SoK: Decentralized Finance (DeFi)

Sam M. Werner*, Daniel Perez*, Lewis Gudgeon*,
Ariah Klages-Mundt†, Dominik Harz*‡, William J. Knottenbelt*
* Imperial College London, † Cornell University, ‡ Interlay

***Abstract*—Decentralized Finance (DeFi), a blockchain powered peer-to-peer financial system, is mushrooming. One year ago the total value locked in DeFi systems was approximately 700m USD, now, as of April 2021, it stands at around 51bn USD. The frenetic evolution of the ecosystem makes it challenging for newcomers to gain an understanding of its basic features. In this Systematization of Knowledge (SoK), we delineate the DeFi ecosystem along its principal axes. First, we provide an overview of the DeFi primitives. Second, we classify DeFi protocols according to the type of operation they provide. We then go on to consider in detail the technical and economic security of DeFi protocols, drawing particular attention to the issues that emerge specifically in the DeFi setting. Finally, we outline the open research challenges in the ecosystem.**

***Index Terms*—Decentralized Finance, DeFi, Ethereum, Cryptocurrencies,**

I. DeFi: Finance 2.0?

Consider two views on the promise of Decentralized Finance (DeFi). For the DeFi Optimist, DeFi amounts to a breakthrough technological advance, offering a new financial architecture that is non-custodial, permissionless, openly auditable, (pseudo)anonymous, and with potentially new capital efficiencies. According to this view, DeFi generalizes the promise at the heart of the original Bitcoin whitepaper [1], extending the innovation of non-custodial transactions to complex financial operations. In contrast, the DeFi Pessimist is concerned that, inter alia, the unregulated, hack-prone DeFi ecosystem serves to facilitate unfettered and novel forms of financial crime. The pseudo-anonymous nature of DeFi permits cryptocurrency attackers, scammers, and money launderers to move, clean, and earn interest on capital. A critical part of the debate between the DeFi Optimist and the DeFi Pessimist, but outside of the scope of this paper, is *moral* in nature. Rather, in this SoK, we seek to synthesize and evaluate the technical innovations of DeFi, allowing newcomers to the field to discover the essential features and problems of the DeFi terrain.

DeFi, in its ideal form, exhibits four properties. First, DeFi is *non-custodial*: participants have full control over their funds at any point in time. Traditional finance is based on a custodial model: banks hold custody of funds, stocks are held at a custodian bank, and collateral of contracts may be held in escrow accounts by a custodian. For better or worse, these custodians have to be trusted and they need to be compensated for their custodial services. In contrast, blockchain mechanisms provide a means for agents who do not trust each other to cooperate without requiring trusted third parties. Holding on-chain assets can be done without a custodian, and general scripting functionality ("smart contracts") can execute deterministically and verifiably on an underlying blockchain. Among many uses, this allows collateral to be escrowed on-chain without a custodian, which opens up a variety of non-custodial applications.

Second, DeFi is *permissionless*: anyone can interact with financial services without being censored or blocked by a third party. Third, DeFi is *openly auditable*: anyone can audit the state of the system, for example to verify that it is fully collateralized/healthy. Fourth, Defi is *composable*: financial services can be arbitrarily *composed* such that new financial products and services can be created similar to how one is able to conceive new Lego models based on a few basic building blocks. This allows capital to be seamlessly rehypothecated while following the protocol collateralization rules.

DeFi has grown rapidly, going from around 600m USD in total value locked (TVL) at the start of 2020 to over 51bn USD as of April 2021, with the most capitalized use cases being collateralized lending, constituting c.48% of the TVL, and decentralized exchange (DEXs), constituting c.38% of the TVL as of April 2021 [2]. In turn this rise led to the 24 hour volume on a decentralized cryptoasset exchange [3], overtaking that of a major centralized cryptoasset exchange [4] for the first time [5].

Yet, as with any nascent technology, DeFi is not without its risks. In the last year alone, DeFi has experienced more than 20 major protocol exploits, resulting in a loss of funds amounting to over 130m USD [6]. An apparent willingness of market participants to take large financial risks coupled with the possibility of any actor writing unaudited and even malicious smart contracts—precisely due to the decentralized nature of such technologies—renders the risks particularly acute. Moreover, due in part to the emergent complexity of smart contracts once composed together, there are even a number of instances (e.g., [7], [8], [9], [10], [11]) of audited protocols being exploited, rendering the audit process an imperfect defence against exploits. Even at the technical level, the blockchains underlying DeFi are facing significant challenges. Blockchain transaction fees have risen considerably during periods of congestion, with the fees for relatively simple smart contract operations running into the hundreds of dollars. Rising transaction costs price out small transactions, in turn restricting the set of transaction types for which the layer-one blockchain can be used.

This Work: We outline the primitives for DeFi in Sec. II and then make the following contributions:

- **Protocol Systematization:** We systematize the existing

DeFi protocols according to six types of operations (Sec. III).

- **Technical Security:** We define technical security in the context of DeFi as a risk-free earning potential and classify the set of technical attacks into three distinct categories. Technical security challenges such as smart-contract vulnerabilities serve to undermine the soundness of the ecosystem, limiting the extent to which it can be entrusted with funds (Sec. IV).
- **Economic Security:** We define economic security in the context of DeFi as secure incentive alignment of agents and organize the set of economic attack vectors into four distinct categories. The economic security risks emerge as the incentive mechanisms encoded in the underlying smart contracts make contact with reality (Sec. V).
- **Holistic Security:** The distinction between technical and economic security illustrates that the development of the DeFi ecosystem is on two fronts. We synthesize insights from both these perspectives and propose a set of six primary open research challenges for DeFi going forward (Sec. VI).

II. DeFi Primitives

DeFi protocols require an underlying distributed ledger such as a blockchain, a peer-to-peer distributed append-only record of transactions. We take the underlying distributed ledger layer solely as an input into DeFi and refer the reader to existing work (notably [12], [13], [14], [15]) for a fuller exposition of the blockchain layer itself. We assume that the ledger has the basic security properties of consistency, integrity and availability [16]. Without these security properties, DeFi protocols built on top of such a ledger would themselves become inherently insecure.

In this section, we draw attention to and outline the essential features of the underlying blockchain layer which have particular relevance to the security of DeFi protocols.

A. Smart Contracts

The most important provision is that the underlying ledger offers the ability to use smart contracts. These are programs that encode a set of rules for processing transactions which are enforced by a blockchain's consensus rules, allowing for trustless economic interactions between parties. Smart contracts rely on blockchains that are transaction-based state machines, whereby an agent can interact with smart contracts via transactions. Once a transaction is confirmed, the contract code is run by all nodes in the network and the state is updated. The underlying cost to state updates comes in the form of transaction fees charged to the sender. For instance, the Ethereum Virtual Machine (EVM) [17] on the Ethereum [18] blockchain is a stack machine which uses a specific set of instructions for task execution. The EVM maintains a fixed mapping of how much gas, an Ethereum-specific unit that denominates computational cost, is consumed per instruction. The total amount of gas consumed by a transaction is then paid for by the sender [19], [20].

In order for DeFi protocols to function on top of them, smart contracts must:

- be expressive enough to encode protocol rules
- allow conditional execution and bounded iteration
- be able to communicate with one-another within the same execution context (typically a transaction)
- support atomicity, i.e., a transaction either succeeds fully (state update) or fails entirely (state remains unaltered), such that no execution can result in an invalid state

In relation to DeFi, the most notable property of smart contracts is that they are able to call each other via message calls. This makes possible *composability*: smart contracts can be snapped together like Lego bricks ("Money Lego" [21]), with the possibility of building complex financial architectures. This is similar to as was envisaged in [22]. While promising, the side-effects of smart contracts interactions and the space of all possible interactions is vast. In a setting focused on financial applications, such complexity brings with it a great burden to understand the emergent security properties of composed smart contracts. We discuss this in more detail in Sections IV and V.

B. Tokens

A common use of smart contracts is to implement *tokens*, which can be used to represent assets, ranging from Ether [23] and other cryptoassets [24] to synthetic assets or derivatives [25], as well as provide some utility, such as the right to participate in an election. Tokens are implemented by contracts adhering to a standard token interface, allowing protocols to easily handle different tokens without having to know about their implementation in advance. In Ethereum, tokens are usually implemented via the standardized ERC-20 [26] and ERC-721 [27] interfaces for fungible and non-fungible tokens, respectively [28], although other token standards exist [29], [30], [31]. A common distinction is between fungible tokens, which are interchangeable [26], and non-fungible tokens which are distinct [27].

C. Transaction Execution

When a blockchain network participant wishes to make a transaction, the details of the unconfirmed transaction (e.g., transaction cost, sender, recipient, data input) are first broadcast to a network of peers, validated, and then stored in a waiting area (the *mempool* of a node). This mempool is then propagated among the network nodes. Participants of the underlying ledger responsible for ensuring consensus, *miners*, then choose which transactions to include in a given block, based in part on the transaction fee attached to each transaction. Transactions in a block are executed sequentially in the order in which the miner of the respective block included them. For a detailed treatment of how this process works, we refer the reader to [1], [17], [32], [33].

Miners' ability to control the sequence in which transactions are executed means that miners can order transactions in ways that will earn them revenues. In addition, they can insert their own transactions to extract further revenues. Miners can

even be bribed to undertake such transaction re-ordering [34], [35]. The value that miners can extract is known as *Miner Extractable Value* (MEV) [36]. We consider these issues in detail in Sec. V-B.

D. Keepers

Protocols may rely on their on-chain state being continually updated for their security. In transaction-based systems, updating the on-chain state requires transactions that are triggered externally. Since smart contracts are not able to create transactions programmatically, protocols must rely on external entities to trigger state updates. These entities, *keepers*, are generally financially incentivized to trigger such state updates. For instance, if for whatever reason a protocol requires a user's collateral to be automatically liquidated under certain conditions, the protocol will incentivize keepers to initiate transactions to trigger such liquidation.

E. Oracles

An oracle is a mechanism for importing off-chain data into the blockchain virtual machine so that it is readable by smart contracts. This includes off-chain asset prices, such as ETH/USD, as well as off-chain information needed to verify outcomes of prediction markets. Oracles are relied upon by various DeFi protocols (e.g. [37], [38], [39], [40], [41]).

Oracle mechanisms differ by design and their risks, as discussed in [42], [43]. A centralized oracle requires trust in the data provider and bears the risk that the provider behaves dishonestly should the reward from supplying manipulated data be more profitable than from behaving honestly. Decentralized oracles offer an alternative. As the correctness of off-chain data is not verifiable on-chain, decentralized oracles tend to rely on incentives for accurate and honest reporting of off-chain data. However, they come with their own shortcomings. We provide a detailed overview of oracle manipulation risks and on the shortcomings of on and off-chain oracles in Sections IV-B and V-D.

F. Governance

Governance refers to the process through which a system is able to effect change to the parameters which establish the terms on which interactions between participants within the system take place [42]. Such changes can be performed either algorithmically or by agents. While there is existing work on governance in relation to blockchains more broadly (e.g. [44], [45], [46]), there is still a limited understanding of the properties of different mechanisms that can be used both for blockchains and DeFi.

Presently, a common design pattern for governance schemes is for a DeFi protocol to be instantiated with a benevolent dictator who has control over governance parameters, with a promise made by the protocol to eventually decentralize its governance process. Such decentralization of the governance process is most commonly pursued through the issuance of a governance token (e.g. [47], [48], [49], [50]), an ERC-20 token which entitles token holders to participate in protocol governance via voting on and possibly proposing protocol updates. We return to governance in Sec. V.

