Disentangling Decentralized Finance (DeFi) Compositions

STEFAN KITZLER, Complexity Science Hub Vienna and AIT - Austrian Institute of Technology, Austria
FRIEDHELM VICTOR, Technische Universität Berlin, Germany
PIETRO SAGGESE, AIT - Austrian Institute of Technology and Complexity Science Hub Vienna, Austria
BERNHARD HASLHOFER, Complexity Science Hub Vienna, Austria

We present a measurement study on compositions of Decentralized Finance (DeFi) protocols, which aim to disrupt traditional finance and offer services on top of distributed ledgers, such as Ethereum. Understanding DeFi compositions is of great importance, as they may impact the development of ecosystem interoperability, are increasingly integrated with web technologies, and may introduce risks through complexity. Starting from a dataset of 23 labeled DeFi protocols and 10,663,881 associated Ethereum accounts, we study the interactions of protocols and associated smart contracts. From a network perspective, we find that decentralized exchange (DEX) and lending protocol account nodes have high degree and centrality values, that interactions among protocol nodes primarily occur in a strongly connected component, and that known community detection methods cannot disentangle DeFi protocols. Therefore, we propose an algorithm to decompose a protocol call into a nested set of building blocks that may be part of other DeFi protocols. This allows us to untangle and study protocol compositions. With a ground truth dataset that we have collected, we can demonstrate the algorithm's capability by finding that swaps are the most frequently used building blocks. As building blocks can be nested, that is, contained in each other, we provide visualizations of composition trees for deeper inspections. We also present a broad picture of DeFi compositions by extracting and flattening the entire nested building block structure across multiple DeFi protocols. Finally, to demonstrate the practicality of our approach, we present a case study that is inspired by the recent collapse of the UST stablecoin in the Terra ecosystem. Under the hypothetical assumption that the stablecoin USD Tether would experience a similar fate, we study which building blocks — and, thereby, DeFi protocols — would be affected. Overall, our results and methods contribute to a better understanding of a new family of financial products.

CCS Concepts: • **Applied computing → Digital cash**; *Electronic funds transfer;*

Additional Key Words and Phrases: Decentralized Finance, DeFi, blockchain, Ethereum, networks

Authors' addresses: S. Kitzler and P. Saggese, Complexity Science Hub Vienna, Josefstädter Straße 39, 1080, Vienna, Austria and AIT - Austrian Institute of Technology, Giefinggasse 4, 1210, Vienna, Austria; emails: kitzler@csh.ac.at, pietro.saggese@ait.ac.at; F. Victor, Technische Universität Berlin, Straße des 17. Juni 135, 10623, Berlin, Germany; email: friedhelm.victor@tu-berlin.de; B. Haslhofer, Complexity Science Hub Vienna, Josefstädter Straße 39, 1080, Vienna, Austria; email: haslhofer@csh.ac.at.

1 INTRODUCTION

Decentralized Finance (DeFi) stands for a new paradigm that aims to disrupt established financial markets. It offers financial services in the form of *smart contracts*, which are executable software programs deployed on top of distributed ledger technologies (DLT) such as Ethereum. Despite being a relatively recent development, we can already observe rapid growth in DeFi protocols, enabling lending of virtual assets, exchanging them for other virtual assets without intermediaries, or betting on future price developments in the form of derivatives such as options and futures. The term "financial lego" is sometimes used because DeFi services can be *composed* into new financial products and services.

As an example of a DeFi composition, consider Figure 1, which illustrates a user interacting with the *1inch* decentralized exchange (DEX) aggregator Web service.[1] The user holds an amount of USDT tokens and wants to swap them to KYL tokens. Using the Web application and the user's *externally owned account (EOA)*, the user creates a transaction against the *1inch* contract. This, in turn, triggers a sequence of two swaps on two DeFi protocols within the same transaction, from USDT to WETH on *SushiSwap* and thereafter from WETH to KYL on *UniSwap*. In this article, we study such single-transaction DeFi interactions and the networks that arise when combining multiple DeFi transactions.

1.1 Motivation

In 2021, the total value of tokens held by smart contracts underlying the DeFi protocols reached 106 billion USD [12], demonstrating rapid growth. As composability of DeFi protocols is frequently seen as one of the main advantages [36], there are multiple reasons why it is interesting to study DeFi compositions.

Ecosystem interoperability. While composability can be seen as an opportunity, single-transaction compositions as shown in Figure 1 currently work only within a single distributed ledger. Most of the emerging DLT scaling solutions, such as sidechains [27, 38], rollups [40], and off-chain networks [15, 37], lead to multiple, somewhat isolated DeFi ecosystems. Hence, composability is disrupted, as smart contracts on one platform cannot invoke contract functions on another platform within a single transaction. Understanding which types of compositions are frequently used may help in developing solutions to cross-chain [4, 20, 46, 52] composability. Until solutions are found, such knowledge can help in deciding which services should be co-located, and which services could be separate.

Integration with Web technologies. Cryptoassets have started integrating with various Web technologies. For example, the Brave[2] browser includes an integrated cryptoasset wallet and native use of Basic Attention Tokens (BATs), and various applications from the commercial BitTorrent[3] ecosystem rely on the BitTorrent Token (BTT). This raises the question regarding the interdependence between DeFi compositions and web technologies. Services such as Furucombo[4] already illustrate that almost arbitrary DeFi compositions are constructed through Web interfaces. In order to develop an understanding of this it is important to identify compositions and their points of interaction in the first place.

Risks through complexity. After its deregulation in the early 2000s, the securitization market became more complex and opaque. Financial institutions used new financial instruments to maximize their exposure in this market. They were based on technical computer models and traded by

---

[1]https://app.1inch.io (this and all subsequent links were accessed on June 15, 2022).
[2]https://brave.com.
[3]https://www.bittorrent.com.
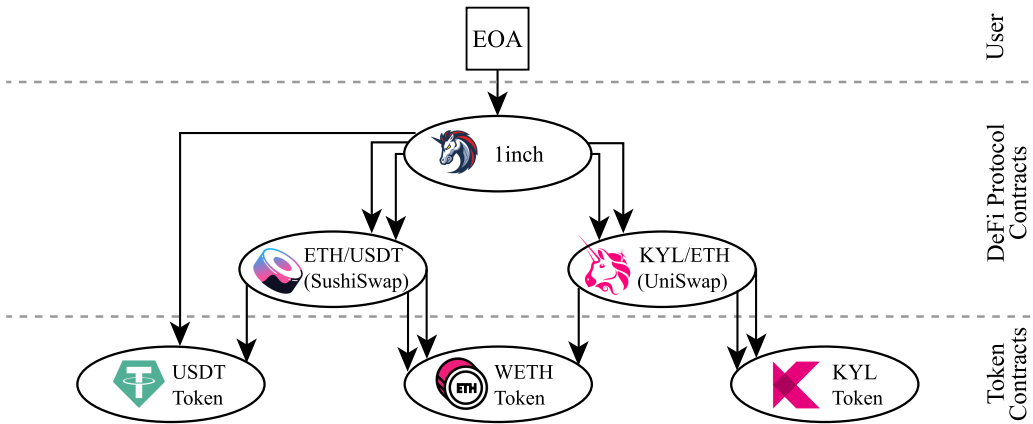[4]https://furucombo.app.

Fig. 1. A DeFi composition in which USDT tokens are swapped against KYL tokens through the DeFi service *1inch* in a single transaction. *1inch* executes the swap sequentially through the DeFi services *SushiSwap* and *UniSwap*, using WETH as an intermediary token. In the transaction trace graph, we can see the user calling the *1inch* smart contract, which, in turn, triggers several calls to DeFi protocol-, and token smart contracts.

highly leveraged institutions, many of whom did not understand the underlying models. These instruments were highly profitable, but the lack of any infrastructure and public information about them created a massive panic in the financial system that began in August 2007 [2]. DeFi protocols may offer opportunities, such as technological innovation or new governance models. However, their composability adds more complexity and opaqueness to an already complex cryptoasset ecosystem, which currently has a market valuation of about 1 trillion USD.[5] If these protocols are not understood and adopted more broadly, they could have unforeseeable systemic effects on financial markets and our society as a whole, as seen in the 2008 financial crisis [21]. A recent example involving DeFi protocols is the collapse of the stablecoin protocol Terra and its associated cryptoassets LUNA and UST. While the protocol did work as designed, its stabilization mechanism was not robust to significant selling pressure in the advent of market participants panicking. This ultimately led to deleveraging spiral effects [22] destroying over 30 billion USD of value within a single week and rendering institutions with large exposures to LUNA or UST insolvent. In addition, the stablecoin UST was used as part of compositions in many other DeFi protocols on the Terra blockchain and through bridges on different blockchains, affecting the entire ecosystem [35].

Previous work [10, 16] has partially studied risks in the DeFi ecosystem, showing possible strategies that allow rational agents to maximize their revenues by subverting the intended design of DeFi protocols, for example, in DEXs and lending protocols. However, none of the existing studies have systematically investigated compositions of DeFi protocols, which form complex, interconnected financial instruments.

1.2 Contributions

Our work aims to analyze DeFi protocols and to develop a novel algorithmic method that helps to understand protocol compositions. We can summarize our contributions as follows:

(1) We provide a manually curated ground truth of 1407 addresses from 23 DeFi protocols and derived 10,663,881 associated Ethereum smart contracts. These are labels that can be

---

   reused in future research. On this basis, we propose two network abstractions representing
   interactions among DeFi protocols and smart contracts (Section 3).

(2) We study intertwined DeFi protocols from a macroscopic perspective by analyzing the
   topology of both networks. We find that DEX and lending protocols have high degree
   and centrality values, and protocol interactions primarily occur in a strongly connected
   component. We also find that known community detection algorithms can indicate DeFi
   compositions but cannot effectively disentangle them (see Section 4).

(3) We address the microscopic transaction level and propose an algorithm for extracting the
   building blocks of DeFi protocols. We apply the algorithm to all protocol transactions in our
   ground truth, identify the most frequent building blocks, and find that swaps are the most
   frequent ones. We show what the observed space of compositions looks like for the *Aave*
   protocol. We also demonstrate, using *1inch* and *Instadapp* as examples, how to disentangle
   and visualize the building blocks of a single protocol as a treemap (Section 5.1).

(4) We present an overall picture of DeFi compositions by extracting and flattening the entire
   nested building block structure across multiple DeFi protocols. The results show that DeFi ag-
   gregation protocols (*1inch*, *0x*, or *Instadapp*) are, as expected, heavily intertwined with many
   other DeFi protocols, which confirms that our algorithm works as intended (Section 5.2).

(5) Finally, we present a case study illustrating how a hypothetical run on the stablecoin USD
   Tether would affect the building blocks of individual DeFi protocols (Section 5.3). We detect
   a comparatively high dependency of *Curvefinance* building blocks to the USDT cryptoasset.

We believe that our results are an essential contribution towards understanding DeFi compositions.
On a microscopic level, our proposed methods can be used to assess the composition of individual
protocols. On a macroscopic level, they show how DeFi protocols and their implementations are
connected with each other. For this article, we limit our scope to the largest Ethereum Virtual
Machine (EVM)–based blockchain Ethereum. However, in principle, the approach can be used and
applied to any other EVM-based platform. For reproducibility of results, we make our ground truth
dataset, including the labels and our source code, openly available at https://github.com/StefanKit/
Untangling_DeFi_Composition.

