FairTraDEX: A Decentralised Exchange Preventing Value Extraction

Conor McMenamin*
Universitat Pompeu Fabra
Barcelona, Spain
Nokia Bell Labs
Nozay, France

Vanesa Daza
Universitat Pompeu Fabra
CYBERCAT – Center for Cybersecurity Research of Catalonia
Barcelona, Spain

Matthias Fitzi
IOG
Zurich, Switzerland

Padraic O'Donoghue
Independent Consultant
Dublin, Ireland

ABSTRACT

We present FairTraDEX, a decentralized exchange (DEX) protocol based on frequent batch auctions (FBAs), which provides formal game-theoretic guarantees against extractable value. FBAs, when run by a trusted third-party, ensure that the unique game-theoretic optimal strategy for all players is to trade at the true market-implied price of the underlying token swap, excluding explicit, pre-determined fees. FairTraDEX replicates the key features of an FBA that provide these game-theoretic guarantees using a combination of set-membership in zero-knowledge protocols and an escrow-enforced commit-reveal mechanism. We extend the results of FBAs to handle monopolistic and/or malicious liquidity providers. We provide real-world examples that demonstrate that the costs of executing orders in existing academic and industry-standard protocols become prohibitive as order size increases due to basic value extraction techniques, popularized as maximal extractable value. We further demonstrate that FairTraDEX protects against these execution costs, guaranteeing a fixed fee model independent of order size, the first guarantee of it's kind for a DEX protocol. We also provide detailed Solidity and pseudo-code implementations of FairTraDEX, making FairTraDEX a novel and practical contribution.

CCS CONCEPTS

• **Security and privacy** → *Privacy-preserving protocols*; *Economics of security and privacy*.

KEYWORDS

blockchain, decentralized exchange, extractable value, incentives

*Author is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 814284.

DeFi '22, November 11, 2022, Los Angeles, CA, USA

ACM ISBN 978-1-4503-9882-4/22/11.
https://doi.org/10.1145/3560832.3563439

ACM Reference Format:
Conor McMenamin, Vanesa Daza, Matthias Fitzi, and Padraic O'Donoghue. 2022. FairTraDEX: A Decentralised Exchange Preventing Value Extraction. In *Proceedings of the 2022 ACM CCS Workshop on Decentralized Finance and Security (DeFi '22), November 11, 2022, Los Angeles, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3560832.3563439

1 INTRODUCTION

One of the most prominent and widely-used classes of protocols being run on smart-contract enabled blockchains is that of decentralized exchange (DEX) protocols. DEX protocols allow a specific set of players, whom we call *retail users*, to exchange one token for another in the presence of *market-makers* (MMs), who provide liquidity-as-a-service to retail users. Interacting with a blockchain-based DEX typically reveals a player's intention to trade before the transaction is confirmed on the blockchain, and in doing so, presents what has become known as an *expected extractable value* (EEV) opportunity [16]. EEV refers to any expected profits a player can extract from other players interacting with the blockchain by manipulating the ordering of, injecting, and/or censoring transactions in prospective blocks.

A significant advancement in DEX protocols was the advent of *automated market makers* (AMMs), with Uniswap [27] being the most prominent of which. Projects like Flashbots [12] (a direct spin-off to [10]) have identified that AMMs are the main source of recorded EEV (> 98%, as seen in the chart labelled "Extracted MEV Split by Protocol" in [12], of the $665M in EEV identified by Flashbots since August, 2020). A peer-reviewed analysis in [25] identified $540.54M in extracted value up to August 2021, indicating the current number provided by Flashbots is significantly lower than the actual amount of extracted value being extracted from DEX protocols. In [18], it has been further highlighted that in Uniswap V3, liquidity providers are losing more to EEV attacks (impermanent loss in that case) than they are collecting in fees. It is clear that the long-term viability of existing DEX protocols is not plausible.

Although many attempts have been made academically to address this source of EEV [3, 4, 9, 13, 17], no satisfactory solution has been found. The protocols presented in these works remain vulnerable to basic EEV-extraction strategies, as outlined in Section 2. Therefore, there is a clear gap, both in literature and in practice, to provide a DEX protocol which definitively eliminates all sources of EEV. In this paper, we provide such a protocol.

1.1 Our Contribution

We introduce width-sensitive frequent batch auctions (WSFBAs), idealised commit-reveal exchange protocols between retail and MMs, based on FBAs [7]. WSFBAs are an important improvement on basic FBAs with respect to decentralised systems. WSFBAs ensure retail submit market orders in the presence of monopolistic MMs, compared to a standard FBA in which retail are required to submit limit orders at their interpretation of the true price of the swap, plus some trade fee. The requirement for retail to submit limit orders leads to worse order execution as trade probability is decreased, while also placing a significant burden on retail to track this market price. This burden is removed in a WSFBA, providing an "obvious optimal" for retail, as coined in [26]. Furthermore, in the case of competing MMs, a WSFBA provides equivalent equilibria to [7].

We then describe FairTraDEX, a blockchain-based implementation of a WSFBA. In FairTraDEX, order commitments are recorded on-chain (to enforce the corresponding escrow punishment). We utilize ZK set-membership proofs to allow retail to commit to their orders anonymously. As such, in FairTraDEX, every retail user must initially register to the protocol, depositing an escrow. Then, whenever a retail user wants to commit to an order, the user only has to prove membership of the player set registered in the protocol. Given enough registrations, the probability that a retail user's ZK set-membership proof/committed order relates to the actual order contents approaches 0 (we formalize this notion in Section 5). In other words, no other player in the system can see the committed order and use it to infer anything about what the order is. To definitively hide retail order information, orders are committed, including the ZK-proof, by using a relayer, a third-party who receives a fee for including relayed transactions in the blockchain (see Appendix A.1 for further details).

