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 600m USD, now, as of January 2021, it stands at around 25bn 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. The contrasting view of the DeFi Pessimist is that the unregulated, hack-prone DeFi ecosystem serves to facilitate unfettered and novel forms of financial crime. For instance, the pseudo-anonymous nature of DeFi permits cryptocurrency attackers, scammers, and money launderers to move, clean, and earn interest on capital. To a certain extent, the debate between the DeFi Optimist and the DeFi Pessimist turns on critical moral issues. We do not contribute to this important debate in this paper. 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.

First, we must be clear about what DeFi is. DeFi, in its ideal form, exhibits four properties. First, *non-custodial* financial services allow participants to exert full control over their funds at any point in time. To illustrate, traditional finance and fintech is based on a custodial model. For instance, your bank holds custody of your funds, your stocks are held at a custodian bank, and collateral of contracts may be held in escrow accounts by a custodian. For better or worse, you have to trust these custodians and they need to be compensated for their custodial services. In contrast, blockchain mechanisms provides a means for agents who do not trust each other to cooperate without requiring trusted third parties. For instance, holding on-chain assets can be done without a custodian, and general scripting functionality ('smart contracts') can execute deterministically and verifiably on-chain. Among many uses, this allows collateral to be escrowed on-chain without a custodian, which opens up a variety of non-custodial applications.

Second, the *permissionless* nature of DeFi allows anyone to interact with financial services without being able to be censored or blocked access by a third party. Third, DeFi is *openly auditable*, which means that anyone has the ability to audit the state of protocols–e.g., that they are fully collateralized/healthy. Fourth, 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. For example, this allows seamless rehypothecation of collateral (and the composability risks therein) 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 25bn USD as of January 2021, with the most capitalized use cases being collateralized lending, constituting c.48% of the TVL, and decentralized exchange (DEXs), constituting c.34% of the TVL as of January 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, the evolution of 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.

Moreover, at a 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: After outlining the primitives for DeFi in Sec. II, we 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. The technical security risks 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 is not merely cosmetic but serves to make clear that the development of the DeFi ecosystem is akin to a 'two-front' war, and moreover one in which the fronts can merge to great effect. We combine both views in proposing a set of seven main open research challenges for DeFi going forward (Sec. VI).

II. DeFi Primitives

DeFi protocols require an underlying distributed ledger such as a blockchain, which is a peer-to-peer distributed append-only record of transactions. In this paper, we primarily treat 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. In particular, 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, thereby allowing for economic interactions between mistrusting 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].

In order for DeFi protocols to function on top of them, smart contracts must possess certain properties. First, they need to be expressive enough to be able to encode protocol rules. Most applications require some support for conditional execution and bounded iterations. Smart contracts also need to be able to communicate with one-another within the same execution context, typically a transaction. Finally, support for atomicity is required to ensure that no execution can result in an invalid state, i.e., a transaction either succeeds fully (state update) or fails entirely (state remains unaltered).

When considered specifically 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 Legos" [20]), with the possibility of building complex financial architectures. This is similar to as was envisaged in [21]. While promising, the side-effects of smart contracts interactions and the space of all possible interactions is likely vast. Such complexity in the context of financial applications brings with it a great burden to understand the emergent security properties of composed smart contracts or else face significant financial risk. 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 [22] and other cryptoassets [23] to synthetic assets or derivatives [24], 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 [25] and ERC-721 [26] interfaces for fungible and non-fungible tokens, respectively [27], although other token standards exist [28], [29], [30]. The commonly agreed definition is that fungible tokens are interchangeable [25] while non-fungible tokens are distinct [26]. To give a simple example, fungible tokens can be used to represent a currency, such as the US dollar, where any two US dollars are equivalent. On the other hand, non-fungible tokens can be used to represent tokenized pieces of arts, where each piece of art is distinct from another.

C. Transaction Execution

A feature of the underlying blockchain which we draw particular attention to is the provision of the ability for users to make transactions. 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 at first broadcast to a network of peers, validated, and then stored in a waiting area (the *mempool* of a node). Consensus

participants of the underlying ledger known as *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], [31].

The ability of miners to choose which transactions are and are not included in a given block means that miners are able to control the sequence in which particular transactions are executed. In turn, this opens up the possibility that miners can arbitrarily include, exclude and order transactions in a way that is beneficial to them, resulting in *miner extractable value* (MEV) [32]. Order optimization fees can be captured by reordering and censoring transactions, while also inserting the miner's own transactions if profitable. The notion of MEV can be further exacerbated by the possibility of bribing miners to undertake such transaction re-ordering [33], [34]. We consider these issues in detail in Section V-B.

D. External Agents and State Updates

Protocols may rely on certain state updates in order to preserve protocol security. In transaction-based systems, a state can not update unless a transaction is triggered externally. However, as smart contracts are not able to create transactions programmatically, protocols rely on external entities to trigger state updates. These entities, called *keepers*, are generally incentivized through profit opportunities to automate certain operations around on-chain protocols and thereby contribute towards maintaining a decentralized system. For instance, if a protocol requires collateral assets to be liquidated to cover a borrow position if certain conditions are met, then the protocol will incentivize keepers to initiate transactions to push these actions.

