A Survey of DEX Security

Jack McPherson*, Dong Bao†
School of Information Technology and Electrical Engineering
University of Queensland
*[REDACTED], †[REDACTED]

***Abstract*—The proliferation of Decentralised Finance (DeFi) has started with the development of entirely Decentralised Exchanges (DEXes): peer-to-peer cryptocurrency protocols built atop Ethereum that facilitate the trade of both Ether and ERC20 tokens. Despite their meteoric success in attracting both capital and user attention, there are substantial lackings in the understanding of the security implications these DEXes present. We adapt an existing security model applied to traditional financial exchanges and extend it to these DEX platforms. We assess three major DEXes of academic interest and use them as proxies to compare their respective design approaches.**

CONTENTS

I. INTRODUCTION

With the rapid development of various blockchain platforms in the last five years, cryptocurrencies

have enjoyed immense popularity amongst investors globally. The demand for cryptocurrency exchanges continues to rise, allowing users to trade one cryptocurrency for another (e.g., trading Bitcoin for Ether). According to Coinmarketcap[1], cryptocurrency exchanges have a total market capitalisation of more than 250 billion US Dollars, with centralised exchanges occupying the vast majority of daily trading volume. When users choose to exchange their digital assets on a centralised exchange platform, they need to trust the company or the platform that provides the exchange service. The drawbacks of centralised exchanges are well discussed and remain a disputed topic amongst blockchain communities.

Due to the loose regulatory environment, instances of malpractice often occur in centralised exchanges, damaging the interests of their users. The most well-known of such instances are usually thefts of users' funds and market manipulation. Additionally, despite the fact that cryptocurrencies are built on decentralised systems, exchanging these cryptocurrencies mainly relies on centralised exchange systems. This has raised concerns amongst both investors and grass-roots users as this clearly harms the decentralization of the underlying blockchain protocols. Consequently, decentralised exchanges (DEXes) have arisen and are a relatively recent development.

DEXes are based on smart contract technology, enabling users to trade directly from their existing wallets. DEXes have a variety of core functionality, often including capital deposit, maintenance of order books, order matching, and of course asset exchange. Asset exchange is decentralised by default, while the other three functions are sometimes centralised. Currently, most DEXes only support the exchange of ERC-20 tokens on Ethereum; however, in the future, DEXes will surely be extended and able to support the exchange of cryptocurrencies from different blockchain systems[2][3]. DEXes seem to be well-designed, contributing to the fair and transparent trade of tokens. Trades are automatically executed by a smart contract - thus there is no third party to control the transaction. Note that different DEXes have different designs for their smart contact implementations.

Despite the *a priori* soundness of DEX designs, vulnerabilities in DEXes do exist. Adversaries can perform various attacks against DEXes - both technical and financial in nature. In this survey, we look into three popular DEXes of various design types: EtherDelta, IDex, and Uniswap - all built upon the Ethereum blockchain. We take advantage of existing work[4] to describe the nature of several recently discovered attacks against them and other members of their design classes. These attacks include frontrunning, time-bandit attacks, and fee-based forking. Moreover, we assess each chosen DEX platform against our adapted list of security properties derived from the literature[5] and provide potential mitigations for each DEX in light of our assessment. We believe our work is the first to leverage security properties which are used for evaluating traditional finance markets to assess DEXes.

II. Background

A. Smart Contracts

Smart contracts are computer programs that are stored on a blockchain and executed by a distributed virtual machine based on the same blockchain. Smart contracts are immutable, which means that once a smart contract is created, it can never be changed again. The execution and subsequent output of smart contracts is verified by nodes on the distributed overlay network backing the host blockchain's consensus protocol.

Ethereum by far is the largest and most prevalent blockchain supporting smart contracts to date[6] and was proposed by Vitalik Buterin in late 2013[7]. Additionally, Solidity is a Turing-complete programming language[8] that is specifically designed for Ethereum, supporting the arbitrary design and implementation of smart contracts. Furthermore, Ethereum is a distributed computing platform upon which many decentralized applications (DApps) are built. There are many kinds of DApp on Ethereum, (e.g. games, gambling, exchanges, etc.). A DEX is one such type of DApp. In addition, some DApps implement their own ERC-20 token which serves the purpose of powering the ecosystem and meeting the application-level needs of the DApp. Depending on the purpose of DApp, such tokens can usually represent any fungible item. For example, customers may need a ticket to watch a movie in a cinema, a ticket here is the token used in a hypothetical cinema DApp. In some DApps, such tokens can even represent a share in a company. A DEX is a type of DApp that is designed to enable users to trade such tokens in a decentralised fashion.