III. DeFi Protocols

We now present DeFi protocols categorized by the type of operation they provide. An overview is shown in Figure 1, while a classification of a selection of existing DeFi protocols is given in Appendix A.

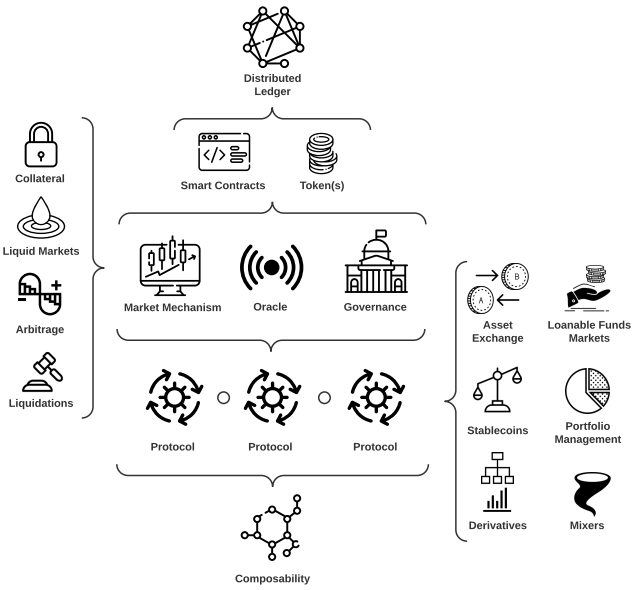

Fig. 1: A conceptual overview of the different constructs within the DeFi ecosystem.

A. On-chain Asset Exchange

Decentralized exchanges (DEXs) [51], [52] are a class of DeFi protocol that facilitate the non-custodial exchange of on-chain digital assets. Apart from being non-custodial, i.e., the exchange does not have ownership over a user's funds at any point in time, a DEX settles all trades on-chain, thereby ensuring public verifiability for all transactions to network participants. While DEXs initially only supported assets native to the chain on which they operate, wrapped tokens, such as wBTC [24] (wrapped Bitcoin), and novel cross chain solutions [53], [15], [54], [55] have enabled DEXs to overcome this limitation. Today, based on the mechanism for price discovery, DEXs come in different variants, such as *order book DEXs* (including individual [56], [57] and batch settlement [58], [59]) and *automated market makers* (AMMs) (e.g., [60], [61], [62]). Due to their widespread adoption and novelty in DeFi, we specifically focus on AMMs.

In traditional finance, market makers are liquidity providers that both quote a bid and ask price, selling from their own book, while making a profit from the bid-ask spread. Optimal market making strategies quickly become sophisticated optimization problems. In contrast, AMMs provide liquidity

algorithmically through simple pricing rules with on-chain liquidity pools in place of order books. AMMs have been studied in algorithmic game theory, e.g., logarithmic market scoring rule (LMSR) [63] in prediction markets. While they have largely remained unimplemented in traditional finance, they have become popular in DeFi for a several reasons: (1) they allow easy provision of liquidity on minor assets, (2) they allow anyone to become a market maker, even if the market making returns are suboptimal, (3) AMM pools can be separately useful as automatically rebalancing portfolios.

In an AMM liquidity pool, reserves for two or more assets are locked into a smart contract, where for a given pool, each liquidity provider receives newly minted liquidity tokens to represent the share of liquidity they have provided. A trade is consequently performed by trading against a smart contract's liquidity reserve for an asset, whereby liquidity is added to the reserves of one token and withdrawn from the reserves of one or more other tokens in the pool. A trading fee is retained by a liquidity pool and paid out proportionally to the amount of liquidity provided by each liquidity token holder. Liquidity providers are required to give up their liquidity tokens in order to redeem their share of liquidity and accrued fees.

With an AMM, the price of an asset is deterministic and decided by a formula, not an order book, and thus depends on the relative sizes of the provided liquidity on each side of a currency pair. If the liquidity is thin, a single trade can cause a significant fluctuation in asset prices relative to the overall market, and arbitrageurs can profit by closing the spread. Arbitrage refers to the process of buying or selling the same asset in different markets to profit from differences in price. Parties who undertake this process are *arbitrageurs*, and play a critical role in DeFi protocols. Arbitrage is used to ensure that the price for an asset on an AMM is at parity with the price on the open market. Note that as the reserve ratios for a pool's assets change as liquidity is added and withdrawn, a liquidity provider may receive a different token ratio upon withdrawing his liquidity share compared to the ratio he initially deposited. For a more focused and formal analysis of AMM design and the underlying market making mechanism, we direct the reader to [64], [65], [66], [67], [68].

Lastly, we touch on new batch settlement systems, like [59]. In Gnosis exchange, trades are matched algorithmically in periodic batches maintained by decentralized keepers. Keepers compete to solve a complicated matching problem. They submit solutions on-chain, from which the protocol executes the best solution, by some metric. If this keeper market is competitive, trades should be settled at fair prices, though issues can arise when the keeper market is not competitive [69] or if the method for choosing the best keeper solution can be gamed [70].

B. Loanable Funds Markets for On-chain Assets

Lending and borrowing of on-chain assets is facilitated through *protocols for loanable funds* (PLFs) [71], which refer to DeFi lending protocols that establish distributed ledger-based markets for loanable funds of cryptoassets by pooling deposited funds in a smart contract. In the context of a PLF, a *market* refers to the total supplied and total borrowed amounts of a token, where the available (i.e., non-borrowed) deposits make up a market's liquidity. An agent may directly borrow against the smart contract reserves, assuming the market for the token is sufficiently liquid. The cost of borrowing is given by an interest rate charged to the borrower, which is determined by a market's underlying interest rate model [72].

On PLFs, loans are generally of two forms: *over-collateralized loans* and *flash loans*. With an over-collateralized loan, a borrower is required to post collateral, i.e., provide something of value as security to cover the value of the debt, where the value of the collateral posted exceeds the value of the debt. In this way, collateralization simultaneously ensures that the lender (likely a smart contract) can recover their loaned value and provides the borrower with an incentive to repay the loan. The idea is to ensure that even if the value of the collateral relative to the debt falls considerably, there would still be sufficient collateral to cover the debt. In case the value of the locked collateral falls below some liquidation threshold, so-called liquidators, a type of keeper, are able to purchase the locked collateral at a discount and close the borrower's debt position. In a liquidation scenario, the liquidated borrower would receive the collateral minus any outstanding debt and incurred penalty charges [73].

An alternative to over-collateralized loans are flash loans. These are uncollateralized loans for the duration of a single transaction, requiring the borrower to repay the full borrowed amount plus interest by the end of the transaction. Flash loans leverage a blockchain's atomicity (i.e., the transaction fails if the loan is not repaid in the same transaction) and offer several use cases, such as decentralized exchange arbitrage and collateral swaps. However, they can also be used in attacks [74]. For a more detailed discussion and formal analysis of PLFs, we direct the reader to [71], [75].

C. Stablecoins

Non-custodial stablecoins are cryptoassets which aim to be price stable relative to a target currency, commonly the USD, and seek to achieve this via additional economic mechanisms. As of the time of writing, there are about a dozen non-custodial stablecoins, of which perhaps the most notable is MakerDAO's Dai [39], which has close to 7.80bn USD in market capitalization as of April 2021[1]. Note that custodial stablecoins, such as USDT [76] are not within the scope of DeFi, since these principally rely on a trusted third-party to operate, though they may be among the assets used in other DeFi protocols.

In the decentralized setting, the challenge for the protocol designer is to construct a stablecoin which achieves price stability in an economically secure and stable way and wherein all required parties can profitably continue to participate [42]. Price-stability is pursued via the use of on-chain collateral,

[1]Source: https://defipulse.com/. Accessed: 07-04-2021.

providing a foundation of secured loans from which the stablecoin derives its economic value.

The core components of a non-custodial stablecoin are as follows [42].

- Collateral. This is the store of primary value for a stablecoin. Collateral can be exogenous (e.g., ETH in Maker [48], where the collateral is primarily used externally to the stablecoin, endogenous (e.g., SNX in Synthetix [25], where the collateral was created to be collateral or implicit (e.g., Nubits [77], where the design lacks an explicit store of collateral.
- Agents. Agents form at least two roles in a non-custodial stablecoin: (1) risk absorption, for instance by providing collateral that is intended to absorb price risk, and (2) stablecoin users.
- Governance. A mechanism and set of parameters that governs the protocol as a whole (either performed by agents or algorithmically).
- Issuance. A mechanism to control the issuance of stablecoins against or using the collateral (either performed by agents or algorithmically).
- Oracles. A mechanism to import data external to the blockchain onto the blockchain, such as price-feeds.

See [42], [78] for a more complete discussion of stablecoin designs, models, and challenges.

D. Portfolio Management

For liquidity providers seeking to maximize their returns, liquidity allocation can be an onerous task given the complex and expansive space of yield-generating options. The management of on-chain assets can thus be automated through DeFi protocols which serve as decentralized investment funds, where tokens are deposited into a smart contract and an investment strategy that entails transacting with other DeFi protocols (e.g., PLFs) is encoded in the contract. Yield in DeFi is generated through interest (including accrued fees earned) and token rewards. For the latter, a protocol (e.g., PLF or AMM) distributes native tokens to its liquidity providers and/or users as rewards for the provision of deposits and/or protocol adoption. These protocol-native token rewards are similar to equity in the sense that they serve as a right to participate in the protocol's governance, as well as often represent a claim on protocol-generated earnings. The distribution model for token rewards in exchange for supplied liquidity may vary across protocols, yet is commonly proportional to how much liquidity an agent has supplied on a protocol. Therefore, smart contract-encoded investment strategies of on-chain assets are tailored around yield generating mechanisms of different protocols with the sole aim of yield aggregation and maximization. In practice, on-chain management of assets may range from automatic rebalancing of a token portfolio [79] to complex yield aggregating strategies [80].

E. Derivatives

Derivatives are financial contracts which derive their value from the performance of underlying assets. As of February 2021, the derivatives market represents about 51% of the entire cryptoassets trading market [81]. While about 99% of the derivative trading volume is achieved on centralized exchanges, a number of DeFi protocols have emerged which provide similar functionality. We lay out the adoption four different basic types of derivatives popular in the cryptoasset space[2]:

- Synthetic assets. In DeFi, synthetic assets typically replicate off-chain assets on-chain (e.g., the USD in protocols like Maker and Synthetix [25]). Though less used at present, another mechanism for constructing synthetic assets is to use AMMs that enact dynamic portfolio rebalancing strategies to replicate derivative payoffs. These bear a resemblance to synthetic portfolio insurance (see Ch. 13 in [82]) in traditional finance and have been explored more specifically using constant product market makers in [83], [84].
- Futures. Futures have seen little adoption in DeFi yet. Likely this is caused by the high volatility of the underlying cryptoassets making it hard to determine the risk taken by traders writing the futures.
- Perpetual Swaps. These are similar to futures, however, they have no set expiry date or settlement and were specifically created and popularized for cryptoasset markets [85]. Perpetuals allow traders to decide (typically on a daily basis, e.g., [86]) to keep the position by providing a funding transaction in case their position is underfunded. Due to the frequent price discovery, the price of perpetuals trades typically closer to the underlying in comparison to futures. Moreover, perpetuals are more capital efficient than trading the underlying itself since platforms require less than 100% collateral be posted by traders.
- Options. Currently, the DeFi market for options is very early with basic call and put options (e.g., [87], [88]). The cause for the limited adoption of options is threefold. First, current option platforms are at least 100% collateralized. In comparison to their centralized counterparts, this represents large capital inefficiency. Second, derivatives with set expiry dates like futures and options are hard to price on AMMs. Most AMM platforms (e.g., Uniswap [3]) do not account for a time dimension in the asset. This causes an issue specifically with option trading since the value of the option is subject to time decay. Possible remedies are more nuanced AMM designs like [89] that aim to incorporate such a time dimension. Also, complex value functions in the AMM like Balancer [50] allow replicating strategies that combine the underlying and a derivative into a single asset [84]. Third, options require a liquid market for efficient price discovery. Adoption will require solving the above problems to bootstrap the required liquidity that allows efficient pricing of those options.

