SoK: Decentralized Exchanges (DEX) with Automated Market Maker (AMM) Protocols

JIAHUA XU, UCL Centre for Blockchain Technologies
KRZYSZTOF PARUCH, Vienna University of Economics and Business, Austria
SIMON COUSAERT, The Block
YEBO FENG, University of Oregon, USA

As an integral part of the decentralized finance (DeFi) ecosystem, decentralized exchanges (DEXs) with automated market maker (AMM) protocols have gained massive traction with the recently revived interest in blockchain and distributed ledger technology (DLT) in general. Instead of matching the buy and sell sides, AMMs employ a peer-to-pool method and determine asset price algorithmically through a so-called conservation function. To facilitate the improvement and development of AMM-based DEXs, we create the first systematization of knowledge in this area. We first establish a general AMM framework describing the economics and formalizing the system's state-space representation. We then employ our framework to systematically compare the top AMM protocols' mechanics, illustrating their conservation functions, as well as slippage and divergence loss functions. We further discuss security and privacy concerns, how they are enabled by AMM-based DEXs' inherent properties, and explore mitigating solutions. Finally, we conduct a comprehensive literature review on related work covering both DeFi and conventional market microstructure.

CCS Concepts: • **Information systems → Digital cash**; *Secure online transactions*; **Electronic funds transfer**; • **General and reference → Surveys and overviews**; • **Social and professional topics → Economic impact**; • **Security and privacy**;

Additional Key Words and Phrases: Decentralized finance, decentralized exchange, automated market maker, blockchain, Ethereum

ACM Reference format:
Jiahua Xu, Krzysztof Paruch, Simon Cousaert, and Yebo Feng. 2023. SoK: Decentralized Exchanges (DEX) with Automated Market Maker (AMM) Protocols. *ACM Comput. Surv.* 55, 11, Article 238 (February 2023), 50 pages.
https://doi.org/10.1145/3570639

1 INTRODUCTION

With the revived interest in blockchain and cryptocurrency among both the general populace and institutional actors, the past year has witnessed a surge in crypto trading activity and accelerated

This material is based upon work partially supported by Ripple under the University Blockchain Research Initiative (UBRI) [79].

Authors' addresses: J. Xu, University College London, 66-72 Gower Street, London, WC1E 6EA; email: jiahua.xu@ucl.ac.uk; K. Paruch, Gebäude AR, 6.OG, Perspektivstraße 4, 1020 Vienna, Austria; email: krzysztof.paruch@wu.ac.at; S. Cousaert, Willem Wenemaerstraat 14 9000, Gent, Oost-Vlaanderen, Belgium; email: scousaert@theblock.co; Y. Feng (corresponding author), 1202 University of Oregon, 1477 E. 13th Ave., Eugene, Oregon, USA, 97403-1202; email: yebof@uoregon.edu.

0360-0300/2023/02-ART238 $15.00
https://doi.org/10.1145/3570639

development in the decentralized finance (DeFi) space. Among all the prominent DeFi applications, decentralized exchanges (DEXs) with automated market maker (AMM) protocols are in the ascendancy, with an aggregate value locked exceeding $100 billion at the time of writing [1].

An AMM-based DEX bears attractive features such as decentralization, automation, and continuous liquidity. With traditional order-book-based exchanges, the market price of an asset is determined by the last matched buy and sell orders, ultimately driven by the supply and demand of the asset. In contrast, on an AMM-based DEX, a liquidity pool acts as a single counterparty for each transaction, with a so-called conservation function that prices assets algorithmically by only allowing the price to move along predefined trajectories. AMMs implement a peer-to-pool method, where liquidity providers (LPs) contribute assets to liquidity pools while individual users exchange assets with a pool or pools containing the input and the output assets. Users obtain immediate liquidity without having to find an exchange counterparty, whereas LPs profit from asset supply with exchange fees from users. Furthermore, by using a conservation function for price setting, AMMs render moot the necessity of maintaining the state of an order book, which would be costly on a distributed ledger.

AMMs benefit both LPs and exchange users with accessible liquidity provision and exchange, especially for illiquid assets. Despite the apparent advantages, AMMs are often characterized by high slippage and divergence loss, two implicit economic risks imposed on the funds of exchange users and LPs, respectively. Moreover, AMM-based DEX are associated with myriads of security and privacy issues. Throughout the past three years, new protocols have been introduced to the market one after another with incremental improvements, attempting to tackle different issues that had been identified as weak spots in a previous version. On top of that, new use cases are being addressed, and new applications of AMM iterations are proposed to the market. While innovative in certain aspects, the various AMM protocols generally consist of the same set of composed mechanisms to allow for multiple functionalities of the system. As such, these systems are structurally similar, and their main differences lie in parameter choices and/or mechanism adaptations.

As one of the earliest DeFi application categories, AMM-based DEXs also constitute the most fundamental and crucial building blocks of the DeFi ecosystem. Their importance prompts the emergence of studies covering various aspects of AMM-based DEXs: economics [25], security [219, 227, 228], privacy [27, 32], and so on. While there has arisen a large number of surveys on general DeFi [177, 180] as well as various DeFi applications [24, 50], there is a dearth of literature that systematically studies and critically examines DEXs with AMM protocols in a comprehensive manner. To the best of our knowledge, this article fills this gap as the first SoK on AMM-based DEX with examples of deployed protocols. We contribute to the body of literature mainly by:

(1) generalizing mechanisms and economics of AMM-based DEXs with a formalized state space modeling framework and a summary of key common properties;
(2) illustrating instantiations of our framework by comparing major AMM-based DEXs with mathematical derivation and parameterized visualization on their conservation function, slippage, and divergence loss functions;
(3) positioning AMM-based DEXs within the broader taxonomy of DeFi and examining their relationships and interactions with other DeFi protocols;
(4) establishing a taxonomy of security and privacy issues concerning AMM-based DEXs and exploring mitigation solutions;
(5) conducting a state-of-the-art literature review summarizing current research priorities as well as existing output in AMM-based DEXs and related fields and identifying potential directions for future research.

The rest of the article is structured as follows: in Section 2, we lay out fundamental concepts and components of AMMs; in Section 3, we formalize AMM mechanisms with a state-space representation; in Section 4, we compare main protocols in terms of conservation function, exchange rates, slippage, and impermanent loss; in Section 5, we address issues related to security and privacy of AMM-based DEX; in Section 6, we indicate several avenues of future AMM research; in Section 7, we present related work; in Section 8, we conclude.

2 AMM PRELIMINARIES

This section presents AMMs-based DEXs' main components, including different actors and assets, as well as their generalized mechanism and economics.

2.1 Actors

2.1.1 Liquidity Provider (LP). A liquidity pool can be deployed through a smart contract with some initial supply of crypto assets by the first LP. Other LPs can subsequently increase the pool's reserve by adding more of the type of assets that are contained in the pool. In turn, they receive pool shares proportionate to their liquidity contribution as a fraction of the entire pool [76]. LPs earn transaction fees paid by exchange users. While sometimes subject to a withdrawal penalty, LPs can freely remove funds from the pool [129] by surrendering a corresponding amount of pool shares [76].

2.1.2 Exchange User (Trader). A trader submits an exchange order to a liquidity pool by specifying the input and output asset and either an input asset or output asset quantity; the smart contract automatically calculates the exchange rate based on the conservation function as well as the transaction fee and executes the exchange order accordingly.

Arbitrageurs compare asset prices across different markets to execute trades whenever closing price gaps can extract profits [125]. AMMs such as DODO (see Section 4.1.5) leverage users' arbitrage behavior through their protocol design.

2.1.3 Protocol Foundation. A protocol foundation consists of protocol founders, designers, and developers responsible for architecting and improving the protocol. The development activities are often funded directly or indirectly through accrued earnings such that the foundation members are financially incentivized to build a user-friendly protocol that can attract high trading volume.

2.2 Assets

Several distinct types of assets are used in AMM protocols for operations and governance; one asset may assume multiple roles.

2.2.1 Risk Assets. Characterized by illiquidity, risk assets are the primary type of assets AMM-based DEXs are designed for. Like centralized exchanges, an AMM-based DEX can facilitate an initial exchange offering (IEO) to launch a new token through liquidity pool creation, a capital raising activity termed "initial DEX offering (IDO)" that is particularly suitable for illiquid assets. To be eligible for an IDO, a risk asset sometimes needs to be whitelisted and must be compatible with the protocol's technical requirements (e.g., ERC20 [2] for most AMMs on Ethereum).

2.2.2 Base Assets. Some protocols require a trading pair always to consist of a risk asset and a designated base asset. In the case of Bancor, every risk asset is paired with BNT, the protocol's native token [22]. Uniswap V1 requires every pool to be initiated with a risk asset paired with ETH. Many protocols, such as Balancer and Curve, can connect two or more risk assets directly in liquidity pools without a designated base asset.

2.2.3 Pool Shares. Also known as "liquidity shares" and "LP shares," pool shares represent ownership in the portfolio of assets within a pool and are distributed to LPs. Shares accrue trading fees proportionally and can be redeemed at any time to withdraw funds from the pool.

2.2.4 Protocol Tokens. Protocol tokens are used to represent voting rights on protocol governance matters and are thus also termed "governance tokens" (see Section 2.4.1). Protocol tokens are typically valuable assets [216] that are tradeable outside of the AMM and can incentivize participation when, e.g., rewarded to LPs proportionate to their liquidity supply. AMMs compete with each other to attract funds and trading volume. To bootstrap an AMM in the early phase with incentivized early pool establishment and trading, a feature called liquidity mining can be installed where the native protocol's tokens are minted and issued to LPs and/or exchange users.

2.3 Fundamental AMM Dynamics

2.3.1 Invariant Properties. The functionality of an AMM depends upon a *conservation function* that encodes a desired invariant property of the system. As an intuitive example, Uniswap's constant product function determines trading dynamics between assets in the pool, as it always conserves the product of value-weighted quantities of both assets in the protocol—each trade has to be made in a way such that the value removed in one asset equals the value added in the other asset. This weight-preserving characteristic is one desired invariant property supported by the design of Uniswap.

2.3.2 Mechanisms. An AMM typically involves two types of interaction mechanism: asset swapping of assets and liquidity provision/withdrawal. Interaction mechanisms have to be specified in a way such that desired invariant properties are upheld; therefore, the class of admissible mechanisms is restricted to the ones that respect the defined conservation function, if one is specified, or conserve the defined properties otherwise.

2.4 Fundamental AMM Economics

2.4.1 Rewards. AMM protocols often run several reward schemes, including liquidity reward, staking reward, governance rights, and security reward distributed to different actors to encourage participation and contribution.

Liquidity reward. LPs are rewarded for supplying assets to a liquidity pool, as they have to bear the opportunity costs associated with funds being locked in the pool. LPs receive their share of trading fees paid by exchange users.

Staking reward. On top of the liquidity reward in the form of transaction income, LPs are offered the possibility to stake pool shares or other tokens as part of an initial incentive program from a certain token protocol. The ultimate goal of the individual token protocols (see, e.g., `GIV` [53] and `TRIPS` [57]) is to further encourage token holding, while simultaneously facilitating token liquidity on exchanges and product usage. These staking rewards are given by protocols other than the AMM.

Governance right. An AMM may encourage liquidity provision and/or swapping by rewarding participants governance rights in the form of protocol tokens (see Section 2.2.4). Currently, governance issues such as protocol treasury management [61] are proposed and discussed mostly on off-chain governance portals such as snapshot (snapshot.org), Tally (tally.xyz), and Boardroom (boardroom.io), where protocol tokens are used as ballots (see Section 2.2.4) to vote on proposals.

Security reward. Just as every protocol built on top of an open, distributed network, AMM-based DEXs on Ethereum suffer from security vulnerabilities. Besides code auditing, a common practice that a protocol foundation adopts is to have the code vetted by a broader developer community and reward those who discover and/or fix bugs of the protocol with monetary prizes, commonly in fiat currencies, through a bounty program [34].

2.4.2 Explicit Costs. Interacting with AMM protocols incurs various costs, including charges for some form of “value” created or “service” performed and fees for interacting with the blockchain network. AMM participants need to anticipate three types of fees: liquidity withdrawal penalty, swap fee, and gas fee.

Liquidity withdrawal penalty. As introduced in Section 3.2 and further discussed in Section 4 of this article, withdrawal of liquidity changes the shape of the conservation function and negatively affects the usability of the pool by elevating slippage. Therefore, AMMs such as DODO [64] levy a liquidity withdrawal penalty.

Swap fee. Users interacting with the liquidity pool for token exchanges have to reimburse LPs for the supply of assets and for the divergence loss (see Section 2.4.3). This compensation comes in the form of swap fees that are charged in every exchange trade and then distributed to liquidity pool shareholders. A small percentage of the swap fees may also go to the foundation of the AMM to further develop the protocol [215].

Gas fee. Every interaction with the protocol is executed in the form of an on-chain transaction and is thus subject to a gas fee applicable to all transactions on the underlying blockchain. In a decentralized network, validating nodes need to be compensated for their efforts, and transaction initiators must cover these operating costs. Interacting with more complex protocols will result in a higher gas fee due to the higher computational power needed for transaction verification.

2.4.3 Implicit Costs. Two essential implicit costs native to AMM-based DEXs are slippage for exchange users and divergence loss for LPs.

Slippage. Slippage is defined as the difference between the spot price and the realized price of a trade. Instead of matching buy and sell orders, AMMs determine exchange rates on a continuous curve, and *every* trade will encounter slippage conditioned upon the trade size relative to the pool size and the exact design of the conservation function. The spot price approaches the realized price for infinitesimally small trades, but they deviate more for bigger trade sizes. This effect is amplified for smaller liquidity pools, as every trade will significantly impact the relative quantities of assets in the pool, leading to higher slippage. Due to continuous slippage, trades on AMMs must be set with some slippage tolerance to be executed, a feature that can be exploited to perform, e.g., sandwich attacks (see Section 5.1.3).

Divergence loss. For LPs, assets supplied to a protocol are still exposed to volatility risk, which comes into play in addition to the loss of time value of locked funds. A swap alters the asset composition of a pool, which automatically updates the asset prices implied by the conservation function of the pool (Equation (3)). This consequently changes the value of the entire pool. Compared to holding the assets outside of an AMM pool, contributing the same amount of assets to the pool in return for pool shares can result in less value with price movement, an effect termed “divergence loss” or “impermanent loss” (see Section 4). This loss can be deemed “impermanent,” because as asset price moves back and forth, the depreciation of the pool value continuously disappears and reappears and is only realized when assets are actually taken out of the pool. Well-devised AMMs charge appropriate swap fees to ensure that LPs are sufficiently compensated for the divergence loss (see Section 4.2.2). Despite the fact that “impermanent loss” is a more widely used term on the Internet, we adhere to the more accurate term “divergence loss” in a scientific context. In fact, for

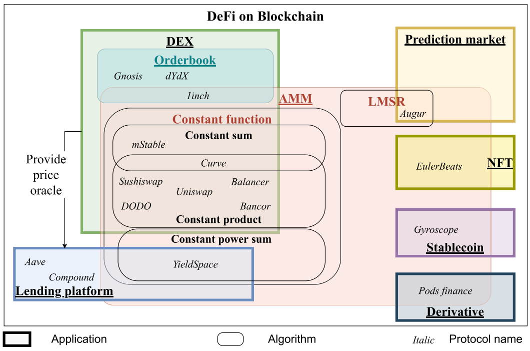


Fig. 1. AMM-based DEX within the broader taxonomy of DeFi on blockchain. AMM as an algorithm and DEX as an application are not mutually inclusive.

the majority of AMM protocols, this "loss" only disappears when the current proportions of the pool assets equal exactly those at liquidity provision, which is rarely the case.

Since assets are bonded together in a pool, changes in prices of one asset affect all others in this pool. For an AMM protocol that supports single-asset supply, this forces LPs to be exposed to risk assets they have not been holding in the first place (see Section 3.3.1).

2.5 AMM-based DEXs within DeFi

For brevity, we use "AMM" or "DEX" to refer to AMM-based DEX throughout the article, unless indicated otherwise. Nevertheless, it is to be noted that the term "AMM" emphasizes the algorithm of a protocol, whereas "DEX" emphasizes the use case, or application, of a protocol. Within the context of blockchain-based DeFi, there also exist orderbook-based DEXs such as Gnosis and dYdX that do not rely on AMM algorithms. Recently, DEX aggregators (Section 4.3.8) such as 1inch have emerged that incorporate both limited order books and AMM pools. However, AMM algorithms are also not exclusively employed by DEXs. DeFi applications such as lending platforms, non-fungible tokens (NFTs), stablecoins, and derivatives all have protocols that make use of different AMM algorithms.

AMMs can also assume various forms (see Section 7.2.2). Prediction markets for example commonly employs logarithmic market scoring rule (LMSR), whereas constant function market maker (CFMM) is the primary underpinning for DEXs. In particular, constant sum and constant product are the most representative forms of CFMM, widely adopted by AMM-based DEX protocols. Figure 1 illustrates AMM-based DEX within the broader taxonomy of DeFi on blockchain.

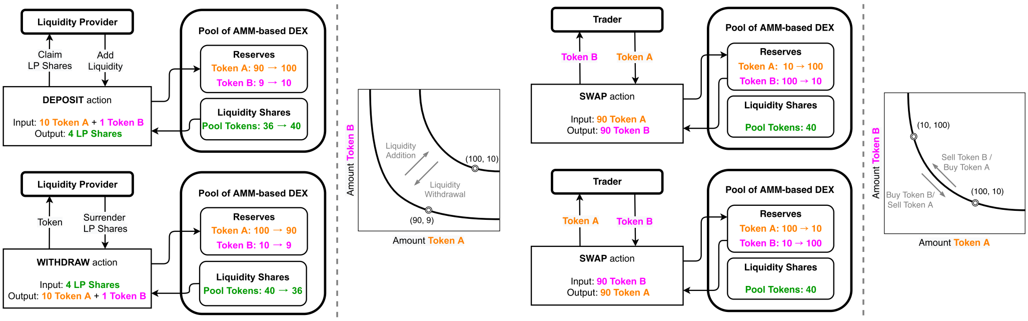
The ensuing sections, Sections 3 and 4, focus on CFMM mechanisms that have been adopted by major DEXs, with their exact formulas derived in Appendix A. Section 4.3 briefly presents other DeFi applications with AMM implementations.

3 FORMALIZATION OF MECHANISMS

Overall, the functionality of an AMM can be generalized formally by a set of few mechanisms. These mechanisms define how users can interact with the protocol and what the response of the protocol will be given particular user actions.

3.1 State Space Representation

The functioning of any blockchain-based system can be modeled using state-space terminology. States and agents constitute the main system components; protocol activities are described as



(a) Liquidity provision and withdrawal.

(b) Swap.

Fig. 2. Stylized AMM mechanisms for LPs (Section 2.1.1) and traders (Section 2.1.2).

actions (Figure 2); the evolution of the system over time is modeled with state transition functions. This can be generalized into a state transition function f encoded in the protocol such that $\chi \xrightarrow[f]{a} \chi'$, where $a \in A$ represents an action imposed on the system, while χ and χ' represent the current and future states of the system, respectively.

The object of interest is the state χ of the liquidity pool, which can be described with

$$\chi = (\{r_k\}_{k=1,\ldots,n}, \{p_k\}_{k=1,\ldots,n}, \mathcal{I}, \Omega), \tag{1}$$

where r_k denotes the quantity of token$_k$ in the pool, p_k the current spot price of token$_k$, $\mathcal{I}$ the conservation function invariant(s), and Ω the collection of protocol hyperparameters. This formalization can encompass various AMM designs.

The most critical design component of an AMM is its conservation function that defines the relationship between different state variables and the invariant(s) $\mathcal{I}$. The conservation function is protocol-specific, as each protocol seeks to prioritize a distinct feature and target particular functionalities (see Section 4).

The core of an AMM system state is the quantity of each asset held in a liquidity pool. Their sums or products are typical candidates for invariants. Examples of a constant-sum market maker include mStable [10]. Uniswap [196] represents constant-product market makers, while Balancer [129] generalizes this idea to a geometric mean. The Curve [68] conservation function is notably a combination of constant-sum and constant-product (see Section 4).

3.2 Generalized Formulas

In this section, we generalize AMM formulas necessary for demonstrating the interdependence between various AMM invariants and state variables, as well as for computing slippage and divergence loss. Mathematical notations and their definitions can be found in Table 1.

Hyperparameter set Ω is determined at pool creation and shall remain the same afterwards. While the value of hyperparameters might be changed through protocol governance activities, this does not and should not occur on a frequent basis.

Invariant $\mathcal{I}$, despite its name, refers to the pool variable that stays constant only with swap actions (see Section 3.3.2) but changes at liquidity provision and withdrawal. In contrast, trading moves the price of traded assets; specifically, it increases the price of the output asset relative to the input asset, reflecting a value appreciation of the output asset driven by demand (see Section 3.3.5). Liquidity provision and withdrawal, however, should not move the asset price (see Section 3.3.1).

Table 1. Mathematical Notations for Pool Mechanisms

Notation	Definition	Applicable protocols
Preset hyperparameters, Ω		
w_k	Weight of asset reserve r_k	Balancer
$\mathcal{A}$	Slippage controller	Uniswap V3, Curve, DODO
n	Number of assets in a pool $\left(n \begin{cases} = 2 \text{ for asset-pair pools} \\ > 2 \text{ for multi-asset pools} \end{cases}\right)$	Curve
Conservation function invariants, $\mathcal{I}$		
$\mathcal{K}$	Conservation function constant	Uniswap V2, Balancer, Curve
$\mathcal{R}_k$	Initial reserve of token$_k$	Uniswap V3, DODO
State variables		
r_k	Quantity of token$_k$ in the pool	all
p_k	Current spot price of token$_k$	all
Process variables		
x_i	Input quantity added to token$_i$ reserve (removed if $x_i < 0$)	all
x_o	Output quantity removed from token$_o$ reserve (added if $x_o < 0$)	all
ρ	Token value change	all
Functions		
C	Conservation function	all
Z	Implied conservation function	all
${}_iE_o$	token$_o$ price in terms of token$_i$	all
S	Slippage	all
V	Reserve value	all
L	Divergence loss	Uniswap, Balancer, Curve

In particular, a *pure* liquidity provision and withdrawal activity requires a proportional change in reserves (Equation (2)).

Formally, the state transition induced by *pure* liquidity change and asset swap can be expressed as follows:

$$(\{r_k\}_{k=1,\dots,n}, \{p_k\}_{k=1,\dots,n}, \mathcal{I}, \Omega) \xrightarrow[f]{\text{liquidity change}} (\{a \cdot r_k\}_{k=1,\dots,n}, \{p_k\}_{k=1,\dots,n}, \mathcal{I}', \Omega), \text{ where } a > 0 \tag{2}$$

$$(\{r_k\}_{k=1,\dots,n}, \{p_k\}_{k=1,\dots,n}, \mathcal{I}, \Omega) \xrightarrow[f]{\text{swap}} (\{r'_k\}_{k=1,\dots,n}, \{p'_k\}_{k=1,\dots,n}, \mathcal{I}, \Omega). \tag{3}$$

3.2.1 Conservation Function. An AMM conservation function, also termed "bonding curve," can be expressed explicitly as a relational function between AMM invariant and reserve quantities $\{r_k\}_{k=1,\dots,n}$:

$$\mathcal{I} = C(\{r_k\}). \tag{4}$$

A conservation function for each token pair, say, r_i—r_o, must be concave, nonnegative, and nondecreasing [12] (see also Figure 3). For complex AMMs such as Curve, it might be convenient to express the conservation function (Equation (4)) implicitly to derive exchange rates between two assets in a pool:

$$Z(\{r_k\}; \mathcal{I}) = C(\{r_k\}) - \mathcal{I} = 0. \tag{5}$$

Equation (5) contains invariants $\mathcal{I}$, whose value is determined by the initial liquidity provision (liquidity pool creation); afterwards, given the change in reserve quantity of one asset, the reserve quantity of the other asset can be solved.