We provide an extensive Ethereum virtual-machine (EVM) compatible proof-of-concept for FairTraDEX [2] including a comparison of protocol running costs with previous solutions in Section 6, which remain constant with respect to order size, price and direction. When compared to potentially percentage-point slippages and EEV-attack costs required to trade on current DEXs, also highlighted in Section 6, FairTraDEX's formal guarantees of protocol-level EEV prevention and up-front, fixed and explicit costs set a new standard for DEX protocols.

2 RELATED WORK

The main works aimed at protecting DEX users from EEV either focus on preventing front-running of orders [4], the fair ordering of transactions based on their delivery time [17], or on hiding client trade information until the trade has been committed to the blockchain [3, 9, 13]. Of these works, the closest to our proposal are [4, 9, 13]. We briefly outline here how, when all players in the respective DEX protocols are rational, EEV opportunities exist.

In P2DEX [4], clients must publicly deposit the tokens with which to trade in the same time frame as the order matching takes place, exposing clients to standard identity- and directional-based EEV exploits. Separating token deposit and identity revelation from a client's commitment to a specific auction are important advancements used in FairTraDEX to protect against EEV.

Similar commit-reveal protocols to FairTraDEX for blockchain-based token-exchange are proposed in [9, 13]. The protocol in [13] is exposed to several game-theoretic exploits which contradict its protection against front-running. These include the necessity to reveal order direction a-priori, and the non-trivial handling of the linkability between commitments and account-balances. In [9], clients commit their own orders to the blockchain, revealing their identities, and corresponding token balances/execution patterns which can be used by a basic professional MM to skew prices and extract value from the client.

3 PRELIMINARIES

This section introduces the terminology and definitions necessary to understand the main results of the paper. By $negl()$ we denote any function $f : \mathbb{N} \to \mathbb{R}$ that decreases faster than any (positive) polynomial p. More formally, $\forall\, p\ \exists\ \lambda_0 \in \mathbb{N} : \forall \lambda > \lambda_0 : f(\lambda) < \frac{1}{p(\lambda)}$. For protocol correctness, we must assume that some of the involved players may be malicious trying to force the protocol into incorrect execution, and without any direct benefit for themselves. However, for the game-theoretic part of the analysis, we assume that all players are rational. Accordingly, the analysis of our protocol is based on two security parameters, a cryptographic security parameter κ used to bound the probability that the protocol execution is incorrect, and a game-theoretic extractable-value parameter ψ used to bound the extractable value of any player. The following are the crucial terms and definitions needed for the remainder of the paper, with supplementary terminology and definitions contained in Appendix A.

- *Notional Value*: The value of a set of tokens expressed in some common reference token. In this paper, we use the symbol ₿ as the reference token in which we measure notional, and with which we reason about utility.
- *External Market Price* (EMP, denoted y): As in [7], the EMP of a token/token swap is a publicly observable signal which is perfectly informative of the fundamental price of the underlying token/token swap. Moreover, a random order of fixed notional generated by a player in the system is equally likely to buy or sell tokens at the EMP, distributed symmetrically around the EMP. Unless otherwise stated, observing the EMP has a prohibitive cost for players in our system.
- *Market*: A market in a DEX between two tokens A_{tkn} and B_{tkn} consists of two limit orders, a *bid* and an *offer*. When the market is quoted from token A_{tkn} to B_{tkn}, the offer price indicates the quantity of token A_{tkn} a player must sell for 1 token B_{tkn}, while the bid price indicates the quantity of token A_{tkn} a player receives for 1 token B_{tkn}. In this paper, we represent such a market as bid @ $offer$, with $0 < bid \leq offer$.
- *Reference Price* (y_{ref}): For a market bid @ $offer$, the reference price y_{ref} is the price such that $\frac{bid}{y_{ref}} = \frac{y_{ref}}{offer}$.
- *(Market) Width* (w): For a market bid @ $offer$, the width is calculated as $w = \frac{offer}{bid}$ (as such $w \geq 1$).
- *Multiplicative Market-Impact Coefficient (δ)*: If the pre-trade EMP for particular swap is y, the expected post-trade EMP given a buy order is δy for some $\delta \geq 1$, while the expected post-trade EMP given a sell order is $\frac{y}{\delta}$. Unless otherwise

stated, a swap from A_{tkn} to B_{tkn} with multiplicative market-impact coefficient δ corresponds to buy orders of B_{tkn} having a multiplicative market-impact coefficient on y_B of $\sqrt{\delta}$ and $\frac{1}{\sqrt{\delta}}$ on y_A. Given our definition of the EMP, this impact function implies an upward drift in y if $\delta > 1$. However, our use of δ is intended to highlight that impact must be considered, with the exact choice of δ for a particular token pair being a complex process and beyond the scope of this paper.

- *Retail user*: Any player in a DEX protocol for whom, for an EMP y, there exists some *minimum retail utility* $f_{mcf} > 1$ such that they have positive expected utility to trade for or below $\sqrt{f_{mcf}}\, y$ as a buyer, and at or above $\frac{y}{\sqrt{f_{mcf}}}$ as a seller, respectively.
- *Market Maker* (MM): A player in a DEX protocol with large supplies of all tokens, who has positive expected utility trading with retail users on markets of any width $w > 1$ with reference price equal to the EMP. MMs can observe the EMP.
- *Strict Nash Equilibrium* (SNE) [22]: Consider a set of non-cooperative players P_1 , ..., P_n, with strategies (series' of actions) str_1 , ..., str_n describing the actions which each player takes throughout a particular protocol. These strategies form a strict Nash Equilibrium if any individual player deviation from these strategies strictly reduces that player's utility.