B. Limit Order Books

A limit order book is an electronic list of bid and ask offers from buyers and sellers, respectively. Usually, there are multiple types of orders accepted by the book. The most common are market orders and limit orders. When a user places a market order (either to buy or sell), the order will be executed immediately by applying the current market price available to the user. A limit order, on the other hand, specifies a maximum or minimum price that a user wants to buy or sell at, respectively. The associated matching engine will then add the limit order to the order book data structure and execute it when the market price satisfies the order conditions.

C. Decentralised Finance (DeFi)

Decentralised Finance, also known as DeFi, is an emerging movement implementing decentralised, cryptocurrency-backed counterparts to existing traditional financial services - all built atop public, permissionless blockchain networks. Different DeFi platforms provide different services, such as lending and borrowing of cryptoassets, insurance, and cryptocurrency exchange.

DeFi has many advantages over both centralised cryptocurrency exchanges and traditional financial services. In both cases, this inherent single point of failure is a serious problem for users. In DeFi, there is no such problem. In DeFi, users have no need to trust a centralised entity and transaction fees are usually lower.

D. Decentralised Exchanges (DEXes)

The development of DEXes is extremely rapid by its very nature, as is the growth of its user base. Compared with centralised exchanges, DEXes are entirely trustless. Users are able to trade their assets directly from their wallets with a smart contract. This provides various advantages to an end user.

Firstly, much lower counterparty risk, as smart contracts play the traditional role of an exchange operator. Secondly, it is possible for users to pay lower transaction fees. Thirdly, users have control of their assets. However, DEXes also have some vulnerabilities; in this work we will apply a security model to assess different DEXes.

Many DEXes manage an on-chain or off-chain order book. On-chain order books are on the distributed system - all the nodes on the network will receive the orders and verify them. However, order books are usually managed off-chain which means a third party will hold the order book. Furthermore, some DEXes with special designs do not have order books. Order matching is an important component of any DEX, as it can automatically match the buy orders with the corresponding sell orders, and vice versa. The usage of different kinds of order matching mechanisms depends on the design of DEX. In this work, we will perform an assessment of three DEXes: EtherDelta, IDex, and Uniswap. We choose these DEXes as they have different order matching mechanisms.

III. Security Model

In order to conduct an analysis of our selected DEXes, we outline a modified set of security properties adapted from[5].

Trading Integrity All orders in a market express the true intent of their owners and are not used to manipulate market microstructure.
There exist a variety of well-documented techniques which violate market microstructure - in both traditional financial exchanges, centralised cryptocurrency exchanges, and DEXes[4]. Perhaps the most well-known is that of quote stuffing.

Operational Transparency All rules governing market operation are followed and known to all market participants.
For applicable types of exchanges (either traditional or cryptocurrency), matching rules and how orders are to be prioritised are the two main categories of market rules of interest to participants. Obviously, in the traditional setting, this requires both trusting the central exchange authority and the existence of external enforcement (i.e., regulatory bodies and courts of law). In the case of DEXes, this property is actually much more likely to hold, given the nature of Ethereum smart contracts.

Fair Market Access All market participants have equal access to the market.
This property often fails to hold for many traditional financial markets[5], primarily due to latency requirements, regulatory constraints, and simple malpractice. In the case of centralised cryptocurrency exchanges, these reasons often hold also. In the case of DEXes, this property

is easier to reason about; however, in practice, latency is often still an issue[4].

Symmetric Information Access All market participants have access to the same information pertaining to the current state of the market and this information is accurate.

Both traditional financial exchanges[5] and centralised cryptocurrency exchanges[9] have historically failed to consistently hold this property. For DEXes, maintenance of this property is largely influenced by the matching scheme the DEX uses - any off-chain logic is likely to hinder this property.

