Blockchain: A Graph Primer

CUNEYT GURCAN AKCORA, University of Texas at Dallas
YULIA R. GEL, University of Texas at Dallas
MURAT KANTARCIOGLU, University of Texas at Dallas

Version 2021: Since we submitted it to Arxiv in August 2017, this primer has been cited widely and served the useful purpose of being the first resource for anyone who wanted to start working on blockchain transaction networks. At the time of submission, academic research on blockchains was in its infancy and there were few articles on transaction network analysis. This is no longer the case, as multiple conferences, workshops and journals have been created to disseminate blockchain research. Access to blockchain data has been facilitated by tools such as Bitcoin-ETL by Google and repositories such as https://github.com/Chartalist.

We were pleasantly surprised to see that our blockchain predictions have come true. For example, Ethereum has become a major blockchain and its asset networks have created a billion-dollar industry in Decentralized Finance. DAOs have become a major development area. Blockchain research is now an important and influential area where futures are made and careers are created (including those of us). With this updated version, we repeat our commitment to the field and hope to help many more researchers.

Changes: We have edited the text to bring the graph primer up-to-date with developments in blockchain. We have expanded Bitcoin address types and removed P2P network aspects. Old content that was no longer relevant, such as the transaction malleability attacks, has been removed. We have added the second-layer solutions and Proof-of-X. The 2017 version can be accessed at https://github.com/cakcora/7570-blockchain/blob/master/files/primer2017.pdf

Abstract

Bitcoin and its underlying technology blockchain have become popular in recent years. Satoshi Nakamoto designed Bitcoin to facilitate a secure distributed platform without central authorities and blockchain has been heralded as a paradigm that will be as powerful as Big Data, Cloud Computing, and Machine learning.

Blockchain incorporates novel ideas from various fields such as public-key encryption and distributed systems. As such, a reader often comes across resources that explain Blockchain technology from a specific perspective only, leaving the reader with more questions than before.

We will offer a holistic view of blockchain. We will start with a brief history and will give the building blocks of the blockchain. As graph mining is a major area in blockchain analysis, we will elaborate on graph-theoretical aspects of Blockchain technology. We also devote a section to the future of blockchain and explain how extensions like Smart Contracts and De-centralized Autonomous Organizations will function.

Without assuming any reader expertise, we aim to provide a concise but complete description of Blockchain technology.

XXXX-XXXX/2021/10-ART1 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Additional Key Words and Phrases: Blockchain, Bitcoin

ACM Reference format:
Cuneyt Gurcan Akcora, Yulia R. Gel, and Murat Kantarcioglu. 2021. Blockchain: A Graph Primer. 1, 1, Article 1 (October 2021), 17 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 BLOCKCHAIN

In simple terms, blockchain is a distributed database that is secure by design. It was proposed by the unknown author Satoshi Nakamoto in 2008 [29]. Blockchain consists of blocks of transactions that can be verified and confirmed without a central authority. The technology has been popularized through its use in the digital currency Bitcoin, where each transaction is financial by nature.

As the origins of blockchain start with Bitcoin, it is easy to confound the two. Most people refer to Bitcoin and blockchain interchangeably.

Blockchain started with Bitcoin but found usage in many new areas. Companies have created blockchain applications ranging from monitors that track diamonds to networks that distribute food products around the globe. Recent years have seen blockchain-based commercial products from established tech companies such as Hyperledger Fabric [19] of IBM. This increasing interest has been fueled further by another popular blockchain application: Ethereum.

In some aspects (e.g., asset transactions), new blockchain applications differ from Bitcoin; we will outline these in Section 4. Nevertheless, these differences are minor; Bitcoin transactions offer excellent examples to explain how blockchain works. Similarly, companies that track and analyze financial transactions on the Bitcoin network use the term *blockchain analysis* to refer to their research efforts.

Another benefit of using Bitcoin is its long history since 2009; Bitcoin usage has shown the utility of blockchain and highlighted its shortcomings in real life (e.g., delays in transactions).

As in the case of creating blocks of transactions, the core ideas behind blockchain directly shape the creation and maintenance of nodes and edges in graphs, and we will explain these ideas thoroughly. Blockchain lends itself easily to graph analysis; public addresses are linked with transactions and transferred amounts correspond to weighted edges. In addition to the behavior imposed by the blockchain protocol, some user practices change how blockchain graphs are updated in time. An example of this is the common practice of moving the remaining coins into a new address at each transaction (i.e., creating a change address).