3.2.2 Spot Exchange Rate. The spot exchange rate between token$_i$ and token$_o$ can be calculated as the slope of the r_i—r_o curve (see examples in Figure 3) using partial derivatives of the conservation function Z.

$$ {}_iE_o(\{r_k\};\mathcal{I}) = \frac{\partial Z(\{r_k\};\mathcal{I})/\partial r_o}{\partial Z(\{r_k\};\mathcal{I})/\partial r_i} \tag{6}$$

Note that ${}_iE_o = 1$ when $i = o$.

3.2.3 Swap Amount. The amount of token$_o$ received x_o (spent when $x_o < 0$) given amount of token$_i$ spent x_i (received when $x_i < 0$) can be calculated following the steps below.

Note that $x_i > -r_i$ and $x_o > -r_o$. Their lower bound corresponds to the case when the received asset is depleted from the pool, i.e., its new reserve becomes 0 (see also Equation (7) below). With common AMM protocols, x_i, x_o theoretically often do not have a upper bound: If the reserve quantity is 1 unit, then a trader can still sell 2 or more units into the pool, but mostly accompanied with a high slippage (see Figure 4 in the next section).

Update reserve quantities. Input quantity x_i is simply added to the existing reserve of token$_i$; the reserve quantity of any token other than token$_i$ or token$_o$ stays the same:

$$ r_i' := R_i(x_i; r_i) = r_i + x_i \tag{7}$$

$$ r_j' = r_j, \qquad \forall j \neq i, o. \tag{8}$$

Compute new reserve quantity of token$_o$. The new reserve quantity of all tokens except for token$_o$ is known from the previous step. One can thus solve r_o', the unknown quantity of token$_o$, by plugging it in the conservation function:

$$ Z(\{r_k'\};\mathcal{I}) = 0. \tag{9}$$

Apparently, r_o' can be expressed as a function of the original reserve composition $\{r_k\}$, input quantity x_i, namely,

$$ r_o' := R_o(x_i, \{r_k\};\mathcal{I}). \tag{10}$$

Compute swapped quantity. The quantity of token$_o$ swapped is simply the difference between the old and new reserve quantities:

$$ x_o := X_o(x_i, \{r_k\};\mathcal{I}) = r_o - r_o'. \tag{11}$$

3.2.4 Slippage. Slippage measures the deviation between effective exchange rate $\frac{x_i}{x_o}$ and the pre-swap spot exchange rate ${}_iE_o$, expressed as:

$$ S(x_i, \{r_k\};\mathcal{I}) = \frac{x_i/x_o}{{}_iE_o} - 1. \tag{12}$$

3.2.5 Divergence Loss. Divergence loss describes the loss in value of all reserves in the pool compared to holding the reserves outside of the pool, after a price change of an asset (see Section 2.4.3). Based on the formulas for spot price and swap quantity established above, the divergence loss can generally be computed following the steps described below. In the valuation, we assign token$_i$ as the denominating currency for all valuations. While the method to be presented can be used for multiple token price changes through iterations, we only demonstrate the case where only the value of token$_o$ increases by ρ, while all other tokens' value stay the same. Token$_i$ is the numéraire. Designating one of the tokens in the pool as a numéraire can also be found in DeFi simulation papers such as Reference [12].

Calculate the original pool value. The value of the pool denominated in token$_i$ can be calculated as the sum of the value of all token reserves in the pool, each equal to the reserve quantity multiplied by the exchange rate with token$_i$:

$$V(\{r_k\};\mathcal{I}) = \sum_j {}_iE_j(\{r_k\};\mathcal{I}) \cdot r_j. \tag{13}$$

Calculate the reserve value if held outside of the pool. If all the asset reserves are held outside of the pool, then a change of ρ in token$_o$'s value would result in a change of ρ in token$_o$ reserve's value:

$$V_{\text{held}}(\rho;\{r_k\},\mathcal{I}) = V(\{r_k\};\mathcal{I}) + [{}_jE_o(\{r_k\};\mathcal{I}) \cdot r_o] \cdot \rho.$$

Obtain re-balanced reserve quantities. Exchange users and arbitrageurs constantly re-balance the pool through trading in relatively "cheap," depreciating tokens for relatively "expensive," appreciating ones. As such, asset value movements are reflected in exchange rate changes implied by the dynamic pool composition. Therefore, the exchange rate between token$_o$ and each other token$_j$ ($j \neq o$) implied by new reserve quantities $\{r'_k\}$, compared to that by the original quantities $\{r_k\}$, can be expressed with Equation set (14). At the same time, the equation for the conservation function must stand (Equation (15)):

$$\rho = \frac{{}_jE_o(\{r'_k\};\mathcal{I})}{{}_jE_o(\{r_k\};\mathcal{I})} - 1, \qquad \forall j \neq o \tag{14}$$

$$0 = Z(\{r'_k\};\mathcal{I}). \tag{15}$$

A total number of n-equations ($n-1$ with Equation set (14), plus 1 with Equation (15)) would suffice to solve n unknown variables $\{r'_k\}_{k=1,\ldots,n}$, each of which can be expressed as a function of ρ and $\{r_k\}$:

$$r'_k := R_k(\rho,\{r_k\};\mathcal{I}). \tag{16}$$

Calculate the new pool value. The new value of the pool can be calculated by summing the products of the new reserve quantity multiplied by the new price (denominated by token$_i$) of each token in the pool:

$$V'(\rho,\{r_k\};\mathcal{I}) = \sum_j {}_iE_j(\{r'_k\};\mathcal{I}) \cdot r'_j. \tag{17}$$

Calculate the divergence loss. Divergence loss can be expressed as a function of ρ, the change in value of an asset in the pool:

$$L(\rho,\{r_k\};\mathcal{I}) = \frac{V'(\rho,\{r_k\};\mathcal{I})}{V_{\text{held}}(\rho;\{r_k\},\mathcal{I})} - 1. \tag{18}$$

3.3 Key Common Properties of AMM-based DEXs

In this section, we summarize key *common* properties featured in AMM-based DEXs. We also clarify that protocol-specific intricacies and real-life implementations may result in "violations" of certain properties listed below.

3.3.1 Zero-impact Liquidity Change. The price of assets in an AMM pool stays constant for *pure*, balanced liquidity provision and withdrawal activities. This feature describes when an LP provides or withdraws liquidity, usually by linearly scaling up or down the existing reserves in the pool, no price impact shall occur (see Equation (2)).

The asset spot price can remain the same only when assets are added to or removed from a pool proportionate to the current reserve ratio ($r_1 : r_2 : \cdots : r_n$). In any other case, a change of

Table 2. Comparison Table of Discussed

Protocol	Value locked ($bn)	Trade volume ($bn)	Market (%)	Governance token	Governance token holders	Fully diluted value ($bn)
Uniswap	6.15	11.4	66.7	UNI	269,923	21.1
Sushiswap	3.92	2.9	14.2	SUSHI	71,007	2.4
Curve	11.64	1.8	6.4	CRV	44,654	4.0
Bancor	1.37	0.4	2.5	BNT	38,124	0.8
Balancer	1.74	0.5	2.2	BAL	37,613	1.0
DODO	0.07	0.4	2.1	DODO	11,330	1.2

DEX: Value Locked, Trade Volume of the Past 7 Days, the Market Share by the Last 30 Days Volume, the Governance Token, the Number of Governance Token Holders and the Fully Diluted Value, as on 21/09/2021. Data retrieved from DeFi Pulse and Dune Analytics.

quantities in any pool would result in changes in relative prices of assets. To manage to uphold the invariances a disproportionate addition or removal can be treated as a combination of two actions: proportionate reserve change plus asset swap (see, e.g., Section 4.1.3).

3.3.2 Path-deterministic Swap. By its algorithm-based pricing nature, how a given swap transitions an AMM pool's reserve balance can be deterministically computed.

In an idealized, frictionless market, an AMM pool's conservation function (see Section 3.2.1) stipulates that the pool's invariant stays constant for *pure* swapping activities (see Equation (3)). Figuratively speaking, absent additional liquidity provision/withdrawal, the coordinates of the reserve quantities in a liquidity pool would always slide up and down along the bonding curve (see Figure 2(b)) through swaps. In reality, swap fees (see Section 2.4.2), when kept within a pool, cause invariant $\mathcal{I}$ to become variant through trading. Also, as float numbers are not yet fully supported by Solidity [71]—the language for Ethereum smart contracts—AMM protocols typically recalculate invariant $\mathcal{I}$ after each trade to avoid the accumulation of rounding errors.

3.3.3 Output-boundedness. With an output-bounded AMM, there is always a sufficient quantity of output tokens for a swap, i.e., a user can never deplete one side of the pool reserve. An AMM with this feature usually constructs its bonding curve such that, when one reserve token is close to depletion (approaching 0), its price—denominated in the other reserve token of the pool—becomes astronomically high (approaching infinity) [25].

Output-boundedness usually applies to continuous AMMs. Hybrid AMMs such as Uniswap V3, which incorporates bounded bonding curves, assimilating an order-book-like mechanism [46], naturally do not carry this property.

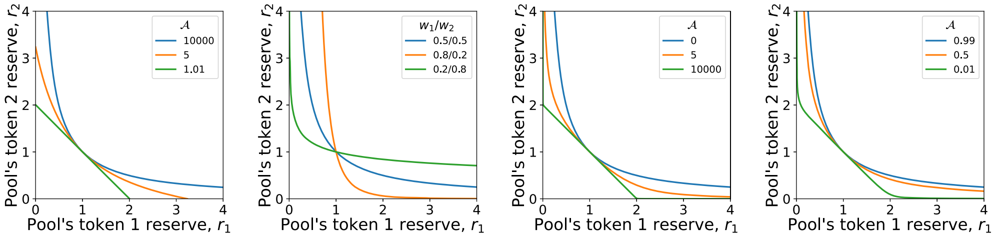
3.3.4 Liquidity Sensitivity. An AMM is liquidity-sensitive when a fixed swap size (same input quantity x_i) makes a larger price impact, i.e., higher slippage (see Section 3.2.4), in a deep liquidity pool than a thin liquidity pool [148, 206].

3.3.5 Demand Sensitivity. An AMM is demand sensitive when the average swap price (i.e., the effective exchange rate $\frac{x_i}{x_o}$) increases as the swap size (input quantity x_i) increases. Intuitively, this suggests that as with the increment of the demand in output token, its price-denominated input token will be driven up.

A constant product AMM is both liquidity-sensitive and demand-sensitive, whereas, strictly speaking, a constant sum one is neither.

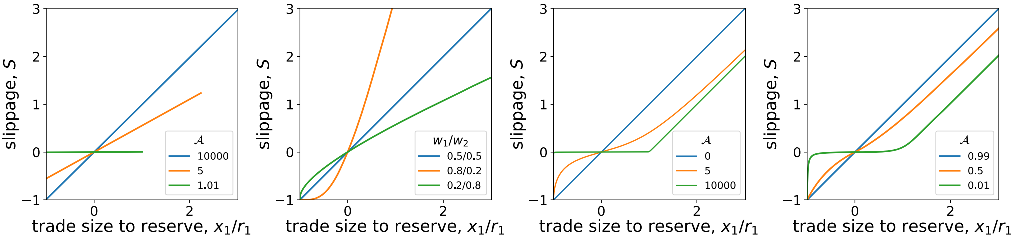
4 COMPARISON OF AMM PROTOCOLS

AMM-based DEXs are home to billions of dollars' worth of on-chain liquidity. Table 2 lists major AMM protocols, their respective value locked, as well as some other general metrics. Uniswap is undeniably the biggest AMM measured by trade volume and the number of governance token holders, although it is remarkable that Curve has more value locked within the protocol. The



(a) Uniswap, Eq. 27 (b) Balancer, Eq. 36 (c) Curve, Eq. 44 (d) DODO, Eq. 49

Fig. 3. Conservation function (see Section 3.2.1) of AMMs with initial reserves of token$_1$ and token$_2$ both equal to 1, namely, $\mathcal{K} = \mathcal{R}_1 = \mathcal{R}_2 = 1$.



(a) Uniswap, Eq. 30 (b) Balancer, Eq. 39 (c) Curve, Eq. 47 (d) DODO, Eq. 52

Fig. 4. Slippage (see Section 3.2.4) of AMMs, depicted with $\frac{x_1}{r_1} \in [-1, 3]$, corresponding to the after-trade token$_1$ reserve quantity $r_1 \in [0, 4]$, which is the x-axis of Figure 3.

number of governance token holders of smaller protocols as Bancor and Balancer is relatively high compared to CRV token holders, as they do approximately a third of the volume but have only slightly fewer governance token holders.

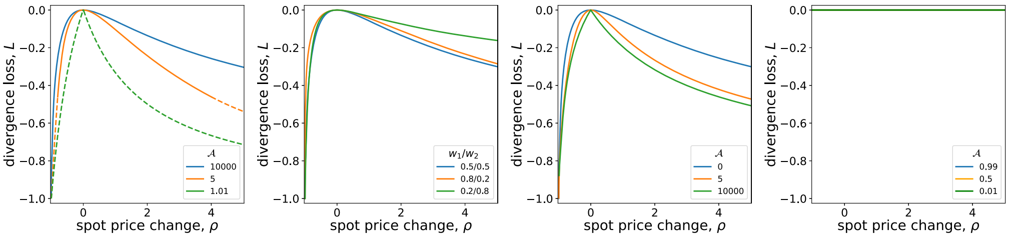
4.1 Major AMM Protocols

This section focuses on the four most representative AMMs: Uniswap (including V2 and V3), Balancer, Curve, and DODO. These protocols were selected based on their market share [97] on the Ethereum blockchain and the representativeness in their overall mechanism.

We describe the liquidity pool structures of those protocols in the main text. We also derive the conservation function, slippage, as well as divergence loss of those protocols. A summary of formulas can be found in Table 4. We refer our readers to Appendix A for a detailed explanation and derivation of those formulas. The protocols' conservation function, slippage, as well as divergence loss under different hyperparameter values are plotted in Figures 3, 4, and 5, respectively. We always use token$_1$ as price or value unit; namely, token$_1$ is the assumed numéraire.

4.1.1 Uniswap V2. The Uniswap protocol prescribes that a liquidity pool always consists of one pair of assets. Uniswap V2 implements a conservation function with a constant-product invariant (see Section A.1.1), implying that the reserves of the two assets in the same pool always have equal value.

Liquidity provision or withdrawal at Uniswap V2 must be balanced and makes no price impact (Section 3.3.1). Swaps with Uniswap V2 are path-deterministic (Section 3.3.2); however, due to the positive swap fee charged and then immediately deposited back into the pool [197], a trading action can be decomposed into asset swap and liquidity provision. This action is, therefore, no longer a *pure* asset swap and would thus move the value of $\mathcal{K}$ [181]. Uniswap V2 carries the properties



(a) Uniswap, Eq. 35 (b) Balancer, Eq. 43 (c) Curve, Eq. 18 (d) DODO, $L \equiv 0$ at equilibrium

Fig. 5. Divergence loss (see Section 3.2.5) of AMMs.

of output-boundedness (Section 3.3.3), liquidity sensitivity (Section 3.3.4), and demand sensitivity (Section 3.3.5).

4.1.2 Uniswap V3. Uniswap V3 enhances Uniswap V2 by allowing liquidity provision to be concentrated on a fraction of the bonding curve [6] (see Section A.2.1), thus virtually amplifying the conservation function invariant and reducing the slippage.

The protocol's slippage controller $\mathcal{A}$ determines the degree of liquidity concentration. Specifically, $\mathcal{A}$ signifies how concentrated the liquidity should be provided around the initial spot price: When $\mathcal{A} \rightarrow \infty$, the covered price range approaches $(0, \infty)$, and the LP's individual conservation function approximates a Uniswap V2 one (approximated with $\mathcal{A} = 10{,}000$ in Figure 3(a)); on the other extreme, when $\mathcal{A} \rightarrow 1$, the liquidity only supports swaps close to the initial exchange rate, and the conservation function approximates a constant-sum one (Figure 3(a)).

Like V2, liquidity provision or withdrawal at Uniswap V3 must respect the existing ratio between two reserve assets, which results in zero price impact (Section 3.3.1). Different from V2 where fees are retained in the pool, Uniswap V3 deducts swap fees as a fraction of the input asset and credit that amount to LP's fee revenue balance; hence, swaps with V3 are path-deterministic with no impact on the bonding curve invariants (Section 3.3.2). By design, Uniswap V3 is not output-bounded (Section 3.3.3)—a sufficiently large swap can deplete one reserve asset and leave the liquidity pool only with the other one in (see Figure 3(a)). Uniswap V3 also features liquidity sensitivity (Section 3.3.4). Thanks to its liquidity-concentrating feature, Uniswap V3 is less demand-sensitive (Section 3.3.5) than V2—a swap with a fixed input quantity would experience a lower slippage at Uniswap V3 than at V2 with the same level of pre-swap reserves (see Figure 4(a)).

4.1.3 Balancer. The Balancer protocol allows each liquidity pool to have more than two assets [129]. Each asset reserve r_k is assigned a weight w_k at pool creation, where $\sum_k w_k = 1$. Weights are pool hyperparameters and do not change with either liquidity provision/withdrawal or asset swap. The weight of an asset reserve represents the value of the reserve as a fraction of the pool value. Balancer can also be deemed a generalization of Uniswap; the latter is a special case of the former with $w_1 = w_2 = \frac{1}{2}$ for asset-pair pools (Figure 3(b)).

Balancer allows both balanced liquidity provision/withdrawal as well as single-asset liquidity change [129]; the former does not cause price impact, whereas the latter does (Section 3.3.1). When only one asset is provided instead of, e.g., eight, the protocol would first execute seven trades to swap this one asset to arrive at a vector of quantities in current proportions and next add this vector to the liquidity pool. Consequently, this sequence of actions is no longer a *pure* liquidity provision/withdrawal and would thus move the asset spot price. As Uniswap V2, swap fees with Balancer are also retained in the pool [20], leading to an update of the bonding curve after each swap (Section 3.3.2). Balancer also features output-boundedness (Section 3.3.3), liquidity sensitivity (Section 3.3.4), and demand sensitivity (Section 3.3.5).

4.1.4 Curve. With the Curve protocol, formerly StableSwap [68], a liquidity pool typically consists of two or more assets with the same peg, for example, `USDC` and `DAI`, or `wBTC` and `renBTC`. Curve approximates Uniswap V2 when its constant-sum component (Section A.4.1) has a near-0 weight, i.e., $\mathcal{A} \rightarrow 0$ (Figure 3(c)).

Like Balancer, Curve allows both proportionate and disproportionate liquidity change to the pool, depending on which the LP's action can induce either zero or some price impact (Section 3.3.1). As Uniswap V2 and Balancer, Curve updates its invariant after each trade to account for swap fees retained in the pool (Section 3.3.2). Curve is also output bounded (Section 3.3.3), liquidity-sensitive (Section 3.3.4) and demand-sensitive (Section 3.3.5).

4.1.5 DODO. DODO supports customized pools [63], where a pool creator provides reserves on both sides of the trading pair with arbitrary quantities, which determines the pool's initial *equilibrium state*. Unlike conventional AMMs such as Uniswap, Balancer and Curve where the exchange rate between two assets in a pool is derived purely from the conservation function, DODO does it the other way around. Resorting to external market data as a major determinant of the exchange rate, DODO has its conservation function (see Section A.5.2) derived from its exchange rate formula (see Section A.5.1).

Specifically, the pool exhibits an arbitrage opportunity—namely, a gap between the price offered by the pool and that from the external market—as soon as the reserve ratio between the two assets in the pool deviates from its *equilibrium state*. Price alignment by arbitrageurs always pulls the reserve ratio back to its equilibrium state set by the LP, thus eliminating any divergence loss. Due to this feature, DODO differentiates itself from other AMMs and terms their pricing algorithm as "proactive market maker," or PMM.

In DODO, a higher slippage controller $\mathcal{A} \in (0, 1)$ results in a greater slippage around the market price—i.e., the equilibrium price. Specifically, when $\mathcal{A} \rightarrow 1$, the DODO bonding curve resembles Uniswap V2 (approximated with $\mathcal{A} = 0.99$ in Figure 3(d)); and with a high slippage around the market price (Figure 4(d)), the pool exhibits a strong tendency to fall back to the equilibrium state. When $\mathcal{A} \rightarrow 0$, the DODO bonding curve resembles a constant sum one (approximated with $\mathcal{A} = 0.01$ in Figure 3(d)); and with a near-flat slippage (Figure 4(d)), the algorithm's force to pull the reserve ratio back to equilibrium is at its weakest due to little arbitrage profitability exhibited.

As with Balancer and Curve, LPs can provide/withdraw both balanced and unbalanced reserves to a DODO pool (Section 3.3.1). Similar to Uniswap V3, swap fees with DODO are recorded outside of the liquidity pool. DODO's bonding curve is thus redrawn not after each swap, but after each price change reported by the oracle (Section 3.3.2). DODO is also output bounded (Section 3.3.3). Although relying on external price feeds for the construction of its conservation function, DODO is still liquidity-sensitive (Section 3.3.4) and demand-sensitivie (Section 3.3.5), since the size of a trade relative to the pool depth determines the magnitude of slippage and the price movement local to the pool.

4.1.6 Other AMM-based DEXs.

Sushiswap. Sushiswap is a fork of Uniswap V2 (see Section 5.1.3). Though the two mainly differ in governance token structure and user experience, Sushiswap shares the same conservation function, slippage, and divergence loss functions as Uniswap.

Kyber Network. Currently in its 3.0 version, the DEX uses a **Dynamic Market Maker (DMM)** mechanism, which allows for dynamic conservation functions based on amplified balances, called "virtual balances" [114]. This is supposed to result in higher capital efficiency for LPs and better slippage for traders. Also, the trading fees are adjusted automatically to market conditions. A volatile market causes increased fees to offset impermanent loss for LPs.

Bancor. While Bancor's white paper [102] gives the impression that a different conservation function is applied, a closer inspection of their transaction history and smart contract leads to the conclusion that Bancor is using the same formula as Balancer (confirmed by a developer in the Bancor Discord community). As the majority of Bancor pools consist of two assets, one of which is usually BNT, with the reserve weights of 50%–50%, Bancor's swap mechanism is equivalent to Uniswap. Bancor V2.1 now allows single-sided asset exposure and provides divergence loss insurance [21] (see Section 4.2.3).

Summary. Each AMM has its quirks. Uniswap V2 implements a rudimentary bonding curve that achieves a low gas fee; Uniswap V3 allows for concentrated liquidity provision that improves capital efficiency; Balancer supports more than two assets in a pool; Curve is suitable for swapping assets with the same peg; DODO proactively reduces divergence loss by leveraging external price feeds. As discussed in Section 4.1, common AMMs can be predominantly seen as a generalization, or an extension, of the most fundamental constant-product protocol that is applied by Uniswap V2.

When hyperparameters such as reserve weights w_k and slippage controller $\mathcal{A}$ are assigned with certain values, various AMMs can be reduced to the basic form equivalent to Uniswap V2 (illustrated with blue curves in Figures 3, 4, and 5). In fact, the majority of top AMMs—including Sushiswap, PancakeSwap, VVS Finance, Quickswap, and BiSwap—are a simple clone of the Uniswap protocol [58] with some adjustment in the fee and reward structure.

When deciding on a new conservation function, AMM developers and designers must consider the tradeoff between different features and properties (Section 3.3). For example, seeking liquidity-insensitivity (Section 3.3.4) and demand-insensitivity (Section 3.3.4) for low slippage leads to higher divergence loss (see Section 2.4.3): Given a range of price movement, traders would be able to swap out an asset with a larger quantity, whereas LPs would suffer a bigger divergence loss. In the extreme case like Uniswap V3, the trader-favoring feature sacrifices the output-bounded property (Section 3.3.3), which is to the detriment of LPs, leaving them completely "rekt"[1] in the case of significant price swings. Seemingly capable of achieving both low slippage and zero divergence loss at equilibrium by setting its $\mathcal{A}$ low, DODO appears to be an exception. Nevertheless, it is to be noted that with a small $\mathcal{A}$, DODO's PMM algorithm is less effective in restoring the pool to its equilibrium state (see Section 4.1.5), thus still exposing LPs to divergence loss risks in non-equilibrated states.