In creating a DEX protocol, a fundamental requirement should be to ensure that there exists an SNE in which retail can trade at the EMP (in expectancy) in exchange for some pre-determined fee, payable to the MMs, which is bounded by the retail utility gain from the swap. In existing AMMs and DEX protocols, this realistic goal remains unachieved, as explained in Section 2. FairTraDEX however, achieves this goal.

Note that MMs differ from liquidity pools in AMMs. The decision logic of AMM liquidity pools is public and deterministic, and any adjustments to liquidity pools must be queued publicly in the mempool, exposing it to EEV attacks. MMs, however, make private trading decisions and communicate them on-chain. One possible action is to add liquidity to an AMM, or in the case of FairTraDEX, add a market to an auction. Following the analysis of [18] and the losses being incurred by liquidity providers in AMMs, players currently providing liquidity in AMMs, although acting honestly, do not fit our rational player model. In FairTraDEX, by ensuring following the protocol forms an SNE, honesty and rationality are equivalent. If players deviate from the protocol in FairTraDEX, this strictly decreases their expected utility, which is further discussed in Appendix D.

3.1 Zero-Knowledge Primitives

This section outlines the *non-interactive zero-knowledge* (NIZK) tools for set membership as used in this paper, such as those stemming from papers like [5, 6, 15, 21]. To participate in FairTraDEX, retail users privately generate two bit strings, the *serial number* S and *randomness* r, with $S,\ r \in \{0,1\}^{\Theta(\kappa)}$. To describe FairTraDEX we define a commitment scheme h, a set-membership proof scheme *SetMembership*, an NIZK proof of knowledge scheme *NIZKPoK* and a NIZK signature of knowledge scheme (*NIZKSoK*). We do not specify which instantiation of these schemes to use, as the exact choice will depend on several factors, such as efficiency, resource limitations and/or the strength of the assumptions used.

- $h(m)$: A deterministic, collision-resistant function taking as input a string $m \in \{0,1\}^*$, and outputting a string $com \in \{0,1\}^{\Theta(\kappa)}$.
- *SetMembership*(com, Com): Compresses a set of commitments Com and generates a membership proof π that com is in Com if $com \in Com$.
- *NIZKPoK*(r, S, Com): For a set of commitments Com, returns a string S and NIZK proof of knowledge that the person running *NIZKPoK*() knows an r producing a proof when running *SetMembership*($h(S||r)$, Com). In FairTraDEX, this revelation identifies to a verifier when a proof has previously been provided for a particular, albeit unknown, commitment as the prover must reproduce S. This is used in FairTraDEX, in conjunction with an escrow, to enforce the correct participation of both retail and MMs.
- *NIZKSoK*(m): Returns a signature of knowledge that the person who chose m can also produce NIZKPoK.

4 WIDTH-SENSITIVE FREQUENT BATCH AUCTIONS

In this section we outline the properties of an idealised variation of an FBA which we define as a *width-sensitive FBA* (WSFBA). WSFBAs maintain the desirable properties of FBAs with respect to optimal strategies for MMs and retail [7], while also adding important protections for retail in a decentralized setting where monopolistic MMs may exist. The important assumption with regard to the guarantees of an FBA is the presence of at least two non-cooperative MMs. In a decentralized setting, this can be seen as insufficient. One of the most desirable properties of FBAs in the presence of 2 non-cooperative MMs is the fact that retail submit market orders. We envisage retail as relatively uninformed players for whom choosing the correct price at which to trade has an implicit cost. Market orders remove this burden, providing an "obvious optimal" for retail as advocated in [26]. To reach a similar equilibrium in the presence of a monopolistic MM, we must amend the basic FBA protocol.

In the presence of a single rational MM, we need to utilize the value gained by retail for exchanging tokens. That is, retail in our protocol observe a positive utility of at least the minimum retail fee f_{mcf} for exchanging tokens. In a WSFBA, this fee is translated to a market width, and input with retail orders as a maximum market width on which retail are willing to trade. This allows us to prove submitting market orders remains a SNE. Conversely in an FBA, if MMs cooperate/are replaced by a monopolistic MM, submitting market orders is a strictly dominated strategy for retail, with retail now required to submit a limit price. WSFBAs avoid this degradation of user experience, and the corresponding reduced probability of execution and quality of liquidity this has on FBAs.

We let D represent the net trade imbalance of retail in a particular instance of a WSFBA in terms of ₿. A positive D indicates a retail buy imbalance (more retail buyers than sellers of the swap with respect to notional), while a negative D indicates a retail sell imbalance. We require a finite bound on the absolute imbalance,

which we denote $Q_{not} < \infty$, for the existence of optimal MM strategies. As in [7], we assume that $|D| \leq Q_{not}$, and in-keeping with the notion of an EMP, D is symmetric around 0 at the EMP. This Q_{not} is used as the lower-bound on the notional of a MM's bid and offer in WSFBA. Importantly, this ensures retail orders submitted to the auction are executed (used in the proof of Theorem 4.3). We now define a WSFBA.

Definition 4.1. A *width-sensitive frequent batch auction* involves MMs submitting markets to the TTP with total notional on the bid and offer of at least Q_{not}. Retail and MMs privately submit limit and market orders to the TTP including a requested maximum width from the tightest MM, above which the order is not executed. Orders are collected until a specified deadline. After this deadline, retail orders with requested width greater than or equal to the tightest MM width, along with a randomly-selected market from the tightest provided markets, are settled at a single clearing price which maximises the total notional traded, and then minimises the net trade imbalance.[1] If there is more supply at the clearing price than demand, sell orders at the highest price at or below the clearing price are pro-rated based on size such that supply equals demand at the clearing price. Similarly, if there is more demand than supply at the clearing price, buy orders at the lowest price at or above the clearing price are pro-rated based on size such that demand equals supply at the clearing price. Any limit buy orders below/sell orders above the clearing price are not executed.