2   BACKGROUND AND DEFINITIONS

We now establish preliminary terms and definitions that are used throughout this work and
introduce the related works.

2.1   Ethereum Account Types

Ethereum is currently the most important distributed ledger technology (blockchain) for DeFi
services [53]. It differs from the Bitcoin blockchain conceptually as it implements the so-called
"account model" with two different account types. An *externally owned account* (*EOA*) is a "regular"
account controlled by a private key held by some user. A *code account* (*CA*), which is synonymous
with the notion "smart contract," is an account controlled by a computer program, which is invoked
by issuing a transaction with the code account as the recipient.

   A *CA* must always be initially called by an *external transaction* originating from an *EOA*, but a
CA can itself trigger other *CAs*. In the latter case, the interaction, which is also known as "message,"
is denoted as an *internal transaction*. Several branches of internal transactions with varying depth
can follow an external transaction, resulting in cascades, which altogether are called *traces*.

   CAs allow users to implement application-layer protocols, which are essentially programs that
can follow some standardized interface. *Tokens* are popular *CA*-based applications and a way to
define arbitrary assets that can be transferred between accounts. The program behind a token

manages token ownership and can implement a standardized interface such as ERC20, which defines functions standardizing token transfer semantics.

2.2 Decentralized Finance (DeFi) Protocol

A *DeFi protocol* is an application-layer program that provides financial service functions such as swapping or lending assets. More technically, we can define it as follows:

Definition 2.1. A DeFi protocol P is a decentralized application that facilitates specific financial service functions defined and implemented by a set of protocol-specific code accounts.

The following properties distinguish DeFi services from traditional financial services. First, they are *non-custodial*, meaning that no intermediary such as a bank or a broker holds custody of users' funds. Second, they are *permissionless*, meaning that anyone can use existing services or implement new services. Third, they are *transparent*, which means that anyone with the necessary technical capabilities and skills can investigate and audit the state of protocols. The fourth is that DeFi protocols are *composable*.

2.3 DeFi Protocol Compositions

The last property, *composability*, is the most crucial for this work and requires more detailed description. *CAs* can call each other, and their individual functions can be arbitrarily composed into new financial products and services ("Financial Lego") [49]. While this analogy is widely used in the literature, to the best of our knowledge, no work investigates *which are* the basic composable building blocks of more complex financial services and how they are related. Harvey et al. [19] refer broadly to composability as asset tokenization and networked liquidity, while Von Wachter et al. [44] conceive composability narrowly as a repeated wrapping operation of tokens resulting in new derivative products. However, as illustrated in Figure 1, we note that DeFi compositions also involve *CAs*, which are not tokens. Also, Engel and Herlihy [13] and Tolmach et al. [41] respectively discuss compositions only in the context of automated market makers (AMMs) and of formal verification of *CAs* related to decentralized exchanges and lending services, which is again a very narrow conception. Thus, there is no comprehensive, technically grounded definition for DeFi compositions to the best of our knowledge. For our work, we define it as follows.

Definition 2.2. A DeFi protocol composition occurs when a protocol-specific account leverages, within a single transaction, one or more accounts belonging to the same or another DeFi protocol to provide a novel financial service.

2.4 Related Work

Others studied networks closely related to the ones we investigated before us: Guo et al. [18] are amongst the first to investigate the Ethereum transaction graph, finding that volumes moved and the numbers of transactions follow a power law distribution, that the component structure follows a bow-tie model, and that negative assortativity is plausibly explained by the presence of service providers such as exchanges. Chen et al. [7] conduct a systematic study of Ethereum between 2015 and 2018 and exploit graph analysis measures to describe three different network constructions (money transfer, smart contract creation, and smart contract invocation). Another systematic study has been conducted by Lee et al. [24], who analyzed the local and global properties of interaction networks extracted from the entire Ethereum blockchain statically, finding heavy-tailed degree distributions. In a follow-up, Zhao et al. [54] analyzed the temporal evolution of Ethereum interaction networks and found that they proliferate and follow the preferential attachment growth model. Several studies focus on the network of Ethereum's tokenized assets. Somin et al. [39], for instance, studied the combined graph of all fungible token networks, whereas Victor and Lüders [43]

explored the networks of the top 1,000 ERC20 tokens individually. Fröwis et al. [14] proposed a method for detecting token systems independent of an implementation standard. Also, Chen et al. [8] conducted a systematic investigation of the whole Ethereum ERC20 token ecosystem and analyzed its activeness, purpose, relationship, and role in token trading. Other studies exploited network methods for the detection of specific nodes using graph-based approaches. Poursafaei et al. [32] developed a method based on graph node feature extraction and graph representation learning techniques to identify illicit nodes. Li et al. [25] and Ofori-Boateng et al. [30], instead, respectively use Topological Data Analysis (TDA) to detect price anomalies and hidden co-movement in pairs of tokens, and for anomalous events detection in a multilayer network. However, none of these related works considers networks that represent DeFi protocols and their relationships.

Another growing body of research concentrates on specific functions offered by individual DeFi protocols or types of protocols. We are aware of many DEX-related measurements focusing on protocol-specific aspects, such as the magnitude of cyclic arbitrage activity [47], the behavior of liquidity providers [48], or the role of oracles as providers of external information [26]. Other studies focus on lending and borrowing services: Perez et al. [31] analyze liquidations and related participants' behavior in the DeFi protocol *Compound*, whereas Gudgeon et al. [17] compare market efficiency, utilization, and borrowing rates in different lending protocols. Also, Wang et al. [45] provide methods to identify flash loans in three different DeFi providers and measure their related activity. Finally, we are aware that von Wachter et al. [44] investigate composability from an asset perspective and measure composability by identifying the number of derivatives produced from an initial root asset. However, we apply a more technical, service-oriented perspective and consider, to put it simply, a DeFi composition as being a computer program utilizing other programs' functions.

Overall, we are not aware of previous studies providing a comprehensive picture of DeFi compositions across various protocols. We also do not know of any work that analyzes in detail the building blocks of individual DeFi protocols. With this work, we want to close this gap.

3 DATASET AND NETWORK CONSTRUCTION

This section describes the data we collected and the network abstractions we constructed for subsequent analysis steps.

3.1 Dataset Collection

To study DeFi compositions, we are interested in transactions between Ethereum code accounts associated with known DeFi protocols. Thus, we used on-chain transaction data from the Ethereum blockchain and built a ground truth of known *CAs* and their associations to DeFi protocols.

3.1.1 On-chain Transaction Data. While Ethereum's history goes as far back as July 2015, DeFi only emerged as a popular term around summer 2020, when these protocols first saw increased usage. This informed our choice of the analysis time frame and the ability to refer to external sources providing information on popular, established DeFi services. We used an OpenEthereum client and ethereum-etl[6] to gather all Ethereum transactions from 01-Jan-2021 (block 11,565,019) to 05-Aug-2021 (block 12,964,999). We collected each external transaction and parsed its cascade of internal transactions, which together gave us the *trace*. For each transaction, we extracted the source and destination account addresses, the transaction hash, the transferred value, the transaction type (call, create, or self-destroy), and the trace ID, which indexes the transactions by their

---

[6]https://github.com/blockchain-etl/ethereum-etl.

execution order. Additionally, we collected the method ID of the 4-byte input sequence, which allows us to identify the signature of called methods using the 4Byte lookup service.[7]

To distinguish between *CAs* and *EOAs*, we gathered all code account creation transactions from the first *CA* created on Ethereum until the end of our observation period. We also use these *creation traces* to associate each *CA* with its creator *CA*. In total, we found 46,112,390 *CAs* and used the output byte sequence to identify 324,143 contracts conforming to the ERC20 standard.

3.1.2 Ground Truth Data. To be able to analyze DeFi protocols, we need a ground truth dataset in which smart contracts are part of a given protocol. We focus on the most relevant protocols regarding valuation and gas-burned between 06-Mar-2021 and 05-Aug-2021, using monthly samples of the top three total-value-locked (TVL) protocols from DeFi Pulse[8] for each financial service category. Additionally, we consider protocols including *CAs* of the top ten gas burner list[9] in the observation period. The result defines the set of DeFi protocols we want to investigate. Table 1 reports summary statistics for the 23 protocols in our sample, divided by category. The last column reports, for each protocol, the average share of each protocol's TVL with respect to the entire DeFi ecosystem between March and August 2021. In total, our 23 DeFi protocols cover more than 81% of the entire DeFi TVL. According to DeFi Pulse, in August 2021 more than 100 DeFi protocols existed, but only around 30 (18 of which are in our sample) had a TVL larger than 200 million USD. Most of the protocols in our sample are still the most relevant ones for TVL as of June 2022.[10] In the following, we briefly introduce the categories and protocols as reported by DeFi Pulse[11]:

- **Assets** identify the category including cryptoasset management protocols, such as yield aggregators, that aim at maximizing the value of a portfolio or basket of underlying assets. *Harvestfinance*, *Yearn*, and *Vesper*, share a similar mechanism, whereby they pool resources that, in turn, are invested in other DeFi platforms according to different optimization strategies. Users are typically rewarded through tokenized assets. *Convex* enables Curvefinance liquidity providers to earn additional rewards. *Badger* allows Bitcoin users to deposit tokenized Bitcoin, such as wBTC, and generate a yield by following programmatic optimization strategies. Similarly, *RenVM* bridges digital assets across DeFi ecosystems by minting ERC20 tokens on Ethereum with 1:1 ratio. *Fei*'s protocol builds on a decentralized stablecoin backed by cryptoassets exploited through yield strategies established by the protocol's governance.
- **Derivatives** protocols allow issuing synthetic financial instruments in the DeFi ecosystem, either tracking other cryptoassets or real-world off-chain assets. *Synthetix*, for instance, supports several real-world assets, such as fiat currencies and metals, while *dYdX* allows investors to trade perpetual positions on the underlying cryptoassets. *Hegic* enables the issuing of ETH and wBTC call and put options. *Futureswap* users can open leveraged long and short positions on cryptoassets. *Nexus*, instead, provides financial insurance instruments that cover potential losses users might incur; similarly, *Barnbridge* offers tools to hedge risk through its financial instruments.
- **DEXs**, that is, Decentralized Exchanges, allow users to exchange cryptoassets. *UniSwap*, *SushiSwap*, *Curvefinance*, and *Balancer* all exploit AMMs, as well as bonding curves and constant functions, to algorithmically set the cryptoasset prices, whereas *0x* is based on the

---

[7]https://www.4byte.directory/.

[8]https://defipulse.com/.

[9]https://ethgasstation.info/gas-burners.

[10]10 out of the first 11 DeFi protocols for TVL in DeFi Pulse are in our dataset.

[11]DeFiPulse reports the protocols divided into five categories. We do not include the *Payment* category because services such as Polygon provide off-chain functionality rather than composable financial services or products.