In a similar vein, users interacting with an AMM-based DEX, including both traders and LPs, form a zero-sum game. They should understand the protocol design and beware of embedded hidden costs such as slippage and divergence loss, which impose economic risks on their funds.

4.2 Additional Features of AMM-based DEXs

4.2.1 Time Component. A time component refers to the ability to change traditionally fixed hyperparameters over time. Balancer V1 and V2 implement this (Table 3) by allowing liquidity pool creators to set a scheme that changes the weights of two pool assets over time. This implementation is called a Liquidity Bootstrapping Pool (see Section 4.3.4).

4.2.2 Dynamic Swap Fee. Dynamic fees are introduced by Kyber 3.0 to reduce the impact of divergence loss for LPs. The idea is to increase swap fees in high-volume markets and reduce

[1] DeFi jargon for "wrecked," in this context, meaning exposed to a single, undiversified asset has depreciated in value.

Table 3. Overview of Major Existing AMM-based DEX on Ethereum, Solana, Polkadot, Tezos, EOS, Polygon, and BNB Chain

			AMM			AMM add-ons				
DEX		**Pool structure**	**CP**	**CS**	**OP**	**CC**	**T**	**Divergence loss compensation**	**Chain**	**Mainnet launch**
Uniswap V1	[4]	asset-pair	●	○	○	○	○	–	Ethereum	11/2018
Uniswap V2	[5]	asset-pair	●	○	○	○	○	–	Ethereum	05/2020
Uniswap V3	[6]	asset-pair	●	○	○	●	○	–	Ethereum	05/2021
Balancer V1	[129]	multi-asset	●	○	○	○	●	–	Ethereum	03/2020
Balancer V2	[128]	multi-asset	●	○	○	○	●	–	Ethereum	–
Curve	[68]	multi-asset	●	●	○	○	○	–	Ethereum	01/2020
DODO	[64]	various	●	○	●	○	○	–	Ethereum, BNB Chain	09/2020
Bancor V1	[102]	asset-pair	●	○	○	○	○	–	Ethereum, EOS	06/2017
Bancor V2	[21]	asset-pair	●	○	●	○	○	–	Ethereum, EOS	04/2020
Bancor V2.1	[22]	asset-pair	●	○	○	○	○	Divergence loss insurance	Ethereum, EOS	10/2020
SushiSwap	[190]	asset-pair	●	○	○	○	○	–	Ethereum	08/2020
Mooniswap	[35]	asset-pair	●	○	○	○	●	–	Ethereum	08/2020
mStable	[10]	asset-pair	○	●	○	○	○	–	Ethereum	07/2020
Kyber 3.0	[115]	multi-asset	●	○	○	●	○	Dynamic swap fee	Ethereum, Tezos	03/2021
Saber	[176]	multi-asset	●	●	○	○	○	–	Solana	06/2021
HydraDX	[106]	multi-asset	●	○	●	●	○	–	Polkadot	–
Uranium Finance	[199]	asset-pair	●	○	○	○	○	–	BNB Chain	05/2021
QuickSwap	[165]	asset-pair	●	○	○	○	○	–	Polygon	10/2020
Burgerswap	[38]	asset-pair	●	○	○	○	○	–	BNB Chain	10/2020

CP: Constant product, CS: Constant sum, OP: Oracle price component, CC: Capital concentration, T: Time component.

them in low-volume markets. This should result in more protection against divergence loss, as during periods of sharp token price movements during a high-volume market, LPs absorb more fees. In low-volume and -volatility markets, trading is encouraged by lowering the fees.

4.2.3 Divergence Loss Insurance. Popularized by Bancor V2.1, LPs are insured against divergence loss after 100 days in the pool, with a 30-day cliff at the beginning. Bancor achieves this by using an elastic BNT supply that allows the protocol to co-invest in pools and pay for the cost of impermanent loss with swap fees from its co-investments [23]. This insurance policy is earned over time, 1% each day that liquidity is staked in the pool.

4.3 Other DeFi Protocols with AMM Implementations

AMMs form the basis of other DeFi applications (see Figure 1) that implement existing or invent newly designed bonding curves, facilitating the functionalities of these implementing protocols. In this section, we present a few examples of projects that use AMM designs under the hood.

4.3.1 Gyroscope. Gyroscope [95] is a stablecoin backed by a reserve portfolio that tries to diversify DeFi tail risks. Gyro Dollars can be minted for a price near $1 and can be redeemed for around $1 in reserve assets, as determined through a new AMM design that balances risk in the system. Gyroscope includes a **Primary-market AMM (P-AMM)**, through which Gyro Dollars are minted and redeemed, and a **Secondary-market AMM (S-AMM)** for Gyro Dollar trading. Similar to Uniswap V3 where a price range constraint is imposed, the P-AMM yields a mint quote and a redeem quote that serves as a price range constraint for the S-AMM to decide upon concentrated liquidity ranges [94].

4.3.2 EulerBeats. EulerBeats [75] is a protocol that issues limited edition sets of algorithmically generated art and music, based on the Euler number and Euler totient function. The project uses self-designed bonding curves to calculate burn prices of music/art prints, depending on the existing supply. The project thus implements a form of AMM to mint and burn NFTs price-efficiently.

4.3.3 Pods Finance. Pods [158] is a decentralized non-custodial options protocol that allows users to create calls and or puts and trade them in the Options AMM. Users can participate as sellers and buy puts and calls in a liquidity pool or act as LPs in such a pool. The specific AMM is one-sided and built to facilitate an initially illiquid options market and price option algorithmically using the Black-Scholes pricing model. Users can effectively earn fees by providing liquidity, even if the options are out-of-the-money, reducing the cost of hedging with options.

4.3.4 Balancer Liquidity Bootstrapping Pool (LBP). LBP are pools where controllers can change the parameters of the pool in controlled ways, unlike immutable pools described in Section 4. The idea of an LBP is to launch a token fairly by setting up a two-token pool with a project token and a collateral token. The weights are initially set heavily in favor of the project token, then gradually "flip" to favor the collateral coin by the end of the sale. The sale can be calibrated to keep the price more or less steady (maximizing revenue) or declining to the desired minimum (e.g., the initial offering price) [19].

4.3.5 YieldSpace. The YieldSpace paper [140] introduces an automated liquidity provision for fixed yield tokens. A formula called the "constant power sum invariant" incorporates time to maturity as input and ensures that the liquidity provision offers a constant interest rate—rather than price—for a given ratio of its reserves. `fyTokens` are synthetic tokens that are redeemable for a target asset after a fixed maturity date [173]. The price of a `fyToken` floats freely before maturity, and that price implies a particular interest rate for borrowing or lending that asset until the `fyToken`'s maturity. Standard AMM protocols as discussed in Section 4 are capital-inefficient. By introducing the concept of a constant power sum formula, the writers want to build a liquidity provision formula that works in "yield space" instead of "price space."

4.3.6 Notional Finance. Notional Finance [143] is a protocol that facilitates fixed-rate, fixed-term crypto-asset lending and borrowing. Fixed interest rates provide certainty and minimize risk for market participants, making this an attractive protocol among volatile asset prices and yields in DeFi. Each liquidity pool in Notional refers to a maturity, holding `fCash` tokens attached to that date. For example, `fDai` tokens represent a fixed amount of `DAI` at a specific future date. The shape of the Notional AMM follows a logit curve to prevent high slippage in normal trading conditions. Three variables parameterize the AMM: the scalar, the anchor, and the liquidity fee [142]; the first and second mentioned allowing for variation in the steepness of the curve and its position in a xy-plane, respectively. By converting the scalar and liquidity fee to a function of time to maturity, fees are not increasingly punitive when approaching maturity.

4.3.7 Gnosis Custom Market Maker (CMM). The Gnosis CMM [86] allows users to set multiple limit orders at custom price brackets and passively provide liquidity on the Gnosis Protocol. The mechanism used is similar to the Uniswap V3 structure, although it allows for even more possibilities to market makers by allowing them to choose price upper and lower limits and a number of brackets within that price range. Uniswap V3 allows LPs to solely choose the upper and lower limits. Because users deposit funds to the assets at different price levels specifically, the protocol behaves more like a central limit order book than an AMM pool.

4.3.8 DEX Aggregators. DEX aggregators are a type of emerging DeFi protocols that connect to various other DEXs and can also have their own liquidity pools [200]. They offer traders superior swap rates through routing across liquidity pools from different DEXs with one single user interface [166]. 1inch and Paraswap are two major DEXs aggregators for Ethereum Virtual Machine (EVM)-compatible chains that incorporate AMMs such as Uniswap and Curve. Rango [169] is an example of cross-chain DEXs aggregators incorporate AMMs, DEX aggregators, and bridges to facilitate token swaps across both EVM and non-EVM blockchains.

4.4 AMMs on Layer 2 Solutions

The growth and success of DeFi on Ethereum have put a strain on the Ethereum network's ability to process transactions, leading to increasing gas [218]. As the Ethereum Network becomes busier, user experience decreases because of increasing gas prices and decreasing transaction speed. Users aim to outbid each other by increasing the gas prices. Also, transaction speed decreases, which results in poor user experience for certain types of decentralized applications (dApps). And as the network gets busier, gas prices increase as transaction senders aim to outbid each other. In an attempt to prevent these consequences, **layer 2 solutions (L2)** are being developed. These solutions handle transactions outside of the Ethereum network, but still rely on the decentralized security model of the mainnet [72]. Examples of layer 2 technologies include Plasma, Sidechains, Optimistic Rollups, and ZK-Rollups. For a more comprehensive reading on this topic, we direct the reader to Reference [73]. Examples of Ethereum layer 2 solutions are Polygon [159], Arbitrum [17], Optimism [146], and Starknet [187].

Characteristics of layer 2 solutions, such as scaling and security, have been well-documented across different sources [92, 96, 108] and are out of scope for this SoK. The advantages and disadvantages of transacting on layer 2 solutions are not specifically related to AMMs, but to all protocols on these technologies. Therefore, we focus on the security issues and user experience of interacting with AMMs on layer 2.

Daian et al. [54] note that abstraction achieved by layer 2 exchange systems is not sufficient to prevent sandwich attacks, and a report of Delphi Digital [60] concludes that there is still frontrunning risk when a protocol wants to aggregate liquidity across layer 1 and layer 2 pools. The frontrunning and sandwich attacks do not seem to be solved by layer 2 solutions. Reference [112] proposes a simple solution for frontrunning attacks on Optimistic Rollups technology.

One of the most important advantages of deploying a protocol on layer 2 is the reduction in gas fees. This dramatically enhances the user experience and opens up ways to introduce new forms of decentralized exchanges. One example is ZKSwap [229], allowing users to trade with zero gas fees. Sushiswap and Curve have deployed their contracts on Polygon, and QuickSwap is a fork of Uniswap on that same layer 2 solution [138]. As a result, users are now able to use the same products on layer 2 solutions, which drastically reduces costs and allows faster transactions. In 2021, dYdX launched its order book-based DEXs on StarkEx [66].

In sum, AMMs on layer 2 solutions result in faster transactions and a reduction of costs due to zero or decreased gas costs, ultimately enhancing the user experience.

5 SECURITY AND PRIVACY CONCERNS

The previous sections focus on implicit economic costs—including slippage and divergence loss—imposed on the funds of users interacting with AMM-based DEXs. Besides those risks, security and privacy matters are also to be taken into account when using AMM-based DEXs.

In particular, as a complex, distributed system with a variety of software and hardware components interacting with each other, AMM-based DEXs are prone to exhibit attack interfaces [116, 117, 130, 223]. With conventional exchanges, the success of market manipulation is uncertain, as each trade must be agreed upon between the sell and buy sides. In contrast, AMM-based DEXs are subject to atomic, risk-free exploits on the protocol's technical structure, such as its algorithmic pricing scheme [177]. Built on top of public blockchain infrastructures featuring transparency and traceability, AMM-based DEXs also expose their users to privacy risks.

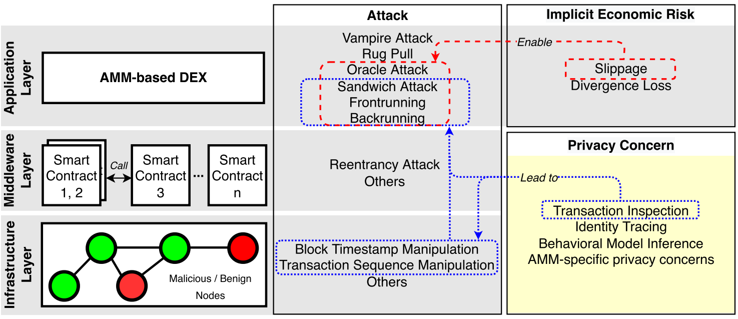


Fig. 6. Architectural layers of an AMM-based DEX with its implicit economic risks, attacks, privacy concerns, and their relationships.

In this section, we define a taxonomy (illustrated in Figure 6) to enumerate potential security and privacy concerns of AMM-based DEXs, expounding their root causes and possible mitigation solutions.

5.1 Associated Attacks

We identify three classes of attacks according to the architectural layer on which they occur: infrastructure-layer attacks, middleware-layer attacks, and application-layer attacks. Sometimes, a certain attack (e.g., frontrunning) can target multiple layers simultaneously. We present known historical attacks affecting AMMs in Table 6.

5.1.1 Infrastructure-layer Attacks. The proper operation of DEXs are based upon healthy and stable blockchain infrastructures (i.e., validators, network, full nodes, etc.). However, since the birth of the blockchain systems, various attacks have threatened their normal operations, potentially affecting the robustness and user experience of DEXs.

Block timestamp manipulation. A timestamp field is set by miners during the validation process. However, malicious miners can manipulate the block timestamps within constraints to win rewards from certain smart contracts [52] or to tamper with the execution order of DEX transactions packed in different blocks [104].

To mitigate the negative impact of such manipulation, DEX contracts should be timestamp-independent [15]. For example, smart contract engineers should avoid using block timestamps as program inputs or make sure a contract function can tolerate variations by a certain time period (e.g., 15 seconds [137]) and still maintain integrity [132]. Besides, DEXs should choose to be built on a blockchain that applies rigorous constraints to the timestamps of committed blocks or uses external timestamp authorities to assert a block creation time [191].

Transaction sequence manipulation. While transactions within a block share the same timestamp, miners can order transactions and choose to include or exclude certain transactions at their discretion. Malicious miners can abuse their "power" to prioritize transactions in their favor, profiting from the miner extractable value (MEV), which is the value that is extractable by miners directly from smart contracts during the validation or mining process [163, 226]. This can be further facilitated by open-source software such as Flashbots [54].

To prevent transaction sequence manipulation, DEXs should first be built upon reputable, frequently used blockchain systems, as they feature high miner/validator participation, making transaction sequence manipulation difficult. Besides, this attack can be mitigated through an enforced transaction sequencing rule that relies on a trusted third party to assign sequential numbers to transactions [70]. We also discuss how DEXs and their transactions can practice transaction sequencing from application-layer in Section 5.1.3, and how privacy-preserving blockchain and DEXs are resistant to this attack in Section 5.2.

Other infrastructure-layer attacks. Aiming to perturb operations of blockchain systems [175], many other attacks do not target AMM-based DEX specifically, but can indirectly affect the service of DEX. For example, attackers can launch spam or distributed denial-of-service (DDoS) attacks towards the blockchain system [91, 152], thereby increasing the latency or even hindering the accessibility of DEX services; blockchain denial-of-service (BDoS) attacks exploit the reward mechanism to discourage miner participation, thereby causing a blockchain to a halt with significantly fewer resources [135]; the 51% attack [175], the most classic blockchain attack, is able to tamper with the blockchain in any way by controlling more than 50% of the network's mining hash rate; network attacks can destroy the network connections between the users and the blockchain system through domain name server (DNS) hijacking [168] or border gateway protocol (BGP) hijacking [16].

In short, AMM-based DEXs should be built upon distributed ledgers with active service, community maintenance, and upgrades, as well as modular security designs. Only by ensuring each module of the blockchain system and the interactions between them are secure can the relative security of the entire blockchain system be ensured.

5.1.2 Middleware-layer Attacks. An AMM-based DEX usually consists of various smart contracts, in which each serves as a middleware that bridges some application-layer functions with blockchain infrastructures and collectively support operations of DEXs. However, the smart contracts and complex collaborations between them can also lead to potential system vulnerabilities [195]. Attackers can exploit such attack interfaces to steal tokens from a DEX or even paralyze it.

Reentrancy attack. Reentrancy attack can happen when two or more entities (e.g., smart contract, side-chain) call or execute certain functions in specific sequences or frequencies. Ever since July 2016, reentrancy attacks have captured the attention of the crypto industry, when the Decentralized Autonomous Organization (DAO), an Ethereum smart contract, was executed maliciously with such an attack, causing a $50 million economic loss in tokens [160]. Afterwards, despite the emergence of various proposals addressing the reentrancy problems [7, 122, 195], reentrancy attacks persist and became particularly threatening to AMM-based DEXs. In January 2019, an audit identified a reentrancy vulnerability in Uniswap [48], which was then exploited by hackers to steal $25 million worth of tokens in April 2020 [149]. Hackers performed this attack by leveraging a subtle interaction between two contracts that were secure in isolation and a third malicious contract [42]. In March 2021, $3.8 million worth of tokens were stolen from DODO through a series of attacks, which began with reentrancies via the `init()` function in a liquidity pool smart contract, followed by frontrunning and honeypot attacks [62].

The security community has proposed a variety of approaches to tackle reentrancy attacks [105]. For example, Rodler et al. [174] protect existing smart contracts on Ethereum in a backwards-compatible way based on runtime monitoring and validation; Das et al. [56] propose Nomos, a reentrancy-aware language that enforces security using resource-aware session types; Cecchetti et al. [42] formalize a general definition of reentrancy and leverage information flow control to solve this problem in general. However, with the increasing complexity of AMM, it can become more difficult for developers to reason about the reentrant interface, thus making reentrancy attacks a more intractable problem for AMM-based DEXs.

Other middleware-layer attacks. On the middleware layer, there are many other attacks and threats that can affect normal operations of smart contracts [179], such as replay attack [167], exception mishandling [161], integer underflow/overflow attacks [189], and so on. These threats are not specifically targeted at DEXs but can potentially be harmful to DEX operation. The security community has proposed a variety of approaches to secure smart contract from these threats, such as Smartshield [224], Zether [36], and NeuCheck [121]. Users may also purchase insurance cover to hedge smart contract risks [49]. To fundamentally counter those attacks, smart contract coders must strictly abide by software development specifications and conduct thorough security tests.

5.1.3 *Application-layer Attacks.*

Oracle attack. A flash loan is a feature provided by lending platforms where an uncollateralized borrow position can be created as long as the borrowed funds can be repaid within one transaction [214]. Flash loans can be used to repay at discount debts that are liquidable without having to acquire borrowed assets in the first place. In this kind of attack, adversaries manipulate lending platforms that use a DEX as their sole price oracle (see Figure 1).

Following Algorithm 1, an attacker profits with Δ_3 token_A less any transaction fees incurred. Utilizing continuous slippage native to an AMM-based DEX (see Section 3.2.4), the attack temporarily distorts the price of token_A relative to token_B. After the prices are arbitraged back, the attack would leave the loan taken from step 3 undercollateralized, jeopardizing the safety of lenders' funds on the lending platform. Examples of such attacks are exploits on Harvest finance [100], Value DeFi [150], and Cheese bank [157].

This broken design can generally be fixed by either providing time-weighted price feeds or using external decentralized oracles. The first solution ensures that a price feed cannot be manipulated within the same block, while the second solution aggregates price data from multiple independent data providers that add a layer of security behind the aggregation algorithm, making sure that prices are not easily manipulated [186].

Rug pull. A rug pull involves the abandonment of a project by the project foundation after collecting investor's funds [211]. One way of doing this is to lure people into buying the coin with no value through a DEX, subsequently swapping this coin for `ETH` or another cryptocurrency with value, as shown in Algorithm 2. DEXs allow users to deploy markets without audit and for free (barring the gas costs), which makes them an excellent target to scam investors. One method is to create a coin with the same name as an existing one. This attracts a lot of attention, since everyone wants to pick up the coin at the lowest price possible. The coin is being bought up, and the original LP swaps his fake coin for `ETH`. In other cases, the creators of the scam token reach out to several prominent people, creating false hype. Once potential buyers see that major players have purchased the token, they start buying themselves, before realizing that the token cannot be swapped back for `ETH`. Sometimes, the attackers let people trade the coin back for `ETH`, but only for a short period, since they are running the risk of losing money. Reference [211] research data on scam tokens on Uniswap and confirm that rug pulls commonly find their victims through DEXs.

In August 2020, a rug puller extracted three `ETH` by imitating the well-known `AMPL` token [9] with a scam token `TMPL`. The token's transaction history shows the provision of 150 `ETH` and `TMPL` tokens to a Uniswap V2 pool by the attacker, who removed 153.81 `ETH` only 35 minutes later [74].

To protect themselves from being rugged, investors should exercise caution and always confirm a project's credibility before investing in its IDO [39, 193]. Usually, reputable IDOs feature high liquidity and a pool lock [3] that disables withdrawal for a fixed period, so LPs are unable to quickly empty the pool once it has absorbed a sufficient amount of valuable assets from investors [136].

Frontrunning. Frontrunning is often enabled through access to privileged market information about upcoming transactions and trades [70]. Since all transactions are visible for a very short period of time before being committed to a block, it is possible for a user to observe and react to a transaction while it is still in the mempool. Those who place their trade immediately before someone else's are called frontrunners [54, 70]. Frontrunners attempt to get the best price of a new coin before selling them onto the market. They can buy up a great portion of the supply of a new token to create exorbitant prices. Due to the hype, this does not stop retail traders from further buying. The frontrunner, who is the seller with the most significant supply, can swap the purchased token for popular coins (e.g., ETH) paid by retail traders. For example, a considerable amount of IDOs on Polkastarter are frontran on Uniswap [192].

Frontrunning can also be achieved through transaction sequence manipulation (see Section 5.1.1) and by exploiting the general mining mechanism. Most mining software, including the vanilla **Go-Ethereum (geth)**, the most popular command-line interface for running Ethereum node, sorts transactions based on their gas price and nonce [54, 70, 228]. This feature can be exploited by malicious users of DEX who broadcast a transaction with a higher gas price than the target one to distort transaction ordering and thus achieve frontrunning.

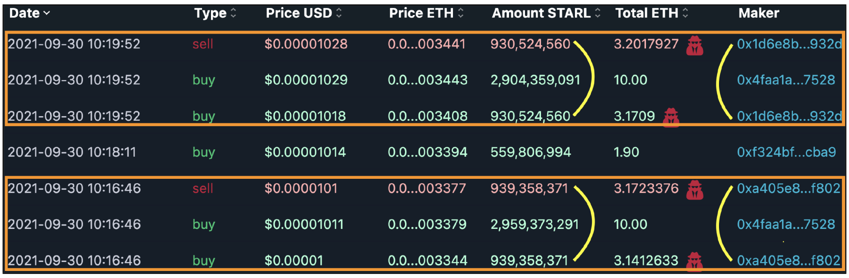
Frontrunning can be avoided with various approaches [26]. Normal exchange users can set a low slippage tolerance to avoid suffering from a price elevated by frontrunners. However, an overly low slippage tolerance may lead a transaction to fail, especially when the trade size is large, resulting in a waste of gas fee [59]. DEXs can enforce transaction sequencing to fundamentally solve frontrunning. Some exchanges, such as EtherDelta [137] and 0xProject [207], utilize centralized time-sensitive functionalities in off-chain order books [70]. In addition, transactions can specify the sequence by including the current state of the contract as the only state to execute on [70], thereby preventing some types of frontrunning attacks. Frontrunning can further be tackled by addressing privacy issues (see Section 5.2) and transaction sequence manipulation on the infrastructure layer (see Section 5.1.1).

