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

Jiahua Xu
UCL CBT
jiahua.xu@ucl.ac.uk

Krzysztof Paruch
Vienna University of Economics and Business
Krzysztof.Paruch@wu.ac.at

Simon Cousaert
UCL CBT
simon.cousaert@outlook.com

Yebo Feng
University of Oregon
yebof@uoregon.edu

***Abstract*—As an integral part of the decentralized finance (DeFi) ecosystem, decentralized exchanges (DEX) 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 DEX, 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 DEX's inherent properties, and explore mitigating solutions. Finally, we conduct a comprehensive literature review on related work covering both DeFi and conventional market microstructure.**

***Index Terms*—Decentralized Finance, decentralized exchange, automated market maker, blockchain, Ethereum**

I. 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 development in the decentralized finance (DeFi) space. Among all the prominent DeFi applications, decentralized exchanges (DEX) with automated market maker (AMM) protocols are in the ascendancy, with an aggregate value locked exceeding $26 billion at the time of writing [1].

Different from order-book-based exchanges where the market price of an asset is determined by the last matched buy and sell orders, each AMM uses a so-called conservation function that price 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. As such, AMM-based DEX allow for accessible liquidity provision and exchange, especially for illiquid assets.

Despite apparent advantages such as decentralization, automation and continuous liquidity, 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 associate with myriads of security and privacy issues.

Contributions: To the best of our knowledge, the paper represents the first systematization of knowledge in AMM-based DEX with deployed protocol examples. We contribute to the body of literature mainly by:

1) generalizing mechanisms and economics of AMM-based DEX with a formalized state space modeling framework;
2) comparing major AMM-based DEX with mathematical derivation and parameterized illustration on their conservation function, slippage and divergence loss functions;
3) positioning AMM-based DEX 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 DEX, and exploring mitigation solutions.

II. AMM Preliminaries

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

A. Actors

a) 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 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 [2]. LPs earn transaction fees paid by exchange users. While sometimes subject to a withdrawal penalty, LPs can freely remove funds from the pool [3] by surrendering a corresponding amount of pool shares [2].

b) Exchange user (Trader): A trader submits an exchange order to a liquidity pool by specifying the input and output asset and one associated 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. AMMs such as DODO (see IV-A5) leverage users' arbitrage behavior to achieve certain protocol design.

c) Protocol foundation: 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.

B. Assets

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

a) Risk assets: Characterized with illiquidity, risk assets are the primary type of assets AMM-based DEX 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 [4] for most AMMs on Ethereum).

b) 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 with an elastic supply [5]. 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 pool without a designated base asset.

c) 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.

d) Protocol tokens: Protocol tokens are used to represent voting rights on protocol governance matters and are thus also termed "governance tokens". Protocol tokens are typically valuable assets that are tradeable outside of the AMM and can incentivize participation, e.g. when they are rewarded to LPs proportionate to their liquidity supply.

C. Fundamental AMM dynamics

1) Invariant properties: The functionality of an AMM depends upon a *conservation function* which 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) 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 which respect the defined conservation function, if one is specified, or conserve the defined properties otherwise.

D. Fundamental AMM economics

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.

a) 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.

b) 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` [6] and `TRIPS` [7]) 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.

c) Governance right: An AMM may encourage liquidity provision and/or swapping by rewarding participants governance rights in the form of protocol tokens (see II-B). 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.

d) Security reward: Just as every protocol built on top of an open, distributed network, AMM-based DEX 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 [8].

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.

a) Liquidity withdrawal penalty: As introduced in III-B and demonstrated in Section IV later in this paper, 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 [9] levy a liquidity withdrawal penalty to discourage this action.

b) Swap fee: Users interacting with the liquidity pool for token exchanges have to reimburse LPs for the supply of assets. 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.

c) Gas fee: Every interaction with the protocol is executed in the form of an on-chain transaction, and is thus subject to 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 results in higher gas fee due to the higher computational power needed for transaction verification.

3) Implicit costs: Two essential implicit costs native to AMM-based DEX are slippage for exchange users and divergence loss for LPs.

a) 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 V-A3).

b) 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 IV). 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.

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 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.

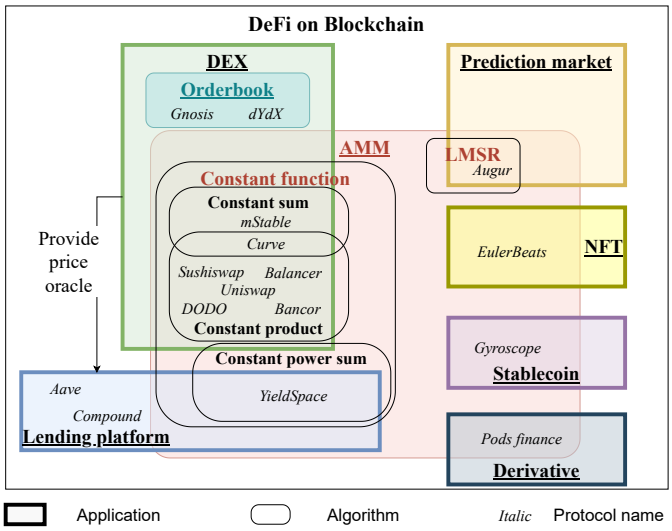


Figure 1: AMM-based DEX within the broader taxonomy of DeFi on blockchain. AMM as an algorithm and DEX as an application are not mutually inclusive.

E. AMM-based DEX within DeFi

For brevity, we use "AMM" or "DEX" to refer to AMM-based DEX throughout the paper, 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 DEX such as Gnosis and dYdX that do not rely on AMM algorithms. On the other hand, AMM algorithms are also not exclusively employed by DEX. 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 VI-B2). Prediction markets for example commonly employs logarithmic market scoring rule (LMSR), whereas constant function market maker (CFMM) is the primary underpinning for DEX. 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, III and IV, focus on CFMM mechanisms which have been adopted by major DEX, with their exact formulas derived in Appendix A. Appendix B briefly presents other DeFi applications with AMM implementations.

III. 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.

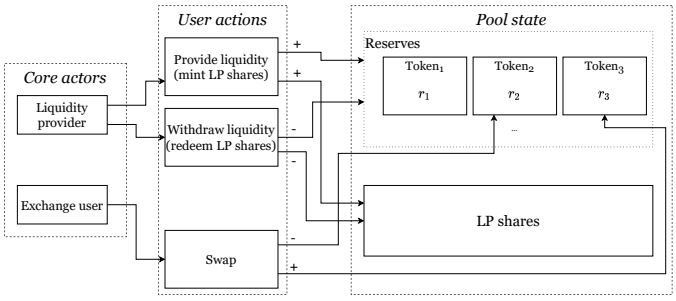


Figure 2: Stylized AMM mechanism

A. 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 actions (Figure 2); the evolution of the system over time is modelled 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

$$\chi = (\{r'_k\}_{k=1,...,n}, \{p_k\}_{k=1,...,n}, \mathcal{C}, \Omega) \tag{1}$$

where r_k denotes the quantity of token k in the pool, p_k the current spot price of token k, $\mathcal{C}$ 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 which defines the relationship between different state variables and the invariant(s) $\mathcal{C}$. The conservation function is protocol-specific as each protocol seeks to prioritize a distinct feature and target particular functionalities (see IV).

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 [11] represents constant-product market makers, while Balancer [3] generalizes this idea to a geometric mean. The Curve [12] conservation function is notably a combination of constant-sum and constant-product (see IV).

B. Liquidity change and asset swap

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

Invariant $\mathcal{C}$, despite its name, refers to the pool variable that stays constant only with swap actions 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. Liquidity provision and withdrawal, on the other hand, should not move the asset price.

General rules of AMM-based DEX

1) The price of assets in an AMM pool stays constant for *pure* liquidity provision and withdrawal activities.
2) The invariant of an AMM pool stays constant for *pure* swapping activities.

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

$$\begin{aligned}&(\{r_k\}_{k=1,...,n}, \{p_k\}_{k=1,...,n}, \mathcal{C}, \Omega)\\ \xrightarrow[f]{\text{liquidity change}}&(\{r'_k\}_{k=1,...,n}, \{p_k\}_{k=1,...,n}, \mathcal{C}', \Omega)\end{aligned} \tag{2}$$

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

Note that protocol-specific intricacies may result in asset price change with liquidity provision/withdrawal, or invariant $\mathcal{C}$ update with trading. This is a consequence of what is described in section II-C regarding each mechanism having to respect imposed system invariances.

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 ($r_1 : r_2 : ... : r_n$). In any other case, a change of 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, such as with Balancer [3]. 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. In other words, adding only one asset to the liquidity pool is *not an eligible state transition* and has to be treated accordingly. Consequently, this sequence of actions is no longer a *pure* liquidity provision/withdrawal and would thus move the asset spot price.

Swap fees (see II-D2b) also cause invariant $\mathcal{C}$ to become variant through trading. Specifically, when swap fees are kept within the liquidity pool, 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 $\mathcal{C}$ (see e.g. [13]). Also, as float numbers are not yet fully supported by Solidity [14]—the language for Ethereum smart contracts, AMM protocols typically recalculate invariant $\mathcal{C}$ after each trade to avoid the accumulation of rounding errors.

C. Generalized formulas

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

Table I: 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
State variables		
$\mathcal{C}$ or $(\mathcal{C}_1, \mathcal{C}_2)$	Conservation function invariant(s)	all
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

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 $\{r_k\}_{k=1,\ldots,n}$:

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

A conservation function for each token pair, say r_i—r_o, must be concave, nonnegative and nondecreasing [15] (see also Figure 5). For complex AMMs such as Curve, it might be convenient to express the conservation function (Equation 4) implicitly in order to derive exchange rates between two assets in a pool:

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

Equation 5 contains a constant $\mathcal{C}$, 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.