The key differences between a conventional FBA and a WSFBA are the specifications of widths by retail, the minimum MM notional requirement on the bid and offer, and the requirement for the clearing price to minimize the imbalance over all prices which maximize the notional traded. Minimizing imbalance is a small optimisation which produces a reasonable and precise clearing price when MMs do not show width 1 markets as in an FBA. An on-chain protocol for verifying a given clearing price satisfies these properties is detailed in [19, 20]. The other amendments are intended to protect retail against monopolistic MMs, and are discussed in the proceeding section.

4.1 Properties of Width-Sensitive Frequent Batch Auctions

In Theorem 4.3, we identify an SNE for WSFBAs, and show that it is equivalent to the SNE of an FBA. The case of a single monopolistic MM is more complex than the case of multiple non-cooperative MMs. First, we observe that an MM in a WSFBA always shows a market with reference price equal to the EMP. Proofs for the proceeding lemmas and theorems are given in the extended version of the paper [19].

Lemma 4.2. For an MM in a WSFBA between A_{tkn} and B_{tkn} with EMP equal to $y_{A\to B} = \frac{y_B}{y_A}$ and a retail order of notional $X\beta > 0$, she strictly maximizes her expected utility by showing a market with reference price $y_{ref} = y_{A\to B}$ for any fixed width $w \geq 1$.

This result is independent of the choice of width and market-impact coefficient. However, it assumes that the MM trades with retail on either the bid or the offer. With respect to a WSFBA without notional restrictions and a monopolistic MM, if retail submit market orders, there are fringe cases (large imbalances) which incentivize MMs to show markets far from the EMP. Removing these restrictions from a WSFBA makes for interesting future work.

Recall retail have a strictly positive utility to exchange tokens described by the minimum retail fee f_{mcf}, which is equivalent to being strongly incentivized to trade on a market with reference price y_{ref} and width $w \leq f_{mcf}$. With this in mind, we can now apply the main result of [7] to a WSFBA.

Theorem 4.3. For a WSFBA, the strict Nash equilibria strategies given the number of non-cooperative MMs submitting markets being N are:

- $N = 1$: Retail submit market orders of requested width f_{mcf} and the MM shows a market of width at most f_{mcf} with reference price equal to the EMP.
- $N \geq 2$: Retail submit market orders of requested width greater than 1 and MMs show a market of width 1 with reference price equal to the EMP.

Theorem 4.3 identifies that retail always submit market orders, and in settings where it is unclear whether there is a single monopolistic MM, or many non-cooperative MMs, it can be seen that retail always submit market orders with requested width f_{mcf}.

5 FairTraDEX

In Section 4 we constructed a WSFBA using a TTP to enforce correct player balances, order sizes, revelation of orders, correct calculation of the clearing price and the settlement of orders. In a decentralized setting, such a TTP does not exist. However, we do have access to censorship-resistant public bulletin boards in the form of blockchain-protocols. If we are able to bound the delay of updates being added to such a bulletin board (transactions are eventually confirmed on the blockchain), we can replicate the key functionalities of the TTP needed to implement a WSFBA in a decentralized setting. In this section we construct the FairTraDEX protocol as a sequence of blockchain-interpretable algorithms. We then provide a series of results regarding the incentive compatibility of these algorithms with the goal of proving FairTraDEX instantiates a WSFBA, and that following the protocol is an SNE.

5.1 System Model

(1) All players $P_1, ..., P_n$ are members of a blockchain-based distributed ledger, and a corresponding PKI.
(2) The ledger is represented by a linear blockchain with its state progressing by having new blocks sequentially appended. For simplicity, we assume instant finality of blocks meaning that such an appended (valid) block cannot be replaced at any later point in time.
(3) A transaction submitted by a player for addition to the blockchain while observing blockchain height H, is included (and thus finalised) in a block of height at most $H + T$, for some known $T > 0$, given that the transaction remains valid for sufficiently many intermediate ledger states.
(4) The public NIZK parameters are set-up in a trusted manner.

A discussion on this model is included in Appendix B.

[1] As Q_{not} is greater than the absolute retail order imbalance, the clearing price must lie between the MM bid and offer

5.2 FairTraDEX Algorithms

Each player P_i owns (has exclusive access to) a set of token balances bal_i which are stored as a global variable. For a token *tkn*, $bal_i(tkn)$ is the amount of token *tkn* that P_i owns. Keeping the notation from Section 3.1, outputs included in round brackets () are known only to the player running the algorithm, with all other outputs posted to the public bulletin board, updating existing variables/balances where appropriate. Algorithm outputs are not signed, so players observing the output of an algorithm instance can only infer information about the player running the algorithm from public outputs and any corresponding global variable updates.

We now outline FairTraDEX as a set of algorithms: Setup(), Register(), CommitRetail(), CommitMM(), RevealRetail(), RevealMM() and Resolution(). A FairTraDEX instance is initialized by running Setup(), and proceeds indefinitely in rounds of three distinct, consecutive phases: *Commit*, *Reveal* and *Resolution*, each of length T blocks (see Section 5.1). For readability, we provide here the intuition to the algorithms of FairTraDEX, with implementations and detailed descriptions provided in [19, 20].

Players in the blockchain protocol can enter FairTraDEX as retail by running an instance of Register(), which for a given retail user deposits an escrow $escrow_{retail}$, and generates private information $(S,\ r \in \{0,1\}^{\Theta(\kappa)})$ which is used in CommitRetail() to prove that the retail user indeed deposited an escrow, without revealing which deposit.