Backrunning. Backrunners place their trade immediately after someone else's trade. The attacker needs to fill up the block with a large number of cheap gas transactions to definitively follow the target's transaction. Compared to frontrunning, which only requires a single high valued transaction and is detrimental to the user being frontrun, backrunning is disastrous to the whole network by hindering the throughput with useless transactions [119].

Backrunning attacks can be mitigated through anti-BDoS solutions such as A^{2}MM [227]. Theoretically, defense solutions of application-layer distributed denial-of-service (L7 DDoS) attacks [78, 203, 212] can also be adopted to tackle backrunning problems, as they all aim to secure usability of web services to legitimate users. In addition, backrunning can be addressed by confidentiality-enhancing solutions that hide the content of a transaction before it is committed to a block (see Section 5.2).

Sandwich attacks. Combining frontrunning and backrunning, an adversary of a sandwich attack places his orders immediately before and after the victim's trade transaction. The attacker uses frontrunning to cause victim losses and then uses backrunning to pocket benefits. While there are endless examples of sandwich attacks, Zhou et al. [228] detail two types that can occur on an AMM: an LP attacking an exchange user (see Algorithm 3) by exploiting AMM's liquidity-sensitive property (Section 3.3.4) and one exchange user attacking another (see Algorithm 4) by taking advantage of AMM's demand-sensitive property (Section 3.3.5). The latter is particularly common. Figure 7 shows two examples of such attacks on Uniswap V2 within a time frame of 3 minutes.

Considering swap fee (see Section 2.4.2), gas fee (see Section 2.4.2), and slippage (see Section 2.4.3), sandwich attacks are only profitable if the size of the target trade exceeds a certain



Date	Type	Price USD	Price ETH	Amount STARL	Total ETH	Maker
2021-09-30 10:19:52	sell	$0.00001028	0.0...003441	930,524,560	3.2017927	0x1d6e8b...932d
2021-09-30 10:19:52	buy	$0.00001029	0.0...003443	2,904,359,091	10.00	0x4faa1a...7528
2021-09-30 10:19:52	buy	$0.00001018	0.0...003408	930,524,560	3.1709	0x1d6e8b...932d
2021-09-30 10:18:11	buy	$0.00001014	0.0...003394	559,806,994	1.90	0xf324bf...cba9
2021-09-30 10:16:46	sell	$0.0000101	0.0...003377	939,358,371	3.1723376	0xa405e8...f802
2021-09-30 10:16:46	buy	$0.00001011	0.0...003379	2,959,373,291	10.00	0x4faa1a...7528
2021-09-30 10:16:46	buy	$0.00001	0.0...003344	939,358,371	3.1412633	0xa405e8...f802

Fig. 7. Two sandwich price attacks (see Attack Algorithm 4), marked with orange □ on 30/09/2021, with the STARL/ETH pair on Uniswap V2. The attack conducted at 10:16:46 by 0xa405e8...f802 resulted in a profit of 0.0310743 ETH; the attack conducted at 10:19:52 by 0x1d6e8b...932d resulted in a profit of 0.0308927 ETH.

threshold, a value that depends on the DEX's design and the pool size. DEXs can thus prevent sandwich attacks by disallowing transactions above the threshold [227]. Naturally, sandwich attacks can also be curbed by deterring either frontrunning (see Section 5.1.3) or backrunning (see Section 5.1.3).

Vampire attack. A vampire attack targets an AMM by creating a more attractive incentive scheme for LPs, thereby siphoning out liquidity from the target AMM [107] to the detriment of the protocol foundation (see Section 2.1.3). In September 2020, Sushiswap gained $830 million of liquidity through a vampire attack [55], where Sushiswap users were incentivized to provide Uniswap LP tokens into the Sushiswap protocol for rewards in SUSHI tokens [190]. A migration of liquidity from Uniswap to protocol Sushiswap was executed by a smart contract that took the Uniswap LP tokens deposited in Sushiswap, redeeming them for liquidity on Uniswap, which was then transferred to Sushiswap and converted to Sushiswap LP tokens.

Legal approaches such as applying a restrictive license to the protocol code base—as done later by Uniswap with its V3 new release [80]—can be employed to hinder vampire attacks.

5.2 Privacy Concerns

Most blockchain systems are open, traceable, and transparent, which can raise severe privacy concerns to DEXs that built upon them. Besides, AMM protocols will reveal real-time DEXs information to the public, bringing additional privacy concerns to AMM-based DEXs. In this section, we introduce the privacy issues that users may face in using AMM-based DEX and discuss their possible solutions.

5.2.1 Transaction Inspection. The transparency and openness of public blockchains, where most AMM-based DEXs are built, allow transactions to be observable to everyone. However, this characteristic enables malicious parties to inspect transactions, thereby seeking profits or even disrupting the market [70]. The inspection activities can occur before or at the moment that the transaction is committed to the blockchain by miners, validators, or even any third parties. For example, the aforementioned frontrunning, backrunning, sandwich attacks, transaction sequence manipulation, and block timestamp manipulation in Section 5.1 are all based on transaction inspections [87]. In fact, major AMM-based DEXs are fraught with bots, constantly monitoring transactions for possible profit opportunities [137].

5.2.2 Identity Tracing. Although most blockchains and AMM-based DEXs usually feature a certain degree of anonymity, the linkabilities of transactions and accounts over time still enable attackers to dig identities information of users [223]. Actually, with some off-chain information (e.g., social network posts, public speak, location [65]), eavesdroppers can launch de-anonymization inference attacks to bridge virtual accounts with real-world individuals or uncover the true identities of traders by linking the transactions of an account together and matching relevant information [137].

5.2.3 Behavioral Model Inference. By collecting data from corresponding blocks and analyzing historical transactions of an account, any third parties infer an account's behavioral model, understanding its active phase, trading frequency, or even preferences. Such activity is called behavioral model inference, which not only compromises user privacy, but can also make preparations for launching honeypot attacks [163] and phishing scams [44, 156, 210, 211].

5.2.4 AMM-specific Privacy Concerns. While it is possible to enable privacy-preservance with non-AMM-based DEXs [27, 89] by hiding all the information such as the transaction, order book, and trading volume, it is challenging to make AMM-based DEXs fully privacy-preserving. Due to their path-deterministic property (see Section 3.3.2), information on swaps with an AMM pool is often reverse-engineerable [14]. Some researchers even argue that complete privacy is impossible once DEXs apply ordinary implementations of CFMMs (e.g., Uniswap, Balancer, Curve) under reasonable adversarial models [13]. The transparency and openness allow any third party to capture rich data about the AMM-based DEXs, such as the overall trading situation per block, asset pool changes over time, or popular tokens on the platform. Under this circumstance, even though the DEXs can hide the detailed information about a transaction, we can still estimate the transaction amount and transaction currencies according to the AMM protocols.

Solution. The confidentiality of AMM-based DEX's pipeline can be enhanced from various angles to limit public access to certain information. Such information includes but is not limited to the transaction amount, asset type, user/pool balance, user identification, order of transactions, MEVs, or protocol-related information. Hiding all the information from both the public and computation parties can fundamentally resolve the privacy concerns, thereby eliminating all the associated attacks and privacy disclosure. However, it may break the market visibility to traders, cause difficulties to governance and regulation, and make the whole system inefficient to operate. However, achieving partial confidentiality may be sufficient to prevent many attacks and provide enough privacy protections to traders. Thus, most existing solutions focus on enhancing the user privacy of certain components in AMM-based DEXs.

On the infrastructure layer, many privacy-preserving blockchain solutions have been proposed to increase confidentiality [30, 37] and anonymity [133, 141, 178] for transactions. These solutions can be based on zero-knowledge proof (ZKP) [87], homomorphic encryption [84], or identity obfuscation [88], aiming to break the linkability of transactions, encrypt transaction content, or anonymize users' accounts. AMM-based DEX built upon these blockchains not only can protect user privacy, but also can defend against attacks discussed in Sections 5.1.1 and 5.1.3. Besides, developers can choose to leverage proposer/builder separation (PBS) [201] to hide transaction details from miners or validators. Although PBS can only achieve miner/validator-specific privacy, it can increase the censorship resistance of transactions against miners and validators, thereby defending against frontrunning attacks effectively and efficiently. Ferveo [28] is a similar approach, which is a fast protocol for Mempool Privacy to avoid transaction censorship.

Even though the blockchain infrastructures are transparent, confidentiality can be achieved through privacy designs on upper layers. On the middleware layer, developers can leverage techniques such as Hawk [113], Ekiden [45], and Submarine Commitments [34] to develop privacy-enhanced smart contracts for AMM-based DEX. On the application layer, privacy-enhanced DEXs are proposed to resolve the privacy concerns. For example, P2DEX [27] harnesses multiparty computation (MPC) [51] to realize efficient privacy-preserving DEX; ZKSwap [229], an AMM-based

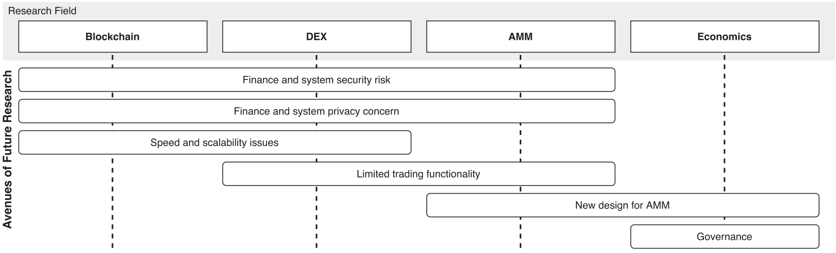

Research Field
Blockchain
DEX
AMM
Economics
Avenues of Future Research
Finance and system security risk
Finance and system privacy concern
Speed and scalability issues
Limited trading functionality
New design for AMM
Governance

Fig. 8. A graphical illustration of future research avenues.

DEX utilizing ZK-Rollup [85] technology, not only provides users with extra privacy protections when withdrawal, deposit, and transfer of tokens by leveraging ZKP, but also significantly reduces gas fees for transactions; ZEXE [32], a ledger-based system, enables users to realize privacy-preserving analogues of popular applications, such as DEX. ZEXE can make it easy to satisfy two main privacy properties. First, transactions hide all information about the offline computations. Second, transactions can be validated in constant time by anyone, regardless of the offline computation; Zswap [69], a zk-SNARKs-based transaction scheme, enables multiple assets and atomic exchanges with sparse homomorphic commitments and Zcash friendly simulation-extractable non-interactive zero-knowledge (NIZK) proofs.

Furthermore, on-chain privacy-preserving services and products are on the rise. For example, Blank [31], a non-custodial Ethereum browser extension wallet, offers transaction obfuscation; Enigma [111] builds a network of "secret nodes" that can perform computations on encrypted data without the necessity to expose original raw data.

However, as mentioned above, even details of an individual transaction will not be disclosed, and it is still possible to infer rough information of transactions through the asset pool state changes and AMM protocols [13]. As the AMM protocols and asset pool state should be accessible to the public, any third party can keep tracking the asset pool changes and then deduce how many assets and which assets were traded during each block. From the perspective of computation parties (i.e., miner), they can fetch even more granular pool state changes to infer detailed transaction information in the mempool. Hence, most existing privacy-preserved DEX designs are hardly compatible with AMM protocols. To overcome this problem, developers can use non-constant function market makers or apply fuzzification to conservation functions. However, this will also bring some side effects to traders, as it can incur larger slippages to the final trading prices. In addition, privacy-enhanced AMM-based DEXs can increase the difficulty of market supervision, creating obstacles for governance, regulation, and financial control. Therefore, how to reach a balance between privacy protection and policy compliance is a question worth exploring for the entire crypto industry.

6 AVENUES OF FUTURE RESEARCH

While DEXs with AMM protocols are maturing, there still exists room for further development, expansion, and exploration. In this section, according to existing literature, our observations about recent trends, and major problems to be solved in this field, we discuss avenues for future research from the perspectives of security, privacy, system design, protocol design, and governance. Figure 8 shows a graphical illustration of future research avenues.

6.1 Finance and System Security Risk

Compared with order-book-based DEX, an AMM-based DEX leverages bonding curves to determine asset prices without reaching any agreements between buyers and sellers. However, this feature can also bring severe security risks and leave many unsolved problems. For example, once attackers successfully manipulate the asset prices in AMM-based DEX, they can theoretically drain all the associated assets from the asset pool, causing severe damage to the trading market. This matter is worthy of deep investigation in the future.

In addition, as a nascent type of trading platform built on complicated distributed systems (i.e., blockchain), AMM-based DEXs operate upon various components run by many different parties. Therefore, any flaws in this system can pose threats to the trading market, potentially causing both economic losses and system dysfunction. As discussed in Section 5, in recent years, AMM-based DEXs suffered from a variety of attacks for different reasons, such as oracle attacks caused by flaws of oracle components in blockchains, vampire attacks caused by the lack of market regulation, flash loan attacks caused by unsound trading rules, and reentrancy attacks caused by inappropriate function calls of smart contracts. Patching those vulnerabilities without bringing significant changes to existing systems requires joint efforts from both computer science, economics, and finance researchers.

6.2 Finance and System Privacy Concern

Privacy is another major concern in AMM-based DEXs (Section 5.2). At present, most DEX systems are built on top of public blockchains such as Ethereum to record transactions in plain text, which enables everyone to observe detailed transactional information. An attacker could inspect an AMM protocol, access the associated transactions from the blockchain, and launch attacks such as frontrunning, backrunning, and sandwich attacks. Even most blockchain systems provide a certain degree of anonymity (i.e., pseudonymous), recent studies have shown that deanonymization attacks can link transactions to users' accounts and reveal users' real identities. Besides, the AMM algorithm itself can leak rich information about the DEX, enabling any third parties to estimate the detailed transaction information.

As discussed in Section 5.2, some general solutions such as Zerocash [178], Hawk [113], and ZEXE [32] have been proposed to achieve ledger privacy in DEX. However, they would substantially decrease the real-time efficiency in AMM-based DEX, thereby being very expensive to deploy in AMM-based DEX. Researchers can try to optimize the system design and the cryptographic algorithm efficiency of those approaches, thus preserving the swap velocity while enhancing user privacy. In addition, existing privacy-preserving blockchains and DEXs are not compatible with AMM protocols. Even though transaction information is protected on blockchains, people can still deduce the transaction's asset type and exchange amount from asset pool transformations. Therefore, a new AMM-oriented DEX that can preserve user privacy at certain degrees is worthy of study. Possible research directions include adding stochasticity to the AMM protocol, developing non-constant function market makers, or applying fuzzification to current AMMs. Moreover, existing privacy solutions could also result in governance issues; since these solutions provide full privacy for transactions, it is almost impossible for law enforcement to investigate cryptocurrency-related crimes in DEX such as theft, money laundering, and illegal transactions in dark markets. Finally, introducing privacy in AMM-based DEX may affect asset prices, asset liquidity, and market predictability across different markets, thus requiring new economic models and protocols to analyze the reward and cost for AMM economics.

6.3 Speed and Scalability Issues

As AMM-based DEXs operate on top of blockchain systems, their transaction speed and scalability are limited by the growth speed and throughput of the blockchain networks.

On the one hand, each DEX transaction takes time to be validated on the blockchain network before it takes effect. The validation speed depends on the miners or validators, rather than the DEX. Compared with centralized exchanges that will immediately finish processing user transactions, DEXs will incur a processing delay ranging from a couple of seconds to several hours. Although some blockchain networks are specifically designed for velocity requirements (e.g., Tezos, XRPL, EOSIO), their delays are still measured in seconds [152].

On the other hand, as the information of DEX transactions must be recorded in blocks, the total number of transactions that a DEX can accept per batch is limited. Although many multi-layer blockchains (e.g., layer two blockchains) were proposed to resolve the throughput issue [182], their throughput is still an order of magnitude behind centralized exchanges.

Therefore, a vital future research direction is to keep improving the validation speed and throughput of blockchain systems, thereby increasing the speed and scalability of the AMM-based DEXs built upon them. Resolving this issue not only can scale AMM-based DEXs to a large group of users, but also can ensure that each transaction can be processed timely. New types of data structure, synchronization mechanism, or validation approaches should be proposed to conquer this problem.

6.4 Limited Trading Functionality

Compared to centralized exchanges nowadays, the trading functionalities of AMM-based DEXs are limited to buying and selling. Due to the nature of AMM-based DEXs, the lack of stop-loss ability, margin trading, and put and call options greatly restrict users' financial operations on the platforms. Besides, it is difficult to implement additional functionalities to existing AMM protocols.

To enhance the trading functionality of AMM-based DEXs, researchers need to bring some structural changes to currently AMM-based DEXs. For example, researchers can try to bridge AMM and order book to provide put and call options; leveraging smart-contract-based lending can also help AMM-based DEXs to provide trading margins to users.

6.5 New Design for AMM

We find that the majority of AMM-based DEX use a CFMM algorithm that can be seen as a variation of Uniswap's constant product protocol. It would be worth exploring novel bonding curves or new AMM designs that can, e.g., balance slippage and impermanent loss in different ways.

Current AMM implementation can be mainly seen with DEX for spot markets. While still at its infancy, DEX for financial derivative markets have also been witnessing an increase in the adoption of AMM algorithms, such as Siren [184] and Hegic [208] for options, the Perpetual Protocol for perpetual contracts [153], and Tracer [83] for swaps. As those protocols are still at their early development phase, they have not been thoroughly tested by the market or scrutinized by academia. As the derivative DEX using AMM becomes more mature, it would be of scholars' and practitioners' interest to see a systematization of these protocols and an investigation on how they advance from the basic AMMs for spot markets.

6.6 Governance

Due to its decentralized and censorship-resistant nature, DEX participants have the liberty to do whatever is permissible by the smart contract code, sowing the seeds for malicious and fraudulent behavior. In such context, governance schemes are essential to ensure the proper operation of AMM-based DEXs. However, centralization of voting power is frequently observable with AMM-based DEX due to the often concentrated nature of protocol token distribution. Furthermore, regulatory concerns often go hand-in-hand with governance issues. For example, the Uniswap community has been debating whether or not to turn the fee switch on for the `UNI` buyback program [198]. In fear of legal consequences in the case of `UNI` being categorized as a security token, the protocol has yet to enable the function due to its similarity to a regular stock buyback exercise, leaving the `UNI` token value unbacked by any monetary flow.

Whether and how the governance mechanism can be improved for more sustainable development of a protocol in a legally compliant way thus merits further research.

7 RELATED WORK

In Table 5, we summarize most SoKs, surveys, and tutorials that investigate the dApp ecosystem on blockchain. Our work differs from those existing works in the following aspects:

(1) we have a clear, focused study subject: DEX-based AMM, while the majority of related survey studies either examines other DeFi applications (e.g., lending protocols [24], yield aggregators [50, 213]) or have a broader coverage;
(2) we use a comprehensive set of methods to generalize and systematize DEX-based AMM protocols, including taxonomization, state space modeling, numerical simulation, and empirical investigation, while most existing related survey studies use only a subset of the aforementioned methods;
(3) we examine an array of aspects of DEX-based AMM, including their architectural design and internal mechanism, financial economics, as well as as associated security and privacy concerns, while most of the related survey papers only cover some of these aspects.

In the following, we discuss more literature related to our study in various ways.

7.1 AMM-based DEX on Blockchain

Our work is first and foremost related to the literature body covering AMM-based DEX on blockchain.

7.1.1 Protocol Mechanism. Angeris et al. [12] discuss arbitrage behavior and price stability in constant product and constant mean markets. Lo et al. [120] empirically evidence that the simplicity of Uniswap ensures the ratio of reserves to match the trading pair price. Despite historical oracle attacks associated with AMMs (see Section 5), Angeris et al. [11, 12] show that CFMM users are incentivized to correctly report the price of an asset, suggesting the suitability for those AMMs to act as a decentralized price oracle for other DeFi protocols. Angeris et al. [14] present a method for constructing CFMM whose portfolio value functions match an arbitrary payoff. Richardson et al. [172] detail the mechanism of the Bancor AMM and its potential implementation in carbon trading markets.

7.1.2 Security. Qin et al. [162] conduct empirical analyses on various AMM attacks, including transaction (re)ordering and frontrunning, and demonstrate the profitability in performing transaction replay through a simple trading bot. Mitigating solutions for frontrunning attacks in DeFi are surveyed in Baum et al. [26]. Security risk in terms of attack vectors in high-frequency trading on DEXs are discussed in Zhou et al. [228] and Qin et al. [164]. Flash loan attacks with the aid of AMMs on Ethereum are described in Cao et al. [40], Perez et al. [151], and Wang et al. [204]. Victor et al. [202] detect self-trading and wash trading activities on order-book-based DEXs. Gudgeon et al. [93] explore design weaknesses and volatility risks in AMM DEXs.

7.1.3 Privacy. Angeris et al. [13] argue that privacy is impossible with typical CFMMs and propose several mitigating strategies. Baum et al. [26] examine input privacy in the context of AMM. Stone et al. [188] describe a protocol that allows trustless, privacy-preserving cross-chain cryptocurrency transfers but is yet susceptible to vampire attacks.

7.2 DEX and AMM in the Context of Market Microstructure

As two core topics of market microstructure [82], decentralized exchange and market-making have been intensively covered in financial economics long before the emergence of blockchain.

7.2.1 DEX. Existing literature primarily suggests the higher efficiency of DEX markets over centralized ones. Perraudin et al. [154] investigate decentralized forex markets and conclude that DEXs are efficient when different market makers can transact with each other and that decentralized markets are more immune to crashes than centralized ones. Nava [139] analyzes quantity competition in the decentralized oligopolistic market and suggests perfect competition can be approximated in large rather than small DEX markets. Malamud et al. [126] develop an equilibrium model of general DEX and prove that decentralized markets can more efficiently allocate risks to traders with heterogeneous risk appetites than centralized ones.

7.2.2 AMM. The concept of automated market making can be traced back to Hanson's logarithmic market scoring rule (LMSR) [98, 99]. LMSR has since been refined and compared to alternative market-making strategies.

Othman et al. [147] address non-sensibility to liquidity and non-profitability of LMSR market making. They propose a bounded, liquidity-sensitive AMM that runs with a profit by levying transaction cost to subsidize liquidity, a strategy later widely implemented by blockchain-based DEXs with AMM protocols to compensate for divergence loss (see Section 2.4.3) experienced by LPs. Brahma et al. [33] propose a Bayesian market maker for binary markets that exhibit better convergent behavior at equilibrium than LMSR.

Jumadinova et al. [109] compare LMSR with different AMM strategies, including myopically optimizing market-maker, reinforcement learning market maker, and utility-maximizing market maker. Simulating empirical market data, they find that reinforcement-learning-based AMM outperforms other strategies in terms of maintaining low spread while simultaneously obtaining high utilities. Slamka [185] compare LMSR with **dynamic parimutuel market (DPM), dynamic price adjustments (DPA)**, and an AMM by the **Hollywood stock exchange (HSX)** in the context of prediction markets. They show that LMSR and DPA generate the highest forecast accuracy and lowest losses for market operators. Today, LMSR has become the de facto AMM for prediction markets [206] and was adopted by the Ethereum-based betting platform Augur [155].

Wang [206] compares mathematical models for AMMs, including LMSR, liquidity-sensitive LMSR (LS-LMSR) and common CFMMs, and proposes constant circle/ellipse-based cost functions for superior computational efficiency. Capponi et al. [41] analyze the market microstructure of constant-product AMMs and predict that AMMs will be used more for low-volatility tokens.

7.3 State Space Modeling Framework

Foundational concepts of the design approach used in tokenized economic systems, which AMMs are an example of, are presented in the following stream of work: The conceptual engineering framework for modeling, analysis, and design of blockchain-based infrastructure is introduced in Zargham et al. [222]. A formalization of the blockchain as a state machine is presented in Shorish [183]. The extension of this framework to dynamical stochastic games is presented in Zhang [225]. A formal discussion on how the evolution of a dynamical system can be constrained to uphold desired system properties is conducted in Zargham et al. [221] using bonding curves as an example. A theoretical framework on estimation properties of aggregated agent signals into systemic statistics in dynamic economic games is provided in Zargham et al. [220].