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 5) using partial derivatives of the conservation function Z.

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

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

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, a trader can still sell 2 or more units into the pool, but mostly accompanied with a high slippage (see Figure 6 in the next section).

a) 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:

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

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

b) 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:

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

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

$$r_o' \coloneqq R_o(x_i, \{r_k\}; \mathcal{C}) \tag{10}$$

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

$$x_o \coloneqq X_o(x_i, \{r_k\}; \mathcal{C}) = r_o - r_o' \tag{11}$$

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:

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

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 II-D3b). 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 [15].

a) 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$_i$:

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

b) 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:

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

Table II: Comparison table of discussed 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 September 2021. Data retrieved from DeFi Pulse and Dune Analytics on 21 September 2021.

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	1.5	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

c) 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).

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

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

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 $\{r_k\}$:

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

d) 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:

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

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

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

IV. Comparison of AMM protocols

AMM-based DEX are home to billions of dollars' worth of on-chain liquidity. Table II 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 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.

A. 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 [35] 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 IV. 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 Figure 5, Figure 6 and Figure 7, respectively. We always use token$_1$ as price or value unit; namely, token$_1$ is the assumed numéraire.

1) Uniswap V2: The Uniswap protocol prescribes that a liquidity pool always consists of one pair of assets. The pool's smart contract always assumes that the reserves of the two assets have equal value. Uniswap implements a conservation function with a constant-product invariant.

2) Uniswap V3: Uniswap V3 enhances Uniswap V2 by allowing liquidity provision to be concentrated on a fraction of the bonding curve [23] (see A2a), thus virtually amplifying the conservation function invariant and reducing the slippage.

The 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 $\mathcal{A} \to \infty$, the covered price range approaches $(0, \infty)$, and the LP's individual conservation function approximates a Uniswap V2 one ($\mathcal{A} = 10000$ in Figure 5a); on the other extreme, when $\mathcal{A} \to 1$, the liquidity only supports swaps close to the initial exchange rate, and the conservation function approximates a constant-sum one (Figure 5a).

3) Balancer: The Balancer protocol allows each liquidity pool to have more than two assets [3]. 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/removal 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 $w_1 = w_2 = \frac{1}{2}$ (Figure 5b).

4) Curve: With the Curve protocol, formerly StableSwap [12], a liquidity pool 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 has a near-0 weight, i.e. $\mathcal{A} \to 0$ (Figure 5c).

5) DODO: DODO supports customized pools [36] where a pool creator provides reserves on both sides of the trading pair with arbitrary quantities, which determines the pool's

Table III: Overview of important existing AMM-based DEX on Ethereum, Solana, Polkadot, Tezos and EOS. CS = Constant sum, CP = Constant product, OP = Oracle price component, CC = Capital concentration, T = Time component.

			AMM			AMM add-ons					
DEX		**Pool structure**	**CP**	**CS**	**OP**	**CC**	**T**	**Divergence loss compensation**	**Associated attacks**	**Chain**	**Mainnet launch**
Uniswap V1	[16]	asset-pair	●	○	○	○	○	—	[17]	Ethereum	11/2018
Uniswap V2	[18]	asset-pair	●	○	○	○	○	—	[19]–[22]	Ethereum	05/2020
Uniswap V3	[23]	asset-pair	●	○	○	●	○	—	—	Ethereum	05/2021
Balancer V1	[3]	multi-asset	●	○	○	○	●	—	[24]	Ethereum	03/2020
Balancer V2	[25]	multi-asset	●	○	○	○	●	—	—	Ethereum	—
Curve	[12]	multi-asset	●	●	○	○	○	—	[19], [21]	Ethereum	01/2020
DODO	[9]	various	●	○	●	○	○	—	[26]	Ethereum, BSC	09/2020
Bancor V1	[27]	asset-pair	●	○	○	○	○	—	—	Ethereum, EOS	06/2017
Bancor V2	[28]	asset-pair	●	○	●	○	○	—	—	Ethereum, EOS	04/2020
Bancor V2.1	[5]	asset-pair	●	○	○	○	○	Divergence loss insurance	—	Ethereum, EOS	10/2020
SushiSwap	[29]	asset-pair	●	○	○	○	○	—	[21], [22], [30]	Ethereum	08/2020
Mooniswap	[31]	asset-pair	●	○	○	○	●	—	—	Ethereum	08/2020
mStable	[10]	asset-pair	○	●	○	○	○	—	—	Ethereum	07/2020
Kyber 3.0	[32]	multi-asset	●	○	○	●	○	Dynamic swap fee	—	Ethereum, Tezos	03/2021
Saber	[33]	multi-asset	●	●	○	○	○	—	—	Solana	06/2021
HydraDX	[34]	multi-asset	●	○	●	●	○	—	—	Polkadot	—

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 A5b) derived from its exchange rate formula (see A5a).

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 $\mathcal{A} \to 1$, the DODO bonding curve resembles Uniswap V2 ($\mathcal{A} = 0.99$ in Figure 5d); and with a high slippage around the market price (Figure 6d), the pool exhibits a strong tendency to fall back to the equilibrium state. When $\mathcal{A} \to 0$, the DODO bonding curve resembles a constant sum one ($\mathcal{A} = 0.01$ in Figure 5d); and with a near-flat slippage (Figure 6d), the algorithm's force to pull the reserve ratio back to equilibrium is at its weakest due to little arbitrage profitability exhibited.

Summary: Each AMM has its quirks: Uniswap V2 implements an rudimentary bonding curve that achieves a low gas fee; Uniswap V3 allows for concentrated liquidity provision which improves capital efficiency; Balancer supports more than 2 assets in a pool; Curve is suitable for swapping assets with the same peg; DODO proactively reduces divergence loss by leveraging external price feeds.

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, different AMMs can be reduced to the basic form equivalent to Uniswap V2 (illustrated with blue curves in Figures 5, 6 and 7).

In other cases, AMMs assume different forms with a trade-off between slippage and divergence loss. Indeed, an AMM with low slippage is in favor of traders but to the detriment of LPs: given a range of price movement, traders would be able to exchange assets in higher volume, whereas LPs would suffer a higher divergence loss. Seemingly with the capability to achieve both low slippage and zero divergence loss at equilibrium (when $\mathcal{A} \to 0$), DODO appears to be an exception. Nevertheless, it is to be noted that with low $\mathcal{A}$, DODO's PMM algorithm is less effective in restoring the pool to its equilibrium state (see IV-A5), thus still exposing LPs to divergence loss risks in non-equilibrated states.

Ultimately, 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.

B. Other AMM-based DEXs

1) Sushiswap: Sushiswap is a fork of Uniswap V2 (see V-A3f). Though the two mainly differ in governance token structure and user experience, Sushiswap share the same conservation function, slippage and divergence loss functions as Uniswap.

2) 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" [37]. 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.

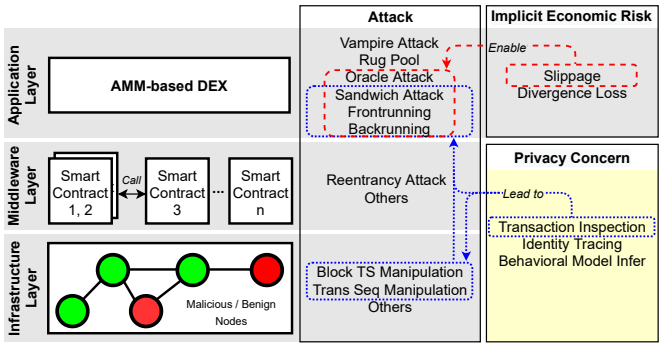


Figure 3: Architectural layers of an AMM-based DEX with its implicit economic risks, attacks, privacy concerns, and their relationships

3) Bancor: While Bancor's white paper [27] 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 [28] (see IV-C3).

C. Additional AMM features

1) Time component: A time component refers to the ability to change traditionally fixed hyperparameters over time. Balancer V1 and V2 implement this (Table III), 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 B4).

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 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.

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 [38]. This insurance policy is earned over time, 1% each day that liquidity is staked in the pool.

V. 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 DEX. Besides those risks, security and privacy matters are also to be taken into account when using AMM-based DEX.

In particular, as a complex, distributed system with a variety of software and hardware components interacting with each other, AMM-based DEX are prone to exhibit attack interfaces [39]–[41]. 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 DEX are subject to atomic, risk-free exploits on the protocol's technical structure such as its algorithmic pricing scheme [42]. Built on top of public blockchain infrastructures featuring transparency and traceability, AMM-based DEX also expose their users to privacy risks.

In this section, we define a taxonomy (illustrated in Figure 3) to enumerate potential security and privacy concerns of AMM-based DEX, expounding their root causes and possible mitigation solutions.

A. 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.

1) Infrastructure-layer attacks: The proper operation of DEX is 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 DEX.

a) 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 [43], or to tamper with the execution order of DEX transactions packed in different blocks [44].

To mitigate the negative impact of such manipulation, DEX contracts should be timestamp independent [45]. 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 [46]) and still maintain integrity [47]. Besides, DEX 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 [48].

b) 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 miner extractable value (MEV). This can be further facilitated by open-source software such as Flashbots [49].

To prevent transaction sequence manipulation, DEX 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 [50]. We also discuss how DEX and its transactions can practice transaction sequencing from application-layer in V-A3c, and how privacy-preserving blockchain and DEX are resistant to this attack in V-B.

c) Other infrastructure-layer attacks: Aiming to perturb operations of blockchain systems [51], 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 [52], [53], 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 [54]; the 51% attack [55], 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 [56] or border gateway protocol (BGP) hijacking [57].

In short, AMM-based DEX 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.