Table 1. Ground Truth Dataset Summary Statistics

Protocol type	DeFi Protocol	Number of addresses		External calls	% TVL
		Seed	Extended seed		
Assets	Badger	64	278	258,773	1.09%
	Convex	22	131	147,855	1.13%
	Fei	40	37	146,691	0.28%
	Harvestfinance	101	803	119,631	0.46%
	RenVM	15	15	234,161	0.86%
	Vesper	44	44	94,189	1.19%
	Yearn	3	3	243,036	3.54%
Derivatives	Barnbridge	40	46	55,588	0.17%
	dYdX	38	38	107,264	0.14%
	Futureswap	9	10	6484	0.04%
	Hegic	8	8	8372	0.03%
	Nexus	24	26	20,067	0.57%
	Synthetix	271	272	611,942	2.55%
DEX	0x	28	50	2,094,335	- %
	1inch	15	10,338,305	1,277,641	0.52%
	Balancer	9	3473	281,530	2.29%
	Curvefinance	163	267	745,672	9.28%
	SushiSwap	12	1705	2,026,674	5.37%
	UniSwap	15	54,038	28,394,798	8.30%
Lending	Aave	157	166	851,578	13.31%
	Compound	67	65	741,069	11.48%
	Instadapp	72	32,770	97,080	7.39%
	Maker	190	231,261	2,992,692	11.77%

Seed addresses were manually collected for each DeFi protocol. The extended seed are heuristically derived and also include further created code accounts from the seed addresses.

order book mechanism. The *1inch* protocol aggregates information on liquidity from several DEXs and routes transactions to those offering the best prices.

- **Lending** protocols provide investors with automated markets for loanable funds: lenders issue interest-bearing instruments and borrowers can take positions, typically conditional to the provision of collateral that covers potential losses. *Aave* and *Compound* follow the model described earlier. *Maker* users lock their cryptoassets as collateral and receive the DAI token in return. *Instadapp* follows a more complex scheme and acts mostly as an aggregator of multiple DeFi protocols.

After identifying the most relevant DeFi protocols, we manually collected the *CAs* associated with each protocol. Since this information is not available on the blockchain, we rely on off-chain and publicly available sources such as protocol websites and available documentation. We resolved conflicts of duplicated *CA* to protocol assignments and identical names by querying *CA* addresses on Etherscan[12] and uniquely assigned each *CA* address to its original protocol and obtained a unique label. We denote these manually collected data points as *seed data* and make them available as part of our source code repository.
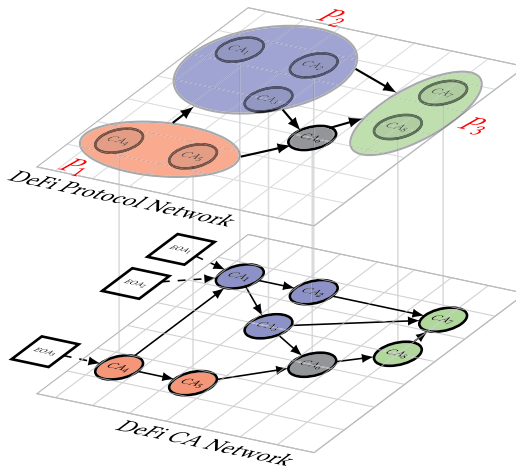
---

[12]https://etherscan.io/.

Fig. 2. Schematic illustration of constructed networks. The lower-level DeFi Code Account (CA) network represents interactions between *CAs*. The higher-level DeFi Protocol Network models relations between DeFi protocols. Lower-level *CA* vertices are associated with higher-level protocol vertices. *CAs* are triggered by *EOAs* or other *CAs*.

Next, we extended our seed data by implementing a heuristic that uses the creation transactions and identifies the *CAs* deployed by each seed address. By default, all extended addresses inherit the label and protocol assignments from the corresponding seed address. If the procedure leads to a conflict of labels for an address, we preserve the one obtained through the heuristic. Combined with our seed data, these extended addresses form our *extended seed data* dataset. Table 1 summarizes the number of seed and extended addresses collected for each DeFi protocol. It shows that our automated expansion does not increase the number of addresses associated with DeFi protocols for assets and derivatives. However, it massively expands the dataset for DEXs and lending protocols utilizing automated factory contract deployments. In particular, more than 10 million additional *CAs* are associated with *1inch* due to the factory contract that deploys gas tokens. The last column shows the number of external transactions directed to each of our DeFi protocols. The distribution is heterogeneous and, again, the most relevant categories are DEX and lending. *UniSwap* is the most frequently appearing one, with a gap of around one order of magnitude to the second one, which is *Maker*.

3.1.3 Dataset Reduction. As we are interested in known DeFi protocols only, we finally limited and reduced the *traces* dataset to the subset of *protocol traces*, in which the initial external transaction originating from an *EOA* triggers a *CA* address in our extended seed dataset. This reduction allows us to investigate and interpret compositions within the context of known protocols.

3.2 Network Construction

In our analysis, we want to understand and discover relations between DeFi protocols and associated *CAs*. For that purpose, as shown in Figure 2, we constructed networks consisting of DeFi traces on two abstraction levels: the lower-level *DeFi Code Account (CA) network* and the higher-level *DeFi Protocol network*.

The DeFi *CA* network includes all known ground truth *CAs* triggered by external transactions from arbitrary *EOA* addresses and all *CAs* subsequently called by cascades of internal transactions. We note that *CAs* in the network can or cannot be associated with a DeFi protocol in our ground

Table 2. Summary Statistics of the Analyzed Networks

	DeFi CA Network	DeFi Protocol Network
Nodes	2,536,371	43,624
Edges	3,472,757	84,789
Self-loops	6668	146
Average degree	1.369	1.944
Density	5.398e-07	4.456e-05

truth dataset. We construct the network by filtering all internal and external transactions between *CAs* from the *protocol traces*. Since repeated usage of DeFi services results in recurring transaction patterns, we aggregate and count transactions with the same source and destination address.

The DeFi Protocol network represents interactions between protocols. We constructed it by merging all DeFi *CA* vertices associated with the same DeFi protocol into a single node. We modeled both networks as a directed graph, in which vertices represent either a protocol or a single *CA*. The weighted edges represent the aggregated set of transactions between DeFi protocols or *CAs*.

4 TOPOLOGY MEASUREMENTS

We now analyze the constructed networks from a macroscopic perspective. Since our research focuses on understanding DeFi compositions, we do not aim to conduct an encompassing study of the entire Ethereum topology as it was done in previous studies (see Section 2.4). This supports our choice to focus on a narrower number of targeted metrics that provide relevant insights on composability aspects. Other approaches that are beyond the scope of our work are discussed in Section 6.2. The analysis of the degree distribution and centrality measures can help identify the CAs implementing core functionalities, and the reciprocity and assortativity provide additional insights on the relationships across such CAs. To understand how CAs associated to the same protocols interact with each other, we investigate how the network is separated in different components and whether known community detection algorithms identify community structures that overlap or not with the protocol structures.

We start by reporting basic summary statistics for the DeFi CA network and the DeFi Protocol network in Table 2. The main difference is in the network dimension, the latter being two orders of magnitude smaller. The presence of self-loops indicates that some contracts include multiple functionalities and, thus, can also call themselves. Both networks are sparse, as shown by the average degree and density measure, suggesting that *CAs* tend to interact with only a few other *CAs*.

4.1 Degree Distribution

Looking at the TVL at DeFi Pulse, we can observe that some DeFi protocols and their contracts play a major role. This observation suggests that they might implement core functionality, which other protocols in DeFi compositions can utilize in turn . Under this assumption, preferential attachment [3, 33] is a plausible generative mechanism for both networks. More generally, networks whose degree distribution follows a power law, that is, the fraction of vertices with degree k is given by $P(k) \sim k^{-\alpha}$ for values of $k \geq k_{min}$, are often associated with this generative mechanism. Thus, we estimate the parameters $\hat{\theta} = (\hat{k}_{min}, \hat{\alpha})$ for our two networks and investigate whether the power law distribution is a good fit.

We rely on the methodology introduced by Clauset et al. [9] and Broido and Clauset [6]: evidence of scale-free properties exist either when no alternative heavy-tailed distribution is
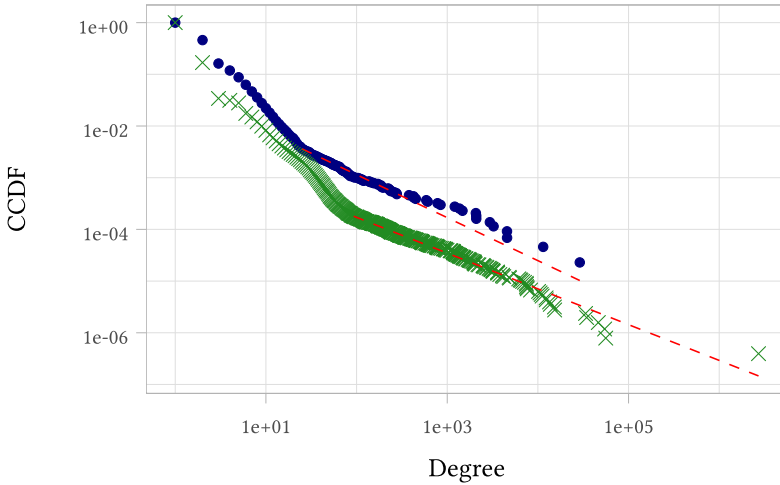
Fig. 3. Degree distribution of the *CA* (×) and Protocol (●) networks are shown in the plot as cumulative distribution function (CCDF). The estimated parameters $\hat{\theta} = (\hat{k}_{min}, \hat{\alpha})$ are $\hat{\theta}_{CA} = (93, 1.69)$ and $\hat{\theta}_P = (25, 1.83)$, respectively. In both networks, high-degree nodes are associated with DEX or lending protocols. For the *CA* network, they are routing contracts or factory contracts that deploy other contracts. Nodes with high degree are likely to contain core functionalities and, thus, to play a relevant role in compositions.

relatively better than the power law or when the power law is a plausible model for the distribution. In the former case, the network exhibits *Super-Weak* scale-free structure. In the latter, evidence of scale-free properties is said to be *Weak* if the tail of the distribution contains at least 50 nodes, and *Strong* if also $2 < \hat{\alpha} < 3$ holds. We start by estimating the parameters $\hat{\theta} = (\hat{k}_{min}, \hat{\alpha})$ by minimizing the Kolmogorov–Smirnov distance between empirical and fitted data for $\hat{k}_{min}$, and exploit it to estimate $\hat{\alpha}$ through the method of maximum likelihood estimation [9]. We then conduct a goodness-of-fit test via a bootstrapping procedure ($N = 5,000$). The resulting p-value indicates whether the power law is a plausible fit ($p \geq 0.1$) for the empirical data or not. Finally, we conduct a log-likelihood ratio ($\mathcal{R}$) test to compare the power law fit against other heavy-tailed distributions (i.e., the Exponential, the Lognormal, and the Weibull). A positive value indicates that the power law distribution is favored over the alternative, and the statistical significance is supported by a p-value that indicates whether the hypothesis $\mathcal{R} = 0$ is rejected ($p < 0.1$) or not ($p \geq 0.1$).

Figure 3 shows the power law fit for both networks and their estimated $\hat{k}_{min}$ and $\hat{\alpha}$. As with other studies on the interaction networks from Ethereum blockchain data [24], α lies around 1.7 and 1.8, which is slightly smaller than the average values usually found for power law distributions. The hypothesis that a power law distribution is a good fit is not plausible for both networks because p-values are 0.020 and 0.035 for the *CA* and Protocol networks, respectively. Table 3 reports the comparisons with other heavy-tailed distributions and shows that the power law is not significantly favored over the Lognormal distribution for both networks, whereas it is a better fit than the Weibull and the Exponential for the Protocol network. In summary, according to the classification proposed in Broido and Clauset [6], both networks have *Super-Weak* scale-free properties. Table 4 inspects the tails of the distributions and reports the top 15 *CAs* sorted by highest degree: most of the *CAs* are associated with a few DEX and lending protocols (*1inch, UniSwap, 0x, Instadapp, Maker*). We can hypothesize that they are part of DeFi compositions, which we will explore further in subsequent sections.