We aim to provide a concise but complete summary of blockchain for graph mining researchers. This manuscript will cover core aspects and user practices in blockchain and explain how they affect research efforts.

We will start with a brief history of Blockchain in Section 2.

2 A BRIEF HISTORY

> Nakamoto may have been the mother of Bitcoin, but it is a child of many fathers: David Chaum's blinded coins and the fateful compromise with DNB, e-gold's anonymous accounts and the post-9/11 realpolitik, the cypherpunks and their libertarian ideals, the banks and their industrial control policies, these were the whole cloth out of which Nakamoto cut the invention.
>
> Ian Grigg [15]

Bitcoin [1] represents the culmination of efforts from many people and organizations to create an online digital currency. Notable currencies from early times are *ecash* from Chaum and *b-money* from Dai. Up to the seminal

[1] In community practice, bitcoin is used for currency units, whereas Bitcoin refers to the software, protocol, and community.

Bitcoin paper by Nakamoto [29], multiple times digital currencies seemed to have finally succeeded in creating a viable payment method [15]. Hindered by laws and regulations but mostly due to technical shortcomings, none of these currencies took root. Each failure, however, allowed digital enthusiasts to learn from the experience and propose a new building block towards a viable currency.

From the beginning, digital currencies ran into several fundamental problems. A significant hurdle was the need for a central authority to keep track of digital payments among users. Companies that invented digital currencies proposed to be the central authorities themselves. In a sense, rather than eliminating financial institutions, currencies tried to replace them. This approach never gained traction.

An alternative to the central authority approach was to use a distributed, public ledger to track user balances; every user stores account balances of all users. Although theoretically possible, this solution is unfeasible in practice because information about transactions cannot be digitally transmitted in fast and reliable ways. After all, the networks are faulty and malicious users benefit from lying about balances in double-spending attacks. The problems with achieving consensus in a distributed system, known as the Byzantine general's problem, has been a well-studied topic in distributed systems [23].

While the central authority problem was still an issue, in the unrelated spam detection domain, a solution was developed for a very different purpose [13]. Email providers had the problem of receiving too many spam emails. Even though emails can be analyzed and marked as spam, this wasted system resources. Researchers were searching for ways to check emails for spam without using too many resources.

Dwork et al. proposed an agreement between email senders and email providers. Email senders were tasked to do the computation for each email and append proof of the computation to the email. The computation itself is designed to be time-consuming, whereas checking the proof is easy. The email provider accepts incoming emails with valid proofs and may discard all others without analyzing them. This scheme is called a **Proof-of-Work** (POW). Note that an email sender can still create a spam email, compute its proof and send it. However, POW makes it costly to send too many such emails.

HashCash or Dwork? Appearing in 1997, HashCash proposed POW to prevent service denial attacks and spam emails [4]. HashCash authors claim that at the time of their writing, they were unaware of [13] who also proposed a POW scheme. Furthermore, the Bitcoin white paper cites the HashCash white paper [4], cementing the importance of HashCash in POW.

Bitcoin uses three data structures:

- The address is a unique identifier of an account.
- The transaction contains coin transfers between addresses.
- The block stores error-free and non-conflicting transactions.

Blockchain users include ordinary nodes in a P2P network and miners who attempt to create those clean and non-conflicting blocks in a process called block mining. In a distributed setting, such as the one used by Bitcoin, conflicts arise due to transaction details and address balances. Nakamoto used POW to create an efficient block mining process.

POW attempts to hinder block creation so that global pool miners can create at most one block approximately every 10 minutes. There are two essential points in POW. First, POW makes lying about blocks difficult; a miner needs to spend considerable power to mine a block. Second, the time it takes to create a new block allows the network to reach a consensus about transactions. Furthermore, each block contains information about the previous block, collating it all in an immutable chain. This chain of blocks is called a **blockchain**.

Bitcoin's popularity brought a flood of alternative digital currencies. The earliest being Litecoin, **alt-coins** propose modifications to Bitcoin in aspects like block creation frequency and block mining. More fundamental changes have come from products not related to digital currencies. Ethereum, specifically, was designed to be the

“World Computer” to de-centralize and democratize the Internet, a network of thousands of linked computers. We will discuss these improvements in Section 4.