8 CONCLUSION

The DeFi ecosystem is a relatively new concept, and innovations within the space are being developed at an incredible speed. As an integral part of that ecosystem, AMM-based DEX are an incredible innovation sprung up by the trustless, verifiable, and censorship-resistant distributed ledger technology.

In this article, we systematize the knowledge around AMM-based DEX and use state-space representation to formalize and generalize the AMM algorithms. We apply our protocol design framework to major exchanges—Uniswap, Balancer, Curve, and DODO—and comment on various other exchanges such as Sushiswap, Kyber Network, and Bancor. We examine the implied economic risks in AMMs, including slippage and divergence loss, and establish a taxonomy covering security and privacy issues associated with AMMs. In particular, AMM-based DEX can be the target of a plethora of infrastructure-, middleware-, and application-layer attacks. Future research into AMM mechanisms can build upon this systematization of knowledge, establish unique ways for differentiating AMM innovations, and expand on our security taxonomy that can help the development of more robust AMM-based DEXs.

APPENDICES

A FORMULAS OF MAJOR AMM-BASED DEX

All formulas provided here adhere to the state space representation introduced in Section 3.1, proving that this generalized framework can be used to present and discuss various AMM protocols. All important properties such as the *conservation function*, *state updates,* and *metrics* can be framed within the defined taxonomy and allow an unified analysis of various specifications. The survey of knowledge presented in this article has been couched in the context introduced in Section 3 to facilitate a qualitative comparison among different protocols without presuming to provide a methodology sufficiently abstract for engineering rigor. This task is on the agenda of researchers in the interdisciplinary space of *Token Engineering,* where methodologies are being developed that allow to perceive such interactions on top of blockchain protocols as *Economic Games* [220] in the context of *Generalized Dynamical Systems.* Those methodologies currently under construction will in the future allow quantitative comparisons within unified frameworks where numerical experiments and system identification techniques support system designers in the construction, design, analysis, and maintenance of *classes of protocols* (such as, e.g., the AMM-protocol-class as one representative from the DeFi space) and where the framework both allows to derive system properties from a given representation or alternatively find a system representation conditioned on desired system requirements.

A.1 Uniswap V2

A.1.1 Conservation Function. The product of reserve quantity of token$_1$, r_1, and reserve quantity of token$_2$, r_2, stays constant with swapping:

$$\mathcal{K} = r_1 \cdot r_2. \tag{19}$$

A.1.2 Spot Exchange Rate. Given the *equal value assumption* encoded in the pool smart contract, the implied spot price of assets in a liquidity pool can be derived based on the ratio between their reserve quantities. Specifically, denominated in token 1, the price of token$_2$ can be expressed as:

$$ {}_1E_2 = \frac{r_1}{r_2}. \tag{20}$$

A.1.3 Swap Amount. Based on the Uniswap conservation function (Equation (27)), the amount of token$_2$ received x_2 (spent when $x_2 < 0$) given amount of token$_1$ spent x_1 (received when $x_1 < 0$)

can be calculated following the steps described in Section 3.2.3:

$$\begin{aligned} r_1' &= r_1 + x_1 \\ r_2' &= \frac{\mathcal{K}}{r_1'} \\ x_2 &= r_2 - r_2'. \end{aligned} \tag{21}$$

A.1.4 Slippage. The slippage that a Uniswap user experiences when swapping x_1 token$_1$ with x_2 token$_2$ can be expressed as:

$$S(x_1) = \frac{x_1/x_2}{{}_1E_2} - 1 = \frac{x_1}{r_1}. \tag{22}$$

Figure 4(a) illustrates the relationship between Uniswap slippage and normalized token$_1$ reserve change $\frac{x_1}{r_1}$.

A.1.5 Divergence Loss. Given the equal value assumption with Uniswap, the reserve value of token 1, V_1, equals exactly half of original value of the entire pool V (token$_1$ being numéraire):

$$\frac{V}{2} = V_1 = V_2 = r_1. \tag{23}$$

Should a LP have held r_1 token$_1$ and r_2 token$_2$, then when token$_2$ appreciates by ρ (depreciates when $\rho < 0$), the total value of the original reserve composition V_{held} becomes:

$$V_{\text{held}} = V + V_2 \cdot \rho = r_1 \cdot (2 + \rho). \tag{24}$$

With r_1 token$_1$ and r_2 token$_2$ locked in a liquidity pool from the beginning, their quantity ratio would have been updated through users' swapping to result in token$_2$'s price change of ρ. The equal value assumption still holds, and the updated pool value V' becomes:

$$\frac{V'}{2} = V_1' = V_2' = r_1' = r_1 \cdot \sqrt{1+\rho}. \tag{25}$$

Note that $r_2' = \frac{r_2}{\sqrt{1+\rho}}$ and $p' = \frac{(1+\rho)r_1}{r_2}$, which preserves the invariance of $\mathcal{K}$ and reflects the change in token$_2$'s spot exchange rate against token$_1$.

As illustrated in Figure 3(a), the divergence loss due to liquidity provision as opposed to holding can thus be expressed as a function of price change:

$$L(\rho) = \frac{V'}{V_{\text{held}}} - 1 = \frac{\sqrt{1+\rho}}{1 + \frac{\rho}{2}} - 1. \tag{26}$$

A.2 Uniswap V3

A.2.1 Conservation Function. The conservation function of a Uniswap V3 pool is an aggregate of all the individual LP's conservation functions, each dependent on the exchange rate range that the LP wants to provide his liquidity for.

Suppose an LP supplies $\mathcal{R}_1$ token$_1$ and $\mathcal{R}_2$ token$_2$, with the restriction that his liquidity is only provided for users swapping within a specific range of exchange rates: $[\frac{\mathcal{R}_1}{\mathcal{R}_2 \cdot \mathcal{A}}, \frac{\mathcal{R}_1 \cdot \mathcal{A}}{\mathcal{R}_2}]$ where $\mathcal{A} > 1$ and the initial exchange rate equals $\frac{\mathcal{R}_1}{\mathcal{R}_2}$.

The shape of the conservation function is then *identical* to liquidity provision of the following amounts under Uniswap V2:

$$r_1^{\text{equiv}} = \frac{\mathcal{R}_1}{1 - \frac{1}{\sqrt{\mathcal{A}}}} \quad \text{and} \quad r_2^{\text{equiv}} = \frac{\mathcal{R}_2}{1 - \frac{1}{\sqrt{\mathcal{A}}}}.$$

The bonding curve of a Uniswap V3 pool is equivalent to that of a Uniswap V2 one moving left along the x-axis by $(r_1^{\text{equiv}} - \mathcal{R}_1)$ and down along the y-axis by $(r_2^{\text{equiv}} - \mathcal{R}_2)$. Thus, Uniswap V3 conservation function can be expressed as:

$$\begin{aligned}[r_1 + (r_1^{\text{equiv}} - \mathcal{R}_1)] \cdot [r_2 + (r_2^{\text{equiv}} - \mathcal{R}_2)] &= r_1^{\text{equiv}} \cdot r_2^{\text{equiv}} \\ \left(r_1 + \frac{\mathcal{R}_1}{\sqrt{\mathcal{A}} - 1}\right) \cdot \left(r_2 + \frac{\mathcal{R}_2}{\sqrt{\mathcal{A}} - 1}\right) &= \frac{\mathcal{A} \cdot \mathcal{R}_1 \cdot \mathcal{R}_2}{(\sqrt{\mathcal{A}} - 1)^2},\end{aligned} \tag{27}$$

where $0 \leq r_1 \leq \mathcal{R}_1 \cdot (\sqrt{A} + 1)$ and $0 \leq r_2 \leq \mathcal{R}_2 \cdot (\sqrt{A} + 1)$.[2]

A.2.2 Exchange Rate.

$$_1E_2 = \frac{r_1 + \frac{\mathcal{R}_1}{\sqrt{\mathcal{A}}-1}}{r_2 + \frac{\mathcal{R}_2}{\sqrt{\mathcal{A}}-1}} \tag{28}$$

Note that when token$_1$ is depleted, i.e., $r_1 = 0$, then

$$\begin{aligned}r_2 + \frac{\mathcal{R}_2}{\sqrt{\mathcal{A}} - 1} &= \frac{\mathcal{A} \cdot \mathcal{R}_2}{\sqrt{\mathcal{A}} - 1} \\ _1E_2 &= \frac{\mathcal{R}_1}{\mathcal{R}_2 \cdot \mathcal{A}}.\end{aligned}$$

Similarly, when token$_2$ is depleted, i.e., $r_2 = 0$, then $_1E_2 = \frac{\mathcal{R}_1 \cdot \mathcal{A}}{\mathcal{R}_2}$. Be reminded that $[\frac{\mathcal{R}_1}{\mathcal{R}_2 \cdot \mathcal{A}}, \frac{\mathcal{R}_1 \cdot \mathcal{A}}{\mathcal{R}_2}]$ is exactly the pre-specified exchange rate range that the liquidity supports.

A.2.3 Swap Amount. The swap amount can be derived from the conservation function Equation (27):

$$\begin{aligned}r_1' &= r_1 + x_1 \\ r_2' &= \frac{\mathcal{R}_1 \mathcal{R}_2}{(1 - \frac{1}{\sqrt{\mathcal{A}}})^2} \bigg/ \left(r_1' + \frac{\mathcal{R}_1}{\sqrt{\mathcal{A}} - 1}\right) - \frac{\mathcal{R}_2}{\sqrt{\mathcal{A}} - 1} \\ x_2 &= r_2 - r_2'.\end{aligned} \tag{29}$$

A.2.4 Slippage. The slippage should have the same magnitude as in Uniswap V2, but with r_1 amplified by an increase of $\frac{\mathcal{R}_1}{\sqrt{\mathcal{A}}-1}$:

$$S(x_1) = \frac{x_1/x_2}{_1E_2} - 1 = \frac{x_1}{r_1 + \frac{\mathcal{R}_1}{\sqrt{\mathcal{A}}-1}}. \tag{30}$$

Again, when $\mathcal{A} \to \infty$, the slippage function approximates a Uniswap V2 one; when $\mathcal{A} \to 1$, slippage is restrained as long as there exists liquidity for both assets (Figure 4(a)).

[2] Equation (27) is equivalent to $(x + \frac{L}{\sqrt{pb}})(y + L\sqrt{pa}) = L^2$, Equation (2.2) from page 2 of the Uniswap V3 whitepaper [6]. This can be seen by equating their notation with ours as follows: $L := \frac{\sqrt{\mathcal{A} \cdot C_1 \cdot C_2}}{\sqrt{\mathcal{A}}-1}, x := r_1, y := r_2, pa := \frac{C_1 \cdot \mathcal{A}}{C_2}, pb := \frac{C_1}{C_2 \cdot \mathcal{A}}$.

A.2.5 Divergence Loss. Using the intermediary results from Section A.1.5, we can easily derive V_{held}, r_1', and r_2', and subsequently V':

$$V_{\text{held}} = C_1 \cdot (2+\rho) \tag{31}$$

$$r_1' = r_1^{\text{equiv}}\sqrt{1+\rho} - \frac{\mathcal{R}_1}{\sqrt{\mathcal{A}}-1} = \frac{\mathcal{R}_1 \cdot (\sqrt{1+\rho} - \frac{1}{\sqrt{\mathcal{A}}})}{1-\frac{1}{\sqrt{\mathcal{A}}}}$$

$$r_2' = \frac{r_2^{\text{equiv}}}{\sqrt{1+\rho}} - \frac{\mathcal{R}_2}{\sqrt{\mathcal{A}}-1} = \frac{\mathcal{R}_2(\frac{1}{\sqrt{1+\rho}} - \frac{1}{\sqrt{\mathcal{A}}})}{1-\frac{1}{\sqrt{\mathcal{A}}}}$$

$$V' = V_1' + V_2' = r_1' + \frac{\mathcal{R}_1(1+\rho)r_2'}{\mathcal{R}_2} = \frac{\mathcal{R}_1(2\sqrt{1+\rho} - \frac{2+\rho}{\sqrt{\mathcal{A}}})}{1-\frac{1}{\sqrt{\mathcal{A}}}}. \tag{32}$$

When $-1 \leq \rho \leq \frac{1}{\mathcal{A}} - 1$, then token$_1$ becomes depleted, and the LP is left with token$_2$:

$$V' = \frac{\mathcal{R}_1(1+\rho)}{\mathcal{R}_2} \cdot r_2' = \mathcal{R}_1 \cdot (1+\rho) \cdot (\sqrt{\mathcal{A}}+1). \tag{33}$$

When $\rho \geq \mathcal{A} - 1$, then token$_2$ becomes depleted, and the LP is left with token$_1$ only:

$$V' = r_1' = \mathcal{R}_1 \cdot (\sqrt{\mathcal{A}}+1). \tag{34}$$

The divergence loss can thus be calculated as:

$$L(\rho) = \frac{V'}{V_{\text{held}}} - 1 = \begin{cases} \frac{(\rho+1)\cdot\sqrt{\mathcal{A}}-1}{2+\rho}, & -1 \leq \rho \leq \frac{1}{\mathcal{A}} - 1 \\ \frac{\frac{\sqrt{1+\rho}}{1+\frac{\rho}{2}}-1}{1-\frac{1}{\sqrt{\mathcal{A}}}}, & \frac{1}{\mathcal{A}} - 1 \leq \rho \leq \mathcal{A} - 1 \cdot \\ \frac{\sqrt{\mathcal{A}}-1-\rho}{2+\rho}, & \rho \geq \mathcal{A} - 1 \end{cases} \tag{35}$$

A.3 Balancer

A.3.1 Conservation Function. Balancer implements a conservation function with a weighted-product invariant (Figure 3(b)). Specifically, the product of reserve quantities each raised to the power of its weight stays constant with swapping:

$$\mathcal{K} = \prod_k r_k^{w_k}. \tag{36}$$

A.3.2 Spot Exchange Rate. Given the quantity ratio $r_1 : r_2$ between token$_1$ and 2 and the implicit assumption on their value ratio $w_1 : w_2$, the price of token$_2$ denominated by token$_1$ can be expressed as:

$${}_1E_2 = \frac{r_1 \cdot w_2}{r_2 \cdot w_1}. \tag{37}$$

A.3.3 Swap Amount. We investigate the case when a user swaps token$_1$ for token$_2$, while the reserves of all other assets remain untouched in the pool. Based on the Balancer conservation function (Equation (36)), the amount of token$_2$ received x_2 (spent when $x_2 < 0$) given amount of token$_1$ spent x_1 (received when $x_1 < 0$) can be calculated following the steps described in

Section 3.2.3:

$$\begin{aligned} r_1' &= r_1 + x_1 \\ r_2' &= r_2 \left(\frac{r_1}{r_1'}\right)^{\frac{w_1}{w_2}} \\ x_2 &= r_2 - r_2'. \end{aligned} \tag{38}$$

A.3.4 Slippage. The slippage that a Balancer user experiences when swapping x_1 token$_1$ with x_2 token$_2$ can be expressed as:

$$S(x_1) = \frac{x_1/x_2}{{}_1E_2} - 1 = \frac{\frac{x_1}{r_1} \cdot \frac{w_1}{w_2}}{1 - \left(\frac{r_1}{r_1'}\right)^{\frac{w_1}{w_2}}} - 1. \tag{39}$$

Figure 4(b) illustrates the relationship between Uniswap slippage and normalized token$_1$ reserve change $\frac{x_1}{r_1}$.

A.3.5 Divergence Loss. Given the constant value ratio assumption with Balancer, the value of the entire pool V can be expressed by the reserve quantity of token$_1$, r_1 divided by its weight w_1 (token$_1$ being numéraire):

$$V = \frac{V_1}{w_1} = \frac{V_2}{w_2} = \frac{V_k}{w_k} = \frac{r_1}{w_1}. \tag{40}$$

If token$_2$ appreciates by ρ (depreciates when $\rho < 0$) while all other tokens' prices remain unchanged, then the total value of the original reserve composition, when held outside of the pool, V_{held} becomes:

$$V_{\text{held}} = V + V_2 \cdot \rho = V \cdot (1 + w_2 \cdot \rho). \tag{41}$$

With r_1 token$_1$ and r_2 token$_2$ locked in a liquidity pool from the beginning, their quantity ratio would have been updated through users' swapping to result in token$_2$'s price change of ρ. The value ratio between the pool, token$_1$ and token$_2$, remains $1 : w_1 : w_2$, and the updated pool value V' becomes:

$$V' = \frac{V_1'}{w_1} = \frac{r_1'}{w_1} = \frac{r_1 \cdot (1+\rho)^{w_2}}{w_1} = V \cdot (1+\rho)^{w_2}. \tag{42}$$

The exchange rate range corresponds to the LP's range requirement. Specifically, when $r_2' = \frac{r_2}{(1+\rho)^{1-w_2}}$ and $r_k' = r_k \cdot (1+\rho)^{w_2}$ for $k \neq 2$, reflecting the assumed scenario that only the value of token$_2$ appreciates by ρ, while the value of all other tokens against token$_1$ remains unchanged.

As illustrated in Figure 5(a), the divergence loss due to liquidity provision as opposed to holding can thus be expressed as a function of price change:

$$L(\rho) = \frac{V'}{V_{\text{held}}} - 1 = \frac{(1+\rho)^{w_2}}{1 + w_2 \cdot \rho} - 1. \tag{43}$$

A.4 Curve

A.4.1 Conservation Function. As assets from the same pool are connected to the same peg by design, the ideal exchange rate between them should always equal 1. Theoretically, this could be achieved by a constant-sum invariant. Nevertheless, Curve seeks to allow an exchange rate to deviate from 1 to reflect the supply-demand dynamic, while simultaneously keeping the slippage low.

Curve achieves this by interpolating between two invariants, constant sum and constant product [68], with hyperparameter $\mathcal{A}$ as the interpolating factor (Equation (44)).[3] When $\mathcal{A} \longrightarrow 0$, the conservation function boils down to a constant-product one, as with Uniswap; when $\mathcal{A} \longrightarrow +\infty$, the conservation function is essentially a constant-sum one with constant exchange rate equal to 1 (Figure 3(c)).

$$\mathcal{A}\left(\frac{\sum_k r_k}{\mathcal{K}} - 1\right) = \frac{\left(\frac{\mathcal{K}}{n}\right)^n}{\prod_k r_k} - 1. \tag{44}$$

A.4.2 Spot Exchange Rate. Rearrange Equation (44) and let

$$Z(r_1, r_2) = \frac{\left(\frac{\mathcal{K}}{n}\right)^n}{r_1 r_2 \prod_{k \neq 1,2} r_k} - 1 - \mathcal{A}\left(\frac{r_1 + r_2 + \sum_{k \neq 1,2} r_k}{\mathcal{K}} - 1\right).$$

Following Section 3.2, the spot exchange rate can be calculated as:

$${}_1E_2 = \frac{\partial Z(r_1, r_2)/\partial r_2}{\partial Z(r_1, r_2)/\partial r_1} = \frac{r_1 \cdot \left[\mathcal{A} \cdot r_2 \cdot \prod_k r_k + \mathcal{K} \cdot \left(\frac{\mathcal{K}}{n}\right)^n\right]}{r_2 \cdot \left[\mathcal{A} \cdot r_1 \cdot \prod_k r_k + \mathcal{K} \cdot \left(\frac{\mathcal{K}}{n}\right)^n\right]}. \tag{45}$$

A.4.3 Swap Amount. We investigate the case when a user swaps token$_1$ for token$_2$, while the reserves of all other assets remain untouched in the pool. Based on the Curve conservation function (Equation (44)), the amount of token$_2$ received x_2 (spent when $x_2 < 0$) given amount of token$_1$ spent x_1 (received when $x_1 < 0$) can be calculated following the steps below:

$$\begin{aligned} r_1' &= r_1 + x_1 \\ r_2' &= \frac{\sqrt{\frac{4\mathcal{K}\left(\frac{\mathcal{K}}{n}\right)^n}{\mathcal{A} \cdot \prod'_{k \neq 2}} + \left[\left(1 - \frac{1}{\mathcal{A}}\right)\mathcal{K} - \sum'_{k \neq 2}\right]^2} + \left(1 - \frac{1}{\mathcal{A}}\right)\mathcal{K} - \sum'_{k \neq 2}}{2} \\ x_2 &= r_2 - r_2', \end{aligned} \tag{46}$$

where $\prod'_{k \neq 2} = r_1' \cdot \prod_{k \neq 1,2} r_k$ and $\sum'_{k \neq 2} = r_1' + \sum_{k \neq 1,2} r_k$.

A.4.4 Slippage. As illustrated in Figure 4(c), the slippage that a Curve user experiences when swapping x_1 token$_1$ with x_2 token$_2$ can be expressed as:

$$S(x_1) = \frac{x_1/x_2}{{}_1E_2} - 1 = \frac{\frac{x_1 \cdot \left[\mathcal{A} \cdot r_1 \cdot \prod_k r_k + \mathcal{K} \cdot \left(\frac{\mathcal{K}}{n}\right)^n\right]}{r_1 \cdot \left[\mathcal{A} \cdot r_2 \cdot \prod_k r_k + \mathcal{K} \cdot \left(\frac{\mathcal{K}}{n}\right)^n\right]}}{1 - \frac{\sqrt{\frac{4\mathcal{K}\left(\frac{\mathcal{K}}{n}\right)^n}{a \cdot \prod'_{k \neq 2}} + \left[\left(1 - \frac{1}{a}\right)\mathcal{K} - \sum'_{k \neq 2}\right]^2} + \left(1 - \frac{1}{a}\right)\mathcal{K} - \sum'_{k \neq 2}}{2r_2}} - 1, \tag{47}$$

A.4.5 Divergence Loss. Curve's divergence loss in full form cannot be easily presented in a concise and comprehensible fashion. Therefore, for Curve, we use the generalized method to calculate its divergence loss as described in Section 3.2. The divergence loss in the case of a 2-asset pool is presented in Figure 5(c).

[3] Note that $\mathcal{A}$ here is equivalent to $A \cdot n^n$ in Curve's white paper [68].

A.5 DODO

A.5.1 Spot Exchange Rate. The exchange rate between the two assets in a DODO pool is set by the market rate with an adjustment based on the pool composition. We denote the market exchange rate as P, namely, 1 token$_2$ = P token$_1$, and the initial reserve for token$_1$ and token$_2$ as $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively. The formula Equation (48) sets the exchange rate ${}_1E_2$ higher than the market rate P—i.e., token$_2$ exhibits higher price in the pool than in the market, when the reserve of token$_1$ r_1 exceeds its initial state $\mathcal{R}_1$, and sets ${}_1E_2$ lower than P—i.e., token$_1$ more expensive than its market value, when r_1 falls short of $\mathcal{R}_1$. Formally,

$$
{}_1E_2 = \begin{cases} P\left[1 + \mathcal{A}\left(\left(\frac{\mathcal{R}_2}{r_2}\right)^2 - 1\right)\right], & r_1 \geq \mathcal{R}_1 \\ P \Bigg/ \left[1 + \mathcal{A}\left(\left(\frac{\mathcal{R}_1}{r_1}\right)^2 - 1\right)\right], & r_1 \leq \mathcal{R}_1 \end{cases}. \tag{48}
$$

A.5.2 Conservation Function. DODO's conservation function can be derived from its exchange formula Equation (48). In particular, the initial state of token$_1$ and token$_2$ reserves, $\mathcal{R}_1$ and $\mathcal{R}_2$ can be regarded as the two invariants of the conservation function. This aligns with the definition according to our framework (Section 3), as $\mathcal{R}_1$ and $\mathcal{R}_2$ remain constant with swapping activities, but get updated with liquidity provision or withdrawal.

$$
r_1 - \mathcal{R}_1 = \int_{r_2}^{\mathcal{R}_2} P\left[1 + \mathcal{A}\left(\left(\frac{\mathcal{R}_2}{\delta}\right)^2 - 1\right)\right] d\delta = P \cdot (\mathcal{R}_2 - r_2) \cdot \left[1 + \mathcal{A} \cdot \left(\frac{\mathcal{R}_2}{r_2} - 1\right)\right], \qquad r_1 \geq \mathcal{R}_1 \tag{49}
$$

$$
r_2 - \mathcal{R}_2 = \int_{r_1}^{\mathcal{R}_1} \frac{1 + \mathcal{A}\left(\left(\frac{\mathcal{R}_1}{\delta}\right)^2 - 1\right)}{P} d\delta = \frac{(\mathcal{R}_1 - r_1) \cdot \left[1 + \mathcal{A} \cdot \left(\frac{\mathcal{R}_1}{r_1} - 1\right)\right]}{P}, \qquad r_1 \leq \mathcal{R}_1. \tag{50}
$$

In the special case of $\mathcal{A} = 1$, when $C_1 = P \cdot C_2$, i.e., liquidity provided on both assets are of equal value, then DODO's conservation function is equivalent to Uniswap, with $r_1 \cdot r_2 = C_1 \cdot P \cdot C_2$. This can be observed from Figure 3, where the DODO's conservation function curve with $\mathcal{A} \to 1$ appears identical to that of Uniswap.