Table 3. Likelihood Ratio and p-Value

	DeFi CA Network	DeFi Protocol Network
Exponential	$\mathcal{R}$: 1.322, p-val: 0.186	$\mathcal{R}$: 4.753, p-val: 0.000
Lognormal	$\mathcal{R}$: −0.406, p-val: 0.685	$\mathcal{R}$: 0.191, p-val: 0.848
Weibull	$\mathcal{R}$: 1.122, p-val: 0.262	$\mathcal{R}$: 2.742, p-val: 0.006

None of the reported heavy-tailed distributions is favored over the power law.

Table 4. First 15 *CAs* by Highest Degree

Address	Label	Protocol	Degree	In degree	Out degree
0x00000000000049…	CHI Token	1inch	2,713,153	305,627	2,407,526
0x7a250d5630b4cf…	UniswapV2Router02	UniSwap	56,007	1711	54,296
0xc02aaa39b223fe…	EtherToken-v4	0x	54,469	45,129	9340
0x5c69bee701ef81…	UniswapV2Factory	UniSwap	46,408	26,576	19,832
0x2971adfa57b20e…	Mainnet-InstaIndex	Instadapp	34,497	18,369	16,128
0x4c8a1beb8a8776…	Mainnet-InstaList	Instadapp	33,551	16,956	16,595
0x5ef30b99863452…	CDP_MANAGER	Maker	15,300	8940	6360
0x35d1b3f3d7966a…	MCD_VAT	Maker	15,214	15,214	0
0xa26e15c895efc0…	PROXY_FACTORY	Maker	13,718	1	13,717
0x0000000000b3f8…	GST2 Token	Unknown	13,447	7644	5803
0x11111112542d85…	contractAddress	1inch	12,371	2073	10,298
0x6b175474e89094…	MCD_DAI	Maker	12,314	12,314	0
0xdef1c0ded9bec7…	ExchangeProxy-v4	0x	11,147	1138	10,009
0x939daad09fc4a9…	mainnet-v1-InstaAccount	Instadapp	10,876	10,876	0
0xfd3dfb524b2da4…	N/A	Unknown	10,554	1547	9007

4.2 Centrality Measures

The results in the previous section highlight the relevant role of DEXs and lending protocols. Network centrality measures are another helpful tool to determine which nodes might implement core functionalities. We consider the In degree centrality, as we are interested in identifying relevant contracts that other protocols may use in DeFi compositions. To add further insights, we also provide the results for the Katz and PageRank algorithms. Katz centrality accounts for the importance of a node's neighbors. It is an extension of the eigenvector centrality that addresses issues arising with directed networks [28] by adding a constant initial weight to each node. PageRank takes into account the Out degree of nodes to control for the drawback of the Katz algorithm that peripheric nodes might get too high values if linked to a very central node. The values of each centrality metric are normalized to the range [0,1].

We find that both networks are dominated by a few nodes with relatively high values (for all centrality measures) with respect to the other nodes. The In degree values are almost always higher than the Katz values, which, in turn, are often slightly larger than the PageRank centrality values. Table 5 reports the values for the nodes with the highest centrality in the Protocol (left) and the DeFi CA (right) networks. We show only the first three nodes because the others have relatively smaller values in comparison. In the Protocol network, the most central nodes are two non-labeled *CAs*. When considering the ranking of the nodes in the highest 10 positions for at least one centrality measure, 10 DeFi protocols appear in the highest positions, and *Uniswap*, in particular, plays an important role. Thus, these protocols are heavily used by other non-labeled *CAs* in our dataset. *Uniswap*, *0x*, and *Maker* have higher centrality values with respect to the other protocols. The DeFi CA network is dominated by the *1inch* factory contract mentioned in Section 3.1.2 that

Table 5. In Degree, Katz, and PageRank Centrality Measures for the Three Most Central Nodes

Protocol Network				DeFi CA Network			
Address/Protocol	In degree	Katz	PageRank	Protocol_Address	In degree	Katz	PageRank
0x0000000000b3f8…	1	1	1	1inch_0x00000…	1	1	1
0xcc88a9d330da11…	0.371	0.168	0.111	0x_0xc02aa…	0.148	0.107	0.053
UniSwap	0.313	0.176	0.092	UniSwap_0x5c69b…	0.087	0.064	0.036

For the Protocol network (left), the column Address/Protocol reports the address of non-labeled *CAs* or the protocol name associated with the node. For the DeFi CA network (right), the column Protocol_Address reports the protocol associated with the *CA* and the *CA* itself.

deploys CHI tokens. Two other nodes with relatively high values are the WETH *CA* related to 0x and another factory contract associated with *Uniswap*. Considering again the nodes ranking in the highest 10 positions for at least one centrality measure, *CAs* associated with *Instadapp* and *Maker* appear repeatedly. Factory deployer contracts play a major role in the DeFi *CA* network. Note that, by definition, such contracts have a high Out degree, as their functional role is to deploy other contracts. Interestingly, the In degree centrality results show that they also have a relevant role as recipients of calls by other contracts of the network. In conclusion, these results are consistent with the findings of Section 4.1 in showing that DEX and lending protocols play a major role and may be involved in compositions.

4.3 Reciprocity and Assortativity

Next, we look at two measures that provide information on the relationship between nodes and their neighbors, that is, reciprocity and assortativity. Reciprocity is the likelihood that nodes are mutually linked. Values range from 0 to 1, the former meaning that the network is purely unidirectional and the latter indicating that all links are reciprocated. For both the DeFi CA and Protocol networks, the values (0.234 and 0.215, respectively) are similar to the one reported in [24]. We follow their interpretation that the presence of reciprocated links is a potential sign of composability, as it shows that smart contracts tend to rely often on each other. The lower value obtained for the Protocol network could be explained by the presence of many non-labeled (non-protocol-specific) *CAs*. If we further reduce the Protocol network by removing all non-labeled *CAs*, obtaining a graph abstraction of 23 nodes, the reciprocity (0.677) is much higher, indicating that protocols interact with each other more often and in a bidirectional way, a sign that compositions exist. Assortativity is a metric that indicates whether nodes with similar degrees tend to interact with each other ($1 > \rho > 0$) or whether nodes with high degrees interact more with low-degree nodes ($0 > \rho > -1$). Consistent with previous results on the Ethereum transaction network, both networks are disassortative (-0.473 for the DeFi CA network and -0.262 for the Protocol network), indicating heterogeneity and a sign that CAs with high degree are leveraged by many other CAs with a less relevant role in the ecosystem. As shown earlier, such nodes are often associated with DEX and lending protocols.

4.4 Components

Reciprocity shows that protocols interact bidirectionally with accounts related to other protocols. Thus, we look at metrics providing further insights on how the (code accounts of) different protocols fall into distinct disconnected components. We distinguish between *weakly* connected components, in which all of the nodes are connected by a path independently of the directions of the edges, and *strongly* connected, which considers the edge direction.

For the Protocol network, we find that the largest weakly connected component is equal to the entire network, while for the *CA* network, only 34 nodes are outside of the largest component.

Table 6. Description of the Three Largest Strongly Connected Components

		Largest		2nd largest		3rd largest	
	# Comp.	Nodes	Edges	Nodes	Edges	Nodes	Edges
Contract	2,155,707	305,581	611,160	69,116	370,833	5622	11,242
Protocol	33,832	5622	11,242	3948	14,264	36	71

For both networks, the pattern is fragmented; but, interestingly, the second largest strongly connected components are remarkably more interconnected, indicating that nodes in these components interact with many other nodes, a prerequisite for composition.
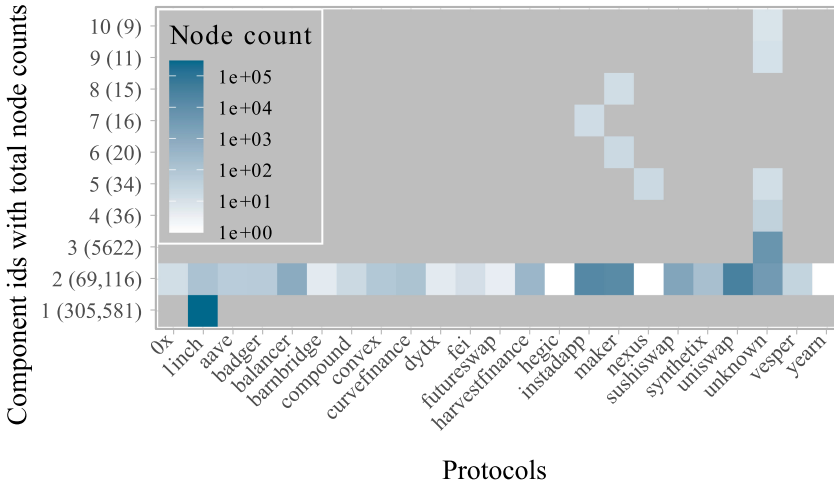

Fig. 4. Heatmap showing how the addresses associated with different protocols fall into the 10 largest strongly connected components. The largest component is uniquely composed of 305,581 *1inch* addresses, while the second collects the vast majority of protocols. Smaller components identify addresses of protocols that do not interact outside of the protocol itself.

The remaining nodes fall into 16 components, with a few nodes each. Table 6 lists the three largest strongly connected components. By comparing the number of edges and nodes, we notice that the second-largest component of both the Protocol and *CA* networks is denser than the other larger components. Additionally, in Figure 4 we illustrate how the *CAs* belonging to different protocols map to the 10 largest strongly connected components of the *CA* network. Interestingly, the second-largest component also encompasses the vast majority of protocol interactions. While the largest component is entirely composed of *CAs* associated with the *1inch* protocol, in the second-largest component, we find addresses of all of the analyzed protocols except for *RenVM*, which is not present in any of the reported large components. We also find that all of the protocols fall into the second-largest strongly connected components regarding the Protocol network. This analysis shows that interactions among protocols primarily occur in a single large component that is more interconnected than average. Notably, such interactions might indicate the existence of compositions due to the overlapping transaction structure of multiple protocols.

4.5 Community Detection

One could naively assume that *CAs* associated with specific DeFi protocols form communities in the *CA* network. However, the previous results suggest that the network topology reflects DeFi

Table 7. Performance Metrics for the Community Detection Algorithms

Algorithms	Communities	Precision	Recall	F1 Score	NMI	$\hat{C}/P^*$
Louvain	14	0.3896	0.7181	0.2917	0.9241	0.6087
Leiden	10	0.3021	0.8589	0.2879	0.9620	0.4348
Label prop.	53	0.7107	0.6009	0.4892	0.9404	2.3043
Eigenvector	4	0.1696	0.9070	0.1776	0.9495	0.1739

Low F1 scores indicate either that the algorithms poorly identify communities or that the network topology reflects a more complex organization at the mesoscopic level.

compositions at the level of the community structure. Thus, we measure how effectively different community detection algorithms detect protocols in the DeFi *CA* network.