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, for instance, prices of off-chain assets, such as ETH/USD, or off-chain information needed to verify outcomes of prediction markets, and is relied upon by various DeFi protocols (e.g. [35], [36], [37], [38], [39]). Such data is not natively accessible on-chain.

Oracle mechanisms differ by design and risk, as discussed in [40], [41]. 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. An alternative is offered by decentralized oracles. 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 come with their own set of 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 [40]. 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. [42], [43], [44]), 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. [45], [46], [47], [48]), an ERC-20 token which entitles token holders to participate in protocol governance via voting on and possibly propose protocol updates. We return to governance in Section V.

III. DeFi Protocols

We now present DeFi protocols categorized by the type of operation they provide. The presented protocol types rely on the previously examined DeFi primitives. A conceptual overview of how DeFi primitives are used in combination with market mechanisms to construct protocols is shown in Figure 1.

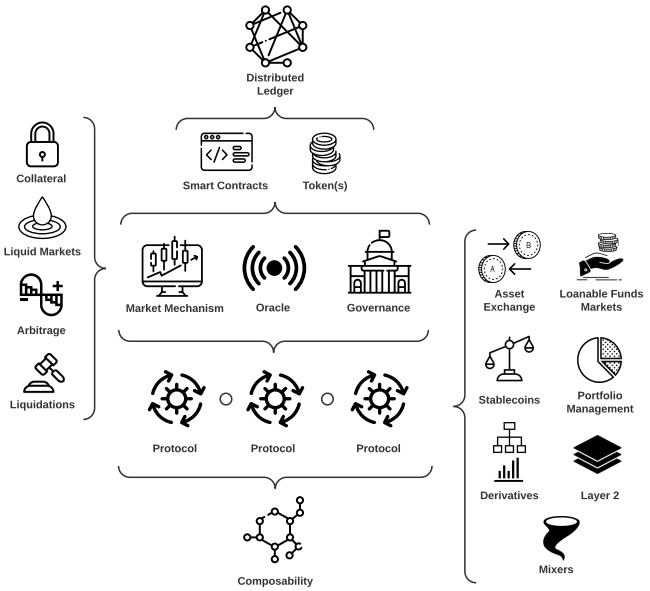

Fig. 1: A conceptual overview of the different constructs within the DeFi ecosystem.

A. On-chain Asset Exchange

Venues facilitating the exchange of digital assets are a crucial part of the wider digital asset ecosystem, with centralized cryptoasset exchanges appearing as early as 2010 [49].

However, centralized cryptoasset exchanges have been repeatedly prone to several major attacks (e.g., [50], [51], [52]) and the absence of public verifiability of trading activity has resulted in reports of fake trading volume [53], [54], undermining centralized exchanges' trustworthiness. A class of DeFi protocols that facilitates the non-custodial exchange of on-chain digital assets exists in the form of decentralized exchanges (DEXs) [55], [56]. Apart from being non-custodial, i.e., the exchange not having 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. A further difference between DEXs and their centralized counterpart is that only assets native to the underlying blockchain, such as ERC-20 tokens on Ethereum, can be traded. This is due to the atomicity of transactions on which DEXs rely to ensure the correctness of their execution [57] and therefore direct interaction with external assets such as Bitcoin [1] or fiat currency is unfeasible.

Some solutions to work around this limitation do exist but have drawbacks limiting their adoption. Wrapped tokens such as wBTC [23] (wrapped Bitcoin) can be used to trade assets which are not directly on Ethereum, but given their often custodian nature, this approach shares similar security concerns as centralized exchanges. Cross-chain solutions have also been designed [58], [15] but at the time of writing, have not yet seen wide adoption in DEXs. For instance, atomic swaps have require inherently high latency, over which a free option is granted to one party, governed systems like [59] essentially require the user to trust the incentive alignment of governance, and methods like [58] require relays, which can be expensive to maintain and require the required intermediary to overcollateralize the wrapped asset.

Based on the mechanism for price discovery, DEXs come in different variants, such as *order book DEXs* (including individual [60], [57] and batch settlement [61], [62]) and *automated market makers* (AMMs) (e.g., [63], [64], [65]).

1) Order Book DEXs: In centralized financial exchanges, an order book is an electronic list of buy and sell orders for a particular financial instrument, where a trade is executed when orders are matched. Maintaining the state of an order book is a computationally expensive task and given the design of blockchains (e.g., the Ethereum virtual machine and its gas price mechanism [66], [67]) it is not practically feasible to host this on-chain. Hence, a decentralized order book exchange may employ off-chain order books and thus involve some level of centralization, where only trade settlement is executed on-chain. A user wanting to execute an order will typically pre-sign a transaction allowing the DEX to execute the trade only if it fulfills the conditions specified by the user.