A.5.3 Swap Amount. The swap amount can be derived directly from the DODO conservation function (Equation (49)):

$$
\begin{aligned}
r_1' &= r_1 + x_1 \\
r_2' &= \begin{cases} \dfrac{\begin{array}{c}\mathcal{R}_1 - r_1' + P\cdot\mathcal{R}_2\cdot(1-2\mathcal{A}) + \\ \sqrt{\left[\mathcal{R}_1 - r_1' + P\cdot\mathcal{R}_2\cdot(1-2\mathcal{A})\right]^2 + 4\mathcal{A}\cdot(1-\mathcal{A})\cdot(P\cdot\mathcal{R}_2)^2}\end{array}}{2P\cdot(1-\mathcal{A})}, & r_1' \geq \mathcal{R}_1 \\ \mathcal{R}_2 + \dfrac{(\mathcal{R}_1 - r_1')\cdot\left[1+\mathcal{A}\cdot\left(\frac{\mathcal{R}_1}{r_1'} - 1\right)\right]}{P}, & r_1' \leq \mathcal{R}_1 \end{cases} \\
x_2 &= r_2 - r_2'.
\end{aligned} \tag{51}
$$

A.5.4 Slippage. As illustrated in Figure 4(d), the slippage that a DODO user experiences when swapping x_1 token$_1$ with x_2 token$_2$ can be expressed as:

$$S(x_1) = \begin{cases} \dfrac{2\cdot(1-\mathcal{A})\cdot x_1}{r_1'-\mathcal{R}_1+\mathcal{R}_2\cdot P-\sqrt{\left[\mathcal{R}_1-r_1'+P\cdot\mathcal{R}_2\cdot(1-2\mathcal{A})\right]^2+4\mathcal{A}\cdot(1-\mathcal{A})\cdot(P\cdot\mathcal{R}_2)^2}} - 1, & r_1' \geq \mathcal{R}_1 \\ \dfrac{x_1}{(r_1'-\mathcal{R}_1)\cdot\left[1+A\cdot\left(\frac{\mathcal{R}_1}{r_1'}-1\right)\right]} - 1, & r_1' \leq \mathcal{R}_1 \end{cases}. \tag{52}$$

A.5.5 Divergence Loss. DODO eliminates the kind of divergence loss seen in previously discussed protocols by setting the ratio between the reserve assets supplied by the LP as the pool's equilibrium state (see Section 4.1.5).

Table 4. Function Comparison Table of Uniswap, Balancer, Curve, and DODO

	Uniswap V2	Uniswap V3	Balancer	Curve	DODO
Conservation function $Z(\{r_k\};\mathcal{I})=0$	$\mathcal{K}=r_1\cdot r_2$	$\left(r_1+\frac{\mathcal{R}_1}{\sqrt{\mathcal{A}}-1}\right)\cdot\left(r_2+\frac{\mathcal{R}_2}{\sqrt{\mathcal{A}}-1}\right)=\frac{\mathcal{A}\cdot\mathcal{R}_1\cdot\mathcal{R}_2}{(\sqrt{\mathcal{A}}-1)^2}$	$\mathcal{K}=\prod_k r_k^{w_k}$	$\mathcal{A}\left(\frac{\sum_k r_k}{\mathcal{K}}-1\right)=\frac{\left(\frac{\mathcal{K}}{n}\right)^n}{\prod_k r_k}-1$	$\begin{cases} r_1-\mathcal{R}_1=P\cdot(\mathcal{R}_2-r_2)\cdot\left[1+\mathcal{A}\cdot\left(\frac{\mathcal{R}_2}{r_2}-1\right)\right], & r_1\geq\mathcal{R}_1 \\ r_2-\mathcal{R}_2=\frac{(\mathcal{R}_1-r_1)\cdot\left[1+\mathcal{A}\cdot\left(\frac{\mathcal{R}_1}{r_1}-1\right)\right]}{P}, & r_1\leq\mathcal{R}_1 \end{cases}$
Spot exchange rate ${}_iE_o(\{r_k\};\mathcal{I})$ $=\frac{\partial Z(\{r_k\};\mathcal{I})/\partial r_o}{\partial Z(\{r_k\};\mathcal{I})/\partial r_i}$	$\frac{r_1}{r_2}$	$\frac{r_1+\frac{\mathcal{R}_1}{\sqrt{\mathcal{A}}-1}}{r_2+\frac{\mathcal{R}_2}{\sqrt{\mathcal{A}}-1}}$	$\frac{r_1\cdot w_2}{r_2\cdot w_1}$	$\frac{r_1\cdot\left[\mathcal{A}\cdot r_2\cdot\prod_k r_k+\mathcal{K}\cdot\left(\frac{\mathcal{K}}{n}\right)^n\right]}{r_2\cdot\left[\mathcal{A}\cdot r_1\cdot\prod_k r_k+\mathcal{K}\cdot\left(\frac{\mathcal{K}}{n}\right)^n\right]}$	$\begin{cases} P\left[1+\mathcal{A}\left(\left(\frac{\mathcal{R}_2}{r_2}\right)^2-1\right)\right], & r_1\geq\mathcal{R}_1 \\ P\Big/\left[1+\mathcal{A}\left(\left(\frac{\mathcal{R}_1}{r_1}\right)^2-1\right)\right], & r_1\leq\mathcal{R}_1 \end{cases}$
Post-swap token$_1$ reserve r_1'	r_1+x_1				
Post-swap token$_2$ reserve r_2'	$\frac{\mathcal{K}}{r_1'}$	$\frac{\frac{\mathcal{R}_1\mathcal{R}_2}{\left(1-\frac{1}{\sqrt{\mathcal{A}}}\right)^2}}{\left(r_1'+\frac{\mathcal{R}_1}{\sqrt{\mathcal{A}}-1}\right)}-\frac{\mathcal{R}_2}{\sqrt{\mathcal{A}}-1}$	$r_2\left(\frac{r_1}{r_1'}\right)^{\frac{w_1}{w_2}}$	$\frac{\sqrt{\frac{4\mathcal{K}\left(\frac{\mathcal{K}}{n}\right)^n}{\mathcal{A}\cdot\prod'_{k\neq2}}+\left[\left(1-\frac{1}{\mathcal{A}}\right)\mathcal{K}-\sum'_{k\neq2}\right]^2}+\left(1-\frac{1}{\mathcal{A}}\right)\mathcal{K}-\sum'_{k\neq2}}{2}$	$\begin{cases} \frac{\mathcal{R}_1-r_1'+P\cdot\mathcal{R}_2\cdot(1-2\mathcal{A})+\sqrt{\left[\mathcal{R}_1-r_1'+P\cdot\mathcal{R}_2\cdot(1-2\mathcal{A})\right]^2+4\mathcal{A}\cdot(1-\mathcal{A})\cdot(P\cdot\mathcal{R}_2)^2}}{2P\cdot(1-\mathcal{A})}, & r_1'\geq\mathcal{R}_1 \\ \mathcal{R}_2+\frac{(\mathcal{R}_1-r_1')\cdot\left[1+\mathcal{A}\cdot\left(\frac{\mathcal{R}_1}{r_1'}-1\right)\right]}{P}, & r_1'\leq\mathcal{R}_1 \end{cases}$
Swap amount x_2	$r_2'-r_2$				
Slippage $S(x_i,\{r_k\};\mathcal{I})$ $=\frac{x_i/x_o}{{}_iE_o}-1$	$\frac{x_1}{r_1}$	$\frac{x_1}{r_1+\frac{\mathcal{R}_1}{\sqrt{\mathcal{A}}-1}}$	$\frac{\frac{x_1}{r_1}\cdot\frac{w_1}{w_2}}{1-\left(\frac{r_1}{r_1'}\right)^{\frac{w_1}{w_2}}}-1$	$\frac{\frac{x_1\cdot\left[\mathcal{A}\cdot r_1\cdot\prod_k r_k+\mathcal{K}\cdot\left(\frac{\mathcal{K}}{n}\right)^n\right]}{r_1\cdot\left[\mathcal{A}\cdot r_2\cdot\prod_k r_k+\mathcal{K}\cdot\left(\frac{\mathcal{K}}{n}\right)^n\right]}}{1-\frac{\sqrt{\frac{4\mathcal{K}\left(\frac{\mathcal{K}}{n}\right)^n}{a\cdot\prod'_{k\neq2}}+\left[\left(1-\frac{1}{a}\right)\mathcal{K}-\sum'_{k\neq2}\right]^2}+\left(1-\frac{1}{a}\right)\mathcal{K}-\sum'_{k\neq2}}{2r_2}}-1$	$\begin{cases} \frac{2\cdot(1-\mathcal{A})\cdot x_1}{r_1'-\mathcal{R}_1+\mathcal{R}_2\cdot P-\sqrt{\left[\mathcal{R}_1-r_1'+P\cdot\mathcal{R}_2\cdot(1-2\mathcal{A})\right]^2+4\mathcal{A}\cdot(1-\mathcal{A})\cdot(P\cdot\mathcal{R}_2)^2}}-1, & r_1'\geq\mathcal{R}_1 \\ \frac{x_1}{(r_1'-\mathcal{R}_1)\cdot\left[1+\mathcal{A}\cdot\left(\frac{\mathcal{R}_1}{r_1'}-1\right)\right]}-1, & r_1'\leq\mathcal{R}_1 \end{cases}$
Divergence loss $L(\rho,\{r_k\};\mathcal{I})$ $=\frac{V'(\rho,\{r_k\};\mathcal{I})}{V_{\text{held}}(\rho;\{r_k\},\mathcal{I})}-1$	$\frac{\sqrt{1+\rho}}{1+\frac{\rho}{2}}-1$	$\begin{cases} \frac{(\rho+1)\cdot\sqrt{\mathcal{A}}-1}{2+\rho}, & -1\leq\rho\leq\frac{1}{\mathcal{A}}-1 \\ \frac{\frac{\sqrt{1+\rho}}{1+\frac{\rho}{2}}-1}{1-\frac{1}{\sqrt{\mathcal{A}}}}, & \frac{1}{\mathcal{A}}-1\leq\rho\leq\mathcal{A}-1 \\ \frac{\sqrt{\mathcal{A}}-1-\rho}{2+\rho}, & \rho\geq\mathcal{A}-1 \end{cases}$	$\frac{(1+\rho)^{w_2}}{1+w_2\cdot\rho}-1$	Complex	0 at equilibrium

Formulas are derived in Appendix A.1–Appendix A.5. Conservation functions are visualized in Figure 3, slippage functions in Figure 4, and divergence loss functions in Figure 5. Python implementation is available on GitHub.

B SELECTED ATTACK ALGORITHMS

In this Appendix, we show four algorithms of selected attacks.

Attack Algorithm 1: Flash-loan-funded price oracle attack

```
**Take a flash loan** to borrow $x_A$ token$_A$ from a lending platform, whose value is equivalent to $x_B$ token$_B$ at market price.
**Swap** $x_A$ token$_A$ for $x_B - \Delta_1$ token$_B$ on an AMM, pushing the new price of token$_A$ in terms of token$_B$ down to $\frac{x_B - \Delta_2}{x_A}$, where $\Delta_2 > \Delta_1 > 0$ due to slippage.
**Borrow** $x_A + \Delta_3$ token$_A$ with $x_B - \Delta_1$ token$_B$ as collateral on a lending platform that uses the AMM as their sole price oracle. To temporarily satisfy overcollateralization, $\frac{x_B - \Delta_2}{x_A} < \frac{x_B - \Delta_1}{x_A + \Delta_3}$.
**Repay the flash loan** with $x_A$ token$_A$.
```

Attack Algorithm 2: Rug Pull

```
**Mint** a new coin XYZ.
**Create** a liquidity pool with $x_{\mathtt{XYZ}}$ XYZ and $x_{\mathtt{ETH}}$ ETH (or any other valuable cryptocurrency) on an AMM, and receive LP tokens.
**Attract** unwitting traders to buy XYZ with ETH from the pool, effectively changing the composition of the pool.
**Withdraw** liquidity from the pool by surrendering LP tokens, and obtain $x_{\mathtt{XYZ}} - \Delta_1$ XYZ and $x_{\mathtt{ETH}} + \Delta_2$ ETH, where $\Delta_1, \Delta_2 > 0$.
```

Attack Algorithm 3: Sandwich LP attack

```
User$_A$ places a transaction order to buy $x_A$ token$_A$ with token$_B$ with a pool containing $r_A$ token$_A$ and $r_B$ token$_B$ with gas fee $g_1$.
LP$_B$ **observes** the mempool and sees the transaction.
LP$_B$ **frontruns** by withdrawing liquidity $k\, r_A$ token$_A$ and $k\, r_B$ token$_B$ with a higher gas fee $g_2 > g_1$.
LP$_B$ and User$_A$'s transactions are executed sequentially, resulting in a new composition of the pool with $(1-k)r_A + x_A$ token$_A$ and $(1-k)r_B - x_B$ token$_B$.
LP$_B$ **backruns** by re-providing $k\, r_A$ token$_A$ and $k \cdot \frac{(1-k)r_B - x_B}{(1-k)r_A + x_A}$ token$_B$.
LP$_B$ **backruns** by selling $(1 - \frac{(1-k)r_B - x_B}{(1-k)r_A + x_A})$ token$_B$ for some token$_A$
```

Attack Algorithm 4: Sandwich price attack

```
User$_A$ wishes to purchase $x_A$ XYZ whose spot price is $P_1$ on an AMM with gas fee $g_1$.
User$_B$ **observes** the mempool and sees the transaction.
User$_B$ **frontruns** by buying $x_B$ XYZ with a higher gas fee $g_2 > g_1$ on the same AMM.
User$_B$ and User$_A$'s transactions are executed sequentially at respective average price of $P_B$ and $P_A$, pushing XYZ's spot price up to $P_2$, where $P_2 > P_A > P_B > P_1$ due to slippage.
User$_B$ **backruns** by selling $x_B$ XYZ at an average price of $P'_B$, with $P_2 > P'_B > P_B$ due to slippage.
```

C OVERVIEW TABLES FOR ATTACKS AND RELATED WORK

Table 5. Overview of Related SoKs, Surveys, and Tutorials

		Subjects covered				Methodology					Aspects			
Reference	Summary	DEX	AMM	broader DeFi	broader DLT	Literature review	Taxonomization	Modeling	Simulation	Empirical investigation	Mechanical / structural	Economic	Security	Privacy
This article	An SoK on AMM-based DEX protocols that uses state space modeling to abstract their mechanics and surveys their security and privacy issues	●	●	●	●	●	●	●	●	●	●	●	●	●
[25]	A theoretical framework with parametric characterization of AMM-based DEX's fundamental properties and behaviors, with a focus on their structural and economic aspects	●	●	○	○	○	●	●	○	○	●	●	○	○
[26]	An SoK on frontrunning mitigation in DeFi especially AMMs	●	●	●	◐	○	○	●	○	○	●	◐	●	●
[130]	An overview of security challenges and design principles of order-book-based DEX	○	●	○	○	○	●	○	○	●	●	○	●	○
[177]	An SoK on various DeFi applications drawing the distinction between their technical and economic security aspects, with a focus on the latter	◐	◐	●	○	○	●	○	○	●	●	●	●	●
[24]	An SoK on decentralized lending protocols proposing a formalization of their archetypal implementations and properties in the context of the broader DeFi ecosystem	◐	◐	●	○	○	●	●	○	○	●	●	○	○
[50]	An SoK on DeFi yield aggregators providing a general framework for yield farming strategies with numerical simulations and an empirical evaluation on their profitability	○	○	●	○	○	◐	●	●	●	●	●	●	○
[70]	An SoK on various frontrunning attacks providing a taxonomy exemplified with case studies and summarizing preventative measurements	◐	○	●	●	○	●	○	●	●	●	○	●	●
[163]	A quantitative study that examines various forms of Blockchain Extractable Value (BEV) and estimate historical attack profit through exploitation of BEV	●	●	●	●	◐	●	●	○	●	●	●	●	●
[162]	A study that systematizes lending protocols with a focus on liquidation mechanisms, and empirically assesses the liquidation risk of decentralized lending protocols	○	○	●	○	○	○	●	○	●	●	●	●	○
[93]	A study that provides a generalization of the financial risk existent on DeFi lending protocols and stress-tests their robustness under various scenarios	◐	◐	●	○	○	●	◐	●	●	●	●	●	○
[18]	A survey of Ethereum smart contract attacks with a taxonomy of common programming pitfalls and their minimum working examples	○	○	○	●	○	●	○	○	○	●	○	●	○
[103]	A survey of vulnerabilities, threats, and defenses of different security reference architecture (SRA) layers of a blockchain using a stacked model	◐	◐	●	●	●	●	○	○	○	●	○	●	●
[43]	A survey of vulnerabilities, attacks, and defenses of decentralized applications (DApps) running on top of the Ethereum blockchain	○	○	◐	●	●	●	●	○	○	●	○	●	○

Table 6. Overview of Theoretical and Anecdotal Attacks Against AMM-based DEXs as well as their Mitigating Solutions

Attack layer	Attacks	Literature	Affected AMMs	Estimated loss	Attack time	Solutions
Infrastructure	Block timestamp manipulation	[15, 52, 104, 132, 217]	—	—	—	[124, 191]
	Transaction sequence manipulation	[15, 70, 97]	—	—	—	[70]
	Other infrastructure	[16, 91, 135, 152, 168, 171, 175]	—	—	—	Practices in traditional cybersecurity
Middleware	Reentrancy attack	[7, 48, 105, 160, 195]	Uniswap V1 [47]	≥25.00m USD	04/2020	[8, 42, 77, 118, 122, 174]
	Other middleware	[161, 167, 179, 189]	—	—	—	[36, 121, 224]
Application	Oracle attack	[93, 100, 145, 150, 157, 186]	Curve [170] QuickSwap [90] DODO [29]	≥30.00m USD ≥2.40m USD ≥0.70m USD	11/2020 07/2021 03/2021	[186, 205]
	Rug pull	[3, 9, 39, 74, 193, 211]	UraniumFinance [123, 127] SushiSwap [110]	≥50.00m USD ≥13.00m USD	04/2021 08/2020	[39, 131, 136, 211]
	Frontrunning	[54, 59, 70, 194, 228]	Various [134] DODO [29]	≥280.00m USD ≥0.70m USD	monthly 03/2021	[26, 59, 70, 227]
	Backrunning	[119, 177, 228]	—	—	—	[78, 203, 227]
	Sandwich attacks	[228, 230]	Various [67]	≥0.12m USD	monthly	[101, 227, 230]
	Vampire attack	[55, 81, 107, 120]	Uniswap V2 [209] Uniswap V2 [144] Uniswap V2 [81]	≥1000.00m USD ≥4300.00m USD ≥239.00m USD	08/2020 09/2020 09/2021	[80]

GLOSSARY

call option. a financial derivative instrument giving its owner the right to buy an asset at a given price

hedging. to invest in offsetting positions of a security to minimize the risk of adverse price movements of an asset

mint-quote. the amount of tokens or currency that is being created by the protocol or more generally the monetary institution

numéraire. the base value for comparing values across multiple items allowing for comparison of products or financial instruments

out-of-the-money. a situation when an option contract is worthless as the underlying asset is under-/overpriced accordingly compared to the strike price of the option

put option. a financial derivative instrument giving its owner the right to sell an asset at a given price and time

redeem-quote. the amount of tokens or currency that is being returned by stakeholders to the protocol

ACRONYMS

AMM automated market maker
BDoS blockchain denial-of-service
BGP border gateway protocol
CFMM constant function market maker
CMM Custom Market Maker
DAO Decentralized Autonomous Organization
dApp decentralized application
DDoS distributed denial-of-service
DeFi decentralized finance
DEX decentralized exchange
DLT distributed ledger technology
DNS domain name server
EVM Ethereum Virtual Machine
IDO initial DEX offering
IEO initial exchange offering
L7 DDoS application-layer distributed denial-of-service
LBP Liquidity Bootstrapping Pool
LMSR logarithmic market scoring rule
LP liquidity provider
MEV miner extractable value
MPC multiparty computation
NFT non-fungible token
NIZK non-interactive zero-knowledge
PBS proposer/builder separation
PMM proactive market maker
SoK systematization of knowledge
ZKP zero-knowledge proof

ACKNOWLEDGMENTS

We are indebted to Nazariy Vavryk for his contribution to the early code base for the numeric illustration of various AMMs and insights into security breaches on AMMs. We thank Haopeng Song and Zehua Zhang for their excellent research assistance.

REFERENCES

[1] 2021. Defi Pulse. Retrieved from https://defipulse.com/.

[2] 2021. Ethereum Improvement Proposals - EIP-20: Token Standard. Retrieved from https://eips.ethereum.org/EIPS/eip-20.

[3] 2021. Rug Pull. Retrieved from https://coinmarketcap.com/alexandria/glossary/rug-pull.

[4] Hayden Adams. 2018. Uniswap Whitepaper (v1). Retrieved from https://hackmd.io/C-DvwDSfSxuh-Gd4WKE_ig.

[5] Hayden Adams, Noah Zinsmeister, and Dan Robinson. 2020. Uniswap v2 Core. Retrieved from https://uniswap.org/whitepaper.pdf.

[6] Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. 2021. Uniswap v3 Core. Retrieved from https://uniswap.org/whitepaper-v3.pdf.

[7] Elvira Albert, Shelly Grossman, Noam Rinetzky, Clara Rodríguez-Núñez, Albert Rubio, and Mooly Sagiv. 2020. Taming callbacks for smart contract modularity. *ACM Program. Lang.* 4, OOPSLA (11 2020), 30. DOI: https://doi.org/10.1145/3428277

[8] Ayman Alkhalifah, Alex Ng, Paul A. Watters, and A. S. M. Kayes. 2021. A mechanism to detect and prevent Ethereum blockchain smart contract reentrancy attacks. *Front. Comput. Sci.* 3 (2021), 1.

[9] Ampleforth. 2021. Ampleforth Home Page. Retrieved from https://www.ampleforth.org/.

[10] Henrik Andersson. 2020. mStable — Introducing Constant Sum Bonding Curves for Tokenised Assets. Retrieved from https://medium.com/mstable/introducing-constant-sum-bonding-curves-for-tokenised-assets-6e18879cdc5b.

[11] Guillermo Angeris and Tarun Chitra. 2020. Improved price oracles: Constant function market makers. In *Advances in Financial Technologies*. ACM, New York, NY, 80–91. DOI: https://doi.org/10.1145/3419614.3423251

[12] Guillermo Angeris, Alex Evans, and Tarun Chitra. 2021. A note on bundle profit maximization. Retrieved from https://angeris.github.io/papers/flashbots-mev.pdf.

[13] Guillermo Angeris, Alex Evans, and Tarun Chitra. 2021. A Note on Privacy in Constant Function Market Makers. Retrieved from http://arxiv.org/abs/2103.01193.