Order Queue Integrity All orders are always prioritised as per the public market rules.

Even if an exchange is compliant with the **Operational Transparency** property, it can fail to hold this property if it simply does not follow the public market rules. Such malicious execution is much harder to achieve in the DEX setting - once again due to the nature of Ethereum smart contracts - but there exist consensus-level threats to **Order Queue Integrity** even in the decentralised case[4][9][10].

Participant Anonymity All market participants are anonymous, both to each other and to the exchange.

This property is much more elusive compared to the previous ones, as no exchanges across our three categories truly guarantee strict participant anonymity. Traditional exchanges may satisfy the weak case of this property (see dark pools[11]), but largely fail to satisfy the strong case due to regulatory requirements. Cryptocurrency exchanges necessarily suffer from attacks against the anonymity of the underlying blockchain technology.

IV. Assessment

A. EtherDelta

1) Description: EtherDelta is a popular decentralized exchange that enables users to trade both Ether and ERC20 tokens with other users, without trusting a third party. It was launched in July 2016 by Zachary Coburn[12].

EtherDelta maintains an off-chain order book and lacks automatic order matching. Users must choose an order from the order book to execute themselves. Unlike IDex, having an arbiter component to maintain a queue of pending transaction, EtherDelta does not support the submission of multiple orders. Users must wait for their trade to be mined and only then can additional orders be placed. Furthermore, this lack of automatic order matching means that users can only match against an order already present in the off-chain order book. Moreover, EtherDelta only supports limit orders - market orders are not supported. For order cancellation, users are required to submit a cancellation transaction to the Ethereum mainnet for mining - which leads to poor latency.

In order to trade on EtherDelta, an initial deposit of capital is required (either in the form of Ether or ERC20 tokens). A market maker creates a buy/sell order - which includes parameters such as price, amount, and expired block number. The market maker then signs the order and submits it to the order book. Note that the expired block number determines how long the submitted order will stay in the order book. When a taker sees the maker's order, the taker can decide to trade with the maker by sending the order, together with taker's signature, to the smart contract. Once the smart contract receives the order, it will first check if the signature matches the signature of the order, that the counterparties have sufficient available funds, and then check if the order is expired. After finishing this verification, the smart contract will process the trade. Finally, the transaction being mined marks the end of the trade.

2) Trading Integrity: Most DEXes, due to their decentralised, pseudo-anonymous nature, cannot realistically enforce **Trading Integrity**. Since the order book is fully controlled by users, it is the users that submit the buy/sell orders to the order book. The system cannot prevent the traders from submitting orders that do not represent honest trading intent. EtherDelta currently supports more than one hundred ERC20 tokens. Most of the tokens have a very small market volume, which means that a predatory trader can easily manipulate the market. However, even though the market volumes of some popular tokens are relatively large, users still have some capacity to manipulate the market. As users cannot submit multiple orders on EtherDelta, this can mitigate some price manipulation techniques, to a certain degree.

3) Operational Transparency: Users of EtherDelta enjoy **Operational Transparency**, provided by EtherDelta's blockchain-based design.

When depositing either Ether or ERC20 tokens, users can always verify transaction details via the public ledger (or via tools such as Etherscan). Furthermore, users can verify the order matching logic due to the open source nature of Ethereum smart contracts.

4) Fair Market Access: Since EtherDelta is an Ethereum-based platform, anyone with an Ethereum wallet can join it without any restrictions. In centralized exchanges, the centralized party more or less has the power to arbitrarily restrict a target victim from entering the market. This is mitigated by the distributed nature of the application.

5) Symmetric Information Access: In theory, users share the same market information on EtherDelta; however, EtherDelta can sometimes suffer from information asymmetry in practical settings. An example of this is that, when matching against an order, multiple users can attempt to match against the same counterparty order (i.e. a data race). Since the order book will only be updated upon transaction confirmation (i.e., when the transaction is actually mined), only one of the traders will successfully match against the target order. As a result, the other traders not only lose out on the trade, but also waste their gas. This kind of gas loss is caused by asymmetric information in that that users do not know who else matches against them. Within this context, EtherDelta largely fails to hold this property. Future improvements should focus on the mechanism that is used by EtherDelta to update the order book.