Orders are matched either manually or algorithmically, where in the case of the former, takers are required to fill resting orders created by makers. While manual order matching offers trustless trading between takers and makers as any centralized intermediary is circumvented, it comes at the cost of increased latency and potentially fragmented liquidity due to inefficient price discovery. More efficient order matching can be achieved algorithmically, however this involves trusting centralized off-chain matching engines [60], which are often prone to manipulation [32], [68], to fill orders at fair prices.

Different order book methods using batch settlement may help to resolve these matching issues algorithmically. For instance, [61] settles the order book in a manner resembling a Dutch auction (with batches settling at gradually decreasing prices until sell orders are filled). This has the downside of potentially long settlement delays. Alternatively, trades can be matched algorithmically in periodic batches maintained by decentralized keepers [62]. Here, the matching problem is solved by competing keepers who submit their solutions on-chain, from which the protocol executes the best solution. 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].

2) Automated Market Makers: 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) [71] 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 a 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've 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 often play a critical role in DeFi protocols. Thereby, 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 analysis of AMM design and the underlying market making mechanism, we direct the reader to [72], [73], [74], [75].

B. Loanable Funds Markets for On-chain Assets

Lending and borrowing of on-chain assets is facilitated through *protocols for loanable funds* (PLFs) [76], [77], which refer to DeFi lending protocols that establish distributed ledger-based markets for loanable funds of cryptoassets. 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. Unlike peer-to-peer lending, where funds are directly lent between individual agents, in a PLF, deposits for a given token market are pooled together in a smart contract. An agent may directly borrow against the smart contract reserves, assuming the market for the token is sufficiently liquid.

Given the pseud-anonymous nature of blockchains, borrowers are required to overcollateralize their borrow position, in order to protect PLFs from sustaining financial losses as a result of borrowers defaulting on their debt. Collateralization is the process in which something of value is provided as security to cover the value of a debt. For example, when obtaining a mortgage for a house, the house is the collateral: if the borrower defaults on their repayment obligations, the house can be sold to pay off the mortgage. In general, collateralization makes it possible for agents to borrow assets without the lender incurring credit risk, i.e., suffering financial losses as a result from a borrower defaulting on a debt obligation. By posting collateral of x in USD, in principle an agent can borrow up to 100% of this collateral value in another asset. If the agent does not repay the debt, the collateral can be liquidated to pay it off. 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. Due to the historical volatility and illiquidity of many cryptoassets, it should be noted that *overcollateralization* is often relied upon, where for, e.g. 100 USD of borrowed value, more than 100 USD must be provided as backing collateral. 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 PLFs, a borrower has to ensure that the value of the locked collateral remains above some liquidation threshold, as otherwise 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 [78].

PLFs may offer functionality beyond overcollateralized borrowing capabilities in the form of so-called *flash loans*. These provide access to uncollateralized loans for the duration of one 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 [79].

The cost of borrowing in a PLF is given by an interest rate charged to the borrower, which is determined by a market's underlying interest rate model. These interest rate models tend to reflect the notion that as liquidity becomes scarcer, a higher interest rate should encourage current borrowers to repay their debts, while incentivizing holders of excess deposits to supply these.

In exchange for depositing funds, a depositor receives a derivative token reflecting his share of the total supplied funds in a market. As interest paid by borrowers is generally retained by the smart contract, the relative share of total funds in a market of a derivative token holder will increase over time. Accrued interest in a market is thereby paid out to the market's depositors as compensation for providing liquidity, while a reserve fraction is retained from the paid out interest by the protocol in order to protect against periods of illiquidity [80] and market stress.

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 [37], which has close to 4.38bn USD in market capitalization as of January 2021[1]. Note that custodial stablecoins, such as USDT [81] 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 [40]. Price-stability is pursued via the use of on-chain collateral, 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 [40].

- Collateral. This is the store of primary value for a stablecoin. Collateral can be exogenous (e.g., ETH in Maker [46], where the collateral is primarily used externally to the stablecoin, endogenous (e.g., SNX in Synthetix [24], where the collateral was created to be collateral or implicit (e.g., Nubits [82], where the design lacks an explicit store of collateral.

[1] Source: https://defipulse.com/. Accessed: 20-01-2021.

- Agents. Agents form at least two roles in a non-custodial stablecoin: (i) risk absorption, for instance by providing collateral that is intended to absorb price risk, and (ii) 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 [40] for a more complete discussion of stablecoin designs, models, and challenges.

D. Portfolio Management

For liquidity providers seeking to maximize their returns, it 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 [83] to complex yield aggregating strategies [84].

