$ consists of an algorithm that takes as input the current state of the Domains (Centralized exchanges, Layer 1 and Layer 2), the mempool and the value of the block produced by the auction mechanism and outputs a bid b. Then the auctioneer construct a batch of transactions $B = [tx_1, ..., tx_k]$ with the gas price b and broadcasts to the trusted builders. If the bidding is too close to the actual fees produced by the block B, then the MEV that can be distributed will decrease. On the other hand, if the bid is too small, the builder will not have incentive to include the batch of transactions in the next block.

3.1 Assumptions

Assumption 1: Searchers are able to frictionlessly trade at the external market price. In case of an arbitrage opportunity of value v between a limit order book (LOBs) and a CFMM, there exist at least two arbitrageurs that can extract the opportunity frictionlessly.

When an arbitrageur interacts with the pool, we assume they maximize their immediate profit by exploiting any deviation from the external market price. In other words, they transfer the pool to a point in the feasible set C that allows them to extract maximum value assuming that they unwind their trade at the external market price P. Equivalently, because trading in the pool is zero sum between the arbitrageur and the liquidity provider, arbitrageurs can be viewed as minimizing the value of the assets in the pool reserves.

Assumption 2: Searchers, Builders and Validators do not collude. That is, players do not generate bidding rings to lower the competition and censor transactions.

In reality, we do not observe collusion on searchers. Also, it is fair to expected that well-known MEV opportunities will be extracted by numerous searchers. Importantly, this assumption does not prevent validators from performing as builders or searchers, and vice versa.

Assumption 3: Liquidity providers are rational and want to maximize their long-term profits.

4 ROLVR Protocol

In this section we describe the ROLVR protocol, which for a set of participating on-chain protocols $\mathfrak{P}$, provides a programmable level of MEV protection up to, and potentially greater than it's maximal LVR.

For a protocol $\mathfrak{P}_i \in \mathfrak{P}$ and any given prospective block to be included in the blockchain, searchers participate in a first-price auction for the right to include he first set of non-composable transactions interacting with $\mathfrak{P}_i$ The auctioneer running the auction for $\mathfrak{P}_i$ is elected by a representative(s) of $\mathfrak{P}_i$ (e.g protocol DAO election, selecting from LPs in $\mathfrak{P}_i$ proportional to their liquidity, etc.). We envision the ability for auctioneers to delegate their auctions to meta-auctioneers who run multiple auctions simultaneously (an optimal solution from a communication complexity standpoint would be for all protocols in $\mathfrak{P}$ trusting one auctioneer, although the trust and centralization trade-offs in that cases are important to consider). For a protocol $\mathfrak{P}_i$ with winning bid b_i, we let αb_i be redistributed to $\mathfrak{P}_i$ for $0 < \alpha < 1$, with the remaining $(1 - \alpha)b_i$ used to reward the respective auctioneer, builder and/or proposer.

Consider a permissionless market between arbitrageurs, and auctioneers who are incentivised to execute the auction correctly (auctioneers are incentivised as the correct running of auctions is a repeated game where honestly executing the auction leads to re-election, and the earning of the fees that come with it). Auctioneers distribute winning bids to the set of builders, including a signature verifying the auctioneer has ran the auction. For any protocol $\mathfrak{P}_i \in \mathfrak{P}$, $\mathfrak{P}_i$ only accepts transactions in a block if the first transaction interacting with $\mathfrak{P}_i$ is accompanied by a valid signature of the auctioneer for $\mathfrak{P}_i$. The remaining architecture is in line with that of PBS and MEV-boost.

Based on this description of ROLVR, we now present the main result of the paper; that ROLVR protects protocols from LVR ($R_t()$), and can be tuned to effectively eliminate LVR. This result is described in Theorem 4.1.

Theorem 4.1. For a protocol $\mathfrak{P}_i \in \mathfrak{P}$, and a winning auction bid for $\mathfrak{P}_i$ of b_i for some block B, $R_t(\mathfrak{P}_i) \leq (1 - \alpha)MR_t(\mathfrak{P}_i)$.

Proof. Recall that $R_t(\mathfrak{P}_i)$ is the realised LVR of $\mathfrak{P}_i$, while $MR_t(\mathfrak{P}_i)$ is the maximal LVR of $\mathfrak{P}_i$. From Observation 2.10, we know that for any protocol $\mathfrak{P}_i \in \mathfrak{D}$ arbitrageurs bid at least $MR_t(\mathfrak{P}_i)$ for the right to exclusively interact with $\mathfrak{P}_i$. As such, $b_i \geq MR_t(\mathfrak{P}_i)$. Furthermore, in ROLVR we know that αb_i is redistributed to $\mathfrak{P}_i$. The result follows. □

5 MEV-capturing AMMs

In [6], an MEV-mitigation protocol called MEV-capturing AMMs (McAMMs) is proposed. In McAMMs, a given AMM protocol $\mathfrak{P}_i$ auctions off the right to submit the first transaction to $\mathfrak{P}_i$ for a series of blocks in the future. More than this, the first transaction interacting with $\mathfrak{P}_i$ must be that of the winning bidder. As such, McAMM auctions are predictive auctions for the value of being the first transaction in the specified series of blocks. In an auction among competing bidders. However, [6] do not explain how the MEV will be distributed to different pools and the as mentioned in section 3.1 this mechanism could be weak against fake-mev strategies. Moreover, is not clear that the winner of the auction (called the leader searcher) is sufficiently efficient, leaving MEV in the table that other searchers can capture and not distribute.