We follow the approach of Yang et al. [51], who provide guidelines for selecting community detection algorithms depending on the size of the network. We analyze the weakest largest component in its unweighted and undirected version with non-overlapping communities using four different algorithms: multilevel or Louvain [5], label propagation [34], leading eigenvector [29], and Leiden [42]. Using the labeled addresses in our *ground truth* dataset, we can verify to what extent $\hat{C}$, the set of communities identified by partitioning algorithms, correspond to P^*, the set of ground truth communities defined by the individual protocols. We quantify their performance through *normalized mutual information (NMI)*, a benchmark measure in the literature [11, 23] that quantifies the similarity between the ground truth communities and the identified communities. We also provide two additional measures: the ratio $\hat{C}/P^*$ for the accuracy of the number of identified communities and the F1 score. We compute the latter in a manner similar to the authors of [50]. First, for each protocol $P_i \in P^*$, we identify the detected community $C_j \in \hat{C}$ that maximizes the F1 score. Then, we report average precision, recall, and F1 scores over all communities $P_i \in P^*$. Note that we compute these metrics only on the labeled *CAs*. The second column of Table 7 reports the total number of communities that include labeled *CAs*. The NMI is high for all of the protocols, indicating that the algorithms overall correctly partition the network. Indeed, all algorithms cluster together the *CAs* created by the *1inch deployer* contract, and *1inch* is by far the largest ground truth community in terms of labeled accounts. On the other hand, the low F1 scores (0.18–0.49) result from a small set of misclassified ground truth communities (e.g., *Compound*, *DyDx*, *Fei*). Upon closer inspection, we noticed that some protocols map entirely into a few communities dominated by larger protocols (such as *UniSwap* or *Maker*), negatively impacting precision, while others are split into different communities, affecting recall. *1inch* itself has a non-marginal number of addresses that map into other communities.

In summary, we see that algorithms work well, with NMI scores above 0.92. However, when considering the imbalance in our dataset (precision, recall), we find that known community detection algorithms cannot effectively identify protocols as distinct communities; rather, they indicate protocol composition patterns. The identified community structure reflects a different organization in which protocols are entangled.

5 MEASURING DEFI COMPOSITIONS

After analyzing the macroscopic network perspective, we now address the microscopic trace level, in which we identify and extract building blocks, that is recurring patterns of internal traces induced by protocol-specific *CAs* that are found as subpatterns within different transactions. The building block detection can help better understand DeFi compositions and identify a variety of risks. We consider a detailed risk analysis to be future work but can motivate some sources of risk: for example, if security vulnerabilities are identified in underlying building blocks, they can

(a) Original transaction trace graph of the composition as shown in Figure 1.

(b) Conversion to execution tree, renaming factory deployed contracts and assets.

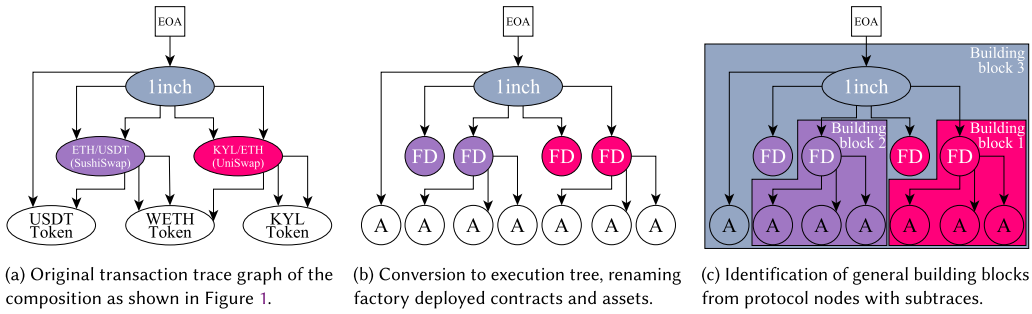(c) Identification of general building blocks from protocol nodes with subtraces.

Fig. 5. A high-level illustration of the building block extraction algorithm. (a) represents the input composition. This graph is then converted into an execution tree, as shown in (b), such that each node can have only one incoming edge, requiring the duplication of nodes. In addition, the underlying assets (tokens) and factory deployed contracts are renamed. In this example, the trading pair contracts are factory deployed (FD). This allows for the identification of generalized building blocks, as each trading pair differentiates itself only by the specific assets it is dealing with. The result of the building block extraction is shown in (c), which is the result of a bottom-up processing of the tree, selecting subtrees of known protocol nodes. See Algorithm 1 for more details.

propagate to higher levels and pose a risk to other DeFi protocols. Atzei et al. [1] analyze the security vulnerabilities of Ethereum code accounts and attacks that exploit them. Legal issues may arise, including licensing issues, thereby limiting usability in other protocols. This phenomenon also exists in traditional software.[13] Finally, the technical evolution of a blockchain can also have an impact on the efficiency or security of an existing building block. Here, too, it is important to identify which protocols are affected.

Thus, we propose an algorithm to extract the possibly nested structure of DeFi protocol calls, which may also be used by other DeFi protocols. In contrast to recent works that have discovered and exposed DeFi compositions, we provide a systematic, automated mechanism to explore them by using building block extraction. We then assess the most frequent building blocks our algorithm identifies and illustrate possible DeFi compositions, showing how the DEX aggregator *1inch* and the *Instadapp* protocols use multiple such building blocks of other protocols. Further, we flatten the nested structure of building blocks and study the interaction of DEX and lending services. Finally, we present in a case study the dependencies of DeFi protocol on stablecoins by using our extracted building block.

5.1 Building Block Extraction Algorithm

In order to detect building blocks, we treat individual transactions as trees of execution traces, that is, as an abstraction in which the external and all internal transactions are represented as an edge to a new node (thus, the same *CA* appears multiple times if executed more than once). We break the trees into subtrees starting from the tree's leaves and identify a building block whenever we encounter a node that is part of a protocol. If multiple protocol nodes exist in a tree, the building blocks can be composed of one another. To obtain the nested structure, we create a hash of each building block and use those hashes to chain nested tree structures. Figure 5 illustrates the process from a high-level perspective. Figure 5(a) represents the input, which corresponds to the original transaction trace graph also shown in Figure 1. We aim to identify building blocks that execute

---

[13]https://www.techradar.com/news/this-popular-code-library-is-causing-problems-for-hundreds-of-thousands-of-devs.

the same logic despite being different instances involving different addresses (i.e., a swap with different tokens). We preprocess and generalize the execution trace trees as follows:

Preprocessing: In contrast to a graph, like in Figure 5(a), an execution tree can have the same node appearing multiple times as a leaf node, effectively having no cycles. Each edge has a trace ID that determines the order of the calls. If a contract address appears in a trace that has been deployed by a factory, we rename it `$protocol-DEPLOYED`. We rename all contract addresses as `ASSET`, which fulfills the criteria that their smart contract code contains the standard ERC20 token method signatures and, if within the trace, the token contract is called with one such method. The result of these preprocessing steps is shown in Figure 5(b). This preprocessing assumes that factory-deployed contracts and ERC20 token contracts provide similar functionality. This allows us to generalize the traces, as many similar interactions with various standardized tokens become identical.

ALGORITHM 1: Building Block Extraction

Inputs: (1) Directed, attributed transaction trace tree $G(V, E, t, m)$ with functions $t : E \to \mathbb{N}$ assigning a unique trace ID, and $m : E \to \mathbb{N}$ assigning a method ID on the edges of the tree,

(2) protocol vertices V_P

Outputs: Lists of building blocks L_B, and hashes L_B^h

1   $L_B \leftarrow (\ )$;   // Init. list of building blocks

2   $L_B^h \leftarrow (\ )$;   // Init. list of building block hashes

3   $L_e \leftarrow (uv) | \forall uv \in E : v \in V_P$;   // Edges to protocol nodes

    // For each edge to a protocol, get subtree

4   $L_E^p \leftarrow (E_i) |$ edges reachable from e_i for each $e_i \in L_e$;

5   $L_G^p \leftarrow (G[E_i]) \mid \forall E_i \in L_E^p$;   // edge induced subtrees

6   $L_G^p \leftarrow$ filter(L_G^p, by=tree-depth, minimum=2);

7   $L_G^p \leftarrow$ sort(L_G^p, by=tree-depth, how=ascending);

8   **for** $G_S(V_S, E_S, t, m) \in L_G^p$ **do**   // for each subtree

9   // Compute building block hash with V_S', D_S, M_S

10     $E_S' \leftarrow$ sort(E_S, by=$t(E_S)$, how=ascending);   // Sort edges

11     $V_S' \leftarrow (v_1, ..., v_n) = v | \forall uv \in E_S' : v \in V_S$;   // Vert. list

12     $D_S \leftarrow deg_{out}(v) | \forall v \in V_S'$;   // Outdegree list

13     $M_S \leftarrow m(E_S')$;   // Method ID list

14     $h_S \leftarrow$ sha256hash(stringify(V_S', D_S, M_S));

15     $B_S \leftarrow G[V_S']$;   // B. block as vertex induced subtree

16   replace(what=G_S, in=G, with=h_S);

17     $L_B \leftarrow L_B \parallel B_S$;   // Append building block

18     $L_B^h \leftarrow L_B^h \parallel h_S$;   // Append building block hash

19   **end**

20   **return** L_B, L_B^h

Algorithm 1 takes as input a transaction trace tree $G(E, V, t, m)$ with two edge attributes: the trace ID t, indicating the order of execution, and m, indicating the method ID of the executed call. The second input is a list of seed protocol nodes, such as those described in Section 3.1.2. The algorithm outputs a list of building blocks and hashes of these building blocks. We first set up the output variables in lines 1 and 2. We then find edges to the protocol nodes in line 3 and
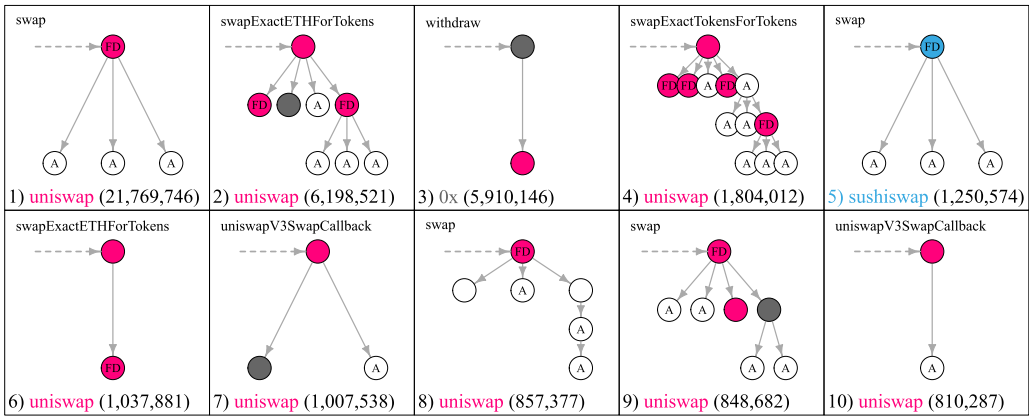
Fig. 6. The 10 most frequently observed building blocks by called root method, root protocol, and count. Nodes marked with FD are generalized factory-deployed contracts and those marked with A are ERC20 assets. The majority of these building blocks originate from *UniSwap*. Note that block 1 of *UniSwap* is equivalent to number 5 of *SushiSwap*. This makes sense, as *SushiSwap* is a fork of *UniSwap*. Number 1 is contained in building blocks 2 and 4 – illustrating an internal composition within the same protocol. Building block 3 represents the withdrawal of Wrapped Ether (WETH) and is associated with the protocol *0x*. Also note that several root methods are identical yet can lead to different types of building blocks.

extract all further reachable edges of these to obtain edge-induced subtrees in lines 4 and 5. We filter them in line 6 to include only those with a minimum depth of 2, such that the protocol node has to make further calls. In line 7, we sort the list of subtrees in ascending order based on their depth. This means that small trees are at the beginning of the list and large trees that may contain these smaller trees are at the end. For each subtree (line 8), we compute a hash in lines 9 to 14, highlighted in gray, akin to a tree kernel. To compute the hash, we first sort the subtree's edges by order of execution in line 10 and then extract the target vertices of each edge in line 11, essentially excluding the original calling node, which could be different in each transaction. For each of those vertices, we compute the outdegree (line 12) and determine the method ID for each edge (line 13). The hash is then computed from the three aforementioned properties in line 14. Using the target vertices, we retrieve the building block from the original tree (line 15), which may contain leaf nodes of building block hashes, as replacing subtrees in line 16 can lead to nested building blocks. Finally, we append building blocks and hashes to their lists in lines 17 and 18. Once all subtrees are processed, the lists are returned in line 20.