3 BUILDING BLOCKS OF BLOCKCHAIN

3.1 Addresses

In essence, a Blockchain address is a unique string of 26-35 characters created from public keys. Currently, Bitcoin uses three kinds of addresses:

- P2PKH (Since 2009): *Pay to PubkeyHash* address that starts with 1 as in 1Pudc88gyFynBVZccRJeYyEV7ZnjfXnfKn.
- P2SH (Since 2012): *Pay to ScriptHash* address that starts with 3 as in 3J4kn4QoYDj95S3fqajUzonFhLyjjfKjP3.
- Bech32 (Since 2017): *Segregated Witness address* that starts with bc1 as in bc1qc7slrfxkknqcq2jevvvkdgvrt8080852dfjewde450xdlk4ugp7szw5tk9.
- Taproot addresses that have not been activated yet (2021 September).

The most common (we may even say the standard) type is the *Pay to PubkeyHash*, where we use a single private key to spend bitcoins received from a transaction. Bitcoin added the *Pay to ScriptHash* functionality later to support m-of-n multi-signature transactions; at the receiving address, bitcoins can be spent only when at least m out of n users sign the transaction. In theory, each of n private keys belongs to different users who must come together to spend the coins. In practice, all keys belong to the same user, and the scheme is used to prevent coin theft through stolen private keys. The last address type, Bech32, is used for Segregated Witness [2] outputs. P2SH and P2PKH are case sensitive and Base58 coded, whereas Bech32 is case insensitive and Base32 coded. We cannot initiate a transaction without a receiver address, and an error checking code appended to the address prevents typo errors in the address. It is not possible to send bitcoins to a malformed address.

Each address type has a creation protocol, but essentially we can use a private key and its public key to create an address uniquely. From a graph perspective, the address type has no importance; we can use all address types in transactions as input/output addresses. However, *Pay to ScriptHash* makes the initiation of a transaction very interesting because it allows users to participate in spending decisions. We can mine this user behavior for various applications such as fraud prevention and marketing.

A user may publish its address on web forums, mailing lists, or other mediums. Coins acquired by transactions can only be spent if the private key associated with the address is known. We cannot initiate a transaction without a valid receiver address, and an error checking code is appended to the address to prevent typos.

3.2 Transaction

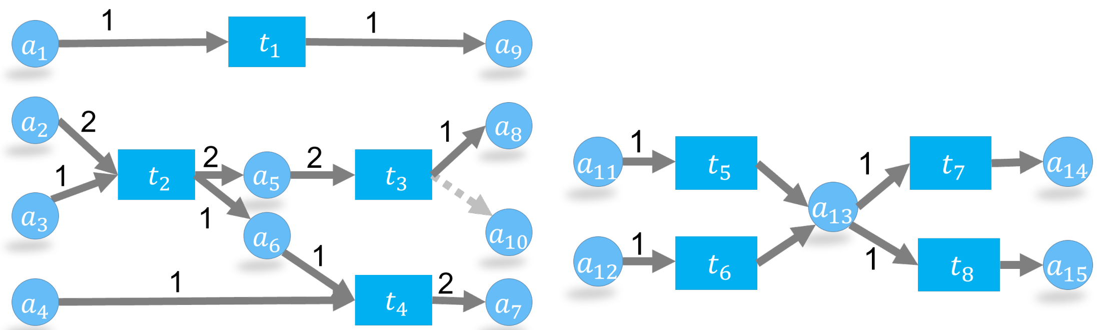
A transaction is a transfer of assets between addresses. Bitcoin allows transferring funds from multiple addresses to multiple addresses. In practice, all input addresses belong to the same user, but output addresses might belong to different users. For each output that is being used as the input of a transaction, three data pieces are required:

- Id of the previous transaction that created the output.
- The index number of the output in the previous transaction.
- Amount to be transferred.

When a transaction has more than one input, each input is signed by the associated private key separately. This signature prevents the transaction from being altered and proves that the user has authorized the transfer.

When a transaction sends assets to an address, the public key of the receiver address is unknown to the network. Only when the received assets are being spent in the future, the receiver shows its public key, and any user in the network can hash the key to verify that the hash equals the address. The matching hash is additional proof that the assets belong to the receiver.

[2] Segregated Witness leaves transaction signatures out of blocks to save space.