E. Derivatives

Derivatives are financial contracts which derive their value from the performance of underlying assets. As of November 2020, the derivatives market represents about 60% of the entire cryptoassets trading market [85]. 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 four different basic types of derivatives:

- Synthetic assets. These aim to replicate the payoffs of another asset without directly taking a position in that asset. In DeFi, synthetic assets typically replicate off-chain assets on-chain (e.g., the USD in protocols like Maker and Synthetix [24]). 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 [86]) in traditional finance and have been explored more specifically using constant product market makers in [87], [88].
- Futures. These facilitate the buying or selling of an underlying asset at an agreed price and time in the future. 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 [89]. These are much more popular as they allow traders to decide (typically on a daily basis, e.g., [90]) 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. These allow the buyer to have a the choice to exercise the contract while it leaves the seller with the obligation to fulfill the contract. For example, a seller can offer to buy Bitcoin at a price of $\$18,000$ two weeks in the future. The buyer of this option contract can then choose to exercise the option after two weeks have passed. If Bitcoin trades at a price of $\$17,000$, the buyer would have a potential earning of $\$1,000$ minus fees. This is an example for a European-style put option. There are many different option types and trading strategies [86]. Currently, the DeFi market for options is very early with basic call and put options (e.g., [91], [92]) but not leveraged positions on options, which present greater capital efficiency issues.

In DeFi derivative design, there are a few particular points and issues to discuss further:

a) Leverage: In DeFi, protocols are typically overcollateralized to reduce the likelihood of defaulting on loans (e.g., in stablecoins or protocols for loanable funds). This makes these protocols capital inefficient as one needs to deposit more value than taking as a loan. Hence, derivatives can form an alternative where traders are only required to provide a fraction of the capital to trade the value of an underlying by, e.g., using a perpetual or an option. Furthermore, platforms like dYdX allow traders to leverage their positions. This elevates the exposure to the price movement of the derivative. However, while centralized alternatives rely on established risk management systems, DeFi alternatives must still rely on higher rates of collateral in absence of other forms of investor verification.

b) Settlement: Derivatives can either be physically settled, i.e., the underlying is transferred, or cash settled, i.e., the price difference at time of exercising the derivative is settled in some currency. Both forms of settlements can be automatically enforced in DeFi by locking the assets at stake in the trading smart contracts. Cash settlement is often more capital efficient as it requires locking only the difference in price movement, e.g., in perpetual and option contracts between different points in time. Physically settled derivatives are mostly possible when the asset is available on-chain (e.g., Ethereum options in Opyn [91]).

c) Trading: Similar to other DeFi assets, derivatives can be traded via an AMM or order book DEX. Order book style DEX trading is very similar to centralized exchanges and the effectiveness of price discovery of the derivatives mostly relies on sufficient liquidity. However, 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 (measured by θ). An option decreases in value over time depending on the price of the underlying. More nuanced AMM designs like [93] aim to incorporate such a time dimension. Bonding curves in AMMs are still not aware of other relationships between the underlying and option value. Hence, the AMM price of the option does not reflect the actual option value as it relies on liquidity providers and traders to correct the price. With more complex value functions in the AMM like Balancer [48] it is possible to replicate strategies that combine the underlying and a derivative into a single asset [88].

F. DeFi on Layer-Two

Layer-two refers to a set of protocols which seek to facilitate the scaling of blockchains (i.e., layer-one) without a change in the trust assumptions at layer-one and without modifying the consensus mechanism. Layer-two protocols have emerged in a variety of guises, perhaps most notably as payment channels and payment channel networks. For a detailed overview of layer-two protocols, we refer the reader to [14].

Rollups are at the center of layer-two based approaches to DeFi scalability. The central idea is that the computation and storage of a would-be layer-one contract is handled on layer-two, with an on-chain assertion made about what the layer-two contract's operations are. Optimistic Rollups are one type of rollup, where each assertion is posted without an accompanying proof to guarantee the validity of the assertion. The assertion can be shown to be incorrect via the posting of a fraud-proof [94], [95]. Arbitrum provides an example of such a rollup mechanism [96]. Additionally, at the time of writing, Bancor [97] is testing a deployment on Arbitrum [96], [98].

zkRollups, rollups which use zero-knowledge proofs, are a further variant. The central idea is similar, using an off-chain prover which is able to compress large computations (i.e., batches of transactions) into smaller validity proofs [99]. Such validity proofs provide evidence that the layer-one state transition was correct. StarkWare's Cairo platform seeks to provide a Turing complete EVM for generating STARK proofs for general computation. An existing integration of STARKs with layer-one can be found via DiversiFi [100], purportedly offering $9,000$ transactions per second. Planned concrete integrations include between StarkWare and dYdX [90], where dYdX's perpetual contracts are to be ported to layer-two leveraging zK-Rollups. These zkRollups, a layer-two transaction compression mechanism where hundreds of transactions are bundled into a single transaction [101], would enable trades to be submitted on chain, with an aim of reducing the gas required per trade. zkRollups are also central to Loopring's layer-two DEX design, which performs most computations off-chain, broadcasting only the state roots of the DEX on chain [102].

G. 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: (i) 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., [103]), or (ii) directly shielding the contents of transactions using zero knowledge proofs of transaction validity (e.g., [104], [105]). Mixers serve as a DeFi-like application itself and additionally as a piece that could be included within other DeFi protocols.[2]

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.