An example of the algorithm's result can be seen in Figure 5(c), showing three building blocks, one each from *SushiSwap*, *UniSwap*, and *1inch*. Note that the building block of *1inch* contains the other two building blocks.

5.2 Building Block Analysis

We execute the algorithm on all transactions in our dataset together with the set of DeFi protocols in our labeled extended seed set (cf. Section 3). We can then count the retrieved building blocks by their hashes, understand their composition, and visualize them. Figure 6 illustrates the top 10 most frequently observed building blocks, 8 of which belong to *UniSwap*. The most frequent building block is a *UniSwap* swap, with more than 21 million occurrences. As *UniSwap* is one of the most popular DeFi protocols, and token swaps are its main functionality, this result shows
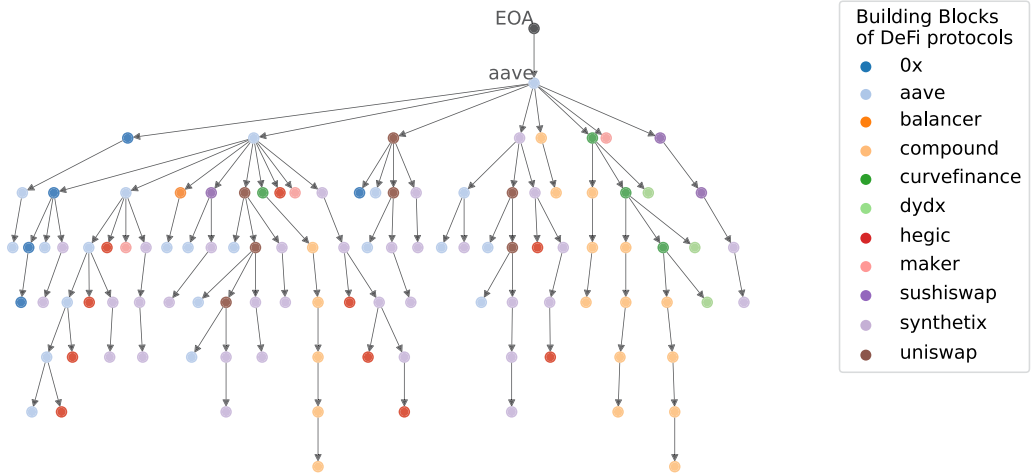
Fig. 7. Illustration of the composition space of *Aave* as a network tree. Each node represents a building block. Each link represents a possible nested building block extracted from all transactions to *Aave*. We observe for this protocol a maximum depth of seven nested DeFi building block levels.

that the building block extraction is meaningful. We further observe that the swap building block is reoccurring and contained in other patterns that appear frequently. Another relevant block is related to *0x*'s Wrapped Ether (WETH), which in our context is not classified as an asset due to its use of withdrawal, a non-ERC20 function. In the following, we will provide more insights into the nested structure from different perspectives and discuss their interpretations.

5.2.1 Protocol Building Block Composition. Starting from the execution tree structure of each trace, the algorithm identifies subtrees. Those building blocks obtained from Algorithm 1 can contain leaves with hashes that point to other building blocks, leading to a nested structure that still preserves the primary tree structure of the traces. However, a single transaction represents only a small snapshot of the entire tree of possible compositions. For a comprehensive image of the DeFi protocols composition space, we have to consider multiple transactions. To observe the space of all possible compositions, we construct a network of overlapping building block trees for all transactions of the same initial (external) DeFi protocol. For an illustrative example, we used the extracted building block structures of all transactions to *Aave*. The network still conserves the tree structure, where each node represents a building block and each link a nested composition, observed in the transactions. Figure 7 shows the *Aave* network and illustrates its multiple nested levels. Starting from the top with external transactions from EOAs to *Aave*, a variety of paths and compositions can be seen, presenting the space of all possible compositions, observed from existing transaction data. Nevertheless, this network illustration does not provide a comprehensive picture of the volume (i.e., number of appearances) of those compositions and the number of branches when a building block calls multiple sub-blocks.

We can inspect the set of contained protocols and the volume of their appearances for each building block: the treemaps in Figure 8 illustrate the shares of protocols appearing in the building block structure of a specific nested level.

In Figure 8(a), we observe the volume of building block calls and associated protocols in the first level for the protocols *1inch*. The largest fraction consists of external transactions that do not contain any other building blocks; this is captured by the box labeled as *NONE*. All other boxes show instead the share of transactions in which one or multiple DeFi service building blocks are
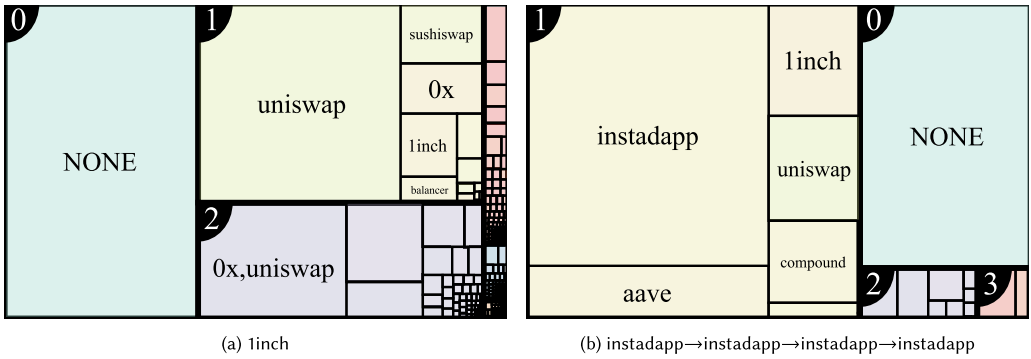
Fig. 8. Inspecting the potentially nested building blocks used by the first level of *1inch* (left) and the fourth level of *Instadapp* (right). The size of each box represents the share of building blocks assigned to one or more unique protocols. For *1inch* transactions, at the first nested level, about a third of the used building blocks are of one (chiefly other) protocol (yellow boxes). An even bigger fraction can be observed for *Instadapp* but in the fourth nested building block level.

nested. We group them using different colors based on the number of unique, distinct protocols that are called in the subsequent building blocks of this level. For instance, yellow boxes indicate the fraction of transactions in which the appearing nested building blocks in the first level are associated with one single DeFi protocol, whereas blue boxes represent the fraction in which the building blocks in the first level are associated with two different protocols. We further observe portions of transactions that contain building blocks assigned to more than two protocols within the first nested level.

The treemap in Figure 8(b) shows branches in a deeper level within *Instadapp* transactions. In the fourth level of self-compositions, in addition to the fraction that does not contain any further block (*NONE*), an even bigger share of building blocks appear that are associated with one single DeFi protocol. We also inspect again the existence of building blocks associated with two and more protocols.

These two illustrations in Figure 8 give insights into our systematic investigation on compositions, and show that looking only to selected compositions or single nested levels of DeFi compositions would return a partial picture: interactions among protocols can be iteratively nested one within the other and can take place in deeply nested levels. Therefore, a further investigation to disentangle and flatten the nested structure is needed.

5.2.2 Flattening Composition Hierarchies. We then want to investigate to what extent the DeFi protocols leverage other protocols to provide their services. That means that we want to identify a mapping of top-level protocols to any of the building blocks they use, whether deeply nested or not. To get an overall picture of the DeFi compositions, we flatten the nested building block structures.

In each transaction, we follow the cascade of nested building blocks and create a mapping from the contained protocol of building blocks to the original DeFi protocol that the external transaction was sent to (the root protocol). If mappings appear multiple times over different transactions, we aggregate them. For each root protocol, we can then compute the frequency of associated protocols to contain building blocks over all transactions. The result is a measure that indicates, for a given root protocol, the probability that a certain building block of a DeFi protocol appears anywhere in the (nested) building block structure. In Figure 9, we show the building block appearances of lending, DEX, derivatives, and asset protocols with a heat map. Each row corresponds to the external calls
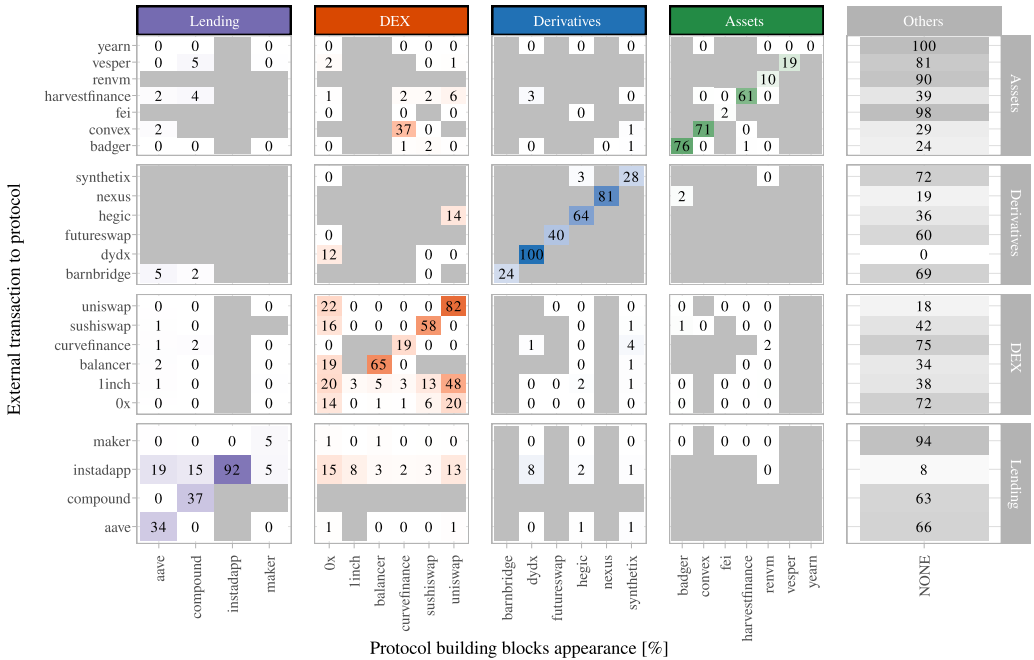
Fig. 9. Appearances of DeFi service building blocks across protocols. The numbers indicate the percentage of transactions in which a building block of a certain protocol is contained. The use of multiple DeFi services can be observed for DeFi aggregation protocols, such as *Instadapp*, *1inch* and *0x*.

to a specific protocol, and the row entries indicate the frequencies of the occurrence of a protocol's building blocks. The relative share measurement is the fraction of internal building blocks based on the number of external transactions. We note that the *NONE* category indicates the share of transactions for which no building blocks have been found. Most protocol interactions exist within each protocol, visible by the highlighted diagonal elements. This pattern is especially remarkable for derivative protocols. For example, consider *dYdX*: all external transactions directed to it contain at least one *dYdX* building block. However, DeFi aggregation protocols such as *Instadapp*, *1inch*, and *0x* in particular show extensive use of other DeFi services and, thus, frequent occurrences of DeFi compositions. This indicates that Algorithm 1 works as intended, as aggregation protocols must call other protocols by definition. The frequent appearance of the *0x* protocol can be attributed to the popular *Wrapped Ether* token and its *withdraw* pattern, already observed and shown in Figure 6. Further, we note that second to *0x*, *UniSwap* building blocks appear in most transactions to the protocols shown in Figure 9. Derivatives protocols have instead little or no further interactions with other protocols, as shown in the row associated with derivatives in the matrix of heat maps, as well as the assets protocols that do not interact heavily with other protocols.