6) Order Queue Integrity: On EtherDelta, users can only submit one order to the blockchain to be mined in a single transaction. This has the effect of enforcing transaction ordering. Additionally, when users try to cancel a transaction, they need to submit a cancel transaction to the blockchain. This exposes EtherDelta users to potential frontrunning attacks. Users who want to cancel their transaction essentially must race against multiple arbitrageurs (who are most likely automated). As a result, due to the aforementioned information asymmetry, an arbitrageur can submit a taker transaction with a higher gas fee than the gas fee paid by the user for cancelling the transaction. This has been documented extensively in previous work[4].

This possibility of frontrunning implies that EtherDelta does not satisfy the **Order Queue Integrity** property.

7) Participant Anonymity: EtherDelta does not enforce any form of Know Your Customer (KYC) procedures and users are only identifiable via their Ethereum wallet addresses. As such, the only anonymity guarantees EtherDelta makes are the same as those made by the Ethereum design itself.

B. IDex

1) Description: IDex is a semi-decentralized exchange that provides users with a high-throughput, real-time trading facility, allowing the exchange of ERC20 tokens on IDex[13]. It was launched in October 2016 and is known as the first DEX on Ethereum that supports real-time trading[14]. Additionally, users are able to submit both market and limit orders to the exchange. Another notable feature of IDex is the absence of order cancellation fees. The design of IDex also enables users to submit multiple orders simultaneously.

IDex improves upon some existing flaws in decentralized exchanges. Firstly, block time can significantly limit trading speed, leading to the inability of users to fill multiple limit orders nor submit market orders (at least without severe slippage). Secondly, users incur a gas cost for cancelling orders. Consequently, this hampers the user experience of such DEXes. IDex solves these problems by centralizing some non-critical components. Namely, IDex controls the process of broadcasting authorized transactions to the network.

IDex has three core components:

Smart contract The smart contract is responsible for trustless storage of all assets and the execution of trade settlement. Users must verify all of their trades via their private keys.

Matching engine When users match against an order in the order book, the matching engine is responsible for both verification of the matched order and submission of the order to the pending order queue (managed by the arbiter). For market orders, the matching engine will automatically match against the top of the counterparty side of the book and then perform ordinary verification and queue submission.

Transaction processing arbiter The arbiter maintains an off-chain queue of pending orders and ensures that these orders are mined into the Ethereum blockchain in the correct order.

IDex maintains an off-chain order book. This system has order makers and order takers. To match an order there are five main steps:

1) Initially, both maker and taker need to deposit the tokens to the IDex contract.
2) The IDex database then updates balance information of the taker and maker.
3) The maker then creates and submits a signed order which contains trade information to their IDex.
4) Subsequently, IDex verifies the maker's account to make sure there are enough funds in it and checks the signed transaction.
5) Afterwards, the order is finally added in the order book.

For example, the maker wants to sell 0.1 ETH for 1000 AURA, so the maker submits a sell order and waits for the taker to buy. If the taker wants to trade AURA for ETH, once they have found a suitable counterparty order to match against, the following step for the taker is to submit a matching order and sign the transaction with the price in the sell order (the amount should be smaller or equal than 0.1 ETH). IDex then verifies the taker's account and checks the signed transaction. Finally, a sell order can be matched. Once every dependent trade has been mined, this matched order will be dispatched to the Ethereum mainnet for mining.

Despite the fact that the advantages of IDex contribute to a better user experience, there are still some risks of using IDex that are worthy of analysis.

2) Trading Integrity: Like many other DEXes that maintain an off-chain order book, users can submit their limit orders directly to the IDex order book. As a result, IDex cannot effectively prevent makers from creating orders to manipulate the quoted market price. Limit orders enable makers to set any price for the cryptocurrency they want to sell, giving participants the ability to conduct price manipulation. For instance, a participant can submit multiple sell orders to influence the price of a particular token (this is known as quote stuffing). Also, the overall trading volume of IDex is not overly large, leaving the price of tokens susceptible to easy manipulation. Thus, IDex does not maintain **Trading Integrity**. In order to make IDex support the property in the future, some potential extensions can be added to the system. For example, adding a module that enables the system to detect the abnormal behavior of creating spurious orders. Additionally, form a mechanism to instantly monitor the price of each token and analyze the dynamic changing of price.