[2]For an introduction to derivatives, we refer to reader to [82]

F. Privacy-preserving Mixers

Mixers are methods to prevent the tracing of cryptocurrency transactions. These are important to preserve user privacy, as the transaction ledger is otherwise public information; however, this also means they could be used to obscure the source of illicit funds. Mixers work by developing a "shielded pool" of assets that are difficult to trace back before entering the pool. They typically take one of two forms: (1) mixing funds from a number of sources so that individual coins can't easily be traced back to address individually (also called a "coinjoin", e.g., [90]), or (2) directly shielding the contents of transactions using zero knowledge proofs of transaction validity (e.g., [91], [92]). Mixers serve as a DeFi-like application itself and additionally as a piece that could be included within other DeFi protocols.

IV. Technical Security

We define a DeFi security risk to be *technical* if an agent can generate a risk-free profit by exploiting the technical structure of a blockchain system, for instance, the sequential and atomic execution of transactions. In current blockchain implementations, this coincides with (1) manipulating an on-chain system within a single transaction, which is risk-free for anyone, and (2) manipulating transactions within the same block, which is risk-free for the miner generating that block. By exploiting technical structure, the underlying blockchain system allows no opportunity for markets or other agents to act in the course of such exploits. We identify three categories of attacks that fall within technical security risks of DeFi protocols: attacks exploiting smart contract vulnerabilities, attacks relying on the execution order of transactions in a block, as well as attacks which are executed within a single transaction.

Technical Security

A DeFi protocol is technically secure if it is not possible for an attacker to obtain a risk-free profit, at the expense of the protocol or its users, by exploiting the technical structure of the protocol, any interacting protocols, or the underlying blockchain. A common property of technical exploits is that they occur within a single block.

An overview of past technical security exploits of DeFi protocols is given in Table I. We discuss a subset of these exploits as practical examples in the context of the attack category the exploit falls under.

A. Smart Contract Vulnerabilities

Smart contracts being at the center of any DeFi protocol, any vulnerabilities in their implementation can cause them to be at loss. Smart contract vulnerabilities have been extensively discussed in the literature [93], [94], [95] and we will therefore not give an extensive list of all the known vulnerabilities but rather focus on the one which have already been exploited in the DeFi context.

Reentrancy. A contract is potentially vulnerable to a reentrancy attack if it delegates control to an untrusted contract, by calling it with a large enough gas limit, while its state is partially modified [96]. A trivial example is a contract with a withdraw function that checks for the internal balance of a user, sends them money and updates the balance. If the receiver is a contract, it can then repeatedly re-enter the victim's contract to drain the funds. Although this attack is already very well-known, it has been successfully used several times against DeFi protocols. We briefly present two of these attacks in more detail.

dForce: One of the most prominent examples of this exploit was against the dForce protocol [97], which features a PLF, in April 2020 to drain around 25 million USD worth of funds [98]. The attacker used imBTC [99], which is an ERC-777 token [29], to perform the attack. A particularity of ERC-777 tokens, as opposed to ERC-20 tokens, is that they have a hook calling the receiver when the receiver receives funds. This means that any ERC-777 tokens will indirectly result in the receiver having control of the execution. In the dForce attack, the attacker used this reentrancy pattern to repeatedly increase their ability to borrow without enough collateral to back up their borrow position, effectively draining the protocol's funds.

imBTC Uniswap Pool: Despite the fact that Uniswap does not support ERC-777 tokens [61], an imBTC Uniswap [3] pool worth roughly 300 000 USD was drained using the reentrancy attack.

Both of these attacks show a common attack pattern in DeFi applications: identifying and exploiting attack vectors which exploit protocols' interconnectedness, where the composability risks therein are often under-examined. In practice, reentrancy vulnerabilities are generally simple to detected and fix by using static analysis tools [95], [100]. There are two main ways to prevent this vulnerability: (1) using a reentrancy guard that prevents any call to a given function until the end of its execution or (2) finalizing all the state updates before passing execution control to an untrusted contract.

Integer Manipulation. Almost every DeFi application manipulates monetary amounts in some way or another. This often involves not only adding and subtracting to balances but also converting into different units or to different currencies. We present the two most common types of integer manipulation issues.

The first issue, which has been extensively studied in the literature [101], [102], is integer over- and underflow. The EVM does not raise any exception in case of over- or underflow and without correct checks, such overflows could stay undetected until the value is used in some sort of action such as, for example, a transaction sending a token amount. This will often result in failed transactions and cause the smart contract to misbehave [94].

The second issue is unit error during integer manipulation. While unit manipulation should in principle be a trivial task, limitations in the expressivity of both the programming language and the virtual machine, as well as poor development practices have caused issues related to this type of arithmetic operations. The main language used to develop DeFi appli-

cations at the time of writing is Solidity [103], which has a limited type system and no support for operator overloading. In addition, the EVM only supports a single type, 32 byte integers, and has no built-in support for fixed-point numbers. To work around this limitation, each protocol decides on an arbitrary power of 10 to use as its base unit, often 10^{18}, and all the computations are performed in terms of this unit. However, given the limitations of the type-system, most programs elect to use exclusively 32 byte integers. Arithmetic on two units accidentally on different scales would not be caught by the compiler. These shortcomings can result in substantial losses in practice, as the following example shows:

YAM: In August 2020, the YAM protocol [104], which had locked almost 500 million USD worth of tokens in a very short period of time, realized that there was an arithmetic-related bug. Two integers scaled to their base unit were multiplied and the result not scaled back, making the result orders of magnitude too large [105], [106]. This prevented the governance to reach quorum and locked all the funds in the protocol's treasury contract, effectively locking over 750 000 USD worth of tokens [107] indefinitely.

Logical Bugs. There are a large number of exploits that are rooted in simple programming errors in the smart contracts. While logical bugs are by no means unique to smart contracts, but common to any type of software, the consequences for smart contracts, where immutability underpins the system, can be much more severe than for many other genres of software and result in unrecoverable financial losses. One such logical bug that resulted in notable financial losses to highlight the often trivial nature of the issue encountered:

bZx: In September 2020, the bZx protocol [108], a lending protocol, suffered a loss of over 8 million USD due to a trivial logic error [109], despite having been through two independent audits. The bZx protocol uses its own ERC-20 tokens, which are minted by locking collateral and repaid to redeem the locked collateral. As other ERC-20 tokens, bZx tokens allow users to transfer the tokens. However, due to a logical bug, when a user transferred tokens to themselves, the amount transferred would effectively only be added to their balance, and not correctly subtract from it, allowing a user to double his amount of tokens at will. The tokens created could then be used to withdraw funds that the attacker never owned or locked.

Although this is only a single instance of smart contract logical bugs, a large share of the other bugs found in Table I are also very simple mistakes that have been overlooked in both the development process and professional contract audits. We discuss in Section VI potential mitigation techniques to these issues.

B. Single Transaction Attacks

We refer to attacks which can be successfully executed, independent of knowing about some other pending transaction, as single transaction attacks. This category of attack is leveraging transaction atomicity and composability of smart contracts.

Governance Attacks. Protocols that implement some decentralized governance mechanisms tend to rely upon governance tokens, which empower token holders to propose and vote on protocol upgrades. Protocol upgrades come through proposals in the form of executable code, on which governance token holders vote. In order to propose protocol updates, the proposer has to hold or have been delegated a required number of governance tokens. For a protocol to be executed, a minimum number of votes is required, commonly referred to as quorum.

An attacker may obtain an amount of governance tokens sufficient to propose and execute malicious contract code and steal a contract's funds [110]. Given the ease with which large quantities of governance tokens can be obtained through flash loans from PLFs and swaps from AMMs, such attacks have been executed in practice [111].

Single Transaction Sandwich Attacks. In a single transaction sandwich attack, an attacker manipulates an instantaneous AMM price in order to exploit a smart contract that uses that price. An attacker first creates an imbalance in an AMM, exploits composable contracts which rely on the manipulated price, and then reverses the imbalance to cancel out the cost of the first step. The whole sequence can be performed atomically in a single transaction risk-free. Creating an imbalance typically requires access to a large amount of capital. In a system with flash loans/minting, all agents effectively have such access, although we stress that these attacks are still possible for large capital holders regardless of whether flash loans/minting are widespread. In practice, this type of attack has occurred multiple times [112], [113]. To protect against such manipulations, AMMs include a limit amount (or maximum slippage) that a trade can incur, though this only prevents manipulations above this amount. The severity single transaction sandwich attacks occurring in practice is highlighted by the following example:

Harvest: The most prominent single transaction sandwich attack in terms of seized funds was performed against the Harvest protocol [114]. The attacker took out a \$50m USDT flash loan from Uniswap and used part of the funds to create an imbalance in the liquidity reserves of USDC and USDT on Curve [49] (an AMM) to increase the AMM's virtual price of USDT. As the price of USDT on Curve was used as an on-chain oracle by the Harvest protocol, the attacker was able to mint Harvest LP tokens (i.e., tokens a liquidity provider receives in exchange for depositing funds into a protocol) by depositing 60.6m USDT, before reversing the imbalance on Curve and withdrawing 61.1m USDT from Harvest. The attacker was able to withdraw more USDT than deposited, as at the time of the withdrawal, the USDT price given by Curve was less than the deposit price, and therefore one Harvest LP token was worth more USDT during withdrawal. The attacker repeated this attack 32 times, draining a total of \$33.8m of the protocol's funds.

C. Transaction Ordering Attacks

In traditional finance, the act of *front-running* refers to taking profitable actions based on non-public information on

upcoming trades in a market. In the context of blockchain, front-running a transaction refers to submitting a transaction which is solely intended to be executed *before* some other pending transaction [115]. As transactions are executed sequentially according to how they have been ordered in a block, an agent may financially benefit from front-running one or more transactions, by having their transaction executed before a victim transaction. Similarly, an agent may pursue *back-running*, whereby a transaction is intended to be executed *after* some designated transaction. As the majority of Ethereum miners order transactions by their gas price [116], an agent can set a higher or lower gas price relative to some target transaction, in order to have his transaction executed before or after the target, respectively. In the case of multiple agents attempting to front-run the same transaction, front-running results in priority gas auctions (PGAs) [36], i.e. the competitive bidding of transaction fees to obtain execution priority.

We refer to attacks which involve front- and/or back-running within a single block, thereby undermining the technical security of DeFi protocols, as transaction ordering attacks. Note that an attacker does not need to be a miner in order to execute the following attacks but such attacks can be undertaken risk-free if the attacker is a miner.