[14] Guillermo Angeris, Alex Evans, and Tarun Chitra. 2021. Replicating Market Makers. Retrieved from http://arxiv.org/abs/2103.14769.

[15] Andreas M. Antonopoulos and Gavin Wood. 2018. *Mastering Ethereum: Building Smart Contracts and Dapps*. O'Reilly Media.

[16] Maria Apostolaki, Aviv Zohar, and Laurent Vanbever. 2017. Hijacking Bitcoin: Routing attacks on cryptocurrencies. In *IEEE Symposium on Security and Privacy (SP)*. 375–392.

[17] Arbitrum. 2021. Arbitrum – Scaling Ethereum. Retrieved from https://arbitrum.io/.

[18] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. 2017. A survey of attacks on Ethereum smart contracts (SoK). In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 10204. Springer Verlag, 164–186. DOI: https://doi.org/10.1007/978-3-662-54455-6_8

[19] Balancer. 2021. Liquidity Bootstrapping Pools (LBPs). Retrieved from https://docs.balancer.fi/products/balancer-pools/liquidity-bootstrapping-pools-lbps.

[20] Balancer. 2022. Swap Fees. Retrieved from https://docs.balancer.fi/concepts/fees#swap-fees.

[21] Bancor. 2020. Announcing Bancor V2. Retrieved from https://blog.bancor.network/announcing-bancor-v2-2f56b515e9d8.

[22] Bancor. 2020. Bancor V2.1 Technical Explainer. (2020). Retrieved from https://drive.google.com/file/d/16EY7FUeS4MXnFjSf-KCgdE-Xyj4re27G/view.

[23] Bancor Network. 2021. FAQs - Bancor Network. Retrieved from https://docs.bancor.network/faqs#how-does-impermanent-loss-insurance-work.

[24] Massimo Bartoletti, James Hsin-yu Chiang, and Alberto Lluch Lafuente. 2021. SoK: Lending pools in decentralized finance. In *Workshop Proceedings of Financial Cryptography and Data Security*. Springer, Berlin, 553–578. DOI: https://doi.org/10.1007/978-3-662-63958-0_40

[25] Massimo Bartoletti, James Hsin-Yu Chiang, and Alberto Lluch-Lafuente. 2021. A theory of automated market makers in DeFi. In *International Conference on Coordination Languages and Models*. 168–187. DOI: https://doi.org/10.1007/978-3-030-78142-2_11

[26] Carsten Baum, James Hsin-yu Chiang, Bernardo David, Tore Kasper Frederiksen, and Lorenzo Gentile. 2021. *SoK: Mitigation of Front-running in Decentralized Finance*. Technical Report. https://eprint.iacr.org/2021/1628.

[27] Carsten Baum, Bernardo David, and Tore Kasper Frederiksen. 2021. P2DEX: Privacy-preserving decentralized cryptocurrency exchange. In *Applied Cryptography and Network Security*. Springer International Publishing, Cham, 163–194. DOI: https://doi.org/10.1007/978-3-030-78372-3_7

[28] Joseph Bebel and Dev Ojha. 2022. Ferveo: Threshold Decryption for Mempool Privacy in BFT Networks. *Cryptology ePrint Archive* (2022).

[29] Rob Behnke. 2021. Explained: The DODO DEX Hack. Retrieved from https://halborn.com/explained-the-dodo-dex-hack-march-2021/.

[30] Jorge Bernal Bernabe, Jose Luis Canovas, Jose L. Hernandez-Ramos, Rafael Torres Moreno, and Antonio Skarmeta. 2019. Privacy-preserving solutions for blockchain: Review and challenges. *IEEE Access* 7 (2019), 164908–164940.

[31] Blank. 2021. Blank Features beyond Basic Privacy (#2): Protecting Your IP in DeFi. Retrieved from https://blankwallet.medium.com/blank-features-beyond-basic-privacy-2-protecting-your-ip-in-defi-11bc76f2d67b.

[32] Sean Bowe, Alessandro Chiesa, Matthew Green, Ian Miers, Pratyush Mishra, and Howard Wu. 2020. ZEXE: Enabling decentralized private computation. In *IEEE Symposium on Security and Privacy*. IEEE, 947–964. DOI: https://doi.org/10.1109/SP40000.2020.00050

[33] Aseem Brahma, Mithun Chakraborty, Sanmay Das, Allen Lavoie, and Malik Magdon-Ismail. 2012. A bayesian market maker. In *ACM Conference on Electronic Commerce*. DOI: https://doi.org/10.1145/2229012.2229031

[34] Lorenz Breidenbach, Philip Daian, Florian Tramèr, and Ari Juels. 2018. Enter the hydra: Towards principled bug bounties and exploit-resistant smart contracts. In *USENIX Security Symposium*. 1335–1352. Retrieved from https://www.usenix.org/conference/usenixsecurity18/presentation/breindenbach.

[35] Anton Bukov and Mikhail Melnik. 2020. Mooniswap by 1inch.exchange. Retrieved from https://mooniswap.exchange/docs/MooniswapWhitePaper-v1.0.pdf.

[36] Benedikt Bünz, Shashank Agrawal, Mahdi Zamani, and Dan Boneh. 2020. Zether: Towards privacy in a smart contract world. In *International Conference on Financial Cryptography and Data Security*. 423–443.

[37] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. 2018. Bulletproofs: Short proofs for confidential transactions and more. In *IEEE Symposium on Security and Privacy (SP)*. 315–334.

[38] Burgerswap. 2020. Burgerswap: Decentralized Finance Platform. (2020). Retrieved from https://burgerswap.org/whitepaper_burgerswap.pdf.

[39] Bybit Learn. 2021. Why Crypto Rug Pulls Happen in DeFi and How to Avoid It. Retrieved from https://learn.bybit.com/investing/why-crypto-rug-pulls-happen-in-defi/.

[40] Yixin Cao, Chuanwei Zou, and Xianfeng Cheng. 2021. Flashot: A Snapshot of Flash Loan Attack on DeFi Ecosystem. (1 2021). Retrieved from http://arxiv.org/abs/2102.00626.

[41] Agostino Capponi and RUIZHE JIA. 2021. The Adoption of Blockchain-based Decentralized Exchanges: A Market Microstructure Analysis of the Automated Market Maker. (2021). DOI: https://doi.org/10.2139/ssrn.3805095

[42] Ethan Cecchetti, Siqiu Yao, Haobin Ni, and Andrew C. Myers. 2021. Compositional security for reentrant applications. In *IEEE Symposium on Security and Privacy*. IEEE, 1249–1267. DOI: https://doi.org/10.1109/SP40001.2021.00084

[43] Huashan Chen, Marcus Pendleton, Laurent Njilla, and Shouhuai Xu. 2020. A survey on Ethereum systems security. *ACM Comput. Surv.* 53, 3 (6 2020). DOI: https://doi.org/10.1145/3391195

[44] Weili Chen, Xiongfeng Guo, Zhiguang Chen, Zibin Zheng, and Yutong Lu. 2020. Phishing scam detection on Ethereum: Towards financial security for blockchain ecosystem. In *International Joint Conferences on Artificial Intelligence*. 4506–4512.

[45] Raymond Cheng, Fan Zhang, Jernej Kos, Warren He, Nicholas Hynes, Noah Johnson, Ari Juels, Andrew Miller, and Dawn Song. 2019. Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts. In *IEEE European Symposium on Security and Privacy (EuroS&P)*. 185–200.

[46] Tarun Chitra, Guillermo Angeris, and Alex Evans. 2021. How Liveness Separates CFMMs and Order Books. Retrieved from https://angeris.github.io/papers/cfmm-ob.pdf.

[47] ConsenSys. 2020. Thoughts on DeFi Security. A deep dive into the Uniswap and… by ConsenSys. ConsenSys Media. Retrieved from https://media.consensys.net/thoughts-on-defi-security-640dde37bb3b.

[48] Consensys Diligence. 2019. ConsenSys/Uniswap-audit-report-2018-12. Retrieved from https://github.com/ConsenSys/Uniswap-audit-report-2018-12#31-liquidity-pool-can-be-stolen-in-some-tokens-eg-erc-777-29.

[49] Simon Cousaert, Nikhil Vadgama, and Jiahua Xu. 2022. Token-based insurance solutions on blockchain. *Blockchains and the Token Economy: Theory and Practice*. Springer International Publishing, Cham, 237–260. https://doi.org/10.1007/978-3-030-95108-5_9

[50] Simon Cousaert, Jiahua Xu, and Toshiko Matsui. 2022. SoK: Yield aggregators in DeFi. In *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 1–14. DOI: https://doi.org/10.1109/ICBC54727.2022.9805523

[51] Ronald Cramer, Ivan Bjerre Damgård, et al. 2015. *Secure Multiparty Computation*. Cambridge University Press.

[52] Crypto Market Pool. 2020. Block Timestamp Manipulation Attack. Retrieved from https://cryptomarketpool.com/block-timestamp-manipulation-attack/.

[53] CryptoLocally. 2020. GIV Balancer Listing and Staking Rewards Updates. Retrieved from https://cryptolocally.medium.com/giv-balancer-listing-and-staking-rewards-updates-81ebb5843e58.

[54] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. 2020. Flash Boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *IEEE Symposium on Security and Privacy (SP)*. IEEE, 910–927. DOI:https://doi.org/10.1109/SP40000.2020.00040

[55] Brady Dale. 2020. SushiSwap Will Withdraw Up to $830M from Uniswap Today: Why It Matters for DeFi. Retrieved from https://www.coindesk.com/sushiswap-uniswap-migration-defi-amm-wars.

[56] Ankush Das, Stephanie Balzer, Jan Hoffmann, Frank Pfenning, and Ishani Santurkar. 2021. Resource-aware session types for digital contracts. In *IEEE 34th Computer Security Foundations Symposium (CSF)*. 1–16.

[57] Luca De Giglio. 2021. Geyser: Staking Rewards for Uniswap Liquidity Providers. Retrieved from https://medium.com/trips-community/geyser-staking-rewards-for-uniswap-liquidity-providers-115afc6f5c07.

[58] DefiLlama. 2022. Dexes TVL Rankings. Retrieved from https://defillama.com/protocols/Dexes.

[59] Degate. 2021. An Analysis of Ethereum Front-running and Its Defense Solutions. Retrieved from https://globalcoinresearch.com/2021/05/04/an-analysis-of-ethereum-front-running-and-its-defense-solutions/.

[60] Alex Gedevani. 2020. Delphi digital. *Layer 2: Rollups*. Technical Report.

[61] demosthenes.eth. 2021. Uniswap proposal: Managing Systemic Risk in Uniswap's Community Treasury using KPI Options. Retrieved from https://gov.uniswap.org/t/temperature-check-should-we-be-managing-systemic-risk-in-uniswaps-community-treasury-using-kpi-options/12624.

[62] DODO. 2021. DODO Pool Incident Postmortem: With a Little Help from Our Friends. Retrieved from https://medium.com/dodoex/dodo-pool-incident-postmortem-with-a-little-help-from-our-friends-327e66872d42.

[63] DODO. 2021. How to Create a Pool? Retrieved from https://dodoexhelp.zendesk.com/hc/en-us/articles/900005558243-How-to-create-a-pool-.

[64] DODO Team. 2020. DODO – A Next-Generation On-chain Liquidity Provider Powered by Pro-active Market Maker Algorithm. Retrieved from https://dodoex.github.io/docs/docs/whitepaper/.

[65] Jules DuPont and Anna Cinzia Squicciarini. 2015. Toward de-anonymizing Bitcoin by mapping users location. In *5th ACM Conference on Data and Application Security and Privacy*. 139–141.

[66] dYdX. 2021. Trade Now on Layer 2. Retrieved from https://dydx.exchange/blog/public.

[67] Anton Dzyatkovskii. 2021. No Sandwich, Please!—Popular DeFi Attack Strategy Analysis. *Hackernoon* (5 2021). Retrieved from https://hackernoon.com/no-sandwich-please-popular-defi-attack-strategy-analysis-jk1734rf.

[68] Michael Egorov. 2019. StableSwap-efficient mechanism for Stablecoin liquidity. Retrieved from https://curve.fi/files/stableswap-paper.pdf.

[69] Felix Engelmann, Thomas Kerber, Markulf Kohlweiss, and Mikhail Volkhov. 2022. Zswap: zk-SNARK based non-interactive multi-asset swaps. *Proc. Privac. Enhanc. Technol.* 4 (2022), 507–527. DOI:https://doi.org/10.2478/popets-2022-0120

[70] Shayan Eskandari, Seyedehmahsa Moosavi, and Jeremy Clark. 2020. SoK: Transparent dishonesty: Front-running attacks on blockchain. In *Lecture Notes in Computer Science*, Vol. 11599. Springer, 170–189. DOI:https://doi.org/10.1007/978-3-030-43725-1_13

[71] Ethereum. 2020. Types. Retrieved from https://docs.soliditylang.org/en/latest/types.html.

[72] ethereum.org. 2021. Layer 2 Rollups. Retrieved from https://ethereum.org/en/developers/docs/scaling/layer-2-rollups/.

[73] ethereum.org. 2021. Scaling. Retrieved from https://ethereum.org/en/developers/docs/scaling/.

[74] Etherscan. 2021. TruAmpl (TMPL) Token Tracker. Retrieved from https://etherscan.io/token/0xfcb755b046ea9b9bc4586db4018b49c5a02e3d1c.

[75] EulerBeats. 2021. About EulerBeats. Retrieved from https://eulerbeats.com/about.

[76] Alex Evans. 2020. Liquidity Provider Returns in Geometric Mean Markets. (6 2020). Retrieved from http://arxiv.org/abs/2006.08806.

[77] Noama Fatima Samreen and Manar H. Alalfi. 2020. Reentrancy vulnerability identification in Ethereum smart contracts. In *IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*. 22–29. DOI:https://doi.org/10.1109/IWBOSE50093.2020.9050260

[78] Yebo Feng, Jun Li, and Thanh Nguyen. 2020. Application-layer DDoS defense with reinforcement learning. In *IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*. 1–10.

[79] Yebo Feng, Jiahua Xu, and Lauren Weymouth. 2022. University blockchain research initiative (UBRI): Boosting blockchain education and research. *IEEE Potentials* (2022). DOI:https://doi.org/10.1109/MPOT.2022.3198929

[80] William Foxley. 2021. Uniswap V3 Introduces New License to Spoil Future SUSHIs. Retrieved from https://www.coindesk.com/tech/2021/03/23/uniswap-v3-introduces-new-license-to-spoil-future-sushis/.

[81] William Foxley. 2021. "Continuous Vampire Attack": The AMM Wars Are Getting Interesting with Integral - CoinDesk. Retrieved from https://www.coindesk.com/tech/2021/03/29/continuous-vampire-attack-the-amm-wars-are-getting-interesting-with-integral/.

[82] Mark B. Garman. 1976. Market microstructure. *J. Finan. Econ.* 3, 3 (6 1976), 257–275. DOI: https://doi.org/10.1016/0304-405X(76)90006-4

[83] Ryan Garner, Webb Mycelium, Jason Potts, Chris Berg, and Sinclair Davidson. 2021. Tracer: Perpetual swaps. Retrieved from https://www.tracer.finance/static/Tracer%20Perpetual%20Swaps-ea826cb7819c7655e078119ee7acf83e.pdf.

[84] Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. In *41st Annual ACM Symposium on Theory of Computing*. 169–178.

[85] Alex Gluchowski. 2019. ZK Rollup: scaling with Zero-knowledge Proofs. Retrieved from https://pandax-statics.oss-cn-shenzhen.aliyuncs.com/statics/1221233526992813.pdf.

[86] Gnosis. 2020. Custom Market Maker—Gnosis Developer Portal Gnosis Protocol. Retrieved from https://docs.gnosis.io/protocol/docs/intro-cmm/.

[87] Oded Goldreich and Yair Oren. 1994. Definitions and properties of zero-knowledge proof systems. *J. Cryptol.* 7, 1 (1994), 1–32.

[88] Shafi Goldwasser and Guy N. Rothblum. 2007. On best-possible obfuscation. In *Theory of Cryptography Conference*. 194–213.

[89] Kavya Govindarajan, Dhinakaran Vinayagamurthy, Praveen Jayachandran, and Chester Rebeiro. 2022. Privacy-preserving decentralized exchange marketplaces. In *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 1–9.

[90] Ana Grabundzija. 2021. BSC DeFi app "Pancakebunny" Releases Post-mortem of $2.4 Million Exploit. Retrieved from https://cryptoslate.com/bsc-defi-app-pancakebunny-releases-post-mortem-of-2-4-million-exploit/.

[91] Richard Greene and Michael N. Johnstone. 2018. An investigation into a denial of service attack on an Ethereum network. In *Proceedings of the 16th Australian Information Security Management Conference*, 90 pages.

[92] Lewis Gudgeon, Pedro Moreno-Sanchez, Stefanie Roos, Patrick McCorry, and Arthur Gervais. 2020. SoK: Layer-two blockchain protocols. In *Financial Cryptography and Data Security*, Vol. 12059 LNCS. Springer, Cham, 201–226. DOI: https://doi.org/10.1007/978-3-030-51280-4_12

[93] Lewis Gudgeon, Daniel Perez, Dominik Harz, Benjamin Livshits, and Arthur Gervais. 2020. The decentralized financial crisis. In *Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 1–15. DOI: https://doi.org/10.1109/CVCBT50464.2020.00005

[94] Gyroscope Finance. 2021. Autonomous Pricing. Retrieved from https://docs.gyro.finance/gyroscope-protocol/stablecoin/autonomous-pricing.

[95] Gyroscope Finance. 2021. Gyroscope, the New All-weather Stablecoin. Retrieved from https://gyro.finance/.

[96] Abdelatif Hafid, Abdelhakim Senhaji Hafid, and Mustapha Samih. 2020. Scaling blockchains: A comprehensive survey. *IEEE Access* 8 (2020), 125244–125262. DOI: https://doi.org/10.1109/ACCESS.2020.3007251

[97] hagaetc. 2021. Weekly DEX Volume. Retrieved from https://dune.xyz/queries/4323/8547https://github.com/flashbots/pm.

[98] Robin Hanson. 2003. Combinatorial information market design. *Inf. Syst. Front.* 5, 1 (2003), 107–119. Retrieved from http://hanson.gmu.edu.

[99] Robin Hanson. 2012. Logarithmic markets scoring rules for modular combinatorial information aggregation. *J. Predict. Mark.* 1, 1 (12 2012), 3–15. DOI: https://doi.org/10.5750/jpm.v1i1.417

[100] Harvest Finance. 2020. Harvest Flashloan Economic Attack Post-mortem. Retrieved from https://medium.com/harvest-finance/harvest-flashloan-economic-attack-post-mortem-3cf900d65217.

[101] Lioba Heimbach and Roger Wattenhofer. 2022. Eliminating sandwich attacks with the help of game theory. In *Asia Conference on Computer and Communications Security*. ACM, New York, NY, 153–167. DOI: https://doi.org/10.1145/3488932.3517390

[102] Eyal Hertzog, Guy Benartzi, and Galia Benartzi. 2018. Bancor Protocol Continuous Liquidity for Cryptographic Tokens through Their Smart Contracts. (2018). Retrieved from https://storage.googleapis.com/website-bancor/2018/04/01ba8253-bancor_protocol_whitepaper_en.pdf.

[103] Ivan Homoliak, Sarad Venugopalan, Daniel Reijsbergen, Qingze Hum, Richard Schumi, and Pawel Szalachowski. 2021. The security reference architecture for blockchains: Toward a standardized model for studying vulnerabilities, threats, and defenses. *IEEE Commun. Surv. Tutor.* 23, 1 (1 2021), 341–390. DOI: https://doi.org/10.1109/COMST.2020.3033665

[104] Sophie Huang. 2019. Will 2020 Be the Year of DEX? Retrieved from https://medium.com/@kidinamoto/will-2020-be-the-year-of-dex-ac7dfb6276e8.

[105] Yongfeng Huang, Yiyang Bian, Renpu Li, J. Leon Zhao, and Peizhong Shi. 2019. Smart contract security: A software lifecycle perspective. *IEEE Access* 7 (2019), 150184–150202.

[106] HydraDX. 2021. Intro. HydraDX Docs. Retrieved from https://docs.hydradx.io/.

[107] Jakub. 2020. What Is a Vampire Attack? SushiSwap Saga Explained. Retrieved from https://finematics.com/vampire-attack-sushiswap-explained/.

[108] Maxim Jourenko, Mario Larangeira, Kanta Kurazumi, and Keisuke Tanaka. 2019. SoK: A Taxonomy for Layer-2 Scalability Related Protocols for Cryptocurrencies. Retrieved from https://eprint.iacr.org/2019/352.pdf.

[109] Janyl Jumadinova and Prithviraj Dasgupta. 2012. A Comparison of Different Automated Market-maker Strategies. 2009–2012. Retrieved from http://www.cs.allegheny.edu/~jjumadinova/market-maker_AMEC.pdf.

[110] Bradley Keoun, Omkar Godbole, and Sebastian Sinclair. 2020. First Mover: SushiSwap's Billion-dollar "Rug Pull" Is Thriller to Crypto Geeks - CoinDesk. Retrieved from https://www.coindesk.com/markets/2020/09/08/first-mover-sushiswaps-billion-dollar-rug-pull-is-thriller-to-crypto-geeks/.

[111] Can Kisagun. 2019. Preventing DEX Front-running with Enigma. Retrieved from https://blog.enigma.co/preventing-dex-front-running-with-enigma-df3f0b5b9e78.

[112] Georgios Konstantopoulos. 2021. (Almost) Everything You Need to Know about Optimistic Rollup. Retrieved from https://research.paradigm.xyz/rollups.

[113] Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. 2016. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *IEEE Symposium on Security and Privacy (SP)*. 839–858.

[114] Bhaskar Krishnamachari, Qi Feng, and Eugenio Grippo. 2021. Dynamic automated market makers for decentralized cryptocurrency exchange. In *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 1–2. DOI: https://doi.org/10.1109/icbc51069.2021.9461100

[115] Kyber Network. 2021. Kyber 3.0: Architecture Revamp, Dynamic MM, and KNC Migration Proposal. Retrieved from https://blog.kyber.network/kyber-3-0-architecture-revamp-dynamic-mm-and-knc-migration-proposal-acae41046513.

[116] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. 2020. A survey on the security of blockchain systems. *Fut. Gen. Comput. Syst.* 107 (2020), 841–853.

[117] Iuon-Chang Lin and Tzu-Chun Liao. 2017. A survey of blockchain security issues and challenges. *Int. J. Netw. Secur.* 19, 5 (2017), 653–659.

[118] Chao Liu, Han Liu, Zhao Cao, Zhong Chen, Bangdao Chen, and Bill Roscoe. 2018. ReGuard: Finding reentrancy bugs in smart contracts. In *IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)*. IEEE, 65–68.

[119] livnev. 2020. Random Ordering of Equally-priced Transactions Incentivises Competitive Spam. Retrieved from https://github.com/ethereum/go-ethereum/issues/21350.

[120] Yuen Lo and Francesca Medda. 2020. Uniswap and the rise of the decentralized exchange. Retrieved from https://mpra.ub.uni-muenchen.de/103925/1/MPRA_paper_103925.pdf.

[121] Ning Lu, Bin Wang, Yongxin Zhang, Wenbo Shi, and Christian Esposito. 2019. NeuCheck: A more practical Ethereum smart contract security analysis tool. *Softw.: Pract. Exper.* 51, 10 (2019).

[122] Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. 2016. Making smart contracts smarter. In *ACM SIGSAC Conference on Computer and Communications Security*. ACM, New York, NY, 254–269. DOI: https://doi.org/10.1145/2976749.2978309

[123] Jordan Lyanchev. 2021. $50M Drained from Uranium Finance: Hack or Rug Pull? (2021). Retrieved from https://cryptopotato.com/50m-drained-from-uranium-finance-hack-or-rug-pull/.