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 the DEX. However, the smart contracts and complex collaborations between them can also lead to potential system vulnerabilities. Attackers can exploit such attack interfaces to steal tokens from a DEX or even paralyze it.

a) 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 [58]. Afterwards, despite the emergence of various proposals addressing the reentrancy problems [59]–[61], reentrancy attacks persist, and became particularly threatening to AMM-based DEX. In January 2019, an audit identified a reentrancy vulnerability in Uniswap [62], which was then exploited by hackers to steal $25 million worth of tokens in April 2020 [63] Hackers performed this attack by leveraging a subtle interaction between two contracts that were secure in isolation, and a third malicious contract [64]. 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 [65].

The security community has proposed a variety of approaches to tackle reentrancy attacks [66]. For example, Rodler et al. [67] protect existing smart contracts on Ethereum in a backwards compatible way based on run-time monitoring and validation; Das et al. [68] propose Nomos, a reentrancy-aware language that enforces security using resource-aware session types; Cecchetti et al. [64] 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 DEX.

b) Other middleware-layer attacks: On the middleware layer, there are many other attacks and threats that can affect normal operations of smart contracts [69], such as replay attack [70], exception mishandling [71], integer underflow/overflow attacks [72], etc. These threats are not specifically targeted at DEX, 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 [73], Zether [74], and NeuCheck [75]. Still, to fundamentally resolve middleware-layer attacks, smart contract developers must strictly abide by software development specifications and conduct comprehensive security tests.

3) Application-layer attacks:

a) 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. 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 Attack 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 III-C4), 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 [19], Value DeFi [21] and Cheese bank [20].

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 [76].

b) Rug pull: A rug pull involves the abandonment of a project by the project foundation after collecting investor's funds [77]. One way of doing is, 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 Attack algorithm 2. DEX 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. [78] research data on scam tokens on Uniswap and confirm that rug pulls commonly find their victims through DEX.

In August 2020, an rug puller extracted 3 ETH by imitating the well-known AMPL token [79] 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 [80].

To protect themselves from being rugged, investors should exercise caution and always confirm a project's credibility before investing in its IDO [81], [82]. Usually, reputable IDOs feature high liquidity and a pool lock [77] that disables withdrawal for a fixed period, so that LPs are unable to quickly empty the pool once it has absorbed a sufficient amount of valuable assets from investors [83].

c) Frontrunning: Frontrunning is often enabled through access to privileged market information about upcoming transactions and trades [50]. 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 [50], [84]. 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 [85].

Frontrunning can also be achieved through transaction sequence manipulation (see V-A1b) 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 [50], [86], [87]. 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.

Normal exchange users can set a low slippage tolerance to avoid suffering from a price elevated by front-runners. 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 [88]. DEX can enforce transaction sequencing

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

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

to fundamentally solve frontrunning. Some exchanges, such as EtherDelta [89] and 0xProject [90], utilize centralized time-sensitive functionalities in off-chain order books [50]. In addition, transactions can specify the sequence by including the current state of the contract as the only state to execute on [50], thereby preventing some types of frontrunning attacks. Frontrunning can further be tackled by addressing privacy issues (see V-B) and transaction sequence manipulation on the infrastructure layer (see V-A1b).

d) 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 [91].

Backrunning attacks can be mitigated through anti-BDoS solutions such as A^{2}MM [92]. Theoretically, defense solutions of application-layer distributed denial-of-service (L7 DDoS) attacks [93]–[95] 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 V-B).

e) Sandwich attacks: Combining front- and back-running, an adversary of a sandwich attack places his orders immediately before and after the victim's trade transaction. The attacker uses front-running to cause victim losses, and then uses back-running to pocket benefits. While there are endless examples of sandwich attacks, Zhou et al. [86] detail two types that can occur on an AMM: an LP attacking an exchange user (see Attack algorithm 3 Sandwich LP attack), and one exchange user attacking another (see Attack algorithm 4 Sandwich price attack). The latter is particularly common. Figure 4 shows two examples of such attacks on Uniswap V2 within a time frame of 3 minutes.

Considering swap fee (see II-D2b), gas fee (see II-D2c) and slippage (see II-D3a), sandwich attacks are only profitable if the size of the target trade exceeds a certain threshold, a value that depends on the DEX's design and the pool size. DEX

can thus prevent sandwich attacks by disallowing transactions above the threshold [86]. Naturally, sandwich attacks can also be curbed by deterring either frontrunning (see V-A3c) or backrunning (see V-A3d).

f) 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 [96] to the detriment of the protocol foundation (see II-A0c).

In September 2020, Sushiswap gained $830 million of liquidity through a vampire attack [30], where Sushiswap users were incentivized to provide Uniswap LP tokens into the Sushiswap protocol for rewards in `SUSHI` tokens [29]. 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 [97]—can be employed to hinder vampire attacks.

B. Privacy concerns

Most blockchain systems are open, traceable, and transparent, which can raise severe privacy concerns to AMM-based DEX that built upon them. In this section, we introduce the privacy issues that users may face in using AMM-based DEX and discuss their possible solutions.

1) Transaction inspection: The transparency and openness of public blockchains, where most AMM-based DEX 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 [50]. For example, the aforementioned frontrunning, backrunning, sandwich attacks, transaction sequence manipulation, and block timestamp manipulation are all based on transaction inspections [98]. In fact, major AMM-based DEX is fraught with bots, constantly monitoring transactions for possible profit opportunities [99].

2) Identity tracing: Although blockchain and AMM-based DEX usually feature a certain degree of anonymity, the linkability of transactions still enables attackers to dig identities information of users [41]. Actually, with a little background knowledge (e.g., social network posts, public speak, location [100]), attackers can launch de-anonymization inference attacks to bridge virtual accounts with individuals or uncover the true identities of traders by linking the transactions of an account together and matching relevant information [101].

3) Behavioral model inference: By analyzing historical transactions of an account, attackers can also infer its behavioral model, understanding its active phase, trading frequency, or even preference. Such activity is called behavioral model inference, which not only compromises user privacy, but can also helps launch honeypot attacks [102] and phishing scams [78], [103]–[105].

Solution: The confidentiality of AMM-based DEX's pipeline can be enhanced from various angles. On the infrastructure layer, many privacy-preserving blockchain solutions have been proposed to increase confidentiality [106], [107] and anonymity [108]–[110] for transactions. These solutions can be based on Zero-knowledge proof (ZKP) [111], homomorphic encryption [112], or identity obfuscation [113], 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 effectively defend against attacks discussed in V-A3c, V-A3d, V-A3e, V-A1a, and V-A1b.

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 [114], Ekiden [115], and Submarine Commitments [8] to develop privacy-enhanced smart contracts for AMM-based DEX. On the application layer, privacy-enhanced DEX are proposed to resolve the privacy concerns. For example, P2DEX [116] harnesses multiparty computation (MPC) [117] to realize efficient privacy-preserving DEX; ZK-Swap [118], an AMM-based DEX utilizing ZK-Rollup [119] technology, not only provides users with extra privacy protections when withdrawal, deposit, and transfer of tokens by leveraging ZKP, but also significantly reduce gas fees for transactions. Furthermore on-chain privacy-preserving services and products are on the rise. For example, Blank [120], a non-custodial Ethereum browser extension wallet, offers transaction obfuscation; Enigma [121] builds a network of "secret nodes" that can perform computations on encrypted data without the necessity to expose original raw data.

However, privacy-enhanced DEX 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.

VI. Related work

A. AMM-based DEX on blockchain

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

1) Security: Qin et al. [122] conduct empirical analyses on various AMM attacks, including transaction (re)ordering and front-running, and demonstrate the profitability in performing transaction replay through a simple trading bot. Security risk in terms of attack vectors in high-frequency trading on DEX are discussed in Zhou et al. [86], and Qin et al. [123]. Flash loan attacks with the aid of AMMs on Ethereum are described in Cao et al. [124], Perez et al. [125] and Wang et al. [126]. Victor et al. [127] detect self-trading and wash trading activities on order-book based DEXs. Gudgeon et al. [128] explore design weaknesses and volatility risks in AMM DEX.

2) Privacy: Angeris et al. [129] argue that privacy is impossible with typical CFMMs and propose several mitigating strategies. Stone et al. [130] describe a protocol that

allows trustless, privacy-preserving cross-chain cryptocurrency transfers but is yet susceptible to vampire attacks.

3) Protocol mechanism: Angeris et al. [15] discuss arbitrage behavior and price stability in constant product and constant mean markets. Lo et al. [131] 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 V), Angeris et al. [15], [132] 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. [133] present a method for constructing CFMM whose portfolio value functions match an arbitrary payoff.

B. DEX and AMM in the context of market microstructure

As two core topics of market microstructure [134], decentralized exchange and market-making have been intensively covered in the discipline of financial economics long before the emergence of blockchain.

1) DEX: Existing literature primarily suggests the higher efficiency of DEX markets over centralized ones.

Perraudin et al. [135] investigate decentralized forex markets and conclude that DEX are efficient when different market makers can transact with each other and that decentralized markets are more immune to crashes than centralized ones. Nava [136] analyzes quantity competition in the decentralized oligopolistic market and suggest perfect competition can be approximated in large rather than small DEX markets. Malamud et al. [137] 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.

2) AMM: The concept of automated market making can be traced back to Hanson's logarithmic market scoring rule (LMSR) [138], [139]. LMSR has since been refined and compared to alternative market-making strategies.

Othman et al. [140] 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 DEX with AMM protocols to compensate for divergence loss (see II-D3b) experienced by LPs. Brahma et al. [141] propose a Bayesian market maker for binary markets which exhibit better convergent behavior at equilibrium than LMSR.

Jumadinova et al. [142] 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 [143] 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 [144] and was adopted by the Ethereum-based betting platform Augur [145].