Displacement Attacks. In a displacement attack, an attacker front-runs some target transaction, where the success of the attack does not depend on whether the target transaction is executed afterwards or not [115]. A simple example of such an attack would be an attacker front-running a transaction that registers a domain name [117]. A further vector for displacement attacks applies to order book DEXs, on which exchange participants are required to submit transactions to cancel existing orders. If a user submits a transaction to cancel an unfilled order due to price changes before the order could be filled, an attacker could front-run the cancel transaction and fill the order. In the context of DEXs, the success of such front-running behavior is particularly likely given the widespread existence of arbitrage bots engaging in PGAs for execution priority [36].

Furthermore, when a sender intends to to make a risk-free profit within a single transaction, it can be vulnerable to displacement attacks by *generalized* front-runners [118]. These bots parse all unconfirmed transactions in the mempool, trying to identify, duplicate, modify and lastly front-run any transaction which would result in a financial profit to the front-runner. Examples of transactions vulnerable to generalized front-runners would be reporting a bug as part of a bug bounty scheme to claim a reward [119] and trying to 'rescue' funds from an exploitable smart contract [118], [120].

Multi-transaction Sandwich Attacks. In a "sandwich attack", an attacker alters the deterministic price on an AMM prior to and after some other target transaction has been executed in order to profit from temporary imbalances in the AMM's liquidity reserves. In simple cases (e.g., Uniswap), the instantaneous AMM price is simply a ratio of AMM reserves and imbalances can be created simply by changing this ratio (e.g., by providing single-sided liquidity or performing a large swap through the AMM). This is how these AMMs are designed to work: swaps create imbalances, which, if left unbalanced, incentivize arbitrageurs to perform the reverse actions to balance the AMM pool.

An attacker may target another user's transaction (e.g., to profit from triggering large slippage in another user's swap) by trying to place adjacent transactions that set up the imbalance right before the swap and close out the imbalance right after the swap [116], [121]. This can be achieved through front-running the user's swap transaction by setting a higher gas price on the transaction creating the imbalance. By setting a lower gas price on the transaction closing the imbalance, the attacker can back-run the user's transaction and complete the attack. Note that setting high and low transaction fees does not guarantee the attack to succeed, as ultimately it is up to a transaction's miner to determine the order of execution.

A variant of this attack [116] can be performed if instead of being a liquidity taker, the attacker is a liquidity provider for the respective AMM. The attacker can front-run a victim transaction that swaps token A for token B and remove liquidity, exposing the victim to higher slippage. Subsequently, the attacker can back-run the victim transaction, and resupply the previously withdrawn liquidity. In a third transaction that swaps B for A, the attacker obtains a profit in B. A formal analysis of sandwich attacks is given in [116].

V. Economic security

We define a DeFi security risk to be *economic* if an exploiting agent can game the incentive structure of the protocol to realize unintended profit at the expense of the protocol or its users. Economic risks are inherently a problem of economic design and cannot be solved by technical means alone. To illustrate, while these attacks could be risk-free within a single transaction or block in a very poorly constructed system that allowed it, they are not solved, for example, just by adding a time delay that ensures they are not executed in the same block (e.g., flash loans used as a way to increase voting weight in governance proposals [110]).

The only way these attacks can be mitigated is by designing better protocol incentive structures. A common property of such attacks is that they are not risk-free and involve the manipulation of systems across many transactions or blocks.

Economic Security

A DeFi protocol is economically secure if the protocol aligns incentives among all interacting agents such that non-technical exploits are economically infeasible.

Economic Rationality. A central assumption in considering the class of economic security attacks is that of economic rationality. Following the standard game theoretic approach, we denote the strategy for player i as s_i. A strategy is a plan for what to do at each decision node (equivalently, information set) that the agent is aware they might reach. For example, a strategy would define what action an agent would take in the event that it finds itself in a protocol that becomes

Protocol	Loss	Audit	Attack	Date	Ref.
bZx	0.35m	✓	TX sandwich	Feb-15-2020	[122]
bZx	0.63m	✓	TX sandwich	Feb-18-2020	[123]
Uniswap	0.30m	✓	Reentrancy	Apr-18-2020	[124]
dForce	25.00m	✗	Reentrancy	Apr-19-2020	[98]
Hegic	0.05m	✗	Logical bug	Apr-25-2020	[125]
Balancer	0.50m	✓	TX sandwich	Jun-28-2020	[126]
Opyn	0.37m	✓	Logical bug	Aug-04-2020	[127]
Yam	0.75m	✗	Logical bug	Aug-12-2020	[105]
bZx	8.10m	✓	Logical bug	Sep-14-2020	[7]
Eminence	15.00m	✗	TX sandwich	Sep-29-2020	[128]
MakerDAO	-	✓	Governance	Oct-26-2020	[111]
Harvest	33.80m	✓	TX sandwich	Oct-26-2020	[10]
Percent	0.97m	✓	Logical bug	Nov-04-2020	[129]
Cheese Bank	3.3m	✓	TX sandwich	Nov-06-2020	[130]
Akropolis	2.00m	✓	Reentrancy	Nov-12-2020	[8]
Value DeFi	7.00m	✗	TX sandwich	Nov-14-2020	[112]
Origin	7.00m	✓	Reentrancy	Nov-17-2020	[11]
88mph	0.01m	✓	Logical bug	Nov-17-2020	[131]
Pickle	19.70m	✗	Logical bug	Nov-21-2020	[132]
Compounder	10.80m	✓	Logical bug	Dec-02-2020	[133]
Warp Finance	7.80m	✓	TX sandwich	Dec-18-2020	[134]
Cover	9.40m	✓	Logical bug	Dec-28-2020	[9]
Yearn	11.00m	✗	TX sandwich	Feb-05-2021	[135]
Growth DeFi	1.30m	✓	Logical bug	Feb-09-2021	[136]
Meerkat Finance	32.00m	✗	Logical bug	Mar-04-2021	[137]
Paid Network	27.00m	✗	Logical bug	Mar-05-2021	[138]
DODO	2.00m	✗	Logical bug	Mar-09-2021	[139]

TABLE I: An overview of empirical technical security exploits in DeFi protocols. The included exploits are explicitly limited to technical exploits and exclude any deliberate protocol scams that may have occurred. Note that the amount of funds seized per exploit is denominated in USD as of the time of the exploit and does not account for any losses that may have been recovered.

undercollateralized. A strategy $s_{1,i} \in \S_i$ for player i strictly dominates another strategy $s_{2,i} \in \S_i$ if regardless of the actions of other agents, strategy $s_{1,i}$ will always result in a higher payoff to the agent. Economic rationality is then defined as follows.

Economic Rationality

An agent is rational iff they will never play a strictly dominated strategy.

Moreover, common knowledge of rationality means that all agents know no agent will play a strictly dominated strategy.

While most economic security analysis ought to consider attackers who have profit-maximizing objectives, it can also be important to consider attackers with other objectives. For instance, an attacker who wishes to shut down the system may decide to attack as long as the cost is of a moderate level. In this sense, the economic security depends on system interruptions being too costly to effect.

Incentive Compatibility. *Incentive compatibility* is originally a concept from game theory (e.g., [140], but as a concept has seen some adaption in the context of cryptoeconomics and in particular DeFi.

In the cryptoeconomic setting, a mechanism is defined as incentive compatible if agents are incentivized to execute the mechanism *as intended* (see e.g. [141]).

Cryptoeconomic Incentive Compatibility

A mechanism (or protocol) is incentive compatible iff agents are incentivized to execute the game as intended by the protocol designer.

A central question in the context of incentive compatibility, considered in [42], is the sustainability of the mechanism implemented by a system (i.e., will the incentives arising from the system allow the system to be economically secure and stable long-term). In [42], for stablecoins, this is separated into a question of incentive security, which is included in our concept of economic security, and a question of economic stability, which is a further question of whether an economically secure system actually plays out to the desired equilibrium envisioned by the designers.

We primarily focus on the direct security questions in this paper; however, similar questions to economic stability apply to protocols other than stablecoins as well. For instance, when designing synthetic derivatives built using dynamic portfolios (and implemented as AMM pools), a lingering question is how well these designs can replicate the derivative payoffs under extreme conditions. As a comparison, synthetic portfolio insurance in traditional markets can break down when markets move too fast for the strategy to rebalance (See Ch. 13 in [82]). AMM pools aim to rebalance over much shorter timescales, and so may have an advantage here, but are also suboptimal in other areas of rebalancing.

A. Overcollateralization as Security

Collateralization is one of the primary devices to ensure economic security in a protocol. As outlined in Section III-B, in a trustless system without strong identities or legal recourse, overcollateralization creates the economic incentive for the loan to be repaid, or at least insures the lender against losses. As asset prices evolve over time, these systems generally allow automated deleveraging: if an agent's level of collateralization (value of collateral / value of borrowing) falls below a protocol-defined threshold, an arbitrager in the system can reduce the agent's borrowing exposure in return for a portion of their collateral at a discounted valuation. This aims to keep the system fully collateralized.

Overcollateralization is not without risks, however. For instance, as explored in [110], [142], times of financial crisis (wherein there are persistent negative shocks to collateral asset prices) can result in thin, illiquid markets, in which loans may become undercollateralized despite an automated deleveraging process. In such settings, it can become unprofitable for liquidators, a type of keeper, to initiate liquidations. Should this occur, rational agents will leave their debt unpaid as that results in a greater payoff.

Another form of deleveraging risk arises when the borrowed asset has endogenous price effects, for instance when its price is affected by other agents' decisions in the system or when it is manipulable. This is the case in non-custodial stablecoins like Dai that are based on leverage markets (Dai is created

by "borrowing" it against collateral and similarly must be returned to later release the collateral). As explored in [143], [144], such stablecoins can have deleveraging feedback effects that lead to volatility in the stablecoin itself. In regions of instability, the stablecoin will tend to become illiquid and appreciate in price (more so as they need to be purchased for liquidations), which can force speculative agents who have leveraged their positions to pay premium prices to deleverage. This causes their collateral to drawdown faster than may be expected, which makes the system in total less healthy and may lead to shortfalls in collateralization. This was later directly observed in Dai on "Black Thursday" [145]. As further discussed in [144], such a stablecoin requires uncorrelated collateral assets to be fully stabilized from such deleveraging effects as stable regions are related to submartingales (i.e., agents expect collateral asset prices to appreciate). However, current uncorrelated assets are primarily centralized/custodial, which poses a challenge for non-custodial designs.

B. Threats from Miner Extractable Value

An assumption by many blockchain protocols is that the block reward is sufficient to incentivize "correct" miner behavior. However, there are consensus layer risks should the MEV exceed the block reward. The simplest example of MEV is double spending of coins, which is commonly considered in base layer incentives. DeFi applications give rise to many new sources of MEV. For instance, (1) DEXs present atomic arbitrage opportunities between different trading pairs, as explored in [36], and (2) stablecoins built on leverage markets (like Dai) present arbitrage opportunities in liquidating leveraged positions, as explored in [143]. Similarly, other protocols, like PLFs, that utilize liquidation mechanisms also create MEV opportunities. Further, MEV can arise when miners are incentivized to re-order or exclude transactions based on cross-chain payments happening on other chains [146]. These are not exhaustive; there are additionally many other ways in which miners could manipulate DeFi protocols to extract value. It's worth noting that these are not just hypothetical concerns, they have actually been observed–e.g., [147], [148]–and shown to be sufficiently profitable, e.g., [149].