(a) Four basic type transactions among 10 addresses. Transaction 3 sends funds to a_{10} for transaction fees. Other transactions do not pay fees.

(b) a_{13} is spending bitcoins received from two transactions in two new transactions. An address can receive inputs from multiple transactions, but it has to spend each input completely when it spends.

Fig. 1. Bitcoin transactions can involve multiple addresses. A many-to-many transaction is an interesting data type that we do not encounter in banking and finance; multiple addresses merge their coins and send them to multiple receiving addresses.

Figure 1a shows four types of transactions. Some edges show coin amounts in Satoshi. A Satoshi is the minimum fraction of a bitcoin, one bitcoin = 10^8 Satoshi. Transaction 1 shows a one-input and one-output transaction. Transactions 2-4 have 2/2, 1/2, 2/1 input/output addresses, respectively. When there is more than one input, the input that specifically funds a given output cannot be distinguished. For example, in transaction 4, we cannot know whether the amount in a_6 comes from a_2 or a_3. These multiple input/output transactions create interesting graphs.

We specifically want to emphasize two aspects of transactions.

First, senders do not specifically indicate a transaction fee; a miner collects the difference between total inputs and outputs as the transaction fee. The miner is a user who mines the block that contains the transaction. Bitcoin transactions can be without fees (i.e., output amount is equal to the input amount), but as we will show in Section 3.3 these fees increase the chance of a transaction to be confirmed in the Bitcoin blockchain. Fees also prevent flooding the network with spam transactions. In Figure 1a only transaction 3 pays a fee. We show the fee payment an edge to the address a_{10}.

Second, an address can receive transfers from multiple transactions and spend the outputs of these transactions separately. For example, in Figure 1b, a_{13} receives transfers from transactions 5 and 6. a_{13} then spends two coins in transactions 7 and 8. This is possible because a_{13} receives each coin in a different transaction. Now consider a_5 in Figure 1a. It also receives coins from transaction 4, but it has to spend both at transaction 3. If it only sends one coin, the miner will take the other as the transaction fee. That is, a_{13} will lose any unspent amount. In practice, when a user has to spend a fraction of the received amount, it sends the remaining balance back to its address, or better, creates a new address, and sends it there.

Manually creating transactions can be too tedious for ordinary users (See righto.com[3] for details.). Companies have created websites, called **online wallets**, that allow users to send, receive and exchange bitcoins without dealing with transaction details. However, these online wallets may leak the private keys of the account to malicious users.

[3] http://www.righto.com/2014/02/bitcoins-hard-way-using-raw-bitcoin.html

3.3 Verification and Confirmation

Having covered addresses and transactions, we now turn our attention to how blockchain, particularly Bitcoin, uses them.

A digital currency, such as Bitcoin, has two main problems: payment verification and payment confirmation.

Payment verification means creating a mechanism to verify that 1) the spender has the necessary balance to make a payment, 2) the spender wants to pay the indicated amount. This is similar to verifying that someone wants to pay 50$ with authentic dollar bills. A user presents its public key (to show that the address belongs to her) and signs a signature with its private key (to confirm the amount). The transaction also lists as input a past transaction that created the output with the associated bitcoin amount. For example, in Figure 1b, transaction 8 lists that its input is from the output of transaction 6.

These measures verify that a user has the balance and wants to make a payment. All network participants could have observed the transaction and recorded the new balances in an ideal peer-to-peer network. Otherwise, the spender can use the same bitcoins to make multiple payments. This issue is known as the **double spending** problem.

In reality, the network is faulty, and the news of a transaction may never reach some users. Furthermore, users could be malicious and lie about balances. Due to these problems, distributed network users may never agree on who owns how many bitcoins and any payment would be a fraud risk. This issue is known as the Byzantine General's problem [23]. As we mentioned earlier in Section 2, early currencies used central authorities to solve this issue, but their fruitless efforts could only replace banks with companies.

Nakamoto proposed a unique solution to the double-spending problem. Bitcoin uses a distributed public ledger to store time-stamped and linked blocks, which contain several transactions. Each block carries a piece of information (i.e., block hash) about the previous block so that users can follow how the blockchain grows in time. Blocks have some constraints: as transferring big blocks among peers would clog the network, Bitcoin limits the block size to 1 MB. In 2017 the **Segregated Witness** concept was activated to exclude transaction signatures from blocks so that miners could fit more transactions into a block. When a new user joins the network, the whole chain is downloaded from peers, and starting from the first block; the user verifies all blocks and transactions. Because of this, it takes considerable time to be up-to-date with the blockchain.