Wang [144] 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. [146] analyze the market microstructure of constant-product AMMs, and predict that AMMs will be used more for low-volatility tokens.

C. 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. [147]. A formalization of the blockchain as a state machine is presented in Shorish [148]. The extension of this framework to dynamical stochastic games is presented in Zhang [149]. 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. [150] 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. [151].

VII. Conclusion

AMM-based DEX are an incredible innovation in the DeFi space sprung up by the trustless, verifiable and censorship-resistant decentralized Turing complete and global execution machine. In this paper, 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 DEX.

Acknowledgments

We thank Nazariy Vavryk for his valuable contribution to the early code base for the numeric illustration of various AMMs and insights into security breaches on AMMs. This material is based upon work partially supported by Ripple under the University Blockchain Research Initiative (UBRI).

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Ripple.

APPENDIX

A. Formulas of major AMM-based DEX

1) Uniswap V2:

a) Conservation function: The product of reserve quantity of token$_1$, r_1, and reserve quantity of token$_2$, r_2, stays constant with swapping:

$$\mathcal{C} = r_1 \cdot r_2 \tag{19}$$

b) 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:

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

c) 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 III-C3:

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

d) 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 6a illustrates the relationship between Uniswap slippage and normalized token$_1$ reserve change $\frac{x_1}{r_1}$.

e) 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 $\mathcal{C}$, and reflects the change in token$_2$'s spot exchange rate against token$_1$.

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

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

2) Uniswap V3:

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

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

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

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

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

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

b) Exchange rate:

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

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

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

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

c) Swap amount: The swap amount can be derived from the conservation function Equation 27:

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

[1] Equation 27 is equivalent to $(x + \frac{L}{\sqrt{pb}})(y + L\sqrt{pa}) = L^2$, equation (2.2) from page 2 of Uniswap v3 whitepaper [23]. This can be seen by equating their notation with ours in the following way: $L \coloneqq \frac{\sqrt{\mathcal{A} \cdot \mathcal{C}_1 \cdot \mathcal{C}_2}}{\sqrt{\mathcal{A}}-1}$, $x \coloneqq r_1$, $y \coloneqq r_1$, $pa \coloneqq \frac{\mathcal{C}_1 \cdot \mathcal{A}}{\mathcal{C}_2}$, $pb \coloneqq \frac{\mathcal{C}_1}{\mathcal{C}_2 \cdot \mathcal{A}}$.

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

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

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 6a).

e) Divergence loss: Using the intermediary results from A1e, we can easily derive V_{held}, r_1' and r_2', and subsequently V':

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

$$\begin{aligned} r_1' &= r_1^{\text{equiv}}\sqrt{1+\rho} - \frac{\mathcal{C}_1}{\sqrt{\mathcal{A}}-1} = \frac{\mathcal{C}_1\cdot(\sqrt{1+\rho}-\frac{1}{\sqrt{\mathcal{A}}})}{1-\frac{1}{\sqrt{\mathcal{A}}}} \\ r_2' &= \frac{r_2^{\text{equiv}}}{\sqrt{1+\rho}} - \frac{\mathcal{C}_2}{\sqrt{\mathcal{A}}-1} = \frac{\mathcal{C}_2(\frac{1}{\sqrt{1+\rho}}-\frac{1}{\sqrt{\mathcal{A}}})}{1-\frac{1}{\sqrt{\mathcal{A}}}} \\ V' &= V_1' + V_2' = r_1' + \frac{\mathcal{C}_1(1+\rho)r_2'}{\mathcal{C}_2} = \frac{\mathcal{C}_1(2\sqrt{1+\rho}-\frac{2+\rho}{\sqrt{\mathcal{A}}})}{1-\frac{1}{\sqrt{\mathcal{A}}}} \end{aligned} \tag{32}$$

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

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

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

$$V' = r_1' = \mathcal{C}_1 \cdot \left(\sqrt{\mathcal{A}}+1\right) \tag{34}$$

The divergence loss can thus be calculated as:

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

3) Balancer:

a) Conservation function: Balancer implements a conservation function with a weighted-product invariant (Figure 5b). Specifically, the product of reserve quantities each raised to the power of its weight stays constant with swapping:

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

b) 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:

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

c) 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 III-C3:

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

d) 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 6b illustrates the relationship between Uniswap slippage and normalized token$_1$ reserve change $\frac{x_1}{r_1}$.

e) 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, 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 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 $k \ne 2$, 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 7b, the divergence loss due to liquidity provision as opposed to holding can thus be expressed as a function of price change:

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

4) Curve:

a) 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, in order 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 [12], with hyperparameter $\mathcal{A}$ as the interpolating factor (Equation 44).[2] When $\mathcal{A} \to 0$, the conservation function boils down to a constant-product one, as with Uniswap; when $\mathcal{A} \to +\infty$, the conservation function is essentially a constant-sum one with constant exchange rate equal to 1 (Figure 5c).

$$\mathcal{A}\left(\frac{\sum\limits_{k} r_k}{\mathcal{C}} - 1\right) = \frac{\left(\frac{\mathcal{C}}{n}\right)^n}{\prod\limits_{k} r_k} - 1 \tag{44}$$

b) Spot exchange rate: Rearrange Equation 44 and let

$$Z(r_1, r_2) = \frac{\left(\frac{\mathcal{C}}{n}\right)^n}{r_1 r_2 \prod\limits_{k\neq 1,2} r_k} - 1 - \mathcal{A}\left(\frac{r_1 + r_2 + \sum\limits_{k\neq 1,2} r_k}{\mathcal{C}} - 1\right)$$

Following III-C, the spot exchange rate can be calculated as:

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

c) 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:

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

where

$$\prod_{k\neq 2}^{'} = r_1' \cdot \prod_{k\neq 1,2} r_k \quad \text{and} \quad \sum_{k\neq 2}^{'} = r_1' + \sum_{k\neq 1,2} r_k \tag{47}$$

d) Slippage: As illustrated in Figure 6c, the slippage that a Curve user experiences when swapping x_1 token$_1$ with x_2 token$_2$ can be expressed as:

$$\begin{aligned} S(x_1) &= \frac{x_1/x_2}{{}_1E_2} - 1 \\ &= \frac{\frac{x_1 \cdot \left[\mathcal{A} \cdot r_1 \cdot \prod\limits_{k} r_k + \mathcal{C} \cdot \left(\frac{\mathcal{C}}{n}\right)^n\right]}{r_1 \cdot \left[\mathcal{A} \cdot r_2 \cdot \prod\limits_{k} r_k + \mathcal{C} \cdot \left(\frac{\mathcal{C}}{n}\right)^n\right]}}{1 - \frac{\sqrt{\frac{4\mathcal{C}\left(\frac{\mathcal{C}}{n}\right)^n}{a \cdot \prod\limits_{k\neq 2}^{'}} + \left[\left(1-\frac{1}{a}\right)\mathcal{C} - \sum\limits_{k\neq 2}^{'}\right]^2} + \left(1-\frac{1}{a}\right)\mathcal{C} - \sum\limits_{k\neq 2}^{'}}{2r_2}} - 1 \end{aligned} \tag{48}$$

[2]Note that $\mathcal{A}$ here is equivalent to $A \cdot n^n$ in Curve's white paper [12].

e) 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 III-C. The divergence loss in the case of a 2-asset pool is presented in Figure 7c.

5) DODO:

a) 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 $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively. The formula Equation 49 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 $\mathcal{C}_1$, and sets ${}_1E_2$ lower than P—i.e. token$_1$ more expensive than its market value, when r_1 falls short of $\mathcal{C}_1$. Formally,

$${}_1E_2 = \begin{cases} P\left[1 + \mathcal{A}\left(\left(\frac{\mathcal{C}_2}{r_2}\right)^2 - 1\right)\right], & r_1 \geq \mathcal{C}_1 \\ P \Big/ \left[1 + \mathcal{A}\left(\left(\frac{\mathcal{C}_1}{r_1}\right)^2 - 1\right)\right], & r_1 \leq \mathcal{C}_1 \end{cases} \tag{49}$$

b) Conservation function: DODO's conservation function can be derived from its exchange formula Equation 49. In particular, the initial state of token$_1$ and token$_2$ reserves, $\mathcal{C}_1$ and $\mathcal{C}_2$ can be regarded as the two invariants of the conservation function. This aligns with the definition according to our framework (Section III), as $\mathcal{C}_1$ and $\mathcal{C}_2$ remain constant with swapping activities, but get updated with liquidity provision or withdrawal.

$$\begin{aligned} r_1 - \mathcal{C}_1 &= \int_{r_2}^{\mathcal{C}_2} P\left[1 + \mathcal{A}\left(\left(\frac{\mathcal{C}_2}{\delta}\right)^2 - 1\right)\right] d\delta \\ &= P \cdot (\mathcal{C}_2 - r_2) \cdot \left[1 + \mathcal{A} \cdot \left(\frac{\mathcal{C}_2}{r_2} - 1\right)\right], \quad r_1 \geq \mathcal{C}_1 \end{aligned} \tag{50}$$

$$\begin{aligned} r_2 - \mathcal{C}_2 &= \int_{r_1}^{\mathcal{C}_1} \frac{1 + \mathcal{A}\left(\left(\frac{\mathcal{C}_1}{\delta}\right)^2 - 1\right)}{P} d\delta \\ &= \frac{(\mathcal{C}_1 - r_1) \cdot \left[1 + \mathcal{A} \cdot \left(\frac{\mathcal{C}_1}{r_1} - 1\right)\right]}{P}, \quad r_1 \leq \mathcal{C}_1 \end{aligned} \tag{51}$$

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 5, where the DODO's conservation function curve with $\mathcal{A} \to 1$ appears identical to that of Uniswap.