The practicality of MEV threats have been highlighted in [36], where the prevalent dangers of *undercutting* and *time-bandit* attacks are presented. In an undercutting attack [150], an adversarial miner would fork off a block with high MEV, while holding back some of the extractable value in order to incentivize other miners to direct their computational efforts towards the adversary's chain. In a time-bandit attack [36], an attacker forks from some previous block and sources *expected* MEV to increase his computational power and pursue a 51% attack until the expected MEV is realized. Hence, time-bandit attacks are a consensus layer risk and can be a direct consequence of historic on-chain actions which could profit a miner at some later point. A further threat is that miners could collude to set up more MEV opportunities over time, for instance by censoring transactions to top up collateral in crises and thus creating more liquidation events, as discussed in [143]. This is very similar to events on Black Thursday, in which mempool manipulations contributed to inefficient liquidation auctions in Maker [147].

C. Governance Risks and Governance Extractable Value

Protocol governance often introduces means to update system parameters and even redefine entire contracts. In many cases, this may be a necessary component for the system to evolve over time. However, governance can also introduce manipulation vectors that affect security. Govenance of a DeFi protocol is typically tied to holders of governance tokens, which can often be thought of as shares in the protocol. In systems where there is large flexibility for governance to change the system, an important question is where governance token value comes from. A typical aim is for the protocol to incentivize good stewardship from its governance token holders by compensating governance with cashflows from the system. In this case, governance token value is derived from future discounted cashflows. Another possibility is that governance is directly aligned with underlying users–e.g., because they are the same.

However, if these incentives aren't of sufficient size, then it may be more profitable for governance token holders to extract value in less desirable ways, which we term *governance extractable value* (GEV).[3] An example of GEV is for governors to effect changes to the protocol in ways that provide outside benefits to themselves but may be harmful to the overall system health. For instance, the Cream protocol governance added high levels of very risky collateral assets that they had an outside stake in, arguably to their benefit but against the interests of the protocol [151]. GEV also includes explicit governance attacks. A hypothetical GEV attack to indirectly extract collateral value is described in [152]. In cases like these, governance may not be incentive compatible. And if the value of governance tokens from incentive compatible sources crashes, the region of incentive compatibility also shrinks, and it may become profitable for a new coalition of governors to form to attack the protocol. This is increasingly problematic given the ease and low cost with which governance tokens may be obtained via flash loans and PLFs. Other complications arise in the need to protect minority rights within the protocol–e.g., building in limitations so that a majority of governors can't unilaterally change the game to, for instance, steal all value of the other minority or users.

The capital structure-like models developed in [42] can be applied more generally to DeFi protocols to model governance security and incentive compatibility around these issues. As can be understood in those models, these issues essentially arise because there may not be outside recourse (e.g., legal) in the pseudo-anonymous setting to disincentivize attacks and manipulations compared to the (idealized) traditional finance setup. Further, [42] conjectures that in the case of a fully decentralized stablecoin with multiple classes of interested

[3]GEV may be interpreted as a generalization of MEV with miners being a specific type of governor tasked with ordering transactions on the base layer.

parties and with a high degree of flexibility for governance design, there exists no long-term incentive compatible equilibrium. Intuitively, there are resulting costs of anarchy in such systems, which can be too much to bear. In such a case, rational agents would choose not to participate. However, they also conjecture that other DeFi systems, such as DEXs, may have wider incentive compatibility in similar situations due to the different structure of such systems.

D. Market and Oracle Manipulation

As the suppliers of off-chain information, oracles pose a fundamental component of DeFi protocols, particularly for sourcing price feeds [153]. However, it is important to distinguish between (1) a price that is manipulated yet correctly supplied by an oracle and (2) an oracle itself being manipulated. While we present each form of manipulation, note that the latter can be essentially modeled as a separate governance-type risk as discussed in [42].

1) Market Manipulation: We wish to quantify economic risks stemming from price manipulations in underlying markets while assuming the oracle follows a best practice implementation and is non-malicious. An adversary may manipulate the market price (on-chain or off-chain) of an asset over a certain time period if a profit can be realized as a consequence of the price manipulation–e.g., by taking positions in a DeFi protocol that uses that market price as an oracle. As discussed in the Section IV, instantaneous AMM prices are easily manipulable with near zero cost and, as a result, should not be used as price oracles. Market manipulation problems persist even when we assume the oracle is not an instantaneous AMM price. In this case, there is a cost to market manipulation related to maintaining a market imbalance over time, whether in an AMM (e.g., to manipulate a time-weighted average price) or through filling unfilled orders in an order book. Depending on whether the market for an asset is thick or thin, the cost for an attacker to significantly change the asset's price will be higher or lower, respectively. An example of such an attack would be to trigger liquidations by manipulating an asset's price, as discussed in the context of stablecoins in [143]. An attacker could profit either by purchasing liquidated collateral at a discount or shorting the collateral asset by speculating on a liquidation spiral. Such attacks are similar to short-squeezes in traditional markets. However, unlike with single transaction sandwich attacks, the aforementioned attack is not risk-free and could bring substantial losses to the attacker should it fail. In particular, markets and agents may react to such attacks in unpredictable ways.

To illustrate the potential of such attacks, the stablecoin DAI, which historically has thin liquidity, traded at a temporary price of $1.30 over a course of about 20 minutes on Coinbase Pro, a major centralized cryptoasset exchange, before returning to its intended $1 peg [154]. As a result, the Compound Open Price Feed [155], a cryptoasset price oracle which is in part based on prices signed by Coinbase, reported a DAI price of $1.23 to Compound for a short period of time. This incident triggered (arguably wrongful) liquidations on collateral worth approximately $89m, costing the liquidated Compound borrowers 23% (from the imbalanced DAI price) plus an additional 5% (the Compound liquidation incentive, i.e., the discount at which collateral is sold at during a liquidation) on their liquidated assets.

2) Oracle Manipulation: Centralized oracles serve as a single point of failure and despite trusted execution environments [156] they remain vulnerable to the provider behaving maliciously if incentives are sufficient for manipulating the source of a data feed. Decentralized price oracles may use on-chain data, most notably on DEXs (specifically AMMs) for crypto-to-crypto price data. However, as outlined in Section IV-B, prices may be manipulable through intentionally created imbalances and thinly traded markets, even after remedying the technical security issues using, for instance, time-weighted average prices. Furthermore, on-chain DEX oracles inherently can not price off-chain assets and fiat currencies. For instance, cryptoasset prices may be quoted in stablecoins through DEX oracles, but this faces the same inherent problem: we then rely on that stablecoin, which may be manipulated or fail, for the data feed.

As discussed in [42], decentralized oracle solutions for off-chain data exist. However, they are yet imperfect solutions, tending to rely on Schelling point games, in which agents vote on the correct price values and are incentivized against having their stake slashed if their vote deviates from the consensus. Tying incentives to consensus, when the correctness of the consensus decision is not objectively verifiable (as in this case), paves a vector for game theoretic attacks, like in Keynesian beauty contests. Widely used decentralized oracles, such as Chainlink [157], try to mitigate this problem by aggregating data feeds from multiple sources (e.g., by calculating the median) and relying on reputation systems to curate reliable sources. These systems may still suffer from similar game theoretic issues, however.

VI. Open Research Challenges

There are many open research challenges in DeFi stemming from the technical and economic security issues presented in Sections IV and V.

A. Composability Risks

Cryptoassets can be easily and repeatedly tokenized and interchanged between DeFi protocols in a manner akin to rehypothecation. This offers the potential to construct complex, inter-connected financial systems, yet bears the danger of exposing agents to composability risks, which are as of yet mostly unquantified. An example of composability risk is the use of flash loans for manipulating instantaneous AMMs and financially exploiting protocols that use those AMMs as price feeds. This has repeatedly been exploited in past attacks (e.g. [10], [158], [130]). Many protocols still struggle to implement sufficient protective measures for addressing this risk.

The breadth of composability risks spans far beyond the negative externalities stemming from instantaneous AMM

manipulations. For instance, there remain open questions about the consequences of the following types of exploitations on connecting systems: the accumulation of governance tokens to execute malicious protocol updates, the failure of non-custodial stablecoin incentives to ensure price stability, and failure of PLF systems to remain solvent. Note, however, that this list is far from exhaustive. These become increasingly important issues as more complex token wrapping structures [159] stimulate higher degrees of protocol interconnectedness. For example, the use of PLF deposit tokens (as opposed to the tokens in their original forms) within AMM pools and strategies to earn yield on underlying assets through leverage by borrowing non-custodial stablecoins and depositing into PLFs or AMMs.

Recent works [71], [160] begin to explore protocol interdependence, while [161] propose a process-algebraic technique that allows for property verification by modeling DeFi protocols in a compositional manner. Nonetheless, a critical gap in DeFi research toward taxonimizing and formalizing models to quantify composability risks remains. This problem is elevated as a holistic view on the integrated protocols is necessary: failures might arise from both technical and economic risks. Ensuring safety of protocol composition will be close to impossible for any protocol designer and forms a major challenge for DeFi going forward.

B. Governance

We identify important research directions in governance and GEV. A general direction is modeling incentive compatibility of governance in various systems with GEV. For instance, setting up models, finding equilibria, and understanding how other agents in the system respond. The models in [42] get this started in the context of stablecoins and additionally discuss how to extend to other DeFi protocols. There is moreover a range of discussions around simulating and formalizing governance incentives through tools like cadCAD [162].

There remain unanswered questions with regards to the general design of governance incentives. For instance, how to structure governance incentives to reward good stewardship: e.g., intrinsic vs. monetary reward, reward per vote vs. reward per token holder, and measures of good stewardship. Furthermore, there lies potential in formally evaluating protection of minority agents in systems with flexible governance and large GEV.

For systems utilizing governance tokens, we identify research gaps rooted in security risks of the ability to borrow governance tokens via flash loans and PLFs. Specifically, we identify opportunities to formally explore how (1) technical security can be compromised and (2) from an economic security point of view, incentive compatibility is further complicated by the borrowing of governance tokens.

C. Oracles

We highlight a few open challenges about oracle design and security. Note that, in many cases, the oracle problem can also be directly related to the governance problem, as typically governors are tasked with choosing the oracles that are used.

A more general open challenge lies in how to structure oracle incentives to maintain incentive compatibility to report correct prices. This is similar to governance design in some ways and needs to take into account the possible game theoretic manipulations that could be profitable.

We identify a further research opportunity in designing and evaluating the security of various oracle strengthening methods. While there exist several works examining oracle designs on both a general and empirical basis–e.g. [43], [153]–a formal security analysis of, e.g., medianizers, reputation systems, and grounding reported prices based on on-chain verifiable metrics is yet to be done.

D. Miner Extractable Value

We identify important research directions in MEV.

While research on MEV and the extraction of it is being put forward [36], methodologies to quantify negative externalities of MEV–e.g., from wasted gas per block, upward gas price pressure–and the full extent of MEV opportunities remain scarce. For the latter, we conjecture that the miner's problem to optimize the MEV they extract in a block is NP-hard and additionally hard to approximate. To support this, it is quite easy to reduce a simplified version of the problem, in which the MEV of each transaction is fixed, to the knapsack problem. Note that while the knapsack problem is NP-hard, it is easy to approximate. In fact, we expect a more realistic version of the miner's problem to be harder than knapsack because the transaction ordering the miner chooses also changes the MEV of the transactions (i.e., swapping two elements might change their weight in knapsack).