3) Operational Transparency: IDex maintains the **Operational Transparency** property by ensuring that all orders are fully transparent. This is largely provided by the underlying blockchain consensus mechanism in Ethereum. When performing token exchange, users are only required to interact with the IDex smart contract. This is contrasted against centralised exchanges where users must trust a third party which processes transactions on their behalf. Often, such users cannot realistically access the details of how such transaction processing takes place.

4) Fair Market Access: The only requirements for market access to IDex are an Ethereum wallet and an IDex account. Given the blockchain-based nature of the platform, market access is open and freely available to all market participants.

5) Symmetric Information Access: In general, all the participants share the same market information on the IDex platform.

The nature of DEXes enables users to check the order book for buy/sell order information in a real-time. Additionally, users can also check pricing data for different tokens. For order matching, takers can choose counterparty buy/sell orders arbitrarily in order to make a transaction. The system subsequently matches the orders and executes a trade. Furthermore, this process is transparent to users.

However, there exist additional sources of information asymmetry. In a real-world case that we observed from public blockchain data, a user traded ETH for ICX via a market order, receiving 99 ICX for 7 ETH, rather than the fair market price of approximately 0.002 ETH. This is likely indicative of manipulative trading practices by predatory traders[15].

As a direction for future improvement of the IDex system, IDex should deliver warnings to users when the system detects atypically large changes in the bid-ask spread.

For these reasons, IDex only partially obeys the principle of **Symmetric Information Access**.

6) Order Queue Integrity: The arbiter component of IDex maintains a queue of pending orders. Only dependent transactions are mined, then the arbiter dispatches the next transaction to be mined into the Ethereum blockchain. This effectively en-

forces transaction ordering and commital. In addition to this practice, the IDex smart contract only accepts signed transactions (due to the nature of the Ethereum design itself). As users use their private keys to sign transactions, this prevents repudiation of previously-confirmed transactions.

As users can cancel existing orders (provided they have not already matched), no mined transaction is needed to cancel. This can significantly decrease the impact of previously documented frontrunning attacks[4].

As such, IDex maintains this property under realistic consensus settings.

7) Participant Anonymity: As IDex is an Ethereum-based platform, it previously supported anonymous trade - users were only identifiable by their Ethereum address. As of August 2019, IDex has implemented both KYC and Anti-Money Laundering (AML) protocols (presumably due to mounting regulatory pressure). Thus, it is now necessary for users to complete KYC procedures to trade via IDex. KYC is used to collect customers' information and verify their identities. Due to such stringent KYC requirements, IDex fails to hold the **Participant Anonymity** property.

C. Uniswap

1) Description: Uniswap is a DEX implementing the Automated Market Maker (AMM) [4] model for matching. As such, Uniswap comprises a liquidity pool and uses a constant product scheme as its pricing mechanism[16] [17].

The Uniswap project has undergone several major revisions, with the current stable version being Uniswap v1[16] (as of the time of writing). As such, we constrain our analysis to this version in this paper. The original prototype was initially deployed to the Ethereum mainnet in November of 2018[18].

To place a trade in Uniswap, a trader interacts with the Uniswap smart contract (either directly or via the website[19]). Unlike other types of DEXes, Uniswap (being an AMM), maintains a liquidity pool of its supported ERC20 tokens, rather than an order book. As such, a trader is performing a direct swap between their currently-held token (e.g. DAI) and their desired token (e.g. SNX). Note that Uniswap also maintains reserves of Ether (ETH) itself, facilitating the trade of ERC20 tokens for Ether and vice versa.

In addition to trading, users can also provide liquidity to the Uniswap protocol as well. Such users are known as *liquidity providers* in the Uniswap literature. Liquidity providers derive rewards from the operation of the protocol proportional to their capital contributions.

Several attractive properties of Uniswap's pricing mechanism (and of constant product markets in general) have been proven[20]. Particularly, Uniswap's tendency to strictly adhere to a reference price oracle (helping to minimise slippage and other *statistical* arbitrage opportunities) and the subsequent convexity of the arbitrage problem in the Uniswap context. Furthermore, these proofs are supported by realistic simulations of Uniswap trading activity.