c) Swap amount: The swap amount can be derived directly from the DODO conservation function (Equation 50):

$$r_1' = r_1 + x_1$$
$$r_2' = \begin{cases} \frac{\mathcal{C}_1 - r_1' + P\cdot\mathcal{C}_2\cdot(1-2\mathcal{A}) + \sqrt{[\mathcal{C}_1 - r_1' + P\cdot\mathcal{C}_2\cdot(1-2\mathcal{A})]^2 + 4\mathcal{A}\cdot(1-\mathcal{A})\cdot(P\cdot\mathcal{C}_2)^2}}{2P\cdot(1-\mathcal{A})}, & r_1' \geq \mathcal{C}_1 \\ \mathcal{C}_2 + \frac{(\mathcal{C}_1 - r_1')\cdot\left[1+\mathcal{A}\cdot\left(\frac{\mathcal{C}_1}{r_1'}-1\right)\right]}{P}, & r_1' \leq \mathcal{C}_1 \end{cases}$$
$$x_2 = r_2 - r_2' \tag{52}$$

d) Slippage: As illustrated in Figure 6d, the slippage that a DODO user experiences when swapping x_1 token$_1$ with x_2 token$_2$ can be expressed as:

$$S(x_1) = \tag{53}$$
$$\begin{cases} \frac{2\cdot(1-\mathcal{A})\cdot x_1}{r_1' - \mathcal{C}_1 + \mathcal{C}_2\cdot P - \sqrt{[\mathcal{C}_1 - r_1' + P\cdot\mathcal{C}_2\cdot(1-2\mathcal{A})]^2 + 4\mathcal{A}\cdot(1-\mathcal{A})\cdot(P\cdot\mathcal{C}_2)^2}} - 1, & r_1' \geq \mathcal{C}_1 \\ \frac{x_1}{(r_1' - \mathcal{C}_1)\cdot\left[1+A\cdot\left(\frac{\mathcal{C}_1}{r_1'}-1\right)\right]} - 1, & r_1' \leq \mathcal{C}_1 \end{cases}$$

e) 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 IV-A5).

B. 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 few examples of projects that use AMM designs under the hood.

1) Gyroscope: Gyroscope [152] 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 an amount worth of near $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 [153].

2) EulerBeats: EulerBeats [154] 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.

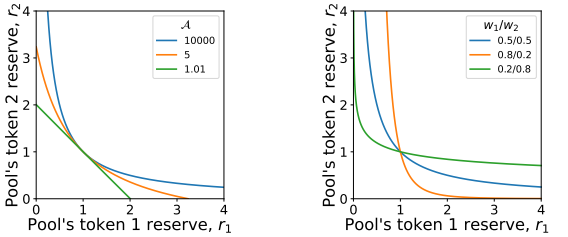
3) Pods Finance: Pods [155] is a decentralized non-custodial options protocol that allows users to create calls and or puts and trade them in the Options AMM. Users can participate as sellers and buy puts and calls in a liquidity pool or act as LPs in such a pool. The specific AMM is one-sided and built 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) Balancer Liquidity Bootstrapping Pool: Liquidity Bootstrapping Pools (LBPs) are pools where controllers can change the parameters of the pool in controlled ways, unlike immutable pools described in section Section IV. 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) [156].

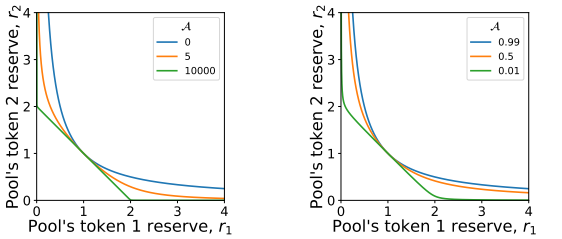
5) YieldSpace: The YieldSpace paper [157] introduces an automated liquidity provision for fixed yield tokens. A formula called the "constant power sum invariant" incorporates time to maturity as input and ensures that the liquidity provision offers a constant interest rate—rather than price—for a given ratio of its reserves. `fyTokens` are synthetic tokens that are redeemable for a target asset after a fixed maturity date [158]. 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 IV 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".

6) Notional Finance: Notional Finance [159] 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 [160]. 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.

7) Gnosis Custom Market Maker (CMM): The Gnosis CMM [161] 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, this specific application behaves more like a central limit order book than an AMM pool.

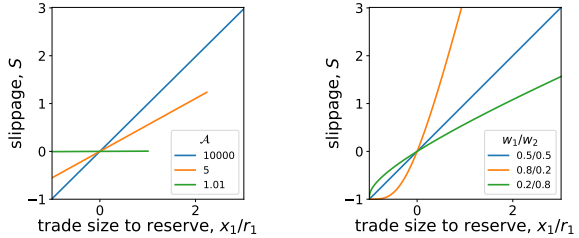


(a) Uniswap, Equation 27 (b) Balancer, Equation 36

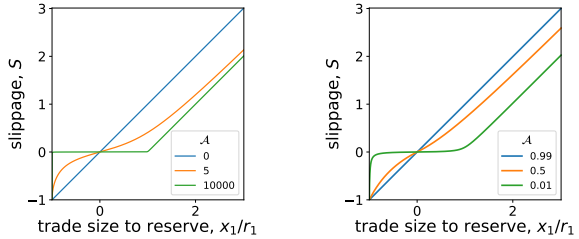


(c) Curve, Equation 44 (d) DODO, Equation 50

Figure 5: Conservation function (see III-C1) of AMMs with initial reserves of token$_1$ and token$_2$ both equal to 1, namely, $\mathcal{C} = \mathcal{C}_1 = \mathcal{C}_2 = 1$.

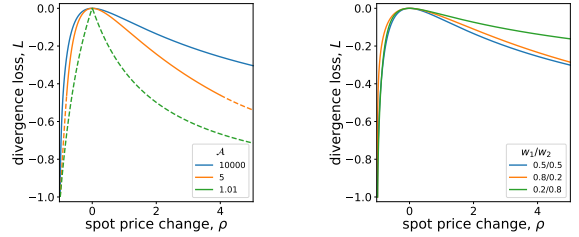


(a) Uniswap, Equation 30 (b) Balancer, Equation 39

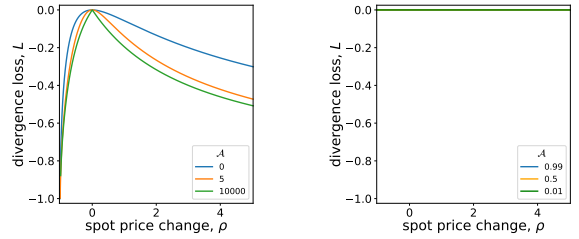


(c) Curve, Equation 48 (d) DODO, Equation 53

Figure 6: Slippage (see III-C4) of AMMs, depicted with $\frac{x_1}{r_1} \in [-1, 3]$, corresponding to the after-trade token$_1$ reserve $r_1 \in [0, 4]$, which is the x-axis of Figure 5.



(a) Uniswap, Equation 35 (b) Balancer, Equation 43



(c) Curve, Equation 18 (d) DODO, $L \equiv 0$ at equilibrium

Figure 7: Divergence loss (see III-C5) of AMMs

Table IV: Function comparison table of Uniswap, Balancer, Curve and DODO. Formulas are derived in Appendix A. Conservation functions are visualized in Figure 5, slippage functions in Figure 6 and divergence loss functions in Figure 7.

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

C. Selected attack algorithms

Attack 1 Flash-loan-funded price oracle attack

1: **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.
2: **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.
3: **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, $\frac{x_B - \Delta_2}{x_A} < \frac{x_B - \Delta_1}{x_A + \Delta_3}$.
4: **Repay the flash loan** with x_A token$_A$.

Attack 2 Rug Pull

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

Attack 3 Sandwich LP attack

1: 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.
2: LP$_B$ **observes** the mempool and sees the transaction.
3: LP$_B$ **front-runs** by withdrawing liquidity $k\,r_A$ token$_A$ and $k\,r_B$ token$_B$ with a higher gas fee $g_2 > g_1$.
4: 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$_B$.
5: LP$_B$ **back-runs** 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$_B$.
6: LP$_B$ **back-runs** by selling $(1 - \frac{(1-k)r_B - x_B}{(1-k)r_A + x_A})$ token$_B$ for some token$_A$

Attack 4 Sandwich price attack

1: User$_A$ wishes to purchase x_A XYZ whose spot price is P_1 on an AMM with gas fee g_1.
2: User$_B$ **observes** the mempool and sees the transaction
3: User$_B$ **front-runs** by buying x_B XYZ with a higher gas fee $g_2 > g_1$ on the same AMM .
4: 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.
5: User$_B$ **back-runs** by selling x_B XYZ at an average price of P'_B, with $P_2 > P'_B > P_B$ due to slippage.