There are interesting questions regarding how the emergence of MEV opportunities endogenously affects agents' behavior within DeFi protocols. Models for this are started in the context of stablecoins in [42].

A further open challenge remains with respect to designing protective mechanisms against (1) consensus layer instability risks that are induced by high MEV incentives and (2) time bandit attacks that seek to rewrite the recent transaction history–for example, which could aim to trigger and profit from increased protocol liquidations. On this point, [143] suggests that oracle price validity could be tied to recent block hashes to prevent such reorderings from extracting the protocol value, though potentially with costs to the economic security of the protocol in other ways.

E. Program Analysis

There exists a large amount of work [163], both in academia [100], [95], [164] and industry [165], [166], to analyze smart contract bugs and vulnerabilities. While smart contracts analysis tools keep improving, the number and scale of smart contract exploits are showing no sign of decrease and are, on the contrary, becoming more frequent. Although program analysis tools are no silver bullet and cannot prevent all exploits, Table I and the discussed exploits in section IV

hint that there are some recurring patterns that could be automatically detected and prevented. We argue that improvements in program analysis could prevent many of the exploits we have seen.

Current program analysis tools can mainly be divided into two categories: (1) fully automatic tools checking for program invariants and (2) semi-automated verification tools checking for user-defined properties [164], [167], [168]. While the latter allows to verify business logic in ways that are not fully automatable, they are typically non-trivial to setup and require knowledge of software verification, which limits their use to projects with enough resources. On the other hand, fully automatic tools, which can be very easily setup and ran, usually focus on checking properties of a single contract in isolation [95], [101], [166], [169], such as unchecked exceptions or integer overflows. However, they have not evolved yet to embrace the composable nature of smart contracts, which makes it impossible for such tools to reason about scenarios where the issue happens due to a change in something external to the smart contracts, such as a sudden change in a price returned by an oracle. Further, most tools reason very little about *semantic* properties of the smart contracts, such as how a particular execution path can influence ERC-20 token balances. We believe that improvements in these areas will allow auditors and developers to analyze and deploy their contracts with more confidence, reducing the number of technical security exploits.

F. Anonymity and Privacy

The anonymity and privacy of DeFi protocols is at present a significantly understudied area. There is a tension between user's privacy being valuable in itself, while at the same time helping malicious users to escape the consequences of their actions. At present, a large proportion of DeFi transactions occurs in protocols built on Ethereum, wherein agents at best have pseudoanonymity. This means that if an agent's real-world identity can be linked to an on-chain address, all the actions undertaken by the agent through that address are observable. While recent advances in zero-knowledge proofs [170], [171] and multi-party computations [172], [173] hold many promises, these technologies are yet to gain traction in the context of DeFi. One of the main friction points is the large computational cost of these technologies, which make them very expensive to use and deploy in the context of DeFi. A decrease of computational cost of the underlying blockchain will be key to how widely privacy-preserving technologies can be deployed by DeFi protocols.

VII. Conclusion

In this SoK we have considered DeFi from two points of view, the DeFi Optimist and the DeFi Pessimist, and examined the workings of DeFi systematically and at length. First, we laid out the primitives for DeFi before categorizing DeFi protocols by the type of operation they provide. We examined the security challenges protocols are exposed to by making a distinction between technical and economic security risks. In so doing, we were able to systematize attacks that have been proposed in theory and/or occurred in practice into categories of attacks that either rely on an agent's ability to generate risk-free profits by exploiting the technical structure of a blockchain or to game the incentive structure of a protocol to obtain a profit at the expense of the protocol. Finally, we drew the attention to open research challenges that require a holistic understanding of both the technical and economic risks.

While DeFi may have the potential to creating a permissionless and non-custodial financial system, the opinion put forward by the DeFi optimist, the open technical and economic security challenges remain strong. The DeFi pessimist is, at least for now, on firm ground: solving these challenges in a robust and scalable way is a central challenge for researchers and DeFi practitioners. In the end, however, it is the blend between promise and challenge — the tension between the views of the DeFi optimist and the DeFi pessimist — that makes DeFi a worthwhile and exciting area for research.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] DeFi Pulse, "The decentralized finance leaderboard at defi pulse," 2020. [Online]. Available: https://defipulse.com/

[3] Uniswap, "Uniswap," 2020. [Online]. Available: https://app.uniswap.org/#/swap

[4] Coinbase, "Coinbase," 2020. [Online]. Available: https://www.coinbase.com/

[5] O. Godbole, "Defi flippening comes to exchanges as uniswap topples coinbase in trading volume," *CoinDesk*, 2020. [Online]. Available: https://www.coindesk.com/defi-flippening-uniswap-topples-coinbase-trading-volume

[6] DeFi Hacks, "Defi hacks," 2021. [Online]. Available: https://defihacks.wiki/

[7] P. Baker, "Defi lender bzx loses $8m in third attack this year," *CoinDesk*, 2020. [Online]. Available: https://www.coindesk.com/defi-lender-bzx-third-attack

[8] T. Wright, "Akropolis defi protocol 'paused' as hackers get away with $2m in dai," 2020, accessed: 29-12-2020. [Online]. Available: https://cointelegraph.com/news/akropolis-defi-protocol-paused-as-hackers-get-away-with-2m-in-dai

[9] K. Reynolds and D. Pan, "Cover protocol attack perpetrated by 'white hat,' funds returned, hacker claims," *CoinDesk*, 2020. [Online]. Available: https://www.coindesk.com/cover-protocol-attack-perpetrated-by-white-hat-all-funds-returned-hacker-claims

[10] Harvest Finance, "Harvest flashloan economic attack post-mortem," 2020, accessed: 29-12-2020. [Online]. Available: https://medium.com/harvest-finance/harvest-flashloan-economic-attack-post-mortem-3cf900d65217

[11] M. Liu, "Urgent: Ousd was hacked and there has been a loss of funds," 2020, accessed: 29-12-2020. [Online]. Available: https://medium.com/originprotocol/urgent-ousd-has-hacked-and-there-has-been-a-loss-of-funds-7b8c4a7d534c

[12] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "Sok: Research perspectives and challenges for bitcoin and cryptocurrencies," in *2015 IEEE symposium on security and privacy*. IEEE, 2015, pp. 104–121.

[13] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis, "Sok: Consensus in the age of blockchains," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, 2019, pp. 183–198.

[14] L. Gudgeon, P. Moreno-Sanchez, S. Roos, P. McCorry, and A. Gervais, "Sok: Off the chain transactions," *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 360, 2019.

[15] A. Zamyatin, M. Al-Bassam, D. Zindros, E. Kokoris-Kogias, P. Moreno-Sanchez, A. Kiayias, and W. J. Knottenbelt, "Sok: communication across distributed ledgers." *IACR Cryptol. ePrint Arch.*, 2020.

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

[17] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[18] V. Buterin, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, 2014.

[19] D. Perez and B. Livshits, "Broken metre: Attacking resource metering in EVM," in *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*. The Internet Society, 2020. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/broken-metre-attacking-resource-metering-in-evm/

[20] S. M. Werner, P. J. Pritz, and D. Perez, "Step on the gas? A better approach for recommending the ethereum gas price," *arXiv preprint arXiv:2003.03479*, 2020.

[21] DeFi Pulse, "What is defi?" 2019. [Online]. Available: https://defipulse.com/blog/what-is-defi/

[22] S. P. Jones, J.-M. Eber, and J. Seward, "Composing contracts: an adventure in financial engineering," *ACM SIG-PLAN Notices*, vol. 35, no. 9, pp. 280–292, 2000.

[23] R. Daniel and B. Roth, "weth — erc20 tradable version of eth," 2020. [Online]. Available: https://weth.io/

[24] W. Bitcoin, "Wbtc wrapped bitcoin an erc20 token backed 1:1 with bitcoin," 2020. [Online]. Available: https://wbtc.network/

[25] Synthetix, "Synthetix — decentralised synthetic assets," 2020. [Online]. Available: https://www.synthetix.io

[26] F. Vogelsteller and V. Buterin, "Eip-20: Erc-20 token standard," 2015. [Online]. Available: https://eips.ethereum.org/EIPS/eip-20

[27] W. Entriken, D. Shirley, J. Evans, and N. Sachs, "Eip-721: Erc-721 non-fungible token standard," 2018. [Online]. Available: https://eips.ethereum.org/EIPS/eip-721

[28] M. Fröwis, A. Fuchs, and R. Böhme, "Detecting token systems on ethereum," in *International conference on financial cryptography and data security*. Springer, 2019, pp. 93–112.

[29] J. Dafflon, J. Baylina, and T. Shababi, "Eip-777: Erc777 token standard," 2017. [Online]. Available: https://eips.ethereum.org/EIPS/eip-777

[30] W. Radomski, A. Cooke, P. Castonguay, J. Therien, E. Binet, and R. Sandford, "Eip-1155: Erc-1155 multi token standard," 2018. [Online]. Available: https://eips.ethereum.org/EIPS/eip-1155

[31] V. Minacori, "Eip-1363: Erc-1363 payable token," 2020. [Online]. Available: https://eips.ethereum.org/EIPS/eip-1363

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

[33] D. Perez, J. Xu, and B. Livshits, "Revisiting transactional statistics of high-scalability blockchains," ser. IMC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 535–550. [Online]. Available: https://doi.org/10.1145/3419394.3423628

[34] P. McCorry, A. Hicks, and S. Meiklejohn, "Smart contracts for bribing miners," in *International Conference on Financial Cryptography and Data Security*. Springer, 2018, pp. 3–18.

[35] F. Winzer, B. Herd, and S. Faust, "Temporary censorship attacks in the presence of rational miners," in *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2019, pp. 357–366.

[36] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges," *arXiv preprint arXiv:1904.05234*, 2019.

[37] R. Leshner and G. Hayes, "Compound: The money market protocol," 2019. [Online]. Available: https://compound.finance/documents/Compound.Whitepaper.pdf

[38] AAVE, "Aave: Protocol whitepaper v1.0," 2020, accessed: 13-08-2020. [Online]. Available: https://github.com/aave/aave-protocol/blob/master/docs/Aave_Protocol_Whitepaper_v1_0.pdf

[39] Maker, "The maker protocol: Makerdao's multi-collateral dai (mcd) system," accessed: 08-06-2020. [Online]. Available: https://makerdao.com/en/whitepaper/

[40] Synthetix, "Litepaper," 2020, accessed: 06-12-2020. [Online]. Available: https://docs.synthetix.io/litepaper/

[41] J. Peterson and J. Krug, "Augur: a decentralized, open-source platform for prediction markets," *arXiv preprint arXiv:1501.01042*, 2015.

[42] A. Klages-Mundt, D. Harz, L. Gudgeon, J.-Y. Liu, and A. Minca, "Stablecoins 2.0: Economic foundations and risk-based models," in *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, 2020, pp. 59–79.

[43] B. Liu and P. Szalachowski, "A first look into defi oracles," 2020.

[44] W. Reijers, F. O'Brolcháin, and P. Haynes, "Governance in blockchain technologies & social contract theories," *Ledger*, vol. 1, pp. 134–151, 2016.

[45] R. Beck, C. Müller-Bloch, and J. L. King, "Governance in the blockchain economy: A framework and research agenda," *Journal of the Association for Information Systems*, vol. 19, no. 10, p. 1, 2018.

[46] B. E. Lee, D. J. Moroz, and D. C. Parkes, "The political economy of blockchain governance," *Available at SSRN 3537314*, 2020.