In the Commit phase, all players can run any number of CommitRetail() and/or CommitMM() instances. CommitRetail() generates a retail order, commits to that order publicly and proves in ZK that the player deposited an escrow. If such a proof cannot be generated, or a proof has already been generated for the same S, no order can be committed. A correctly run CommitMM() instance generates a market for a prospective MM, commits to that market publicly and deposits an escrow $escrow_{MM}$.

In the Reveal phase, players can run any number of RevealRetail() and/or RevealMM() instances. RevealRetail() publishes an order generated through CommitRetail(), returning the escrow corresponding to the CommitRetail() instance, and as such the Register() instance, to the retail user. RevealMM() publishes a market corresponding to a CommitMM() instance, and returns the corresponding escrow. Both Reveal phase algorithms assert that the retail user and MM have sufficient token balances to submit their order and market respectively.

In the Resolution phase, any number of Resolution() instances can be run. The first correct Resolution() instance selects the tightest market from the set of revealed markets, *revealedMkts*, for inclusion in order settlement, and any tie-breaks settled using $h(revealedMkts)$, as a random seed. The clearing price which maximizes notional traded, and then minimizes the notional imbalance of the remaining market and orders is computed. A precise algorithm for the on-chain verification of the clearing price is provided in the implementation of FairTraDEX [20], and described in [19]. Orders and markets are then settled based on this clearing price. Finally, the arrays tracking active commitments, orders and markets (*retailCommits*, *MMCommits*, *revealedOrders*, and *revealedMkts* respectively) are cleared, so unsuccessfully revealed commitments during this round cannot be used to run RevealRetail() or RevealMM() in future rounds. This effectively destroys the deposited escrows of such commitments.

5.3 Properties of FairTraDEX

Towards proving FairTraDEX forms an SNE for rational retail and MMs, we provide a series of Lemmas that we use to prove the main result of the section, Theorem 5.1. Due to space restrictions, we only provide an intuition for these Lemmas. We first prove that some player in the blockchain protocol runs a Resolution() instance every round. Then, we prove a series of Lemmas demonstrating that given a rational retail user (resp. MM) runs an instance of Register() (resp. CommitMM()), that same player correctly runs CommitRetail() and RevealRetail() (resp. RevealMM()) in the proceeding phases. Finally, we show that it is indeed an SNE for a Retail (resp. MM) to run Register() (resp. CommitMM()).

With these results in hand, we have that rational retail and rational MMs correctly execute all algorithms as outlined by FairTraDEX. We now show that with at least n_ψ Register() calls, the optimal strategy for retail is to submit market orders, while the optimal strategy for a MM with EMP $y_{A\to B}$ is to show a market *bid* @ *offer* with $bid \approx y_{A\to B} \approx offer$ in the case where there are at least 2 non-cooperative MMs, and of width $w \leq f_{mcf}$ otherwise.

Theorem 5.1. Consider an instance of FairTraDEX between A_{tkn} and B_{tkn} with EMP $y_{A\to B}$ and at least n_ψ previously called instances of Register(). For N non-cooperative MMs, the following strategies form strict Nash equilibria:

- $N = 1$: Retail run Register(), followed by CommitRetail() producing market orders of width f_{mcf}. The MM runs CommitMM() producing a market of width at most f_{mcf} with reference price equal to $y_{A\to B}$ in size Q_{not}. Retail and MMs then run RevealRetail() and RevealMM() respectively.
- $N \geq 2$: Retail run Register(), followed by CommitRetail() producing market orders of width greater than 1. MMs run CommitMM() producing markets of width 1 with reference price equal to $y_{A\to B}$ in size of at least Q_{not}. Retail and MMs then run RevealRetail() and RevealMM() respectively.

Although providing width-1 markets may seem prohibitive for MMs, the unique guarantees of FairTraDEX ensure that no players external to the protocol can extract value from players within the protocol. As player value is being retained within the FairTraDEX protocol, fees can be introduced to compensate MMs. Given the potential value retention of FairTraDEX (see Section 6, Table 2), these fees can be substantial while still ensuring FairTraDEX provides retail with best-in-class liquidity.

6 COST-BENEFIT ANALYSIS OF FairTraDEX

The aim of this section is to demonstrate the contributory significance of FairTraDEX vs. current state-of-the-art protocols as introduced in Section 2. Our results are based on the Solidity implementation of the protocol provided in [20]. In Table 1 we include an overview of the gas costs for running FairTraDEX compared to the previous blockchain-based attempts to implement batch auctions of [9, 13], with numbers taken from the respective papers. These are the fixed costs for including and executing the transactions on

	FairTraDEX [2]	Uniswap	[9]	[13]
Register	112,800	-	87,000	-
Commit Retail	344,500	-	52,000	276,150
Commit MM (per order)	24,300	-	52,000	276,150
Reveal (per order)	172,000	190,000	171,000	48,750
Settle (per order) [3]	45,500	-	122,500	54,000
Total Retail	674,800	190,000	432,500	378,900
Total MM	266,100	-	397,500	649,050
Total Retail (USDC)	7.27	2.05	4.66	4.08
Total MM (USDC)	2.87	-	4.09	7

Table 1: Comparison of gas costs in batch-auction implementations. [2] Costs provided for FairTraDEX are amortised over 128 retail orders and 8 markets. [3] We add an estimated cost for token transfer from smart contract to player of 40,000 to the figures provided in [13] to standardise the costs therein with those of FairTraDEX and [9].

a blockchain. It can be seen that FairTraDEX has a slightly greater upfront gas cost for retail, but a lesser cost for MMs. This is directly related to the added costs of correctly using ZK tools to hide retail identity and order information until all orders have been committed. Compared to the variable costs of revealing this information, we see these costs as acceptable.