GLOSSARY

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

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
DAO Decentralized Autonomous Organization
DDoS distributed denial-of-service
DeFi decentralized finance
DEX decentralized exchange
DLT distributed ledger technology
DNS domain name server
IDO initial DEX offering
IEO initial exchange offering
L7 DDoS application-layer distributed denial-of-service
LMSR logarithmic market scoring rule
LP liquidity provider
MEV miner extractable value
MPC multiparty computation
NFT non-fungible token
PMM proactive market maker
ZKP Zero-knowledge proof

REFERENCES

[1] "Defi pulse," 9 2021. [Online]. Available: https://defipulse.com/
[2] A. Evans, "Liquidity Provider Returns in Geometric Mean Markets," 6 2020. [Online]. Available: http://arxiv.org/abs/2006.08806
[3] F. Martinelli and N. Mushegian, "Balancer: A non-custodial portfolio manager, liquidity provider, and price sensor," 2019. [Online]. Available: https://balancer.finance/whitepaper/
[4] "Ethereum improvement proposals - eip-20: Token standard," 9 2021. [Online]. Available: https://eips.ethereum.org/EIPS/eip-20
[5] Bancor, "Bancor V2.1 Technical Explainer," 2020. [Online]. Available: https://drive.google.com/file/d/16EY7FUeS4MXnFjSf-KCgdE-Xyj4re27G/view
[6] CryptoLocally, "GIV Balancer Listing and Staking Rewards Updates," 2020. [Online]. Available: https://cryptolocally.medium.com/giv-balancer-listing-and-staking-rewards-updates-81ebb5843e58
[7] L. De Giglio, "Geyser: Staking Rewards For Uniswap Liquidity Providers," 2021. [Online]. Available: https://medium.com/trips-community/geyser-staking-rewards-for-uniswap-liquidity-providers-115afc6f5c07

[8] L. Breidenbach, P. Daian, F. Tramèr, and A. Juels, "Enter the Hydra: Towards principled bug bounties and exploit-resistant smart contracts," in *USENIX Security Symposium*, 2018, pp. 1335–1352. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/breindenbach

[9] DODO Team, "DODO – A Next-Generation On-Chain Liquidity Provider Powered by Pro-active Market Maker Algorithm," 2020. [Online]. Available: https://dodoex.github.io/docs/docs/whitepaper/

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

[11] Uniswap, "Flash Swaps," 2020. [Online]. Available: https://uniswap.org/docs/v2/core-concepts/flash-swaps/

[12] M. Egorov, "StableSwap-efficient mechanism for Stablecoin liquidity," 2019.

[13] D. Senchenko, "Impermanent Losses in Uniswap-Like Markets," 2020. [Online]. Available: https://dsenchenko.medium.com/impermanent-losses-in-uniswap-like-markets-4315359ea9b1

[14] Ethereum, "Types," 2020. [Online]. Available: https://docs.soliditylang.org/en/latest/types.html

[15] G. Angeris, H.-T. Kao, R. Chiang, C. Noyes, and T. Chitra, "An analysis of Uniswap markets," 2019.

[16] H. Adams, "Uniswap Whitepaper (v1)," 2018. [Online]. Available: https://hackmd.io/C-DvwDSfSxuh-Gd4WKE_ig

[17] P. Baker, "DeFi Project bZx Exploited for Second Time in a Week, Loses $630K in Ether," 2 2020. [Online]. Available: https://www.coindesk.com/defi-project-bzx-exploited-for-second-time-in-a-week-loses-630k-in-ether

[18] H. Adams, N. Zinsmeister, and D. Robinson, "Uniswap v2 Core," 2020.

[19] Harvest Finance, "Harvest Flashloan Economic Attack Post-Mortem ," 10 2020. [Online]. Available: https://medium.com/harvest-finance/harvest-flashloan-economic-attack-post-mortem-3cf900d65217

[20] B. Pirus, "Cheese Bank's multi-million-dollar hack explained by security firm," 11 2020. [Online]. Available: https://cointelegraph.com/news/cheese-bank-s-multi-million-dollar-hack-explained-by-security-firm

[21] PeckShield, "Value DeFi Incident: Root Cause Analysis," 11 2020. [Online]. Available: https://peckshield.medium.com/value-defi-incident-root-cause-analysis-fbab71faf373

[22] M. Young, "Warp Finance reportedly loses up to $8M in flash loan attack," 12 2020. [Online]. Available: https://cointelegraph.com/news/warp-finance-reportedly-loses-up-to-8m-in-flash-loan-attack

[23] H. Adams, N. Zinsmeister, M. Salem moody, u. River Keefer, and D. Robinson, "Uniswap v3 Core," 2021.

[24] 1inch Network, "Balancer Pool with STA Deflationary Token Incident," 2020. [Online]. Available: https://blog.1inch.io/balancer-hack-2020-a8f7131c980e

[25] F. Martinelli, "Introducing Balancer V2: Generalized AMMs," 2 2021. [Online]. Available: https://medium.com/balancer-protocol/balancer-v2-generalizing-amms-16343c4563ff

[26] DODOEX, "The Math Behind PMM," 2020. [Online]. Available: https://dodoex.github.io/docs/docs/math

[27] E. Hertzog, G. G. G. Benartzi, and G. G. G. Benartzi, "Bancor Protocol Continuous Liquidity for Cryptographic Tokens through their Smart Contracts," 2018. [Online]. Available: https://storage.googleapis.com/website-bancor/2018/04/01ba8253-bancor_protocol_whitepaper_en.pdf

[28] Bancor, "Announcing Bancor V2," 4 2020. [Online]. Available: https://blog.bancor.network/announcing-bancor-v2-2f56b515e9d8

[29] Sushiswap, "The SushiSwap Project," 9 2020. [Online]. Available: https://sushiswapchef.medium.com/the-sushiswap-project-dd6eb80c6ba2

[30] B. Dale, "SushiSwap Will Withdraw Up to $830M From Uniswap Today: Why It Matters for DeFi," 9 2020. [Online]. Available: https://www.coindesk.com/sushiswap-uniswap-migration-defi-amm-wars

[31] A. Bukov and M. Melnik, "Mooniswap by 1inch.exchange," 2020.

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

[33] Saber, "Saber — Solana AMM and DEX," 2021. [Online]. Available: https://saber.so/

[34] HydraDX, "Intro — HydraDX Docs," 2021. [Online]. Available: https://docs.hydradx.io/

[35] hagaetc, "Weekly dex volume," 10 2021. [Online]. Available: https://dune.xyz/queries/4323/8547

[36] DODO, "How to create a pool?" 2021. [Online]. Available: https://dodoexhelp.zendesk.com/hc/en-us/articles/900005558243-How-to-create-a-pool-

[37] B. Krishnamachari, Q. Feng, and E. Grippo, "Dynamic Automated Market Makers for Decentralized Cryptocurrency Exchange," in *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 5 2021, pp. 1–2. [Online]. Available: https://ieeexplore.ieee.org/document/9461100/

[38] Bancor Network, "FAQs - Bancor Network," 2021. [Online]. Available: https://docs.bancor.network/faqs#how-does-impermanent-loss-insurance-work

[39] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, pp. 841–853, 2020.

[40] I.-C. Lin and T.-C. Liao, "A survey of blockchain security issues and challenges." *Int. J. Netw. Secur.*, vol. 19, no. 5, pp. 653–659, 2017.

[41] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1–34, 2019.

[42] S. M. Werner, D. Perez, L. Gudgeon, A. Klages-Mundt, D. Harz, and W. J. Knottenbelt, "SoK: Decentralized Finance (DeFi)," Tech. Rep., 2021.

[43] Crypto Market Pool, "Block timestamp manipulation attack," 2020. [Online]. Available: https://cryptomarketpool.com/block-timestamp-manipulation-attack/

[44] S. Huang, "Will 2020 be The Year of DEX?" 2019. [Online]. Available: https://medium.com/@kidinamoto/will-2020-be-the-year-of-dex-ac7dfb6276e8

[45] A. M. Antonopoulos and G. Wood, *Mastering ethereum: building smart contracts and dapps*. O'reilly Media, 2018.

[46] ConsenSys, "Ethereum smart contract best practices—secure development recommendations." [Online]. Available: https://consensys.github.io/smart-contract-best-practices/recommendations/

[47] A. Mense and M. Flatscher, "Security vulnerabilities in ethereum smart contracts," in *Proceedings of the 20th International Conference on Information Integration and Web-Based Applications & Services*, 2018, pp. 375–380.

[48] P. Szalachowski, "(short paper) towards more reliable bitcoin timestamps," in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2018, pp. 101–104.

[49] "Weekly dex volume," 8 2019. [Online]. Available: https://github.com/flashbots/pm

[50] S. Eskandari, S. Moosavi, and J. Clark, "SoK: Transparent Dishonesty: Front-Running Attacks on Blockchain," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11599 LNCS. Springer, 2 2020, pp. 170–189. [Online]. Available: https://doi.org/10.1007/978-3-030-43725-1_13

[51] M. Saad, J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang, and A. Mohaisen, "Exploring the attack surface of blockchain: A systematic overview," *arXiv preprint arXiv:1904.03487*, 2019.

[52] R. Greene and M. N. Johnstone, "An investigation into a denial of service attack on an ethereum network," 2018.

[53] D. Perez, J. Xu, and B. Livshits, "Revisiting transactional statistics of high-scalability blockchains," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 535–550.

[54] M. Mirkin, Y. Ji, J. Pang, A. Klages-Mundt, I. Eyal, and A. Juels, "Bdos: Blockchain denial-of-service," in *Proceedings of the 2020 ACM SIGSAC conference on Computer and Communications Security*, 2020, pp. 601–619.

[55] L. Rembert, "The 51% attack," 2021. [Online]. Available: https://privacycanada.net/cryptocurrency/51-attack/

[56] A. Ramdas and R. Muthukrishnan, "A survey on dns security issues and mitigation techniques," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. IEEE, 2019, pp. 781–784.

[57] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 375–392.

[58] Nathaniel Popper, "A Hacking of More Than $50 Million Dashes Hopes in the World of Virtual Currency," 2016. [Online]. Available: https://www.nytimes.com/2016/06/18/business/dealbook/hacker-

may-have-removed-more-than-50-million-from-experimental-cybercurrency-project.html

[59] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making Smart Contracts Smarter," in *ACM SIGSAC Conference on Computer and Communications Security*. New York, NY, USA: ACM, 10 2016, pp. 254–269. [Online]. Available: https://dl.acm.org/doi/10.1145/2976749.2978309

[60] P. Tsankov, A. Dan, D. Drachsler-Cohen, A. Gervais, F. Bünzli, and M. Vechev, "Securify: Practical security analysis of smart contracts," *ACM Conference on Computer and Communications Security*, pp. 67–82, 10 2018.