[2]We plan to discuss these further in a subsequent version of this paper.

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 [106], [107], [108] 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 [109]. A trivial example is a contract with a withdraw function that checks for the internal balance of a user, sends him 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 [110], which features a PLF, in April 2020 to drain around 25 million USD worth of funds [111]. The attacker leveraged imBTC [112], which is an ERC-777 token [28], to perform his 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: Another example of a reentrancy attack was on an imBTC Uniswap [3] pool. Despite the fact that Uniswap does not support ERC-777 tokens [64], an imBTC pool worth roughly 300 000 USD worth of tokens was drained using the above reentrancy attack.

Both of these attacks show a common attack pattern in DeFi applications: identifying and exploiting attack vectors which are based on leveraging 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 [108], [113]. 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 [114], [115], 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 [107].

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 applications at the time of writing is Solidity [116], which has a limited type system and no support for operator overloading. In addition, the EVM only supports a single type, 32 bytes 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 end up using exclusively 32 bytes integers and arithmetic on two units scaled differently 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 [117], 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 [118], [119]. 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 [120] 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.

We will present some of the logical bugs that resulted in notable financial losses to highlight the often trivial nature of

the issue encountered:

bZx: In September 2020, the bZx protocol [121], a lending protocol, suffered a loss of over 8 million USD due to a trivial logic error [122], 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 himself, the amount transferred would effectively only be added to his 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.

Opyn: In August 2020, the Opyn [91] protocol, an options trading protocol, suffered a loss of over 370 000 USD due to a logical bug that allowed a user to re-use the same funds multiple times [123]. Opyn allows users to exercise their put options by requiring them to sell tokens, as a proof of ownership of the option, and the amount of underlying asset to sell. In return, the users receive collateral, typically in a stable coin, from vaults acting as liquidity providers. The smart contract handling the logic to exercise options allowed users to exercise from multiple vaults but failed to correctly update the amount of underlying assets received after exercising from a vault. As a result, an attacker could send a very small amount of underlying asset to the contract and sell as much as his option would allow him to, resulting in a direct loss of money.

Although these are only two instances 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 [124]. 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 [125].

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. Instead of front- and back-running another user's transaction, the attacker sets up the imbalance, 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. Setting up the imbalance requires access to large 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 [126], [127]. 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 [128]. 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 [47] (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 [68]. 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 his 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 [129], 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 re-

sults in priority gas auctions (PGAs) [32], 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 [68]. A simple example of such an attack would be an attacker front-running a transaction that registers a domain name [130].

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 [32].

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 [131]. 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 [132] and trying to 'rescue' funds from an exploitable smart contract [131], [133].

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 [129], [134]. 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.

Protocol	Loss	Audit	Attack	Date	Ref.
bZx	0.35m	✓	TX sandwich	Feb-15-2020	[135]
bZx	0.63m	✓	TX sandwich	Feb-18-2020	[136]
Uniswap	0.30m	✓	Reentrancy	Apr-18-2020	[137]
dForce	25.00m	✗	Reentrancy	Apr-19-2020	[111]
Hegic	0.05m	✗	Logical bug	Apr-25-2020	[138]
Balancer	0.50m	✓	TX sandwich	Jun-28-2020	[139]
Opyn	0.37m	✓	Logical bug	Aug-04-2020	[123]
Yam	0.75m	✗	Logical bug	Aug-12-2020	[118]
bZx	8.10m	✓	Logical bug	Sep-14-2020	[7]
Eminence	15.00m	✗	TX sandwich	Sep-29-2020	[140]
MakerDAO	-	✓	Governance	Oct-26-2020	[125]
Harvest	33.80m	✓	TX sandwich	Oct-26-2020	[10]
Percent	0.97m	✓	Logical bug	Nov-04-2020	[141]
Cheese Bank	3.3m	✓	TX sandwich	Nov-06-2020	[142]
Akropolis	2.00m	✓	Reentrancy	Nov-12-2020	[8]
Value DeFi	7.00m	✗	TX sandwich	Nov-14-2020	[126]
Origin	7.00m	✓	Reentrancy	Nov-17-2020	[11]
88mph	0.01m	✓	Logical bug	Nov-17-2020	[143]
Pickle	19.70m	✗	Logical bug	Nov-21-2020	[144]
Compounder	10.80m	✓	Logical bug	Dec-02-2020	[145]
Cover	9.40m	✓	Logical bug	Dec-28-2020	[9]

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.

A variant of this attack [129] 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 [129].

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 [124]).

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 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., [146], but as a concept has seen some adaption in the context of cryptoeconomics and in particular DeFi.