The entirely on-chain nature of Uniswap (achieved by eliding order books of any kind) and its strong pricing guarantees provide extremely desirable qualities for traders and investors alike.

Despite these properties, Uniswap is still susceptible to both systemic risk in the DeFi ecosystem[21] and also DEX-specific weaknesses[4] [22][9].

2) Trading Integrity: As traders can submit orders to Uniswap smart contracts (either directly or indirectly) without any external oversight, Uniswap (like many DEXes) cannot maintain **Trading Integrity**.

Several strategies have been published which exploit this weakness of Uniswap and DEXes in general[4] [9].

Potential mitigations for this weakness are difficult to prescribe, due to the symmetry of dishonesty that exists. Further complicating this is, perhaps surprisingly, Uniswap's lack of an order book. This is because current literature for countering price spoofing in financial markets largely describes techniques requiring order books to be maintained[23]. Additionally, other literature focuses on legal frameworks for dealing with price manipulation[24] [25].

Due to these difficulties, we leave mitigations for violations of **Trading Integrity** with respect to Uniswap as a direction for future work.

3) Operational Transparency: Uniswap enjoys excellent **Operational Transparency** due to its smart contract-based implementation on the Ethereum mainnet.

In addition to this transparency, security audits of the Uniswap v1 implementation have been under-

taken[26], further strengthening the trustworthiness of the Uniswap exchange rules.

4) Fair Market Access: The lack of any authentication requirements and the open nature of blockchain technology in general means traders (and liquidity providers) all share equal access to the Uniswap markets.

While this is true at the application-level, there exist results in the literature describing application-level incentives for consensus-level security issues[4]. These issues mean that less technologically sophisticated traders face the potential risk of being frontrun - even after their trades have been settled.

For the purposes of our analysis, we consider these issues to be out-of-scope for Uniswap. As such, we consider Uniswap to obey the **Fair Market Access** principle.

5) Symmetric Information Access: As with many of the previous properties described, Uniswap's blockchain-based nature provides strong guarantees as to the transparency and accuracy of market activity due to the nature of the underlying consensus mechanism.

Traders can always view the public blockchain data (via Etherscan, for example) and determine the current balances of each of Uniswap's liquidity pools, every transaction any of the Uniswap smart contracts has had with other users, and any other recorded information.

While the open-access properties provided by blockchain technology provide for *eventual* consistency, there is the risk of state changes prior to block confirmation. Quantification of such risk to an individual trader is highly latency-sensitive and is more an issue of Ethereum network topology and the trader's specific infrastructure.

As such, we also consider such latency issues to be out-of-scope and leave this as a direction for future work.

6) Order Queue Integrity: At the application-level, Uniswap maintains **Order Queue Integrity** by simply not maintaining an order queue. As discussed above, Uniswap is an AMM, maintaining liquidity pools for direct swaps with traders using capital provided by liquidity providers. In this way, the integrity of the order queue is provided by the distributed ledger.

At the consensus-level, orders to Uniswap (any transaction, in fact) can theoretically be reordered either by gas-based frontrunning or potentially malicious miners (both cases are described at length in[4] but also in [9]).

Much like any consensus-level threats, such weaknesses are hard to mitigate against - especially at an application-level, or even user-level.

7) Participant Anonymity: As Uniswap is Ethereum-based, users are only identified by their Ethereum addresses - derived from their public keys.

This is an instance of the classic blockchain anonymity problem[27][28] [29][30]. Because of this, the anonymity provided by Uniswap to its users is limited by the overall anonymity of the Ethereum blockchain system and the individual operational security of the users themselves. As with any blockchain transaction, if an address can be associated with an external identity (e.g. an IP address, name, etc.) then both the trading activity and balance of the user's account (not only for Ether, but all held ERC20 tokens also) can be determined.

The mitigations for these threats are the typical mitigations for conventional blockchain anonymity. As such, we defer to the existing body of academic work.