Block creation is limited to one per ten minutes (this is more a wish than a rule) through the mechanism known as **Proof-of-Work**. Each block is computationally costly to create but easy to validate once created. Proof-of-Work requires finding a 32-bit number known as the *nonce* through trial and error. In a sense, it is similar to buying a ticket to win a lottery. Using more computing power increases the probability of finding the number but does not guarantee it. Currently, (September 2021) difficulty is such that it takes around 10^{21} attempts to find a valid nonce value. Users who work on finding confirmed blocks are called block **miners**.

Proof-of-Work entails these steps: Each miner receives a list of transactions from its peers waiting to be confirmed. The list may be different for each miner. The miner picks a number of them to include in a block. While doing this, it may prefer transactions that pay a higher fee. First, several block-specific data pieces are concatenated, such as the block's time, the previous block's hash, and a Merkle root hash of the contained transactions. Then in each trial, a different nonce (i.e., an integer value) is appended to the end of the string. The SHA-256 hash value of this string is compared against the **target**. If the hash value is smaller than the target, the user has found a working nonce and mined a block.

The difficulty *target* is a globally determined value that is periodically adjusted depending on how long it took to find the last 2016 blocks. In theory, Bitcoin requires that a block take around 10 minutes, so 2016 blocks must take two weeks. However, depending on how many miners there are, it may take more or less than the predicted

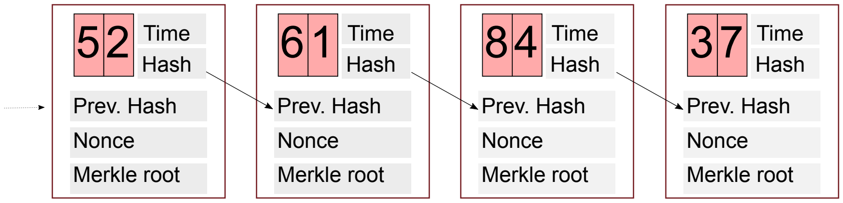


Fig. 2. A toy blockchain for the transactions given in Figure 1. Each block is limited to have two transactions (shown with rectangles 1 to 7) and the coinbase transaction (not shown here).

two weeks.[4] The new difficulty value is increased or decreased accordingly to achieve 10 minutes per block. The difficulty tends to increase over time. However, several short-term decreases have occurred.

Once a miner finds a block b, she sends it to its peers in the network and hopes it is included in the chain. As the information about the block propagates in the network, users update their chains and set b as the latest block. Once notified about this new block, miners (should) stop their ongoing computations. If changed, the miners update the difficulty target and remove transactions that are already included in b. Furthermore, miners change the *hash of the previous block* with the hash of b and resume their mining efforts.

The first block of the chain, known as the *genesis* block, was mined by Nakamoto. Miner or not, a user in the system considers the longest chain the valid blockchain; it is also called the canonical chain.

In some cases, news about a block's discovery may reach users late. Meanwhile, another miner may have created a new block. In this case, users receive two alternative blocks that **fork** from a previous block. In the main blockchain, these **forks** get into a race. As miners choose one block to build on, the fork that is shown to require "most effort" becomes the main chain. The longest chain has the most effort if the latest difficulty level is the same at both forks. Although the last block may change, longer forks are not very common [24]. [5]

Figure 2 shows how blocks can be created from the transactions that we gave in Figure 1. Initially assume that the latest block is block 5-2 (i.e., Chain ..., 5-2). Assume that after the block 5-2 is appended to the blockchain, block 6-1 and 8-4 are mined at the same time. At this point, users and miners can choose any fork and continue their efforts because there are two forks of equal length and difficulty level.

While miner m builds on Chain ..., 5-2, 6-1, miner n searches for a new block on Chain ..., 5-2, 8-4. After all, both chains are the same length, and choosing one over another is a kind of gambit. Next, miner m also mines block 8-4, so the fork becomes Chain ..., 5-2, 6-1, 8-4. Note from Figure 1 that blocks 6-1 and 8-4 contain different transactions, so they can be mined one after another. At this point, other miners in the system see that there are two forks, and Chain ..., 5-2, 6-1, 8-4 is the longest. They choose to build on this long chain. Otherwise, they risk their resources to be wasted on the shorter chain. Once the network miners start building on chain..., 5-2, 6-1, 8-4, the chances of miner n mining a longer fork get even slimmer. No rule stops miner n from continuing to work with its fork. Blockchain assumes that this effort will be futile because the rest of the network will create a much longer chain than that of miner n.