Following [147], agents can be considered to be of different *types*, which are commonly denoted $\theta \in \Theta$. Agents report their type to the game designer, with the reported type conventionally denoted $\hat{\theta}$. A mechanism is a mapping from the set of reported agent types to a set of outcomes Y, i.e. $f(\theta) : \Theta \to Y$, where an outcome is taken to comprise an allocation of goods $x \in X$ and a transfer of money $t \in T$. In the case of full information, the social choice function maps agents' true types θ to an allocation of goods $x \in X$. A mechanism is incentive compatible if agents can do no better than report their true type to the game designer, i.e. $\hat{\theta}(\theta) = \theta$.

In the cryptoeconomic setting, incentive compatibility takes an adapted form: a mechanism is incentive compatible if agents are incentivized to execute the mechanism *as intended* (see e.g. [148]).

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 [40], 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 [40], 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 [86]). 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 or *solvent*.

Overcollateralization is not without risks, however. For instance, as explored in [124], [149], 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. For instance, 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 type 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. For instance, 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 [150], [151], 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' [152]. As further discussed in [151], 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 [32], and (2) stablecoins built on leverage markets (like Dai) present arbitrage opportunities in liquidating leveraged positions, as explored in [150]. 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 [153]. 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., [154], [155].

The practicality of MEV threats have been highlighted in [32], where the prevalent dangers of *undercutting* and *time-bandit* attacks are presented. In an undercutting attack [156], 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 [32], 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 [150]. This is very similar to events on Black Thursday, in which mempool manipulations contributed to inefficient liquidation auctions in Maker [154].

C. Governance Risks

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 dicounted 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 the governance token value may come from less desirable token uses–e.g., to effect changes to the protocol in ways that provide governors outside benefit but may harm the system. For instance, Cream governance added very risky but closely held collateral assets, arguably to their benefit but against the interests of the protocol [157]. Another hypothetical governance attack to indirectly extract collateral value is described in [158]. 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 [40] 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, [40] conjectures that in the case of a fully decentralized stablecoin with multiple classes of interested 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. 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 [40].

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 [150]. 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 unpredicted 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 [159]. As a result, the Compound Open Price Feed [160], 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 $\$89$m, 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 [161] 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 [40], decentralized oracle solutions for off-chain data exist. However, they are yet imperfect solutions. These tend 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. However, 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 [162], 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], [163], [142]). 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 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 [76], [164] begin to explore protocol interdependence; however there remains a critical gap in DeFi research toward taxonimizing and formalizing models to quantify composability risks. 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:

- Generally, modeling incentive compatibility of governance in various systems. For instance, setting up models, finding equilibria, and understanding how other agents in the system respond. The models in [40] 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 [165].
- Formally exploring how technical security can be compromised by borrowing of governance tokens via flash loans and PLFs.
- From an economic security perspective, formally exploring how incentive compatibility is further complicated by the borrowing of governance tokens via PLFs.
- Generally, 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.
- Formally evaluating protection of minority agents in systems with flexible governance.

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.

- 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.
- Designing and evaluating the security of various oracle strengthening methods: e.g., medianizers, reputation systems, and grounding reported prices based on on-chain verifiable metrics.

D. Miner Extractable Value

We identify important research directions in MEV:

- Developing methodology for quantifying the level of MEV opportunities. As we expand on below, we expect this problem to be computationally difficult.
- Developing methodology to quantify negative externalities of MEV–e.g., from wasted gas per block, upward gas price pressure.
- Designing mechansism that protect against consensus layer instability risks that are induced by high MEV incentives.
- How the emergence of MEV opportunities endogenously affects agents' behavior within DeFi protocols. Models for this are started in the context of stablecoins in [40].
- Developing mechanisms to secure protocols against time bandit attacks that seek to rewrite the recent transaction history–for example, which could aim to trigger and profit from increased protocol liquidations.

Toward the last point, [150] 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.

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).

Several protective measures against MEV-based attacks have emerged. One takes the form of trust-based dark pools whereby unconfirmed transactions are routed to permissioned mempools hosted by a mining pool [166], [167], which is trusted to not extract MEV. Note, however, that this undermines the system's decentralization objective and may thereby introduce issues of its own. A second approach tries to contain MEV by restricting certain DeFi roles that rebalance systems, and so could extract value, to a permissioned agent set (e.g., [168]). For instance, only permissioned agents may be able to perform liquidations in a protocol. Note that this similarly introduces decentralization issues and trust assumptions. A third proposal has been to create MEV auctions that sell the right to decide transaction ordering ahead-of-time, and so put an expected price on front-running [169]. This provides value that potentially goes back to the network. This is not without downsides as well, however. As discussed in [170], such a protocol would reduce frictions to turning MEV extraction, which is in general a very hard optimization problem, into a specialized industry that would end up extracting more MEV long-term. The concept of MEV auctions was further developed by the Flashbots research initiative [171], which proposes a mechanism by which miners delegate the task of finding the most profitable ordering of a transaction set to third party agents called searchers. Subsequently, searchers participate in a sealed bid auction and bid for their transaction bundle to be included by a miner in the next block. A first proof-of-concept implementation of a MEV Ethereum client implementing the proposed mechanism has been developed [172]. It remains an open problem to develop and evaluate the trade-offs of such

mechanisms.