TABLE I
SUMMARY OF ASSESSMENT FINDINGS

Property	EtherDelta	IDex	Uniswap
Trading Integrity	No	No	No
Operational Transparency	Yes	Yes	Yes
Fair Market Access	Yes	Yes	Yes
Symmetric Information Access	No	Partially	Yes
Order Queue Integrity	No	Yes	Yes
Participant Anonymity	Yes	No	Yes

D. Summary

We summarise the findings of our assessment in IV-D;

Note that, as DEX technology improves over time, higher assurance of our model's properties are achieved - with Uniswap achieving the best out of our three DEXes. This is a testament to the Automated Market Maker (AMM) design used by Uniswap.

V. CONCLUSION

We have presented an adapted form of an existing security model applied to traditional financial

markets and extended it to three decentralised exchanges operating on the Ethereum mainnet (each of unique design category).

In assessing each DEX, we observe that the fundamental nature of DApps imbues them with various attractive security properties. Namely, the pervasive transparency, high availablity, and easily verifiable behaviour of blockchain platforms and Ethereum smart contracts.

Despite these obvious advantages, there exist several classes of threats, both technical and financial, that affect users of DeFi applications. These are both numerous and intricate; they are thus left to future work (see VI).

In general, the problem of DEX security - and the security of DeFi in the broader sense - is an extremely challenging problem, both commercially and academically. The rapid pace of development, the intricacy of the protocol designs, and the relative infancy of practical smart contract implementations greatly complicate such analyses.

VI. Future Work

A. Open Questions

In light of our presented findings, there still remain several open questions that would be of interest to every major stakeholder in the DeFi space.

- How do other DEXes perform in the context of our security model?
- How will future versions of these DEXes change their performance with respect to our model?
- Are there obvious benefits of a particular DEX design (e.g. Automated Market Maker, etc.)?
- What threats are inherent to DEXes by their very nature?
- How do these threats alter trading strategies, market microstructure, and consensus-level security?
- How do these threats influence the growing DeFi landscape?

B. Directions for Future Work

1) Survey of Smart Contract Security: While security audits of several DEXes have been undertaken (often on behalf of their respective vendors), to the best of our knowledge, no widespread survey of the security of every major DEX has been compiled. Such a survey would enable a comparative analysis to be performed and thus a more holistic overview of the risks affecting the DeFi sector as a whole could be achieved.

Despite the obvious benefits of a broad scope, there exist obvious challenges in producing such work; namely, the timeliness of audit results. Due to the rapid pace of development in the DeFi space (and, indeed, the blockchain field in general), such findings can easily become obsolete prior to publication - and especially peer-review.

2) Open Collection of DeFi Market Manipulation Data: While blockchains themselves act as public append-only ledgers and have enjoyed long-standing searchable Web interfaces, the same open-access searchability would benefit collections of cryptocurrency market manipulation data. Such a resource would ideally collate transactions on the Ethereum mainnet that had been annotated as likely suspects for predatory trading activity.

Despite previous work[4] into assembling and processing transaction data in an effort to gain insight into *how* market manipulation occurs in decentralised cryptocurrency exchanges, we believe this can be extended to provide more comprehensive and automated access to price manipulation activity occurring in real-world DeFi deployments.

References

[1] "Cryptocurrency market capitalizations — coinmarketcap." (), [Online]. Available: https : / / coinmarketcap . com/ (visited on 05/15/2020).

[2] H. Tian, K. Xue, S. Li, J. Xu, J. Liu, and J. Zhao, *Enabling cross-chain transactions: A decentralized cryptocurrency exchange protocol*, 2020.

[3] "Tbtc: A decentralized redeemable btc-backed erc-20 token," Keep Network, May 7, 2020.

[4] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, *Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges*, 2019.

[5] V. Mavroudis, *Market manipulation as a security problem*, 2019.

[6] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts sok," in *Proceedings of the 6th International Conference on Principles of Security and Trust - Volume 10204*, Berlin, Heidelberg: Springer-Verlag, 2017, pp. 164–186, ISBN: 9783662544549. DOI: 10.1007/978-3-662-54455-6_8.

[7] V. Buterin. "Ethereum: A next-generation generalized smart contract and decentralized application platform." (Jan. 11, 2014), [Online]. Available: https://web.archive.org/web/20140111180823/http://ethereum.org/ethereum.html (visited on 05/18/2020).