[124] Guangkai Ma, Chunpeng Ge, and Lu Zhou. 2020. Achieving reliable timestamp in the bitcoin platform. *Peer-to-peer Netw. Applic.* 13, 6 (2020), 2251–2259.

[125] Igor Makarov and Antoinette Schoar. 2020. Trading and arbitrage in cryptocurrency markets. *J. Finan. Econ.* 135, 2 (2020), 293–319.

[126] Semyon Malamud and Marzena Rostek. 2017. Decentralized exchange. *Amer. Econ. Rev.* 107, 11 (11 2017), 3320–3362. DOI: https://doi.org/10.1257/aer.20140759

[127] Shaurya Malwa. 2021. DeFi "Rug Pull" Scams Pulled in $2.8B This Year: Chainalysis. (12 2021). Retrieved from https://www.coindesk.com/markets/2021/12/17/defi-rug-pull-scams-pulled-in-28b-this-year-chainalysis/.

[128] Fernando Martinelli. 2021. Introducing Balancer V2: Generalized AMMs. Retrieved from https://medium.com/balancer-protocol/balancer-v2-generalizing-amms-16343c4563ff.

[129] Fernando Martinelli and Nikolai Mushegian. 2019. Balancer: A Non-custodial Portfolio Manager, Liquidity Provider, and Price Sensor. (2019). Retrieved from https://balancer.finance/whitepaper/.

[130] Fabio Massacci and Chan Nam Ngo. 2021. Distributed financial exchanges: Security challenges and design principles. *IEEE Secur. Priv.* 19, 1 (1 2021), 54–64. DOI: https://doi.org/10.1109/MSEC.2020.2994826

[131] Bruno Mazorra, Victor Adan, and Vanesa Daza. 2022. Do Not Rug on Me: Zero-dimensional Scam Detection. *arXiv preprint arXiv:2201.07220* (2022).

[132] Alexander Mense and Markus Flatscher. 2018. Security vulnerabilities in Ethereum smart contracts. In *Proceedings of the 20th International Conference on Information Integration and Web-Based Applications & Services (iiWAS'18)*. Association for Computing Machinery, New York, NY, 375–380. https://doi.org/10.1145/3282373.3282419

[133] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. 2013. Zerocoin: Anonymous distributed e-cash from Bitcoin. In *IEEE Symposium on Security and Privacy*. 397–411.

[134] Edvardas Mikalauskas. 2021. $280 million stolen per month from crypto transactions. *Cybernews* (2021). Retrieved from https://cybernews.com/crypto/flash-boys-2-0-front-runners-draining-280-million-per-month-from-crypto-transactions/.

[135] Michael Mirkin, Yan Ji, Jonathan Pang, Ariah Klages-Mundt, Ittay Eyal, and Ari Juels. 2020. BDoS: Blockchain denial-of-service. In *ACM SIGSAC Conference on Computer and Communications Security*. ACM, New York, NY, 601–619. DOI: https://doi.org/10.1145/3372297.3417247

[136] Mudra Manager. 2021. Why Locking Liquidity Is Important for Cryptocurrency. Retrieved from https://hackernoon.com/why-locking-liquidity-is-important-for-cryptocurrency-qv4d37hd.

[137] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. 2016. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction.* Princeton University Press.

[138] Nasdaq. 2021. How Ethereum Layer 2's Are Leveling Up DeFi. Retrieved from https://www.nasdaq.com/articles/how-ethereum-layer-2s-are-leveling-up-defi-2021-06-08.

[139] Francesco Nava. 2015. Efficiency in decentralized oligopolistic markets. *J. Econ. Theor.* 157 (5 2015), 315–348. DOI: https://doi.org/10.1016/j.jet.2015.01.009

[140] Allan Niemerg, Dan Robinson, and Lev Livnev. 2020. YieldSpace: An Automated Liquidity Provider for Fixed Yield Tokens. (2020). Retrieved from https://yield.is/Yield.pdf.

[141] Shen Noether. 2015. Ring Signature confidential transactions for Monero. *IACR Cryptol. ePrint Arch.* 2015, 1098.

[142] Notional Finance. 2020. Notional AMM. Retrieved from https://docs.notional.finance/traders/technical-topics/notional-amm.

[143] Notional Finance. 2021. Notional Finance. Retrieved from https://notional.finance/.

[144] Jeremy Ong. 2021. PancakeSwap: A Perpetual Vampire? - Delphi Digital. Retrieved from https://members.delphidigital.io/reports/pancakeswap-a-perpetual-vampire/.

[145] Kris Oosthoek. 2021. Flash Crash for Cash: Cyber Threats in Decentralized Finance. (6 2021). Retrieved from https://arxiv.org/abs/2106.10740v1.

[146] Optimism. 2021. Optimism home page. Retrieved from https://optimism.io/.

[147] Abraham Othman, David M. Pennock, Daniel M. Reeves, and Tuomas Sandholm. 2013. A practical liquidity-sensitive automated market maker. *ACM Trans. Econ. Computat.* 1, 3 (9 2013), 1–25. DOI: https://doi.org/10.1145/2509413.2509414

[148] Abraham Othman and Tuomas Sandholm. 2011. Liquidity-sensitive automated market makers via homogeneous risk measures. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* Vol. 7090. Springer, Berlin, 314–325. DOI: https://doi.org/10.1007/978-3-642-25510-6_27

[149] PeckShield. 2020. Uniswap/Lendf.Me Hacks: Root Cause and Loss Analysis. Retrieved from https://peckshield.medium.com/uniswap-lendf-me-hacks-root-cause-and-loss-analysis-50f3263dcc09.

[150] PeckShield. 2020. Value DeFi Incident: Root Cause Analysis. Retrieved from https://peckshield.medium.com/value-defi-incident-root-cause-analysis-fbab71faf373.

[151] Daniel Perez, Sam M. Werner, Jiahua Xu, and Benjamin Livshits. 2021. Liquidations: DeFi on a knife-edge. In *Financial Cryptography and Data Security*. Retrieved from http://arxiv.org/abs/2009.13235.

[152] Daniel Perez, Jiahua Xu, and Benjamin Livshits. 2020. Revisiting transactional statistics of high-scalability blockchains. In *ACM Internet Measurement Conference.* ACM, New York, NY, 535–550. DOI: https://doi.org/10.1145/3419394.3423628

[153] Perpetual Protocol. 2021. vAMM. Retrieved from https://docs.perp.fi/getting-started/how-it-works/vamm.

[154] William Perraudin and Paolo Vitale. 1996. Interdealer trade and information flows in a decentralized foreign exchange market. In *The Microstructure of Foreign Exchange Markets.* University of Chicago Press, 73–106.

[155] Jack Peterson and Joseph Krug. 2015. Augur: a decentralized, open-source platform for prediction markets. Retrieved from https://cryptochainuni.com/wp-content/uploads/Augur-A-Decentralized-Open-Source-Platform-for-Prediction-Markets.pdf.

[156] Ross Phillips and Heidi Wilder. 2020. Tracing cryptocurrency scams: Clustering replicated advance-fee and phishing websites. In *IEEE International Conference on Blockchain and Cryptocurrency (ICBC).* 1–8.

[157] Benjamin Pirus. 2020. Cheese Bank's Multi-million-dollar Hack Explained by Security Firm. Retrieved from https://cointelegraph.com/news/cheese-bank-s-multi-million-dollar-hack-explained-by-security-firm.

[158] Pods Finance. 2021. The Easiest Way to Hedge Crypto. Retrieved from https://www.pods.finance/.

[159] Polygon. 2021. Ethereum's Internet of Blockchains. Retrieved from https://polygon.technology/.

[160] Nathaniel Popper. 2016. A Hacking of More Than $50 Million Dashes Hopes in the World of Virtual Currency. Retrieved from https://www.nytimes.com/2016/06/18/business/dealbook/hacker-may-have-removed-more-than-50-million-from-experimental-cybercurrency-project.html.

[161] Praitheeshan Purathani, Lei Pan, Jiangshan Yu, Joseph Liu, and Robin Doss. 2019. Security analysis methods on Ethereum smart contract vulnerabilities: A survey. *arXiv preprint arXiv:1908.08605* (2019).

[162] Kaihua Qin, Liyi Zhou, Pablo Gamito, Philipp Jovanovic, and Arthur Gervais. 2021. An empirical study of DeFi liquidations. In *21st ACM Internet Measurement Conference*. ACM, New York, NY, 336–350. DOI: https://doi.org/10.1145/3487552.3487811

[163] Kaihua Qin, Liyi Zhou, and Arthur Gervais. 2022. Quantifying blockchain extractable value: How dark is the forest? In *IEEE Symposium on Security and Privacy*.

[164] Kaihua Qin, Liyi Zhou, Benjamin Livshits, and Arthur Gervais. 2021. Attacking the DeFi ecosystem with flash loans for fun and profit. *Financial Cryptography and Data Security*. Borisov Nikita and Diaz Claudia (Eds). Springer, Berlin, Heidelberg, 3–32.

[165] QuickSwap Official. 2020. QuickSwap FAQ. Retrieved from https://quickswap-layer2.medium.com/welcome-to-quickswap-exchange-93d47e057633.

[166] Mayank Raikwar and Danilo Gligoroski. 2021. Aggregation in blockchain ecosystem. In *International Conference on Software Defined Systems (SDS)*. IEEE, 1–6. DOI: https://doi.org/10.1109/SDS54264.2021.9732100

[167] Paritosh Ramanan, Dan Li, and Nagi Gebraeel. 2021. Blockchain-based decentralized replay attack detection for large-scale power systems. *IEEE Trans. Syst. Man Cyber. Syst.* 52, 8 (2021).

[168] Anju Ramdas and Ramakrishnan Muthukrishnan. 2019. A survey on DNS security issues and mitigation techniques. In *International Conference on Intelligent Computing and Control Systems (ICCS)*. IEEE, 781–784.

[169] Rango. 2022. Rango Docs. Retrieved from https://docs.rango.exchange/.

[170] Jamie Redman. 2020. Report: Blockchain Price Oracle Manipulation Produces Millions in Losses, Shows No Signs of Slowing – Altcoins Bitcoin News. Retrieved from https://news.bitcoin.com/report-blockchain-price-oracle-manipulation-produces-millions-in-losses-shows-no-signs-of-slowing/.

[171] Ludovic Rembert. 2021. The 51% Attack. Retrieved from https://privacycanada.net/cryptocurrency/51-attack/.

[172] Andreas Richardson and Jiahua Xu. 2020. Carbon trading with blockchain. In *Mathematical Research for Blockchain Economy*. Springer, 105–124. DOI: https://doi.org/10.1007/978-3-030-53356-4_7

[173] Dan Robinson and Allan Niemerg. 2020. *The Yield Protocol: On-chain Lending with Interest Rate Discovery*. Technical Report. Yield Protocol.

[174] Michael Rodler, Wenting Li, Ghassan O. Karame, and Lucas Davi. 2019. Sereum: Protecting existing smart contracts against re-entrancy attacks. In *Network and Distributed System Security Symposium*. Internet Society. DOI: https://doi.org/10.14722/ndss.2019.23413

[175] Muhammad Saad, Jeffrey Spaulding, Laurent Njilla, Charles Kamhoua, Sachin Shetty, DaeHun Nyang, and Aziz Mohaisen. 2019. Exploring the attack surface of blockchain: A systematic overview. *arXiv preprint arXiv:1904.03487* (2019).

[176] Saber. 2021. Saber. Solana AMM and DEX. Retrieved from https://saber.so/.

[177] Sam M. Werner, Daniel Perez, Lewis Gudgeon, Ariah Klages-Mundt, Dominik Harz, and William J. Knottenbelt. 2022. SoK: Decentralized finance (DeFi). arXiv. https://arxiv.org/abs/2101.08778.

[178] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. 2014. Zerocash: Decentralized anonymous payments from Bitcoin. In *IEEE Symposium on Security and Privacy*. 459–474.

[179] Sarwar Sayeed, Hector Marco-Gisbert, and Tom Caira. 2020. Smart contract: Attacks and protections. *IEEE Access* 8 (2020), 24416–24427.

[180] Fabian Schär. 2021. Decentralized finance: On blockchain-and smart contract-based financial markets. *Fed. Res. Bank St. Lou. Rev.* 103, 2 (2021), 153–174. DOI: https://doi.org/10.20955/r.103.153-74

[181] Dmitri Senchenko. 2020. Impermanent Losses in Uniswap-Like Markets. Retrieved from https://dsenchenko.medium.com/impermanent-losses-in-uniswap-like-markets-4315359ea9b1.

[182] Cosimo Sguanci, Roberto Spatafora, and Andrea Mario Vergani. 2021. Layer 2 Blockchain Scaling: A Survey. *arXiv preprint arXiv:2107.10881* (2021).

[183] Jamsheed Shorish. 2018. Blockchain State Machine Representation. (2018). DOI: https://doi.org/10.31235/osf.io/eusxg

[184] Siren. 2021. SIREN Markets Summary. (2021). Retrieved from https://siren.xyz/whitepaper.

[185] Christian Slamka, Bernd Skiera, and Martin Spann. 2013. Prediction market performance and market liquidity: A comparison of automated market makers. *IEEE Trans. Eng. Manag.* 60, 1 (2 2013), 169–185. DOI: https://doi.org/10.1109/TEM.2012.2191618

[186] SmartContent. 2021. TWAP Oracles vs. Chainlink Price Feeds: A Comparative Analysis. Retrieved from https://smartcontentpublication.medium.com/twap-oracles-vs-chainlink-price-feeds-a-comparative-analysis-8155a3483cbd.

[187] StarkWare Industries Ltd. 2021. StarkNet. Retrieved from https://starkware.co/product/starknet/.

[188] Drew Stone. 2021. Trustless, Privacy-preserving Blockchain Bridges. (2021). Retrieved from http://arxiv.org/abs/2102.04660.

[189] Jinlei Sun, Song Huang, Changyou Zheng, Tingyong Wang, Cheng Zong, and Zhanwei Hui. 2021. Mutation testing for integer overflow in Ethereum smart contracts. *Tsinghua Sci. Technol.* 27, 1 (2021), 27–40.

[190] Sushiswap. 2020. The SushiSwap Project. Retrieved from https://sushiswapchef.medium.com/the-sushiswap-project-dd6eb80c6ba2.

[191] Pawel Szalachowski. 2018. (Short paper) towards more reliable Bitcoin timestamps. *Crypto Valley Conference on Blockchain Technology (CVCBT'18)*, 101–104. DOI: 10.1109/CVCBT.2018.00018

[192] Dan Taylor. 2021. Privacy First DeFi Sienna Network Raises $11.2 million, Takes Front-running Head on. Retrieved from https://tech.eu/brief/privacy-first-defi-sienna-network-raises-11-2-million-takes-front-running-head-on/.

[193] The European Business Review. 2021. What Is a "Rug Pull" in Crypto? DeFi Exploits Explained. Retrieved from https://www.europeanbusinessreview.com/what-is-a-rug-pull-in-crypto-defi-exploits-explained/.

[194] Christof Ferreira Torres, Ramiro Camino, et al. 2021. Frontrunner Jones and the raiders of the dark forest: An empirical study of frontrunning on the Ethereum blockchain. In *30th USENIX Security Symposium (USENIX Security'21).* 1343–1359.

[195] Petar Tsankov, Andrei Dan, Dana Drachsler-Cohen, Arthur Gervais, Florian Bünzli, and Martin Vechev. 2018. Securify: Practical security analysis of smart contracts. In *ACM Conference on Computer and Communications Security* (10 2018), 67–82. DOI: https://doi.org/10.1145/3243734.3243780

[196] Uniswap. 2020. Flash Swaps. Retrieved from https://uniswap.org/docs/v2/core-concepts/flash-swaps/.

[197] Uniswap. 2022. Liquidity provider fees. Retrieved from https://docs.uniswap.org/protocol/V2/concepts/advanced-topics/fees#liquidity-provider-fees.

[198] Uniswap Governance. 2021. Temperature Check - [Fee Switch V2 should be turned on]. Retrieved from https://gov.uniswap.org/t/temperature-check-fee-switch-v2-should-be-turned-on/13537.

[199] Uranium.finance. 2021. *How It Works - OUSD.* Technical Report. Origin Protocol. Retrieved from https://docs.ousd.com/how-it-works.

[200] Ryosuke Ushida and James Angel. 2021. Regulatory considerations on centralized aspects of defi managed by DAOs. In *FC International Workshops.* Vol. 12676 LNCS. Springer, 21–36. DOI: https://doi.org/10.1007/978-3-662-63958-0_2

[201] Vbuterin. 2022. State of research: increasing censorship resistance of transactions under proposer/builder separation (PBS). Retrieved from https://notes.ethereum.org/@vbuterin/pbs_censorship_resistance.

[202] Friedhelm Victor and Andrea Marie Weintraud. 2021. Detecting and Quantifying Wash Trading on Decentralized Cryptocurrency Exchanges. DOI: https://doi.org/10.1145/3442381.3449824

[203] Chenxu Wang, Tony T. N. Miu, Xiapu Luo, and Jinhe Wang. 2017. SkyShield: A sketch-based defense system against application layer DDoS attacks. *IEEE Trans. Inf. Forens. Secur.* 13, 3 (2017), 559–573.

[204] Dabao Wang, Siwei Wu, Ziling Lin, Lei Wu, Xingliang Yuan, Yajin Zhou, Haoyu Wang, and Kui Ren. 2020. Towards Understanding Flash Loan and Its Applications in DeFi Ecosystem. Retrieved from http://arxiv.org/abs/2010.12252.

[205] Shih-Hung Wang, Chia-Chien Wu, Yu-Chuan Liang, Li-Hsun Hsieh, and Hsu-Chun Hsiao. 2021. ProMutator: Detecting vulnerable price oracles in DeFi by mutated transactions. In *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW).* IEEE, 380–385.

[206] Yongge Wang. 2020. Automated Market Makers for Decentralized Finance (DeFi). Retrieved from http://arxiv.org/abs/2009.01676.

[207] Will Warren and Amir Bandeali. 2017. 0x: An open protocol for decentralized exchange on the Ethereum blockchain. Retrieved from https://github.com/0xProject/whitepaper/blob/master/0x_white_paper.pdf.

[208] Molly Wintermute. 2020. *Hegic: On-chain Options Trading Protocol on Ethereum Powered by Hedge Contracts and Liquidity Pools.* Technical Report. Hegic. Retrieved from https://github.com/hegic/whitepaper/blob/master/HegicProtocolWhitepaper.pdf.

[209] Joon Ian Wong. 2021. SushiSwap Drained UniSwap of $1 Billion in Liquidity and No One Knows Who Was Behind It to This Day. The Business of Business. Retrieved from https://www.businessofbusiness.com/articles/satoshi-30-billion-bitcoin-sushiswap-uniswap-defi-summer-crypto-anonymity-sybil-attacks/.

[210] Jiajing Wu, Qi Yuan, Dan Lin, Wei You, Weili Chen, Chuan Chen, and Zibin Zheng. 2020. Who are the phishers? Phishing scam detection on Ethereum via network embedding. *IEEE Trans. Syst. Man Cyber. Syst.* 52, 2 (2020).

[211] Pengcheng Xia, Haoyu Wang, Bingyu Gao, Weihang Su, Zhou Yu, Xiapu Luo, Chao Zhang, Xusheng Xiao, and Guoai Xu. 2021. Trade or trick? Detecting and characterizing scam tokens on Uniswap decentralized exchange. *Proc. ACM Measur. Anal. Comput. Syst.* 5, 3 (12 2021), 1–26. DOI: https://doi.org/10.1145/3491051

[212] Yi Xie and Shun-Zheng Yu. 2008. Monitoring the application-layer DDoS attacks for popular websites. *IEEE/ACM Trans. Netw.* 17, 1 (2008), 15–25.

[213] Jiahua Xu and Yebo Feng. 2022. Reap the harvest on blockchain: A survey of yield farming protocols. *IEEE Transactions on Network and Service Management* (2022). DOI: 10.1109/TNSM.2022.3222815

[214] Jiahua Xu and Nikhil Vadgama. 2022. From banks to DeFi: The evolution of the lending market. In *Enabling the Internet of Value.* Springer, 53–66. DOI: https://doi.org/10.1007/978-3-030-78184-2_6

[215] Teng Andrea Xu and Jiahua Xu. 2022. A short survey on business models of decentralized finance (DeFi) protocols. In *Workshop Proceedings of Financial Cryptography and Data Security*. DOI: https://doi.org/10.48550/arxiv.2202.07742

[216] Teng Andrea Xu, Jiahua Xu, and Kristof Lommers. 2022. DeFi vs TradFi: Valuation Using Multiples and Discounted Cash Flow. DOI: https://doi.org/10.48550/arxiv.2210.16846

[217] Aviv Yaish, Saar Tochner, and Aviv Zohar. 2022. Blockchain stretching & squeezing: Manipulating time for your best interest. In *Proceedings of the 23rd ACM Conference on Economics and Computation*. 65–88.

[218] YCharts. 2021. Ethereum Average Gas Price. Retrieved from https://ycharts.com/indicators/ethereum_average_gas_price.

[219] Akif Yüksel, Oguzhan Ersoy, and Zekeriya Erkin. 2021. Mitigating Sandwich Attacks in Kyber DMM. (2021). Retrieved from https://repository.tudelft.nl/islandora/object/uuid%3A58ac3b00-10fb-44cd-b1eb-1e1139c39fd7.

[220] Michael Zargham, Krzysztof Paruch, and Jamsheed Shorish. 2020. Economic games as estimators. In *Mathematical Research for Blockchain Economy*. Springer, Cham, 125–142. DOI: https://doi.org/10.1007/978-3-030-53356-4_8

[221] Michael Zargham, Jamsheed Shorish, and Krzysztof Paruch. 2020. From curved bonding to configuration spaces. In *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 1–3. DOI: https://doi.org/10.1109/ICBC48266.2020.9169474

[222] Michael Zargham, Zixuan Zhang, and Victor Preciado. 2018. A State-space Modeling Framework for Engineering Blockchain-enabled Economic Systems. Retrieved from http://arxiv.org/abs/1807.00955.

[223] Rui Zhang, Rui Xue, and Ling Liu. 2019. Security and privacy on blockchain. *ACM Comput. Surv.* 52, 3 (2019), 1–34.

[224] Yuyao Zhang, Siqi Ma, Juanru Li, Kailai Li, Surya Nepal, and Dawu Gu. 2020. SMARTSHEILD: Automatic smart contract protection made easy. In *IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. 23–34.

[225] Zixuan Zhang, Michael Zargham, and Victor M. Preciado. 2020. On modeling blockchain-enabled economic networks as stochastic dynamical systems. *Appl. Netw. Sci.* 5, 1 (12 2020), 19. DOI: https://doi.org/10.1007/s41109-020-0254-9

[226] Liyi Zhou, Kaihua Qin, Antoine Cully, Benjamin Livshits, and Arthur Gervais. 2021. On the just-in-time discovery of profit-generating transactions in DeFi protocols. In *IEEE Symposium on Security and Privacy (SP)*. IEEE, 919–936.

[227] Liyi Zhou, Kaihua Qin, and Arthur Gervais. 2021. A2MM: Mitigating Frontrunning, Transaction Reordering and Consensus Instability in Decentralized Exchanges. Retrieved from http://arxiv.org/abs/2106.07371.

[228] Liyi Zhou, Kaihua Qin, Christof Ferreira Torres, Duc V. Le, and Arthur Gervais. 2021. High-frequency trading on decentralized on-chain exchanges. In *IEEE Symposium on Security and Privacy*. 428–445. DOI: https://doi.org/10.1109/SP40001.2021.00027

[229] ZKSwap. 2021. ZKSwap home page. Retrieved from https://zks.org/en.

[230] Patrick Züst, Tejaswi Nadahalli, and Ye Wang Roger Wattenhofer. 2021. Analyzing and preventing sandwich attacks in Ethereum. (2021). Retrieved from www.DeFi-Sandwi.ch.

Received 30 January 2022; revised 24 July 2022; accepted 1 November 2022