To demonstrate the benefits of FairTraDEX, Table 2 compares specific swaps that allow for EEV attacks in existing state-of-the-art protocols. We perform our analysis on ETH/USDC swaps, as this is the highest volume pool on Uniswap, which at time of writing had pool sizes of 120k ETH and 185M USDC, an indicative EMP of 1 ETH equal to 1,540 USDC [27]. Furthermore, we use a gas cost of 7 gwei [11]. Consider 3 buy ETH orders of 10k, 500k and 10M USDC from 2 different players who are known to need to trade at any price. P_1 has large quantities of both ETH and USDC, and buys or sells ETH pseudo-randomly, while P_2 only owns USDC/only buys ETH. We take the estimated impact for each order to be 0, 0.15% and 1% respectively, numbers taken from the Uniswap V3 API [27] (these are more realistic impacts than those implied by the constant product impact [1] of 0, 0.54% and 11.1% respectively). Although this is a simplification of order impact, true impact is likely some multiple/factor of this impact. Protocol fees incentivizing MMs to provide liquidity are omitted as they are not considered in the provided academic protocols. After gas costs, this fee should be approximately equal for all protocols (the Uniswap fee for this pool is 0.3%).

When P_1 submits an order in FairTraDEX or [13], no information is gained about the direction of the trade. However, in [9], direction is revealed. As such, any blockchain participant can front run that impact on all other markets, and thus the EMP for any MM responding to the order will be the impacted EMP. When P_2 submits an order in either of [9, 13] the direction is known, and the EMP is impacted in the same way as for P_1. Crucially, this impact takes place before any player interacts with P_2, giving P_2 a worse price. Using estimated price impacts of 0, 0.15% and 1%, Table 2 demonstrates the costs of executing these swaps, excluding transaction fees, in these protocols, and Uniswap. For Uniswap, we must also add the recommended slippage, an additional 0.5% of the order size, as it is always in a block producers interest to give Uniswap players worst execution. It can be seen that these costs become

	FairTraDEX	Uniswap	[9]	[13]
P_1-10,000	7	52	5	4
P_2-10,000	7	52	5	4
P_1-500,000	7	3002	755	4
P_2-500,000	7	3002	755	754
P_1-10,000,000	7	150,002	100,005	4
P_2-10,000,000	7	150,002	100,005	100,004

Table 2: Comparison of execution costs in USDC of batch-auction implementations, including the transaction fees of Table 1.

increasingly more significant as order size increases, dominating the differences in gas costs of Table 1.

Although Table 2 can be seen as simplifying how orders are handled, it demonstrates two crucial motivators for our work. Firstly, any information revealed about retail before a trade is agreed can, is and will continue to be used against retail. Furthermore, this cost is not necessarily paid to the MM. As orders are committed in public, any blockchain participant can use the committed information to front run the impact on the EMP before the MM or retail have an opportunity to trade, extracting money from the DEX protocol. Secondly, as the effects of these value-extraction techniques increase super-linearly in order-size, a protocol with the value-extraction guarantees of FairTraDEX is needed to allow typically large retail users to utilise the benefits of DEXs, and blockchain protocols in as a whole, at a fixed cost, as demonstrated in Table 1, without incurring the prohibitive execution costs of previous solutions, as demonstrated in Table 2.

7 CONCLUSION

We provide FairTraDEX, a blockchain-based DEX protocol based on WSFBAs for which we formally prove that the strategies of rational participants have strict Nash equilibria in which all trades occur at the external market price plus or minus bounded upfront costs (specified market widths) which approach 0 in the presence of non-cooperative MMs. This is an attractive alternative to existing mainstream protocols such as AMMs where rational players effectively and systematically prevent such an equilibrium from happening, all at the expense of retail users and MMs. Compared to previous blockchain-based attempts to implement EEV-proof DEXs, FairTraDEX is the first to practically allow for indistinguishable retail order submissions by decoupling order submission from token deposit and order revelation. The FairTraDEX benefits formalized in Section 5.3, and demonstrated in Section 6 provide important improvements on previous protocols regarding EEV protection, setting a new standard for EEV protection in DEXs.

As stated in the comparisons of Section 6, protocol fees are omitted for all protocols. Given the total retention of value within the FairTraDEX protocol (no extractable value), fees in line with the utility gained by retail for exchanging their tokens can be charged to incentivize the long-term participation of MMs in FairTraDEX. These fees should reflect the need to incentivize MMs while retaining the unique retail-side benefit of trading at the external market price in expectation, which is proven to occur in FairTraDEX. An analysis of these fees makes for interesting future work.

REFERENCES

[1] Guillermo Angeris, Hsien-Tang Kao, Rei Chiang, Charlie Noyes, and Tarun Chitra. An analysis of uniswap markets. In Cryptoeconomic Systems Journal, 2019.

[2] Anon. https://anonymous.4open.science/r/FairTraDEX-4D29/README.md, 2022.

[3] Avi Asayag, Gad Cohen, Ido Grayevsky, Maya Leshkowitz, Ori Rottenstreich, Ronen Tamari, and David Yakira. Helix: A fair blockchain consensus protocol resistant to ordering manipulation. IEEE Transactions on Network and Service Management, 18(2):1584–1597, 2021.

[4] Carsten Baum, Bernardo David, and Tore Frederiksen. P2DEX: Privacy-preserving decentralized cryptocurrency exchange. In Kazue Sako and Nils Ole Tippenhauer, editors, Applied Cryptography and Network Security, pages 163–194. Springer International Publishing, 2021.

[5] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In 2014 IEEE Symposium on Security and Privacy, pages 459–474, New York, NY, USA, 2014. IEEE Computer Society.