[61] E. Albert, S. Grossman, N. Rinetzky, C. Rodríguez-Núñez, A. Rubio, and M. Sagiv, "Taming callbacks for smart contract modularity," *ACM on Programming Languages*, vol. 4, no. OOPSLA, p. 30, 11 2020. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3428277

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

[63] PeckShield, "Uniswap/Lendf.Me Hacks: Root Cause and Loss Analysis," 2020. [Online]. Available: https://peckshield.medium.com/uniswap-lendf-me-hacks-root-cause-and-loss-analysis-50f3263dcc09

[64] E. Cecchetti, S. Yao, H. Ni, and A. C. Myers, "Compositional Security for Reentrant Applications," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 5 2021, pp. 1249–1267. [Online]. Available: https://ieeexplore.ieee.org/document/9519436/http://arxiv.org/abs/2103.08577

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

[66] Y. Huang, Y. Bian, R. Li, J. L. Zhao, and P. Shi, "Smart contract security: A software lifecycle perspective," *IEEE Access*, vol. 7, pp. 150 184–150 202, 2019.

[67] M. Rodler, W. Li, G. O. Karame, and L. Davi, "Sereum: Protecting Existing Smart Contracts Against Re-Entrancy Attacks," in *Proceedings 2019 Network and Distributed System Security Symposium*. Reston, VA: Internet Society, 2019. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_09-3_Rodler_paper.pdf

[68] A. Das, S. Balzer, J. Hoffmann, F. Pfenning, and I. Santurkar, "Resource-aware session types for digital contracts," in *2021 IEEE 34th Computer Security Foundations Symposium (CSF)*. IEEE, 2021, pp. 1–16.

[69] S. Sayeed, H. Marco-Gisbert, and T. Caira, "Smart contract: Attacks and protections," *IEEE Access*, vol. 8, pp. 24 416–24 427, 2020.

[70] P. Ramanan, D. Li, and N. Gebraeel, "Blockchain-based decentralized replay attack detection for large-scale power systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[71] P. Praitheeshan, L. Pan, J. Yu, J. Liu, and R. Doss, "Security analysis methods on ethereum smart contract vulnerabilities: a survey," *arXiv preprint arXiv:1908.08605*, 2019.

[72] J. Sun, S. Huang, C. Zheng, T. Wang, C. Zong, and Z. Hui, "Mutation testing for integer overflow in ethereum smart contracts," *Tsinghua Science and Technology*, vol. 27, no. 1, pp. 27–40, 2021.

[73] Y. Zhang, S. Ma, J. Li, K. Li, S. Nepal, and D. Gu, "Smartshield: Automatic smart contract protection made easy," in *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2020, pp. 23–34.

[74] B. Bünz, S. Agrawal, M. Zamani, and D. Boneh, "Zether: Towards privacy in a smart contract world," in *International Conference on Financial Cryptography and Data Security*. Springer, 2020, pp. 423–443.

[75] N. Lu, B. Wang, Y. Zhang, W. Shi, and C. Esposito, "Neucheck: A more practical ethereum smart contract security analysis tool," *Software: Practice and Experience*, 2019.

[76] SmartContent, "TWAP Oracles vs. Chainlink Price Feeds: A Comparative Analysis," 4 2021. [Online]. Available: https://smartcontentpublication.medium.com/twap-oracles-vs-chainlink-price-feeds-a-comparative-analysis-8155a3483cbd

[77] "Rug pull," 2021. [Online]. Available: https://coinmarketcap.com/alexandria/glossary/rug-pull

[78] P. Xia, H. Wang, B. Gao, W. Su, Z. Yu, X. Luo, C. Zhang, X. Xiao, and G. Xu, "Demystifying Scam Tokens on Uniswap Decentralized Exchange," 9 2021. [Online]. Available: http://arxiv.org/abs/2109.00229

[79] Ampleforth, "Ampleforth Home Page," 2021. [Online]. Available: https://www.ampleforth.org/

[80] Etherscan, "TruAmpl (TMPL) Token Tracker," 2021. [Online]. Available: https://etherscan.io/token/0xfcb755b046ea9b9bc4586db4018b49c5a02e3d1c

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

[82] The European Business Review, "What is a "rug pull" in crypto? defi exploits explained," 7 2021. [Online]. Available: https://www.europeanbusinessreview.com/what-is-a-rug-pull-in-crypto-defi-exploits-explained/

[83] Mudra Manager, "Why Locking Liquidity is Important for Cryptocurrency," 2021. [Online]. Available: https://hackernoon.com/why-locking-liquidity-is-important-for-cryptocurrency-qv4d37hd

[84] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 910–927.

[85] D. Taylor, "Privacy first defi sienna network raises $11.2 million, takes front-running head on," 5 2021. [Online]. Available: https://tech.eu/brief/privacy-first-defi-sienna-network-raises-11-2-million-takes-front-running-head-on/

[86] L. Zhou, K. Qin, C. F. Torres, D. V. Le, and A. Gervais, "High-Frequency Trading on Decentralized On-Chain Exchanges," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 5 2021, pp. 428–445.

[87] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability," in *IEEE Symposium on Security and Privacy*, vol. 2020-May. Institute of Electrical and Electronics Engineers Inc., 5 2020, pp. 1106–1120.

[88] DeGate, "An Analysis of Ethereum Front-Running and its Defense Solutions," 5 2021. [Online]. Available: https://globalcoinresearch.com/2021/05/04/an-analysis-of-ethereum-front-running-and-its-defense-solutions/

[89] W. Warren, "Front-running, griefing and the perils of virtual settlement (part 1)," 12 2017. [Online]. Available: https://blog.0xproject.com/front-running-griefing-and-the-perils-of-virtual-settlement-part-1-8554ab283e97

[90] W. Warren and A. Bandeali, "0x: An open protocol for decentralized exchange on the ethereum blockchain," 2 2017. [Online]. Available: https://github.com/0xProject/whitepaper/blob/master/0x_white_paper.pdf

[91] livnev, "Random ordering of equally-priced transactions incentivises competitive spam," 2020. [Online]. Available: https://github.com/ethereum/go-ethereum/issues/21350

[92] L. Zhou, K. Qin, and A. Gervais, "A2MM: Mitigating Frontrunning, Transaction Reordering and Consensus Instability in Decentralized Exchanges," 6 2021. [Online]. Available: https://arxiv.org/abs/2106.07371v2

[93] Y. Feng, J. Li, and T. Nguyen, "Application-layer ddos defense with reinforcement learning," in *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*. IEEE, 2020, pp. 1–10.

[94] Y. Xie and S.-Z. Yu, "Monitoring the application-layer ddos attacks for popular websites," *IEEE/ACM Transactions on networking*, vol. 17, no. 1, pp. 15–25, 2008.

[95] C. Wang, T. T. Miu, X. Luo, and J. Wang, "Skyshield: A sketch-based defense system against application layer ddos attacks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 559–573, 2017.

[96] jakub, "What is a Vampire Attack? SushiSwap Saga Explained," 2020. [Online]. Available: https://finematics.com/vampire-attack-sushiswap-explained/

[97] W. Foxley, "Uniswap V3 Introduces New License to Spoil Future SUSHIs," 2021. [Online]. Available: https://www.coindesk.com/tech/2021/03/23/uniswap-v3-introduces-new-license-to-spoil-future-sushis/

[98] Degate, "An analysis of ethereum front-running and its defense solutions," 5 2021. [Online].

Available: https://globalcoinresearch.com/2021/05/04/an-analysis-of-ethereum-front-running-and-its-defense-solutions/

[99] D. Robinson and G. Konstantopoulos, "Ethereum is a dark forest," 8 2020. [Online]. Available: https://www.paradigm.xyz/2020/08/ethereum-is-a-dark-forest/

[100] J. DuPont and A. C. Squicciarini, "Toward de-anonymizing bitcoin by mapping users location," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, 2015, pp. 139–141.

[101] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016.

[102] K. Qin, L. Zhou, and A. Gervais, "Quantifying blockchain extractable value: How dark is the forest?" *arXiv preprint arXiv:2101.05511*, 2021.

[103] W. Chen, X. Guo, Z. Chen, Z. Zheng, and Y. Lu, "Phishing scam detection on ethereum: Towards financial security for blockchain ecosystem." in *IJCAI*, 2020, pp. 4506–4512.

[104] R. Phillips and H. Wilder, "Tracing cryptocurrency scams: Clustering replicated advance-fee and phishing websites," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2020, pp. 1–8.

[105] J. Wu, Q. Yuan, D. Lin, W. You, W. Chen, C. Chen, and Z. Zheng, "Who are the phishers? phishing scam detection on ethereum via network embedding," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[106] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 315–334.

[107] J. B. Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. T. Moreno, and A. Skarmeta, "Privacy-preserving solutions for blockchain: Review and challenges," *IEEE Access*, vol. 7, pp. 164 908–164 940, 2019.

[108] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *2013 IEEE Symposium on Security and Privacy*. IEEE, 2013, pp. 397–411.

[109] S. Noether, "Ring signature confidential transactions for monero." *IACR Cryptol. ePrint Arch.*, vol. 2015, p. 1098, 2015.

[110] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*. IEEE, 2014, pp. 459–474.

[111] O. Goldreich and Y. Oren, "Definitions and properties of zero-knowledge proof systems," *Journal of Cryptology*, vol. 7, no. 1, pp. 1–32, 1994.

[112] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 169–178.

[113] S. Goldwasser and G. N. Rothblum, "On best-possible obfuscation," in *Theory of Cryptography Conference*. Springer, 2007, pp. 194–213.

[114] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 839–858.