Theorem 5.1. For a McAMM auction in a censorship resistant blockchain between non-cooperating arbitrageurs on a protocol $\mathfrak{P}_i$ for blocks $B_t, ..., B_{t+k}$, the winning bid $b_i \geq \sum_{j=0}^{k} E(MR_{t+j}(\mathfrak{P}_i))$.

Proof. By definition, an arbitrageur with the ability to submit the first transaction to protocol $\mathfrak{P}_i$ in block B_t has an expected profit of at least $MR_{t+j}(\mathfrak{P}_i)$, $\forall j \geq 0$. As such, any strategy bidding less than $E(MR_{t+j}(\mathfrak{P}_i))$ for the right to submit the first transaction to $\mathfrak{P}_i$ in B_t is strictly dominated. The result follows. □

5.1 Comparison of ROLVR and McAMMs

McAMMs and ROLVR have interesting trade-offs. McAMMs auctions are ex-ante, requiring bidders to accurately price the expected volatility of the tradeable assets in $\mathfrak{P}_i$, while ROLVR auctions are ex-post, with bidders instead pricing the value of interacting immediately with $\mathfrak{P}_i$.

A crucial drawback of McAMMs versus ROLVR is the requirement for bidders to correctly price expected volatility over multiple time-step into the future, and for bidders to be able to handle the risk associated with the volatility of volatility. Due to the distribution of asset price movements, the pricing of tail movements (large moves up or down), the median outcome for a bidder will be to lose money. As such, this has a centralizing force on bidders, with bidders restricted to entities with significant capital and risk-tolerance. Such a centralizing force leads to the monopolization of the bidding process, the under-pricing of $MR_{t+j}(\mathfrak{P}_i)$ and the increasing of LVR.

Both protocols have single points of failures. ROLVR can be seen as vulnerable to off-chain agreements between auctioneers and proposers, as proposers have a clear incentive to minimise the reported bids, and maximize their own revenue. However, the auctioneer process can be made auditable by requiring bids to be broadcast publicly, and as auctioneers must behave correctly to be reinstated, off-chain agreements can effectively be prevented. The requirement for McAMM auctions to contain competitive arbitrageurs means the expectancy of the winning bidder approaches 0, with any rebates or fees gained by the winning bidder eventually priced in to the bid. In ROLVR however, the payoff for an auctioneer is increasing in MEV opportunities (proportional to $(1 - \alpha)b_i$).

References

[1] Kushal Babel et al. "Clockwork finance: Automated analysis of economic security in smart contracts". In: *arXiv preprint arXiv:2109.04347* (2021).

[2] Philip Daian et al. "Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability". In: *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2020, pp. 910–927.

[3] Philip Daian et al. *Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges*. https://arxiv.org/abs/1904.05234. Retrieved: 19/01/2022. 2019. arXiv: 1904.05234.

[4] Flashbots. https://explore.flashbots.net.

[5] Lewis Gudgeon et al. *The Decentralized Financial Crisis*. 2020. DOI: 10.48550/ARXIV.2002.08099. URL: https://arxiv.org/abs/2002.08099.

[6] Josojo. *MEV capturing AMMs*. 2022. URL: https://ethresear.ch/t/mev-capturing-amm-mcamm/13336.

[7] Bruno Mazorra, Michael Reynolds, and Vanesa Daza. "Price of MEV: Towards a Game Theoretical Approach to MEV". In: *arXiv preprint arXiv:2208.13464* (2022).

[8] Jason Milionis et al. *Automated Market Making and Loss-Versus-Rebalancing*. 2022. DOI: 10.48550/ARXIV.2208.06046. URL: https://arxiv.org/abs/2208.06046.

[9] Alexandre Obadia et al. "Unity is Strength: A Formalization of Cross-Domain Maximal Extractable Value". In: *arXiv preprint arXiv:2112.01472* (2021).

[10] Alejo Salles. "On the Formalization of MEV". In: *https://writings.flashbots.net/research/formalization-mev/* (2021).

[11] thegostep. *MEV Boost Archive*. 2022. URL: https://github.com/thegostep/awesome-mev-boost#articlespapers.