A transaction is tentatively considered *confirmed* when it appears in a block. In practice, we deem a transaction to be secure after six confirmations, i.e., six blocks.

With Proof-of-Work, Bitcoin developers ensure that deliberately creating a fork to replace the main blockchain eventually becomes very expensive; a malicious user must mine new blocks faster than all other miners in the network combined. This means that the malicious miner must hold at least 51% of all mining resources, which is not probable.

[4] See https://www.blockchain.com/charts for the latest values.

[5] The most extended fork was created due to a version mistake of the Bitcoin protocol but was solved after 24 blocks.

Although transactions are secured by the mechanisms we mentioned so far, malicious users have found multiple ways to scam wallets and users (See an excellent on all attacks in [35]).

In transactions, we briefly mentioned that miners who mine blocks receive transaction fees. Transactions themselves do not have to pay fees, and miners can put non-paying transactions in a block as well. However, setting aside a fee increases the chances of a transaction being mined. As Bitcoin receives more transactions, waiting queues have become longer. For the future, a possible remedy is to increase the block size from 1MB to contain more transactions. Increasing the block size to reduce the fees has always been a contentious issue; after a period of much debate in 2017, Bitcoin Cash was created to use 8MB blocks. Bitcoin itself has, so far, resisted all calls to increase the block size.

In addition to transaction fees, the Bitcoin protocol creates a mining reward for each mined block. The mining reward plus the sum of transaction fees are given to the miner in a particular coinbase transaction, which appears first in a block. When Nakamoto was doing all the mining himself in the early days, a coinbase transaction was the only transaction in a block.

Starting with 50 coins for each block and halving every 210K blocks, the total number of created bitcoins will pass 19M in 2022. This geometric series of rewards converges to a maximum of 21 million bitcoins. [6] Eventually, when rewards become very small, transaction fees are expected to provide a sufficient incentive for miners.

4 CHANGES AND IMPROVEMENTS IN BLOCKCHAIN

As Bitcoin became popular, its limitations also became more visible. New generation blockchains learned from these limitations and improved over Bitcoin. Results range from minor tweaks to revolutionary ideas. In this section, we will go over the most important ones.

4.1 Platform over cryptocurrency

Bitcoin uses transactions to transfer an asset: the bitcoin currency. A transaction consists of addresses, the hash of the asset, and a few other data pieces such as time. The key point is that its hash value represents the asset, implying that any hashable digital asset could appear in transactions. Blockchain can store law documents, contracts, agreements, wills, and many other types of assets. These blockchain platforms, however, would still need to use a currency to facilitate block mining. For example, although the Ethereum blockchain is a blockchain platform, it uses a currency named ether. Beyond digitized assets, some companies also give ids to physical objects such as diamonds and use blockchain to track their movements in time (See Section 3.4 of [25] for examples). From a graph perspective, the proliferation of exchanged assets will create a network that is best described by multilayer networks [10], where edges of different natures connect the same set of nodes.

An example is the districts (nodes) of a city being connected by bus, electricity, and subway edges. On the blockchain, edges will be exchanges of various assets. On a worldwide blockchain, this multilayer network will contain invaluable data about how assets move, change hands, and increase/reduce the demand of each other. By 2021, Ethereum has already developed a large ecosystem of smart contract-based ERC20 tokens, ERC721 tokens (called non-fungible tokens or NFTs), decentralized exchanges, data oracles, and much more.