[6] Daniel Benarroch, Matteo Campanelli, Dario Fiore, Kobi Gurkan, and Dimitris vKolonelos. Zero-knowledge proofs for set membership: Efficient, succinct, modular. In Nikita Borisov and Claudia Diaz, editors, Financial Cryptography and Data Security, pages 393–414, Berlin, Heidelberg, 2021. Springer Berlin Heidelberg.

[7] Eric Budish, Peter Cramton, and John Shim. The High-Frequency Trading Arms Race: Frequent Batch Auctions as a Market Design Response *. The Quarterly Journal of Economics, 130(4):1547–1621, 07 2015.

[8] Jing Chen and Silvio Micali. Algorand: A secure and efficient distributed ledger. Theoretical Computer Science, 777:155–183, 2019.

[9] Theodoros Constantinides and John Cartlidge. Block auction: A general blockchain protocol for privacy-preserving and verifiable periodic double auctions. In 2021 IEEE International Conference on Blockchain (Blockchain), pages 513–520, United States, 2021. IEEE Computer Society.

[10] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges. https://arxiv.org/abs/1904.05234, 2019. Retrieved: 19/01/2022.

[11] Etherscan. https://etherscan.io/gastracker. Retrieved: 25/07/2022.

[12] Flashbots. https://explore.flashbots.net. Retrieved: 25/07/2022.

[13] Hisham S. Galal and Amr M. Youssef. Publicly verifiable and secrecy preserving periodic auctions. In Matthew Bernhard, Andrea Bracciali, Lewis Gudgeon, Thomas Haines, Ariah Klages-Mundt, Shin'ichiro Matsuo, Daniel Perez, Massimiliano Sala, and Sam Werner, editors, Financial Cryptography and Data Security. FC 2021 International Workshops, pages 348–363, Berlin, Heidelberg, 2021. Springer Berlin Heidelberg.

[14] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In Advances in Cryptology - EUROCRYPT 2015, pages 281–310, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[15] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, Advances in Cryptology – EUROCRYPT 2016, pages 305–326, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[16] Aljosha Judmayer, Nicholas Stifter, Philipp Schindler, and Edgar Weippl. Estimating (miner) extractable value is hard, let's go shopping!, 2021. https://ia.cr/2021/1231.

[17] Mahimna Kelkar, Fan Zhang, Steven Goldfeder, and Ari Juels. Order-fairness for byzantine consensus. In Daniele Micciancio and Thomas Ristenpart, editors, Advances in Cryptology - CRYPTO 2020, pages 451–480, Cham, 2020. Springer International Publishing.

[18] Stefan Loesch, Nate Hindman, Mark B Richardson, and Nicholas Welch. Impermanent loss in uniswap v3, 2021.

[19] Conor McMenamin, Vanesa Daza, Matthias Fitzi, and Padraic O'Donoghue. Fairtradex: A decentralised exchange preventing value extraction. arXiv preprint arXiv:2202.06384, 2022.

[20] Conor McMenamin and Padraic O'Donoghue. https://github.com/MEVProof/Contracts, 2022.

[21] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In 2013 IEEE Symposium on Security and Privacy, pages 397–411, United States, 2013. IEEE Computer Society.

[22] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. Algorithmic Game Theory. Cambridge University Press, Cambridge, 2007.

[23] Perpetual Powers of Tau. https://zkproof.org/2021/06/30/setup-ceremonies/. Retrieved: 02/08/2022.

[24] Rafael Pass, Lior Seeman, and abhi shelat. Analysis of the blockchain protocol in asynchronous networks. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, Advances in Cryptology – EUROCRYPT 2017, pages 643–673, Cham, 2017. Springer International Publishing.

[25] Kaihua Qin, Liyi Zhou, and Arthur Gervais. Quantifying blockchain extractable value: How dark is the forest?, 2021.

[26] Tim Roughgarden. Transaction fee mechanism design for the ethereum blockchain: An economic analysis of eip-1559. https://arxiv.org/pdf/2012.00854, 2020. Retrieved: 18/05/2021.

[27] Uniswap. https://app.uniswap.org/. Retrieved: 25/07/2022.

A TERMINOLOGY AND USEFUL DEFINITIONS

This section contains additional financial and game-theoretical terms used in this paper. Although not mandatory for all readers, this section serves as a useful reference point towards understanding the results and discussions that follow.

- *Decentralized Exchange* (DEX): A distributed marketplace which allows players to swap one token for another.
- *Limit Order*: Specifies an amount of tokens to be bought (sold), and a maximum (minimum) price at which to buy (sell) these tokens. This price is known as the *limit price*.
- *Market Order*: Specifies an amount of tokens to be sold, but no limit price. Market orders are to be executed immediately at the best available price based on the liquidity of buy orders.
- *Direction*: With respect to an order on a market quoted from token A_{tkn} to B_{tkn}, if the order is trying to buy token B_{tkn}, the direction is *buying*, while if the order is trying to sell token B_{tkn}, the direction is *selling*.
- *Forward Price*: This is the price at which a seller delivers a token to the buyer at some predetermined date. In any exchange protocol without instantaneous delivery, the forward price at expected delivery time is the price at which trades should happen. The difference between current (spot) price and forward price is known as *carry*, and can be due to storage/opportunity costs, interest rates, etc. In this paper, we set carry to 0 for complexity and ease-of-notation purposes.

The following definition of expected extractable value is translated from [16] using the terminology of this paper.

Definition A.1. The expected extractable value EEV_i, describes the total value in value units, which is transferred to player P_i in expectation using a certain strategy which produces a transaction, sequence of transactions, or blocks that later become part of the main chain with some probability.