[8] G. Wood. "Solidity." (Aug. 29, 2014), [Online]. Available: https://stackedit.io/viewer#!url=https://gist.githubusercontent.com/gavofyork/31b35cd2252a00d0d057/raw/16de06189d2175d2e31b300f1f8531e20c927635/solidity-original (visited on 05/18/2020).

[9] S. Eskandari, S. Moosavi, and J. Clark, *Sok: Transparent dishonesty: Front-running attacks on blockchain*, 2019.

[10] I. Bentov, L. Breidenbach, P. Daian, A. Juels, L. Yunqi, and Z. Xueyuan. "The cost of decentralization in 0x and etherdelta." (Aug. 13, 2017), [Online]. Available: https://hackingdistributed.com/2017/08/13/cost-of-decent/ (visited on 05/04/2020).

[11] H. Zhu, "Do Dark Pools Harm Price Discovery?" *The Review of Financial Studies*, vol. 27, no. 3, pp. 747–789, Dec. 2013, ISSN: 0893-9454. DOI: 10.1093/rfs/hht078.

[12] "Sec.gov — sec charges etherdelta founder with operating an unregistered exchange." (Nov. 8, 2018), [Online]. Available: https://www.sec.gov/news/press-release/2018-258 (visited on 05/27/2020).

[13] "Idex: A real-time and high-throughput ethereum smart contract exchange," Aurora Labs, Jan. 23, 2019.

[14] "Faq - idex decentralized ethereum asset exchange." (), [Online]. Available: https://idex.market/faq (visited on 05/20/2020).

[15] "Idex market order : Cryptocurrency." (Jan. 20, 2018), [Online]. Available: https://www.reddit.com/r/CryptoCurrency/comments/7ro2lf/idex_market_order/ (visited on 05/20/2020).

[16] H. Adams. "Uniswap whitepaper." (Apr. 29, 2018), [Online]. Available: https://hackmd.io / @Uniswap / HJ9jLsfTz (visited on 05/11/2020).

[17] Y. Zhang, X. Chen, and D. Park, "Formal Verification of Constant Product ($x \times y = k$) Market Maker Model and Implementation," Runtime Verification, Incorporated, Tech. Rep., Oct. 24, 2018.

[18] H. Adams. "A short history of uniswap." (Nov. 2, 2019), [Online]. Available: https://uniswap.org/blog/uniswap-history/ (visited on 05/14/2020).

[19] Uniswap, *Uniswap exchange*.

[20] G. Angeris, H.-T. Kao, R. Chiang, C. Noyes, and T. Chitra, *An analysis of uniswap markets*, 2019.

[21] L. Gudgeon, D. Perez, D. Harz, A. Gervais, and B. Livshits, *The decentralized financial crisis: Attacking defi*, 2020.

[22] K. Qin, L. Zhou, B. Livshits, and A. Gervais, *Attacking the defi ecosystem with flash loans for fun and profit*, 2020.

[23] X. Wang, Y. Vorobeychik, and M. P. Wellman, "A cloaking mechanism to mitigate market manipulation," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, International Joint Conferences on Artificial Intelligence Organization, Jul. 2018, pp. 541–547. DOI: 10.24963/ijcai.2018/75.

[24] S. D. Ledgerwood and P. R. Carpenter, "A framework for the analysis of market manipulation," *Review of Law & Economics*, vol. 8, no. 1, pp. 253–295, 2012.

[25] "Cp 202 dark liquidity and high frequency trading: Proposals," Australian Securities & Investments Commission, Mar. 18, 2013.

[26] C. Diligence. "Uniswap audit." (Jan. 11, 2019), [Online]. Available: https://github.com/ConsenSys/Uniswap-audit-report-2018-12 (visited on 05/15/2020).

[27] M. Moser, "Anonymity of bitcoin transactions," 2013.

[28] W. Ladd, *Blind signatures for bitcoin transaction anonymity*, 2012.

[29] J. Herrera-Joancomart, "Research and challenges on bitcoin anonymity," in *Data Privacy Management, Autonomous Spontaneous*

Security, and Security Assurance, Springer, 2014, pp. 3–16.

[30] M. S. Ortega, "The bitcoin transaction graph—anonymity," Ph.D. dissertation, Master's thesis, Universitat Oberta de Catalunya, 2013.