E. Program Analysis

There exists a large amount of work [173], both in academia [113], [108], [174] and industry [175], [176], to analyze smart contract bugs and vulnerabilities. While smart contracts analysis tools keep improving, the number and scale of smart contracts 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 [174], [177], [178]. 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 [108], [114], [176], [179], 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 can a particular execution path influence its 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. Undercollateralized Loans

As a means of protecting protocols from losses incurred by pseudonymous agents not repaying loans, overcollateralization serves as a fundamental building block in DeFi protocols. However, overcollateralized loans are expensive from an access to capital point of view, especially when compared to the cost of obtaining funds via traditional bank loans. Overcollateralization restricts capital access to a set of wealthy agents and incurs the opportunity cost of being unable to employ excess collateral elsewhere.

In DeFi, non-collateralized loans only exist in the form of flash loans and credit delegation lines, where the former is only available for the duration of a single transaction and the latter requires strong identities and trust in some legal system which holds parties liable in case of contract breach (e.g. Aave's credit delegation [180]). While there has been research into collateral reduction mechanisms [181], [182], we identify a research gap for novel approaches that provides an agent with the ability to source liquidity via other forms of loans. A simple form is just to change the asset that is used as the security, for example, borrowers could collateralize non-fungible assets like domains or other digital goods.

However, another interesting form of undercollaterized loans can come in the form of start-up funding. The ICO wave of 2017/18 produced very few projects that managed to ship products to market. Improvements were suggested following this. One example is the DAICO, in which investors would fund a new organization by providing funds into a smart contract that has a predetermined 'tap' from which funds can be extracted over time. The investors would be able to withdraw the funds in case progress of the project failed to meet expectations [183]. This could be further combined with distributing governance tokens to investors over time.

G. 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 [184], [185] and multi-party computations [186], [187] hold many promises, these technologies are yet to gain traction in the context of DeFi. One of the main friction point 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 paper, we provided the first SoK on DeFi, an increasingly complex system of financial applications which is exposed to its own classes of security risks. We introduced DeFi from two point of views: the DeFi Optimist and the DeFi Pessimist and subsequently examined the workings of DeFi systematically and at length. First, we laid out the primitives for DeFi to then categorize the existing 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. By doing so, 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.

Referring back to the views of the DeFi Optimist and Pessimist, in this paper, we are not weighing in on the moral trade-off, but present a set of tools to be able to evaluate it. While DeFi may have the potential for creating a permissionless and non-custodial financial system, the headwinds in the form of open security challenges remain strong. In the end, however, it is the blend between promise and challenge what makes DeFi an area worthwhile for technical and economic security 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] DeFi Pulse, "What is defi?" 2019. [Online]. Available: https://defipulse.com/blog/what-is-defi/

[21] 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.

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

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

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

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

[26] 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

[27] 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.

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

[29] 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

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

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

[32] 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.

[33] 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.

[34] 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.

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

[36] 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

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

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

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

[40] 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.

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

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

[43] 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.

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

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

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

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

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

[49] G. Hileman and M. Rauchs, "Global cryptocurrency benchmarking study," *Cambridge Centre for Alternative Finance*, vol. 33, pp. 33–113, 2017.

[50] T. Moore and N. Christin, "Beware the middleman: Empirical analysis of bitcoin-exchange risk," in *International Conference on Financial Cryptography and Data Security*. Springer, 2013, pp. 25–33.

[51] C. Decker and R. Wattenhofer, "Bitcoin transaction malleability and mtgox," in *European Symposium on Research in Computer Security*. Springer, 2014, pp. 313–326.

[52] P. Rizzo, "Poloniex loses 12.3% of its bitcoins in latest bitcoin exchange hack," *CoinDesk*, 2014. [Online]. Available: https://www.coindesk.com/poloniex-loses-12-3-bitcoins-latest-bitcoin-exchange-hack

[53] T. Fusaro and M. Hougan, "Bitwise asset management: Presentation to the us securities and exchange commission," 2019. [Online]. Available: https://www.sec.gov/comments/sr-nysearca-2019-01/srnysearca201901-5164833-183434.pdf

[54] Alameda Research, "Investigation into the legitimacy of reported cryptocurrency exchange volume," 2019. [Online]. Available: https://ftx.com/volume-report-paper.pdf

[55] 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.

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

[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] 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.

[59] Ren, "Ren," 2021. [Online]. Available: https://renproject.io/

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

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

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

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

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

[65] 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/

[66] A. A. Zarir, G. A. Oliva, Z. M. J. Jiang, and A. E. Hassan, "Developing cost-effective blockchain-powered applications: A case study of the gas usage of smart contracts transactions in the ethereum blockchain platform," *ACM Trans. Softw. Eng. Methodol*, vol. 1, no. 1, 2020.

[67] 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.

[68] 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.

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

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

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

[74] 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

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

[76] 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.