A.1 Relayers

A typical requirement for transaction submission in blockchains is the payment of some transaction fee to simultaneously incentivize block producers to include the transaction, and to prevent denial-of-service/spamming attacks. However, in both the UTXO- and account-based models, this allows for the linking of player transactions, balances, and their associated transaction patterns. With respect to DEX protocols, if clients are required to deposit money into a UTXO/account before initiating a trade, any other player in the system can infer who the client is, what balances the client owns, what transactions the client usually performs, etc., and use this information to give the client a worse price.

To counteract this, we utilise the concept of *transaction relayers*[4]. In the smart-contract encoding of FairTraDEX [20], clients must publicly register to a smart contract, and in doing so, deposit some escrow. In addition to this escrow, we also require the clients to

[4]Ox https://0x.org/docs/guides/v3-specification, Open Gas Station Network https://docs.opengsn.org/, Rockside https://rockside.io/, Biconomy https://www.biconomy.io/

deposit a relayer fee. When the client wishes to submit a transaction anonymously to the blockchain, the client publishes a proof of membership in the set of registered clients to the relayer mempool, as well as the desired transaction and a signature of knowledge cryptographically binding the membership proof to the transaction, preventing tampering. As the relayer can verify the proof of membership, the relayer can also be sure that if the transaction is sent to the FairTraDEX contract, the relayer will receive the corresponding fee. With this in mind, a relayer observing the client transaction includes it in a normal blockchain transaction, with the first relayer to include the transaction receiving the fee. As such, relayers are a straightforward extension of the standard transaction-submission model. Furthermore, if the proof of membership is NIZK and the message is broadcast anonymously (using the onion routing (Tor) protocol[5] for example), the relayer can only infer that the player sending the transaction is a member of the set of clients.

B MODEL DISCUSSION

We do not make any assumptions regarding transaction ordering in blocks. Specifically, the order in which transactions are executed is at the discretion of the block proposer.

If block producers are participating as MMs/retail, we need to adjust T. Let $0 < \alpha < 1$ bound the fraction of blocks produced over chains of length greater than T by a MM responding to/the set of retail users requesting trades in a particular instance of a FBA (we need to consider all retail in a request phase, as they may all have the same direction, and as such, some positive expectancy to preventing a MM revelation). We need to increase T by a factor of $\frac{1}{1-\alpha}$ (similar to the methodology behind the Chain Quality property in [14, 24]). Moreover, our property can be seen as a 'block-based' variant of the time-based *liveness* property defined in [14, 24]. An example for instant finality is Algorand [8] which stands in contrast to, e.g., Bitcoin which only guarantees eventual finality, while example of a public NIZK parameter setup is a Perpetual Powers of Tau ceremony, as used in Zcash [23].

C FairTraDEX VS. WSFBA

The main differences between FairTraDEX and a WSFBA are as follows:

- Escrows are used to enforce the correct revelation of players who commit to orders or markets. Escrows are only returned to players if orders are revealed and correspond to a valid commit. Furthermore, escrows are chosen large enough to ensure the reclamation of escrows has strictly higher utility than not, ensuring rational players follow the protocol.
- FairTraDEX requires an algorithm involving deposits and/or withdrawals updates the set of balances for all players, identifying the player calling the algorithm.
- FairTraDEX separates the depositing of retail escrow and retail order commitments. This is a key functionality necessary to preserve retail anonymity and the guarantees of a WSFBA. If retail deposits an escrow in the same instance as committing to an order, that information can be used to identify the player, and imply information about the player's order. By separating the two, commitment does not require the update of global variables that can be used to identify the retail user.
- Set-membership proofs in ZK in the CommitRetail() algorithm are used to prove that a player committing to a retail order has deposited a retail escrow. As FairTraDEX separates the deposit and commitment steps, these proofs allow retail that deposited an escrow to generate one (and only one, as ZK proofs reveal S) order per escrow, while only revealing that the order corresponds to a deposited escrow. As the number of deposited escrows increases, the probability that an order commitment matches any particular escrow approaches 0. This replicates the anonymous order submission of a WSFBA.
- Tokenized incentives are used to ensure some player in the blockchain calculates the clearing price, and settles orders correctly.

[5] https://www.torproject.org/

D EXISTENCE OF IRRATIONAL PLAYERS AND COALITIONS

When analysing the optimal strategies of players in WSFBAs, our results are based on all players being rational and that n_ψ instances of Register() are called. If we consider the presence of irrational players in the system, we can apply the following adjustments:

- **Irrational MM**: In Lemma 4.2, it is shown that the optimal strategy for a MM is to show markets centred around the EMP. Any other (irrational) strategy must therefore result in reduced expectancy for the MM, and higher expectancy for retail. Therefore, given the presence of irrational MMs, submitting market orders maximises retail expectancy (with greater expected utility than in the presence of rational MMs, although with increased variance).
- **Irrational retail**: Given the optimal strategy for rational retail is to submit market orders, irrational retail may then submit limit orders. This merely reduces irrational retail's chance of trading vs. other retail. This would not change the strategy of non-colluding rational MMs, but may have some affect on a monopolistic MM's interpretation of f_{mcf}.

Furthermore, if less than n_ψ instances of Register() are called, retail resort to submitting limit orders. This can be seen by examining the proof of Theorem 4.3, as contained in [19]. In the proof, if retail are not sure that a MM will show a market with reference price equal to the EMP, the case when less than n_ψ instances of Register() are called, the optimal strategy for retail is to submit limit orders, which only stands to reduce retail's probability of trading. As the number of non-cooperative players in FairTraDEX decreases towards two, the guarantees of FairTraDEX approach those of an AMM. However, as retail price and order size remain hidden until the counterparty chooses her strategy, and before the clearing price is fixed (end of the Commit phase), FairTraDEX maintains advantages over AMMs against retail-based EEV attacks, such as price/order-size specific front-running and selective participation.