[47] Compound, "Compound finance," 2019. [Online]. Available: https://compound.finance/

[48] MakerDAO, "Makerdao," 2019. [Online]. Available: https://makerdao.com/en/

[49] Curve Finance, "Curve.fi," 2020, accessed: 20-08-2020. [Online]. Available: https://www.curve.fi/

[50] Balancer Labs, "BAL – balancer governance token," 2020, accessed: 20-08-2020. [Online]. Available: https://docs.balancer.finance/protocol/bal-balancer-governance-token

[51] L. X. Lin, E. Budish, L. W. Cong, Z. He, J. H. Bergquist, M. S. Panesir, J. Kelly, M. Lauer, R. Prinster, S. Zhang *et al.*, "Deconstructing decentralized exchanges," *Stanford Journal of Blockchain Law & Policy*, 2019.

[52] Index, "Index: A comprehensive list of decentralized exchanges (dex)." [Online]. Available: https://distribuyed.github.io/index/

[53] A. Zamyatin, D. Harz, J. Lind, P. Panayiotou, A. Gervais, and W. Knottenbelt, "Xclaim: Trustless, interoperable, cryptocurrency-backed assets," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 193–210.

[54] A. Dubovitskaya, D. Ackerer, and J. Xu, "A game-theoretic analysis of cross-ledger swaps with packetized payments," 2021.

[55] J. Xu, D. Ackerer, and A. Dubovitskaya, "A game-theoretic analysis of cross-chain atomic swaps with htlcs," 2020.

[56] IDEX, "Idex 2.0: The next generation ofnon-custodial trading," *URL: https://idex.io/document/IDEX-2-0-Whitepaper-2019-10-31.pdf*, 2019.

[57] W. Warren and A. Bandeali, "0x: An open protocol for decentralized exchange on the ethereum blockchain," *URL: https://github.com/0xProject/whitepaper*, 2017.

[58] N. Beneš, "Introducing the dutchx," 2017. [Online]. Available: https://blog.gnosis.pm/introducing-the-gnosis-dutch-exchange-53bd3d51f9b2

[59] Gnosis, "Introduction to gnosis protocol," 2020. [Online]. Available: https://docs.gnosis.io/protocol/docs/introduction1/

[60] M. Egorov, "Stableswap - efficient mechanism for stablecoin liquidity," 2019. [Online]. Available: https://www.curve.fi/stableswap-paper.pdf

[61] Uniswap, "Uniswap whitepaper," 2020, accessed: 26-08-2020. [Online]. Available: https://hackmd.io/@HaydenAdams/HJ9jLsfTz#%F0%9F%A6%84-Uniswap-Whitepaper

[62] F. Martinelli and N. Mushegian, "Balancer whitepaper: A non-custodial portfolio manager, liquidity provider, and price sensor." 2019, accessed: 26-08-2020. [Online]. Available: https://balancer.finance/whitepaper/

[63] R. Hanson, "Combinatorial information market design," *Information Systems Frontiers*, vol. 5, no. 1, pp. 107–119, 2003.

[64] G. Angeris and T. Chitra, "Improved price oracles: Constant function market makers," *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, 2020.

[65] G. Angeris, H.-T. Kao, R. Chiang, C. Noyes, and T. Chitra, "An analysis of uniswap markets," *Cryptoeconomic Systems Journal*, 2019.

[66] G. Angeris, A. Evans, and T. Chitra, "Replicating market makers," *arXiv preprint arXiv:2103.14769*, 2021.

[67] Y. Zhang, X. Chen, and D. Park, "Formal specification of constant product (xy= k) market maker model and implementation," 2018. [Online]. Available: https://github.com/runtimeverification/verified-smart-contracts/blob/uniswap/uniswap/x-y-k.pdf

[68] G. Angeris, A. Evans, and T. Chitra, "When does the tail wag the dog? Curvature and market making," *arXiv preprint arXiv:2012.08040*, 2020.

[69] M. Koeppelmann, "Tweet," 18 July 2020. [Online]. Available: https://twitter.com/koeppelmann/status/1284502534208528385

[70] Gnosis, "API3 IDO incident - post mortem," 2020. [Online]. Available: https://hackmd.io/@n6YCqowrQduQ5u25wSoRXw/Hylnk7SjD

[71] L. Gudgeon, S. M. Werner, D. Perez, and W. J. Knottenbelt, "Defi protocols for loanable funds: Interest rates, liquidity and market efficiency," in *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, 2020, p. 92–112.

[72] J. Xu and N. Vadgama, "From banks to defi: the evolution of the lending market," 2021.

[73] D. Perez, S. M. Werner, J. Xu, and B. Livshits, "Liquidations: Defi on a knife-edge," *arXiv preprint arXiv:2009.13235*, 2020.

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

[75] M. Bartoletti, J. H.-y. Chiang, and A. Lluch-Lafuente, "Sok: Lending pools in decentralized finance," *arXiv preprint arXiv:2012.13230*, 2020.

[76] T. Limited, "Tether: Fiat currencies on the bitcoin blockchain," 2016, accessed: 08-06-2020. [Online]. Available: https://tether.to/wp-content/uploads/2016/06/TetherWhitePaper.pdf

[77] J. Lee, "Nubits," 2014. [Online]. Available: https://nubits.com/NuWhitepaper.pdf

[78] W. Zhao, H. Li, and Y. Yuan, "Understand volatility of algorithmic stablecoin: Modeling, verification and empirical analysis," *arXiv preprint arXiv:2101.08423*, 2021.

[79] F. Feng and B. Weickmann, "Set: A protocol for baskets of tokenized assets," 2019. [Online]. Available: https://www.setprotocol.com/pdf/set_protocol_whitepaper.pdf

[80] A. Cronje, "yEARN," 2020. [Online]. Available: https://yearn.finance

[81] CryptoCompare, "Cryptocompare exchange review, february 2021," 2021. [Online]. Available: https://www.cryptocompare.com/media/37746440/cryptocompare_exchange_review_2021_02.pdf

[82] J. Hull *et al.*, *Options, futures and other derivatives/John C. Hull*. Upper Saddle River, NJ: Prentice Hall,, 2009.

[83] J. Clark, "The replicating portfolio of a constant product market," *Available at SSRN 3550601*, 2020.

[84] A. Evans, "Liquidity provider returns in geometric mean markets," *arXiv preprint arXiv:2006.08806*, 2020.

[85] BitMEX, "Bitmex perpetual contracts guide," 2020. [Online]. Available: https://www.bitmex.com/app/perpetualContractsGuide

[86] dYdX, "dydx," 2019. [Online]. Available: https://dydx.exchange/

[87] Opyn, "Opyn," 2020. [Online]. Available: https://opyn.co/#/

[88] M. Wintermute, "Hegic: On-chain options trading protocol on ethereum powered by hedge contracts and liquidity pools," 2020, accessed: 13-11-2020. [Online]. Available: https://ipfs.io/ipfs/QmWy8x6vEunH4gD2gWT4Bt4bBwWX2KAEUov46tCLvMRcME

[89] A. Niemerg, D. Robinson, and L. Livnev, "Yieldspace," https://yield.is/YieldSpace.pdf, 2020.

[90] W. Wallet, "Wasabi wallet," 2021. [Online]. Available: https://wasabiwallet.io/

[91] Tornado, "Tornado," 2021. [Online]. Available: https://tornado.cash/

[92] Zcash, "Zcash," 2021. [Online]. Available: https://z.cash/

[93] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts (sok)," in *International conference on principles of security and trust*. Springer, 2017, pp. 164–186.

[94] D. Perez and B. Livshits, "Smart contract vulnerabilities: Does anyone care?" *arXiv preprint arXiv:1902.06710*, 2019.

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

[96] M. Rodler, W. Li, G. O. Karame, and L. Davi, "Sereum: Protecting existing smart contracts against re-entrancy attacks," in *Proceedings of 26th Annual Network & Distributed System Security Symposium (NDSS)*, February 2019. [Online]. Available: http://tubiblio.ulb.tu-darmstadt.de/111410/

[97] dForce, "dforce," 2020. [Online]. Available: https://dforce.network/

[98] W. Foxley and N. De, "Weekend attack drains decentralized protocol dforce of $25m in crypto," *CoinDesk*, 2020. [Online]. Available: https://www.coindesk.com/attacker-drains-decentralized-protocol-dforce-of-25m-in-weekend-attack

[99] Tokenlon, "imbtc," 2020. [Online]. Available: https://tokenlon.im/imBTC#/

[100] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 254–269.

[101] C. F. Torres, J. Schütte, and R. State, "Osiris: Hunting for integer bugs in ethereum smart contracts," in *Proceedings of the 34th Annual Computer Security Applications Conference*, ser. ACSAC '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 664–676. [Online]. Available: https://doi.org/10.1145/3274694.3274737

[102] S. Kalra, S. Goel, M. Dhawan, and S. Sharma, "ZEUS: analyzing safety of smart contracts," in *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society, 2018. [Online]. Available: http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018_09-1_Kalra_paper.pdf

[103] E. Foundation, "Solidity v0.8.0 documentation," 2020, accessed: 12-01-2020. [Online]. Available: https://docs.soliditylang.org/en/v0.8.0/index.html

[104] YAM, "Yam finance," 2020. [Online]. Available: https://yam.finance/

[105] T. Claburn, "Single-line software bug causes fledgling yam cryptocurrency to implode just two days after launch," 2020. [Online]. Available: https://www.theregister.com/2020/08/13/yam_cryptocurrency_bug_governance/

[106] CertiK, "Yam finance smart contract bug analysis & future prevention," 2020. [Online]. Available: https://certik.io/blog/technology/yam-finance-smart-contract-bug-analysis-future-prevention

[107] YAM Finance, "Yam post-rescue attempt update," 2020. [Online]. Available: https://medium.com/@yamfinance/yam-post-rescue-attempt-update-c9c90c05953f

[108] bZx Network, "bZx, The most powerful open finance protocol," 2020. [Online]. Available: https://bzx.network/

[109] PeckShield, "bzx hack full disclosure (with detailed profit analysis)," 2020. [Online]. Available: https://medium.com/@peckshield/bzx-hack-full-disclosure-with-detailed-profit-analysis-e6b1fa9b18fc

[110] L. Gudgeon, D. Perez, D. Harz, B. Livshits, and A. Gervais, "The decentralized financial crisis," in *2020 Crypto Valley Conference on Blockchain Technology (CVCBT)*, 2020, pp. 1–15.

[111] LongForWisdom, "[urgent] flash loans and securing the maker protocol," 2020. [Online]. Available: https://forum.makerdao.com/t/urgent-flash-loans-and-securing-the-maker-protocol/490

[112] Peckshield, "Value defi incident: Root cause analysis," 2020, accessed: 13-01-2021. [Online]. Available: https://peckshield.medium.com/value-defi-incident-root-cause-analysis-fbab71faf373

[113] Rekt, "Harvest finance - rekt," 2020. [Online]. Available: https://rekt.ghost.io/harvest-finance-rekt/

[114] ETH Tx Decoder, "Transaction analysis," 2020, accessed: 13-01-2021. [Online]. Available: https://ethtx.info/mainnet/0x9d093325272701d63fdafb0af2d89c7e23eaf18be1a51c580d9bce89987a2dc1

[115] S. Eskandari, S. Moosavi, and J. Clark, "Sok: Transparent dishonesty: front-running attacks on blockchain," in *International Conference on Financial Cryptography and Data Security*. Springer, 2019, pp. 170–189.