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

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

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

[80] Alethio, "Illiquidity and bank run risk in defi," 2019. [Online]. Available: https://medium.com/alethio/overlooked-risk-illiquidity-and-bank-runs-on-compound-finance-5d6fc3922d0d

[81] 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

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

[83] 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

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

[85] CryptoCompare, "Cryptocompare exchange review, november 2020," 2020. [Online]. Available: https://www.cryptocompare.com/media/37621821/cryptocompare_exchange_review_2020_11.pdf

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

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

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

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

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

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

[92] 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

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

[94] M. Al-Bassam, A. Sonnino, and V. Buterin, "Fraud and data availability proofs: Maximising light client security and scaling blockchains with dishonest majorities," *arXiv preprint arXiv:1809.09044*, 2018.

[95] A. Zamyatin, Z. Avarikioti, D. Perez, and W. J. Knottenbelt, "Txchain: Efficient cryptocurrency light clients via contingent transaction aggregation." *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 580, 2020.

[96] H. Kalodner, S. Goldfeder, X. Chen, S. M. Weinberg, and E. W. Felten, "Arbitrum: Scalable, private smart contracts," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 1353–1370.

[97] Bancor, "Bancor," 2021. [Online]. Available: https://blog.bancor.network/

[98] O. Labs, "Offchain labs," 2021. [Online]. Available: https://offchainlabs.com/

[99] StarkWare, "Hello, cairo!" 2020. [Online]. Available: https://medium.com/starkware/hello-cairo-3cb43b13b209

[100] DiversiFi, "Diversifi," 2020. [Online]. Available: https://www.deversifi.com/

[101] Ethhub, "Zk-rollups," 2021. [Online]. Available: https://docs.ethhub.io/ethereum-roadmap/layer-2-scaling/zk-rollups

[102] Loopring, "Loopring zkrollup exchange and payment protocol," 2021. [Online]. Available: https://loopring.org/#/

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

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

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

[106] 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.

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

[108] 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.

[109] 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/

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

[111] 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

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

[113] 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.

[114] 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

[115] 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

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

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

[118] 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/

[119] 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

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

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

[122] 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

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

[124] 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.

[125] 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

[126] 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

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

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

[129] 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.

[130] 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.

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

[132] 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.

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

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

[135] 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

[136] 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

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

[138] 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/

[139] 1inch, "Balancer pool with sta deflationary token incident," 2020. [Online]. Available: https://1inch-exchange.medium.com/balancer-hack-2020-a8f7131c980e

[140] 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

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

[142] 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

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

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

[145] 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

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

[147] J. C. Harsanyi, "Games with incomplete information played by "bayesian" players, i–iii part i. the basic model," *Management science*, vol. 14, no. 3, pp. 159–182, 1967.

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

[149] 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.

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

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

[152] 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.

[153] 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

[154] 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

[155] 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

[156] 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.

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

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

[159] 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

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

[161] 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.

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

[163] 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

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

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

[166] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Financial Cryptography and Data Security*, N. Christin and R. Safavi-Naini, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 436–454.

[167] S. M. Werner and D. Perez, "Poolsim: A discrete-event mining pool simulation framework," in *Mathematical Research for Blockchain Economy*, P. Pardalos, I. Kotsireas, Y. Guo, and W. Knottenbelt, Eds. Cham: Springer International Publishing, 2020, pp. 167–182.

[168] A. Cornje, "Keep3r network," 22 Oct. 2020. [Online]. Available: https://andrecronje.medium.com/keep3r-network-ba5af26c1f24

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

[169] K. Floersch, "Mev auction: Auctioning transaction ordering rights as a solution to miner extractable value," 2020, accessed: 18-12-2020. [Online]. Available: https://ethresear.ch/t/mev-auction-auctioning-transaction-ordering-rights-as-a-solution-to-miner-extractable-value/6788

[170] E. Felten, "Front-running as a service," 29 Jun. 2020. [Online]. Available: https://medium.com/offchainlabs/front-running-as-a-service-334c929c945a

[171] thegostep, "Flashbots: Frontrunning the mev crisis," 2020, accessed: 18-12-2020. [Online]. Available: https://ethresear.ch/t/flashbots-frontrunning-the-mev-crisis/8251

[172] Flashbots, "Mev-geth," 2020, accessed: 18-12-2020. [Online]. Available: https://github.com/flashbots/mev-geth

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

[174] 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.

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

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

[178] 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.

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

[180] AAVE, "Aave," 2020. [Online]. Available: https://aave.com/

[181] D. Harz, L. Gudgeon, A. Gervais, and W. J. Knottenbelt, "Balance: Dynamic adjustment of cryptocurrency deposits," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1485–1502.

[182] D. Harz, L. Gudgeon, R. Khalil, and A. Zamyatin, "Promise: Leveraging future gains for collateral reduction." *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 532, 2020.

[183] V. Buterin, "Explanation of daicos," 2018. [Online]. Available: https://ethresear.ch/t/explanation-of-daicos/465

[184] 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.

[185] 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.

[186] 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.

[187] 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.