5.3 Case Study: A Hypothetical Run on the Tether

In May 2022, we witnessed the collapse of the Terra ecosystem and its stablecoin TerraUSD (UST), which maintained its peg to the US dollar through an arbitrage mechanism with the token LUNA. This triggered a so-called stablecoin run and destroyed over 30 billion USD of value within a single week. Motivated by this recent demonstration of systemic risk associated with stablecoins, we apply our building block extraction and analysis methods to measure how a hypothetical run on
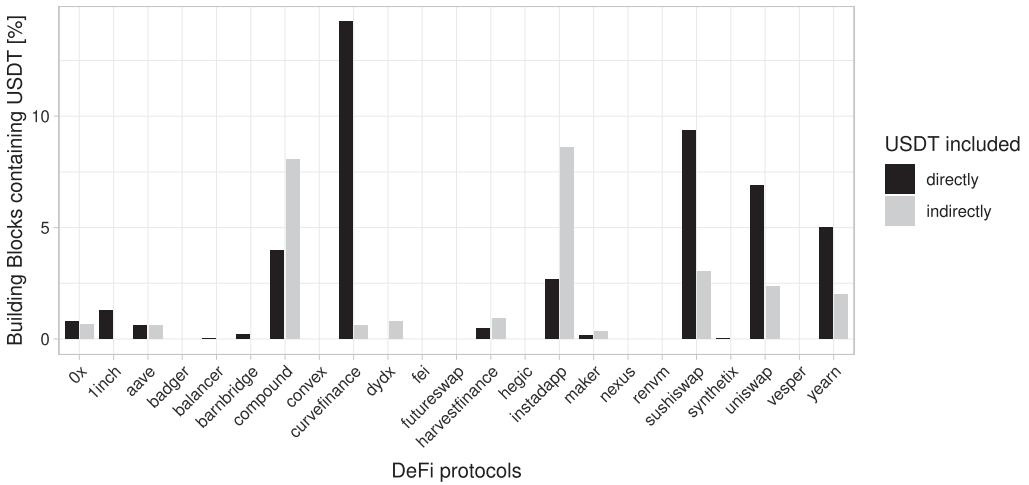
Fig. 10. Dependencies of building blocks on the USDT crypto asset for each DeFi protocol, distinguished between direct included asset or indirect through other nested contained blocks.

the stablecoin Tether (USDT),[14] which is the most widely adopted stablecoin in Ethereum, would affect known DeFi protocols based on building block dependencies. We distinguish between *direct* dependencies, in which USDT is an explicit part of a building block, and *indirect* dependencies, in which USDT appears somewhere in its nested building blocks. Starting with the most frequent building blocks (see Figure 6), we analyzed the occurrence of USDT in the regularly used sub-patterns of transactions. We detected USDT in 10.6% of 'swap' building blocks from *UniSwap* (1) and 16.2% from *SushiSwap* (5). For the 'swapExactTokensForTokens' building block from *UniSwap* (2), we find an even higher direct occurrence of 22.7% and an indirect dependency of an additional 21.2% with the nested block structure, containing the aforementioned 'swap' building blocks from *UniSwap* (1).

In order to obtain a broader picture of the dependencies in the DeFi ecosystem, we also analyzed, for each protocol, the fraction of building blocks containing the USDT asset directly or indirectly in more deeply nested blocks. Our results, which are summarized in Figure 10, show that most protocols have rather low dependencies (<10%). However, 14.2% of *Curvefinance* building blocks include the USDT asset directly as well as the two DEX protocols *UniSwap* and *SushiSwap*, which strongly depend on that asset. This is in line with our previous finding that 'swap' is the most frequent building block. We further find that *Compound* and *Instadapp* building blocks have, in comparison, high indirect dependencies on the USDT asset. These dependencies indicate how a shock in the DeFi ecosystem, such as a run on a stablecoin, could affect DeFi protocols directly and indirectly through their services. Since USDT has become a multi-chain asset, which is also traded and used on other blockchains (e.g., Binance Smart Chain and Avalanche), such shocks could also spread across chains and lead to systemic failures. However, we consider this analysis a first step towards a deeper investigation of systemic risk and plan a deeper investigation for future work.

6 DISCUSSION

In this section, we discuss some of the insights from our analyses as well as the limitations of our work.

---

[14]*0xdAC17F958D2ee523a2206206994597C13D831ec7.*

6.1 Insights

Cryptoassets are no longer a niche phenomenon. They reached an overall market capitalization of more than 2 trillion USD by November 2021 and are increasingly interconnected with the traditional financial systems. With DeFi, we now see the introduction of leveraged financial products and assets that are backed with some poorly understood virtual securities. Our results provide initial insights into the motivating questions mentioned in the introduction.

Concerning ecosystem interoperability, we found that compositions between DEX protocols are particularly frequent in our dataset (see Figure 9). From this, we can conclude that these protocols should ideally be deployed on the same DLT platform as long as single-transaction cross-chain compositions are not possible. At the same time, however, we also found that derivative protocols in particular still contain relatively few compositions. This suggests that a protocol-type specific scaling solution could be useful, for example, a sidechain for derivative protocols. Fewer compositions would still be possible but not with a significant negative impact as when separating DEX protocols.

As far as integration with web technologies is concerned, the versatile use of building blocks shows that elementary constructs are already reused and integrated by various applications without this necessarily being transparent to the users. The view is further reinforced when considering that various assets are already integrated into web technologies, but their simultaneous inclusion in financial instruments and compositions is barely obvious. An example of this is the BAT token, which is integrated into the Brave browser but is also used in various DeFi protocols.

Finally, turning to risks through complexity, we recall that the financial crisis in 2008 has shown that a lack of understanding and lack of regulation can have unforeseeable risks for the financial markets and our society as a whole. Whilst composability unleashes unexplored possibilities, it may also lead to unforeseen risks. Indeed, despite DeFi protocols being aware of and often even facilitating the use of their own *CAs* in composition with those of other protocols, these interconnected novel financial services lack a form of coordination on the resulting compositions. Thus, unintended forms of interaction across protocols could take place, exposing users to risk, even more so when calls are iteratively nested and several protocols are indirectly involved. If the DeFi ecosystem evolves at the current pace and integrates closely with the traditional financial sector, associated systemic risks must be understood and mitigated. Our work shows how DeFi protocols can be decomposed, and the share of protocol interactions can be visualized (see Figure 8). With our case study, we simulate a hypothetical run on Tether and show how our method can provide first insights into how DeFi protocols and their services could be affected, along with through cascading effects from other protocols. Our work highlights the potential and possibilities for further studies to evaluate systemic risk.

6.2 Limitations

We acknowledge and describe some limitations of our work. First, our results naturally reflect only the compositions of the protocols and labeled addresses contained in our ground truth dataset. Since the DeFi landscape is evolving rapidly, extending our seed data and the observation period as well as investigating the temporal evolution of the DeFi protocols are obvious next steps. One can then re-run our generally applicable analytics procedures. However, we emphasize that while a longitudinal analysis of DeFi usage in a longer time frame would be of interest, our main contribution is the devised methodology to uncover compositions. The time frame and extent of the DeFi protocol activity that we investigated are sufficiently large for this (static) analysis. Second, as we focused on composability, we did not investigate some features of the network topology, such as their small-world properties (e.g., clustering coefficients and path lengths). We

studied recurrent patterns by decomposing individual transactions as nested building blocks rather than studying triadic (or higher-order) motifs and core decomposition methods. Topological Data Analysis (TDA) has been exploited in the literature mostly in predictive models to identify anomalous patterns, which is beyond the scope of our work. Similarly, temporal aspects are left for future work, as discussed previously. In our network analysis, we currently neglect edge weights between *CAs*, which may indicate the strength of composition. Including them could also be part of future work. Third, our building block extraction algorithm currently yields the building blocks of known DeFi protocols. We believe that future work should aim at a more systematic evaluation using a curated ground truth of DeFi compositions. Finally, we point out that currently we mainly focus on single-transaction interactions between *CAs*. However, DeFi compositions could also be constructed by *EOAs* over time using multiple transactions. We do not yet consider this aspect in our analysis, but we deem it one of the most promising avenues for future work.

7 CONCLUSION

The overall goal of our work is to provide methods and results that contribute to a better understanding of DeFi protocols, which are a new family of financial products. We manually curated a ground truth set of 23 DeFi protocols, which can be reused in future research. We constructed network abstractions representing the interactions between smart contracts (*CAs*) and DeFi protocols and conducted a topology analysis in the time span from January 2021 to August 2021. The results indicate the existence of compositions, which is further supported by our finding that known community detection algorithms cannot disentangle DeFi protocols. Therefore, we proposed an algorithm that extracts the building blocks of DeFi protocols from transactions. We assessed the most frequent blocks and found that swaps play an essential role. We also analyzed individual DeFi protocols by disentangling their building blocks and flattened the composition hierarchies of all DeFi protocol transactions in our dataset. We provide a case study that discovers how the building blocks depend on the USDT stablecoin. This shows how the proposed method can help identify potential systemic risk, by measuring to what extent each protocol is affected by propagating shock of a single entity, originated from vulnerabilities, legal issues, or technical advances. Finally, we have discussed the implications and limitations of our work, providing first insights into questions about interoperability, integration with Web technologies, and systemic risks that may arise in complex financial systems.

In summary, our work is the first that investigates DeFi compositions across multiple protocols, both from a network perspective and at the level of individual transactions. We believe that our methods make an essential contribution to understanding the bigger picture and the basic building blocks of individual DeFi protocols and their relationships across protocols.

REFERENCES

[1] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. 2017. A survey of attacks on Ethereum smart contracts SoK. In *Proceedings of the 6th International Conference on Principles of Security and Trust - Volume 10204*. Springer, 164–186.

[2] Martin Neil Baily, Robert E. Litan, and Matthew S. Johnson. 2008. *The Origins of the Financial Crisis*. Technical Report. Brookings Institution.

[3] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *Science* 286, 5439 (1999), 509–512.

[4] Rafael Belchior, André Vasconcelos, Sérgio Guerreiro, and Miguel Correia. 2021. A survey on blockchain interoperability: Past, present, and future trends. *ACM Computing Surveys (CSUR)* 54, 8 (2021), 1–41. https://doi.org/10.1145/3471140

[5] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (2008), P10008.