[115] R. Cheng, F. Zhang, J. Kos, W. He, N. Hynes, N. Johnson, A. Juels, A. Miller, and D. Song, "Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts," in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2019, pp. 185–200.

[116] C. Baum, B. David, and T. K. Frederiksen, "P2dex: privacy-preserving decentralized cryptocurrency exchange," in *International Conference on Applied Cryptography and Network Security*. Springer, 2021, pp. 163–194.

[117] R. Cramer, I. B. Damgård *et al.*, *Secure multiparty computation*. Cambridge University Press, 2015.

[118] ZKSwap, "ZKSwap home page," 2021. [Online]. Available: https://zks.org/en

[119] A. Gluchowski, "Zk rollup: scaling with zero-knowledge proofs," 2019. [Online]. Available: https://pandax-statics.oss-cn-shenzhen.aliyuncs.com/statics/1221235526992813.pdf

[120] Blank, "Blank features beyond basic privacy (#2): Protecting your IP in DeFi," 2021. [Online]. Available: https://blankwallet.medium.com/blank-features-beyond-basic-privacy-2-protecting-your-ip-in-defi-11bc76f2d67b

[121] C. Kisagun, "Preventing DEX Front-running with Enigma," 2019. [Online]. Available: https://blog.enigma.co/preventing-dex-front-running-with-enigma-df3f0b5b9e78

[122] K. Qin, L. Zhou, and A. Gervais, "Quantifying Blockchain Extractable Value: How dark is the forest?" 1 2021. [Online]. Available: http://arxiv.org/abs/2101.05511

[123] K. Qin, L. Zhou, B. Livshits, and A. Gervais, "Attacking the DeFi Ecosystem with Flash Loans for Fun and Profit," Tech. Rep., 2020. [Online]. Available: http://arxiv.org/abs/2003.03810

[124] Y. Cao, C. Zou, and X. Cheng, "Flashot: A Snapshot of Flash Loan Attack on DeFi Ecosystem," 1 2021. [Online]. Available: http://arxiv.org/abs/2102.00626

[125] D. Perez, S. M. Werner, J. Xu, and B. Livshits, "Liquidations: DeFi on a Knife-edge," in *Financial Cryptography and Data Security*, 2021. [Online]. Available: http://arxiv.org/abs/2009.13235

[126] D. Wang, S. Wu, Z. Lin, L. Wu, X. Yuan, Y. Zhou, H. Wang, and K. Ren, "Towards understanding flash loan and its applications in defi ecosystem," 10 2020. [Online]. Available: http://arxiv.org/abs/2010.12252

[127] F. Victor and A. M. Weintraud, "Detecting and Quantifying Wash Trading on Decentralized Cryptocurrency Exchanges," p. 10, 2 2021. [Online]. Available: http://dx.doi.org/10.1145/3442381.3449824

[128] L. Gudgeon, D. Perez, D. Harz, B. Livshits, and A. Gervais, "The Decentralized Financial Crisis," in *Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 6 2020, pp. 1–15. [Online]. Available: https://ieeexplore.ieee.org/document/9150192/

[129] G. Angeris, A. Evans, and T. Chitra, "A Note on Privacy in Constant Function Market Makers," 3 2021. [Online]. Available: http://arxiv.org/abs/2103.01193

[130] D. Stone, "Trustless, privacy-preserving blockchain bridges," 2021. [Online]. Available: http://arxiv.org/abs/2102.04660

[131] Y. Lo and F. Medda, "Uniswap and the rise of the decentralized exchange," 11 2020.

[132] G. Angeris and T. Chitra, "Improved Price Oracles: Constant Function Market Makers," in *Advances in Financial Technologies*. New York, NY, USA: ACM, 10 2020, pp. 80–91. [Online]. Available: https://dl.acm.org/doi/10.1145/3419614.3423251

[133] G. Angeris, A. Evans, and T. Chitra, "Replicating Market Makers," 3 2021. [Online]. Available: http://arxiv.org/abs/2103.14769

[134] M. B. Garman, "Market microstructure," *Journal of Financial Economics*, vol. 3, no. 3, pp. 257–275, 6 1976. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/0304405X76900064

[135] W. Perraudin and P. Vitale, "Interdealer Trade and Information Flows in a Decentralized Foreign Exchange Market," in *The Microstructure of Foreign Exchange Markets*. University of Chicago Press, 1996, pp. 73–106.

[136] F. Nava, "Efficiency in decentralized oligopolistic markets," *Journal of Economic Theory*, vol. 157, pp. 315–348, 5 2015. [Online]. Available: www.sciencedirect.comwww.elsevier.com/locate/jet

[137] S. Malamud and M. Rostek, "Decentralized Exchange," *American Economic Review*, vol. 107, no. 11, pp. 3320–3362, 11 2017. [Online]. Available: https://pubs.aeaweb.org/doi/10.1257/aer.20140759http://pubs.aeaweb.org/doi/10.1257/aer.20140759

[138] R. Hanson, "Combinatorial Information Market Design," *Information Systems Frontiers*, vol. 5, no. 1, pp. 107–119, 2003. [Online]. Available: http://hanson.gmu.edu

[139] ——, "Logarithmic Markets Scoring Rules for Modular Combinatorial Information Aggregation," *The Journal of Prediction Markets*, vol. 1, no. 1, pp. 3–15, 12 2012. [Online]. Available: http://www.bjll.org/index.php/jpm/article/view/417

[140] A. Othman, D. M. Pennock, D. M. Reeves, and T. Sandholm, "A Practical Liquidity-Sensitive Automated Market Maker," *ACM Transactions on Economics and Computation*, vol. 1, no. 3, pp. 1–25, 9 2013. [Online]. Available: https://dl.acm.org/doi/10.1145/2509413.2509414

[141] A. Brahma, M. Chakraborty, S. Das, A. Lavoie, and M. Magdon-Ismail, "A bayesian market maker," in *Proceedings of the ACM Conference on Electronic Commerce*. New York, New York, USA: ACM Press, 2012, p. 215. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2229012.2229031

[142] J. Jumadinova and P. Dasgupta, "A Comparison of Different Automated Market-Maker Strategies," 2012, pp. 2009–2012. [Online]. Available: http://www.cs.allegheny.edu/~jjumadinova/market-maker_AMEC.pdf

[143] C. Slamka, B. Skiera, and M. Spann, "Prediction Market Performance and Market Liquidity: A Comparison of Automated Market Makers," *IEEE Transactions on Engineering Management*, vol. 60, no. 1, pp. 169–185, 2 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6189381/

[144] Y. Wang, "Automated market makers for decentralized finance (DeFi)," 9 2020. [Online]. Available: http://arxiv.org/abs/2009.01676

[145] J. Peterson and J. Krug, "Augur: a decentralized, open-source platform for prediction markets," 2015.

[146] A. Capponi and R. JIA, "The Adoption of Blockchain-based Decentralized Exchanges: A Market Microstructure Analysis of the Automated Market Maker," 2021. [Online]. Available: https://ssrn.com/abstract=3805095

[147] M. Zargham, Z. Zhang, and V. Preciado, "A State-Space Modeling Framework for Engineering Blockchain-Enabled Economic Systems," 7 2018. [Online]. Available: http://arxiv.org/abs/1807.00955

[148] J. Shorish, "Blockchain State Machine Representation," 2018. [Online]. Available: https://osf.io/preprints/socarxiv/eusxg/

[149] Z. Zhang, M. Zargham, and V. M. Preciado, "On modeling blockchain-enabled economic networks as stochastic dynamical systems," *Applied Network Science*, vol. 5, no. 1, p. 19, 12 2020. [Online]. Available: https://appliednetsci.springeropen.com/articles/10.1007/s41109-020-0254-9

[150] M. Zargham, J. Shorish, and K. Paruch, "From Curved Bonding to Configuration Spaces," in *EEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 5 2020, pp. 1–3. [Online]. Available: https://ieeexplore.ieee.org/document/9169474/

[151] M. Zargham, K. Paruch, and J. Shorish, "Economic Games as Estimators," in *Mathematical Research for Blockchain Economy*. Springer, Cham, 2020, pp. 125–142. [Online]. Available: http://link.springer.com/10.1007/978-3-030-53356-4_8

[152] Gyroscope Finance, "Gyroscope, the new all-weather stablecoin — Gyroscope Protocol," 2021. [Online]. Available: https://gyro.finance/

[153] ——, "Gyroscope AMMs - Gyroscope Protocol," 2021. [Online]. Available: https://docs.gyro.finance/learn/gyro-amms

[154] EulerBeats, "EulerBeats — About," 2021. [Online]. Available: https://eulerbeats.com/about

[155] Pods Finance, "Pods Finance — The easiest way to hedge crypto," 2021. [Online]. Available: https://www.pods.finance/

[156] Balancer, "Liquidity Bootstrapping Pool - Balancer," 2021. [Online]. Available: https://docs.balancer.finance/guides/smart-pool-templates-gui/liquidity-bootstrapping-pool

[157] A. Niemerg, D. Robinson, and L. Livnev, "YieldSpace: An Automated Liquidity Provider for Fixed Yield Tokens," 2020. [Online]. Available: https://yield.is/Yield.pdf

[158] D. Robinson and A. Niemerg, "The Yield Protocol: On-Chain Lending With Interest Rate Discovery," Tech. Rep., 2020.

[159] Notional Finance, "Notional Finance," 2021. [Online]. Available: https://notional.finance/

[160] ——, "Notional AMM," 2020. [Online]. Available: https://docs.notional.finance/traders/technical-topics/notional-amm

[161] Gnosis, "Custom Market Maker · Gnosis Developer Portal Gnosis Protocol," 2020. [Online]. Available: https://docs.gnosis.io/protocol/docs/intro-cmm/