[116] L. Zhou, K. Qin, C. F. Torres, D. V. Le, and A. Gervais, "High-frequency trading on decentralized on-chain exchanges," *arXiv preprint arXiv:2009.14021*, 2020.

[117] H. A. Kalodner, M. Carlsten, P. Ellenbogen, J. Bonneau, and A. Narayanan, "An empirical study of namecoin and lessons for decentralized namespace design." in *WEIS*. Citeseer, 2015.

[118] D. Robinson, "Etherum is a dark forest," 2020, accessed: 24-11-2020. [Online]. Available: https://medium.com/@danrobinson/ethereum-is-a-dark-forest-ecc5f0505dff

[119] L. Breidenbach, P. Daian, F. Tramèr, and A. Juels, "Enter the hydra: Towards principled bug bounties and exploit-resistant smart contracts," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 1335–1352.

[120] samczsun, "Escaping the dark forest," 2020, accessed: 24-11-2020. [Online]. Available: https://samczsun.com/escaping-the-dark-forest

[121] M. Swende, "Blockchain frontrunning," 2017. [Online]. Available: https://swende.se/blog/Frontrunning.html

[122] W. Foxley, "Exploit during ethdenver reveals experimental nature of decentralized finance," *CoinDesk*, 2020. [Online]. Available: https://www.coindesk.com/exploit-during-ethdenver-reveals-experimental-nature-of-decentralized-finance

[123] P. Baker, "Defi project bzx exploited for second time in a week, loses $630k in ether," *CoinDesk*, 2020. [Online]. Available: https://www.coindesk.com/defi-project-bzx-exploited-for-second-time-in-a-week-loses-630k-in-ether

[124] T. Cooper, "imbtc uniswap pool drained for ∼$300k in eth," 2020, accessed: 20-01-2021. [Online]. Available: https://defirate.com/imbtc-uniswap-hack/

[125] A. Tarasov, "Millions lost: The top 19 defi cryptocurrency hacks of 2020," 2020. [Online]. Available: https://cryptobriefing.com/50-million-lost-the-top-19-defi-cryptocurrency-hacks-2020/

[126] 1inch, "Balancer pool with sta deflationary token incident," 2020. [Online]. Available: https://1inch-exchange.medium.com/balancer-hack-2020-a8f7131c980e

[127] opyn, "Opyn eth put exploit," 2020. [Online]. Available: https://medium.com/opyn/opyn-eth-put-exploit-c5565c528ad2

[128] C. Harper, "Defi degens hit hard by eminence exploit will be partially compensated," *CoinDesk*, 2020. [Online]. Available: https://www.coindesk.com/eminence-exploit-defi-compensated

[129] Percent Finance, "Important announcement," 2020. [Online]. Available: https://percent-finance.medium.com/important-announcement-d35f9a0df112

[130] B. Pirus, "Cheese bank's multi-million-dollar hack explained by security firm," 2020, accessed: 29-12-2020. [Online]. Available: https://cointelegraph.com/news/cheese-bank-s-multi-million-dollar-hack-explained-by-security-firm

[131] PeckShield, "88mph incident: Root cause analysis," 2020. [Online]. Available: https://peckshield.medium.com/88mph-incident-root-cause-analysis-ce477e00a74d

[132] P. Thompson, "Defi project pickle finance exploited for $20 million," 2020. [Online]. Available: https://coingeek.com/defi-project-pickle-finance-exploited-for-20-million/

[133] W. Foxley, "$10.8m stolen, developers implicated in alleged smart contract 'rug pull'," *CoinDesk*, 2020. [Online]. Available: https://www.coindesk.com/compounder-developers-implicated-alleged-smart-contract-rug-pull

[134] Rekt, "Warp finance - rekt," 2020. [Online]. Available: https://rekt.eth.link/warp-finance-rekt/

[135] Rekt, "Yearn - rekt," 2021. [Online]. Available: https://rekt.eth.link/yearn-rekt/

[136] Rekt, "The big combo (growth defi - rekt)," 2021. [Online]. Available: https://rekt.eth.link/the-big-combo/

[137] Rekt, "Meerkat finance - bsc - rekt," 2021. [Online]. Available: https://rekt.eth.link/meerkat-finance-bsc-rekt/

[138] Rekt, "Paid network - rekt," 2021. [Online]. Available: https://rekt.eth.link/paid-rekt/

[139] Rekt, "Dodo - rekt," 2021. [Online]. Available: https://rekt.eth.link/au-dodo-rekt/

[140] T. Roughgarden, "Algorithmic game theory," *Communications of the ACM*, vol. 53, no. 7, pp. 78–86, 2010.

[141] T. Roughgarden, "Transaction fee mechanism design for the ethereum blockchain: An economic analysis of eip-1559," *arXiv preprint arXiv:2012.00854*, 2020.

[142] H.-T. Kao, T. Chitra, R. Chiang, and J. Morrow, "An analysis of the market risk to participants in the compound protocol," in *Third International Symposium on Foundations and Applications of Blockchains*, 2020.

[143] A. Klages-Mundt and A. Minca, "(in) stability for the blockchain: Deleveraging spirals and stablecoin attacks," *arXiv preprint arXiv:1906.02152*, 2019.

[144] A. Klages-Mundt and A. Minca, "While stability lasts: A stochastic model of stablecoins," *arXiv preprint arXiv:2004.01304*, 2020.

[145] E. Frangella, "Crypto black thursday: The good, the bad, and the ugly," https://medium.com/aave/crypto-black-thursday-the-good-the-bad-and-the-ugly-7f2acebf2b83, 2020, accessed: 20-01-2021.

[146] A. Judmayer, N. Stifter, A. Zamyatin, I. Tsabary, I. Eyal, P. Gazi, S. Meiklejohn, and E. Weippl, "Pay to win: Cheap, crowdfundable, cross-chain algorithmic incentive manipulation attacks on pow cryptocurrencies," Cryptology ePrint Archive, Report 2019/775, 2019. [Online]. Available: https://eprint.iacr.org/2019/775

[147] Blocknative, "Evidence of mempool manipulation on black thursday: Hammerbots, mempool compression, and spontaneous stuck transactions," 2020. [Online]. Available: https://www.blocknative.com/blog/mempool-forensics

[148] P. Baker, "Miners trick stablecoin protocol pegnet, turning 11 into almost 7m hoard," *CoinDesk*, 2020. [Online]. Available: https://www.coindesk.com/miners-trick-stablecoin-protocol-pegnet-turning-11-into-almost-7m-hoard

[149] L. Zhou, K. Qin, A. Cully, B. Livshits, and A. Gervais, "On the just-in-time discovery of profit-generating transactions in defi protocols," *arXiv preprint arXiv:2103.02228*, 2021.

[150] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 154–167.

[151] D. Rate, "Cream finance partially delists ftt amidst governance contention," 2021. [Online]. Available: https://defirate.com/cream-ftt-delisting/

[152] A. Klages-Mundt, "Vulnerabilities in maker: oracle-governance attacks, attack daos, and (de)centralization," Nov. 14, 2019. [Online]. Available: https://link.medium.com/VZG64fhmr6

[153] M. Kaleem and W. Shi, "Demystifying pythia: A survey of chainlink oracles usage on ethereum," *arXiv preprint arXiv:2101.06781*, 2021.

[154] Y. Khatri, "Dai price increase led to a massive $88 million worth of liquidations at defi protocol compound," 2020, accessed: 14-01-2021. [Online]. Available: https://www.theblockcrypto.com/post/85850/dai-compound-dydx-liquidations-defi

[155] Compound, "Open price feed," 2020, accessed: 06-12-2020. [Online]. Available: https://compound.finance/prices

[156] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi, "Town crier: An authenticated data feed for smart contracts," in *Proceedings of the 2016 aCM sIGSAC conference on computer and communications security*, 2016, pp. 270–282.

[157] S. Ellis, A. Juels, and S. Nazarov, "A decentralized oracle network," 2017.

[158] A. Thurman, "Value defi protocol suffers $6 million flash loan exploit," 2020, accessed: 29-12-2020. [Online]. Available: https://cointelegraph.com/news/value-defi-protocol-suffers-6-million-flash-loan-exploit

[159] V. von Wachter, J. R. Jensen, and O. Ross, "Measuring asset composability as a proxy for ecosystem integration," *arXiv preprint arXiv:2102.04227*, 2021.

[160] M. Nadler and F. Schär, "Decentralized finance, centralized ownership? an iterative mapping process to measure protocol token distribution," *arXiv preprint arXiv:2012.09306*, 2020.

[161] P. Tolmach, Y. Li, S.-W. Lin, and Y. Liu, "Formal analysis of composable defi protocols," *arXiv preprint arXiv:2103.00540*, 2021.

[162] OpenCollective, "cadcad," 2020. [Online]. Available: https://cadcad.org/

[163] D. Harz and W. Knottenbelt, "Towards safer smart contracts: A survey of languages and verification methods," *arXiv preprint arXiv:1809.09805*, 2018.

[164] A. Permenev, D. Dimitrov, P. Tsankov, D. Drachsler-Cohen, and M. Vechev, "Verx: Safety verification of smart contracts," in *2020 IEEE Symposium on Security and Privacy, SP*, 2020, pp. 18–20.

[165] Consensys, "Mythx: Smart contract security service for ethereum," 2021. [Online]. Available: https://mythx.io/

[166] J. Feist, "Slither – a solidity static analysis framework," 2018. [Online]. Available: https://blog.trailofbits.com/2018/10/19/slither-a-solidity-static-analysis-framework/

[167] D. Annenkov and B. Spitters, "Towards a smart contract verification framework in coq," *arXiv preprint arXiv:1907.10674*, 2019.

[168] X. Chen, D. Park, and G. Roşu, "A language-independent approach to smart contract verification," in *International Symposium on Leveraging Applications of Formal Methods*. Springer, 2018, pp. 405–413.

[169] ConsenSys, "Mythril," 2021. [Online]. Available: https://github.com/ConsenSys/mythril

[170] S. Panja and B. K. Roy, "A secure end-to-end verifiable e-voting system using zero knowledge based blockchain." *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 466, 2018.

[171] Y. Wang and A. Kogan, "Designing confidentiality-preserving blockchain-based transaction processing systems," *International Journal of Accounting Information Systems*, vol. 30, pp. 1–18, 2018.

[172] R. K. Raman, R. Vaculin, M. Hind, S. L. Remy, E. K. Pissadaki, N. K. Bore, R. Daneshvar, B. Srivastava, and K. R. Varshney, "Trusted multiparty computation and verifiable simulations: A scalable blockchain approach," *arXiv preprint arXiv:1809.08438*, 2018.

[173] F. Benhamouda, S. Halevi, and T. Halevi, "Supporting private data on hyperledger fabric with secure multiparty computation," *IBM Journal of Research and Development*, vol. 63, no. 2/3, pp. 3–1, 2019.

APPENDIX A
DEFI PROTOCOLS

TABLE II: DeFi Protocols: A selection of prominent DeFi protocols classified according to the proposed protocol types.

Exchanges	PLFs	Stablecoins	Portfolio Managers	Derivatives
Curve	Compound	Maker	Harvest	Opyn
Uniswap	Aave	Unit	Yearn	Hegic
Sushiswap	dYdX	Reflexer	Set	Synthetix
Balancer	Cream	Fei	Alpha	
Bancor				
1inch				