[6] Anna D. Broido and Aaron Clauset. 2019. Scale-free networks are rare. *Nature Communications* 10, 1 (2019), 1–10.

[7] Ting Chen, Zihao Li, Yuxiao Zhu, Jiachi Chen, Xiapu Luo, John Chi-Shing Lui, Xiaodong Lin, and Xiaosong Zhang. 2020. Understanding Ethereum via graph analysis. *ACM Transactions on Internet Technology (TOIT)* 20, 2 (2020), 1–32.

[8] Weili Chen, Tuo Zhang, Zhiguang Chen, Zibin Zheng, and Yutong Lu. 2020. Traveling the token world: A graph analysis of Ethereum ERC20 token ecosystem. In *Proceedings of The Web Conference 2020 (WWW'20)*. Association for Computing Machinery, 1411–1421. https://doi.org/10.1145/3366423.3380215

[9] Aaron Clauset, Cosma Rohilla Shalizi, and Mark E. J. Newman. 2009. Power-law distributions in empirical data. *SIAM Review* 51, 4 (2009), 661–703.

[10] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. 2020. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE Symposium on Security and Privacy (SP'20)*. IEEE, 910–927. https://doi.org/10.1109/SP40000.2020.00040

[11] Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. 2005. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment* 2005, 09 (2005), P09008.

[12] DeFi Pulse. 2021. Total Value Locked (USD) in DeFi. Retrieved July 2021 from https://defipulse.com/.

[13] Daniel Engel and Maurice Herlihy. 2021. Composing networks of automated market makers. *arXiv preprint arXiv:2106.00083* (2021).

[14] Michael Fröwis, Andreas Fuchs, and Rainer Böhme. 2019. Detecting token systems on Ethereum. In *Financial Cryptography and Data Security*, Ian Goldberg and Tyler Moore (Eds.). Springer International Publishing, Cham, 93–112.

[15] Lewis Gudgeon, Pedro Moreno-Sanchez, Stefanie Roos, Patrick McCorry, and Arthur Gervais. 2020. SoK: Layer-two blockchain protocols. In *International Conference on Financial Cryptography and Data Security*. Springer, 201–226.

[16] Lewis Gudgeon, Daniel Perez, Dominik Harz, Benjamin Livshits, and Arthur Gervais. 2020. The decentralized financial crisis. In *2020 Crypto Valley Conference on Blockchain Technology (CVCBT'20)*. 1–15. https://doi.org/10.1109/CVCBT50464.2020.00005

[17] Lewis Gudgeon, Sam Werner, Daniel Perez, and William J. Knottenbelt. 2020. DeFi protocols for loanable funds: Interest rates, liquidity and market efficiency. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*. 92–112.

[18] Dongchao Guo, Jiaqing Dong, and Kai Wang. 2019. Graph structure and statistical properties of Ethereum transaction relationships. *Information Sciences* 492 (2019), 58–71.

[19] Campbell R. Harvey, Ashwin Ramachandran, and Joey Santoro. 2021. *DeFi and the Future of Finance*. John Wiley & Sons.

[20] Maurice Herlihy. 2018. Atomic cross-chain swaps. *CoRR* abs/1801.09515 (2018). arXiv:1801.09515. http://arxiv.org/abs/1801.09515.

[21] Andrei Kirilenko, Albert S. Kyle, Mehrdad Samadi, and Tugkan Tuzun. 2017. The flash crash: High-frequency trading in an electronic market. *The Journal of Finance* 72, 3 (2017), 967–998.

[22] Ariah Klages-Mundt and Andreea Minca. 2021. (In)stability for the blockchain: Deleveraging spirals and stablecoin attacks. *Cryptoeconomic Systems* 1, 2 (Oct. 22 2021). https://cryptoeconomicsystems.pubpub.org/pub/klages-mundt-blockchain-instability.

[23] Andrea Lancichinetti and Santo Fortunato. 2009. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E* 80, 1 (2009), 016118.

[24] Xi Tong Lee, Arijit Khan, Sourav Sen Gupta, Yu Hann Ong, and Xuan Liu. 2020. Measurements, analyses, and insights on the entire Ethereum blockchain network. In *Proceedings of The Web Conference 2020 (WWW'20)*. ACM, 155–166. https://doi.org/10.1145/3366423.3380103

[25] Yitao Li, Umar Islambekov, Cuneyt Akcora, Ekaterina Smirnova, Yulia R. Gel, and Murat Kantarcioglu. 2020. Dissecting Ethereum blockchain analytics: What we learn from topology and geometry of the Ethereum graph?. In *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 523–531.

[26] Bowen Liu, Pawel Szalachowski, and Jianying Zhou. 2020. A first look into DeFi oracles. *arXiv preprint arXiv:2005.04377* (2020).

[27] Debasis Mohanty, Divya Anand, Hani Moaiteq Aljahdali, and Santos Gracia Villar. 2022. Blockchain interoperability: Towards a sustainable payment system. *Sustainability* 14, 2 (2022). https://doi.org/10.3390/su14020913

[28] Mark Newman. 2018. *Networks*. Oxford University Press.

[29] Mark E. J. Newman. 2006. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E* 74, 3 (2006), 036104.

[30] Dorcas Ofori-Boateng, I. Segovia Dominguez, C. Akcora, M. Kantarcioglu, and Yulia R. Gel. 2021. Topological anomaly detection in dynamic multilayer blockchain networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 788–804.

[31] Daniel Perez, Sam M. Werner, Jiahua Xu, and Benjamin Livshits. 2020. Liquidations: DeFi on a knife-edge. *arXiv preprint arXiv:2009.13235* (2020).

[32] Farimah Poursafaei, Reihaneh Rabbany, and Zeljko Zilic. 2021. SigTran: Signature vectors for detecting illicit activities in blockchain transaction networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 27–39.

[33] Derek de Solla Price. 1976. A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science* 27, 5 (1976), 292–306.

[34] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E* 76, 3 (2007), 036106.

[35] Rahul Rai. 2022. The Death Spiral: How Terra's Algorithmic Stablecoin Came Crashing Down. Retrieved June 5, 2022 from https://www.forbes.com/sites/rahulrai/2022/05/17/the-death-spiral-how-terras-algorithmic-stablecoin-came-crashing-down/?sh=41275c6a71a2.

[36] Fabian Schär. 2021. Decentralized finance: On blockchain- and smart contract-based financial markets. *Federal Reserve Bank of St. Louis Review* 2 (2021), 153–74. https://doi.org/10.20955/r.103.153-74

[37] Cosimo Sguanci, Roberto Spatafora, and Andrea Mario Vergani. 2021. Layer 2 blockchain scaling: A survey. *arXiv preprint arXiv:2107.10881* (2021).

[38] Amritraj Singh, Kelly Click, Reza M. Parizi, Qi Zhang, Ali Dehghantanha, and Kim-Kwang Raymond Choo. 2020. Sidechain technologies in blockchain networks: An examination and state-of-the-art review. *Journal of Network and Computer Applications* 149 (2020), 102471.

[39] Shahar Somin, Goren Gordon, and Yaniv Altshuler. 2018. Network analysis of ERC20 tokens trading on Ethereum blockchain. In *Unifying Themes in Complex Systems IX*, Alfredo J. Morales, Dan Gershenson, Carlosand Braha, Ali A. Minai, and Yaneer Bar-Yam (Eds.). Springer International Publishing, Cham, 439–450.

[40] Louis Tremblay Thibault, Tom Sarry, and Abdelhakim Senhaji Hafid. 2022. Blockchain scaling using rollups: A comprehensive survey. *IEEE Access* (2022).

[41] Palina Tolmach, Yi Li, Shang-Wei Lin, and Yang Liu. 2021. Formal analysis of composable DeFi protocols. *CoRR* abs/2103.00540. (2021). arXiv:2103.00540. https://arxiv.org/abs/2103.00540.

[42] Vincent A. Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports* 9, 1 (2019), 1–12.

[43] Friedhelm Victor and Bianca Katharina Lüders. 2019. Measuring Ethereum-based ERC20 token networks. In *Financial Cryptography and Data Security — 23rd International Conference (FC'19), Frigate Bay, St. Kitts and Nevis, February 18-22, 2019, Revised Selected Papers (Lecture Notes in Computer Science)*, Ian Goldberg and Tyler Moore (Eds.), Vol. 11598. Springer, 113–129. https://doi.org/10.1007/978-3-030-32101-7_8

[44] Victor von Wachter, Johannes Rude Jensen, and Omri Ross. 2021. Measuring asset composability as a proxy for DeFi integration. In *International Conference on Financial Cryptography and Data Security*. Springer-Verlag, Berlin, Heidelberg, 109–114. https://doi.org/10.1007/978-3-662-63958-0_9

[45] Dabao Wang, Siwei Wu, Ziling Lin, Lei Wu, Xingliang Yuan, Yajin Zhou, Haoyu Wang, and Kui Ren. 2021. Towards a first step to understand flash loan and its applications in DeFi ecosystem. In *Proceedings of the 9th International Workshop on Security in Blockchain and Cloud Computing (Virtual Event, Hong Kong) (SBC'21)*. Association for Computing Machinery, New York, NY, 23–28. https://doi.org/10.1145/3457977.3460301

[46] Gang Wang. 2021. SoK: Exploring blockchains interoperability. *Cryptology ePrint Archive* (2021).

[47] Ye Wang, Yan Chen, Haotian Wu, Liyi Zhou, Shuiguang Deng, and Roger Wattenhofer. 2022. Cyclic arbitrage in decentralized exchanges. *Available at SSRN 3834535* (2022), 12–19. https://doi.org/10.1145/3487553.3524201

[48] Ye Wang, Lioba Heimbach, and Roger Wattenhofer. 2021. Behavior of liquidity providers in decentralized exchanges. *arXiv preprint arXiv:2105.13822* (2021).

[49] Sam M. Werner, Daniel Perez, Lewis Gudgeon, Ariah Klages-Mundt, Dominik Harz, and William J. Knottenbelt. 2021. SoK: Decentralized Finance (DeFi). (2021). arXiv:cs.CR/2101.08778.

[50] Jaewon Yang and Jure Leskovec. 2012. Community-affiliation graph model for overlapping network community detection. In *IEEE 12th International Conference on Data Mining*. IEEE, 1170–1175. https://doi.org/10.1109/ICDM.2012.139

[51] Zhao Yang, René Algesheimer, and Claudio J. Tessone. 2016. A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports* 6, 1 (2016), 1–18.

[52] Alexei Zamyatin, Mustafa Al-Bassam, Dionysis Zindros, Eleftherios Kokoris-Kogias, Pedro Moreno-Sanchez, Aggelos Kiayias, and William J. Knottenbelt. 2021. SoK: Communication across distributed ledgers. In *International Conference on Financial Cryptography and Data Security*. Springer, Springer-Verlag, Berlin, Heidelberg, 3–36. https://doi.org/10.1007/978-3-662-64331-0_1

[53] Dirk A. Zetzsche, Douglas W. Arner, and Ross P. Buckley. 2020. Decentralized finance. *Journal of Financial Regulation* 6, 2 (2020), 172–203.

[54] Lin Zhao, Sourav Sen Gupta, Arijit Khan, and Robby Luo. 2021. Temporal analysis of the entire Ethereum blockchain network. In *Web Conference 2021 (WWW'21)*. Association for Computing Machinery, New York, NY, 2258–2269. https://doi.org/10.1145/3442381.3449916