4.2 Smart contracts

Ethereum was created in 2015 to implement transactions and software code that execute conditions and rules. Those so-called smart contracts are written in proprietary coding languages, such as Solidity, and put to a network address by everyone to see and analyze. An analogy is the MYSQL snippets stored on a database. A contract clearly defines the functions that can be used and are guaranteed to work in a specified way. A simple example is an exchange service of assets $\mathcal{A}$ and $\mathcal{B}$. The order of transactions is as follows:

[6] 21M is chosen because when each coin is divided by 10^8 Satoshis, there will be as much Satoshis as there are state-issued M1 currencies

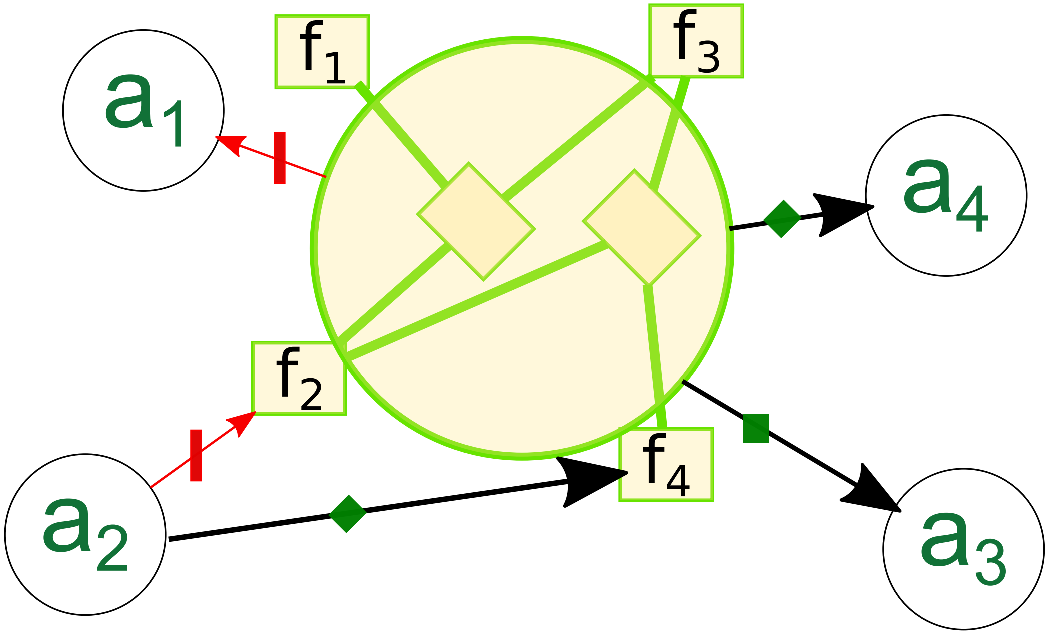


Fig. 3. Four addresses and a contract (it is an address as well). The contract has four functions and two decision boxes. Depending on the function used by a_2, and even the amount of transferred asset, a_1, or a_4 and a_5 can be linked to a_2.

(1) Alice writes a contract with three functions: withdraw and exchange, and puts this to an address by including the contract code in an Ethereum transaction. Alice creates 100 $\mathcal{A}$ in the contract. The contract specifies that it will do an exchange for $1ether = 5\mathcal{A}$. Alice signs the contract transaction with her private key.
(2) Bob has ethers and wants to exchange them for $\mathcal{A}$s. Bob sends 20 ethers to the contract's address.
(3) The contract automatically accepts 20 ethers and assigns 100 $\mathcal{A}$s to Bob's address. This assignment is internal to the contract; a key-value pair of Bob-100 is added to the smart contract data storage.
(4) The contract can accept sell/buy orders from other Ethereum addresses and act as a bridge between buyers and sellers.
(5) Alice may use the withdraw function and receive 100 ethers from the contract.

In this contract, Alice is selling an imaginary asset $\mathcal{A}$ that she created. The price that is specified in the contract reflects Alice's view of the value of A. Although Alice wrote the contract, the contract does not require any involvement from Alice before sending $\mathcal{B}$s to Bob. Even if Alice is malicious, she cannot intervene, take Bob's ethers, and disappear. In reality, contracts are more complex, and there have been misuses. Formal verification of contracts to make sure that they work as intended is a developing research field [6].

Smart contracts bring an exciting notion to graph analysis. A contract can be thought of as a template for future edges. Once a node creates an edge with a contract (e.g., Step 1), the contract will create more edges (e.g., Step 3) that are predefined but constrained by conditions (if Bob sends less than 5 $\mathcal{B}$s, exchange will be refused). These edges are called internal transactions on Ethereum. Figure 3 shows such a contract with its created edges. Edge templates such as this bring cascading behavior into mind. How can the graph be manipulated by planning to facilitate or restrict cascading behaviors? Such topics have been studied in Decentralized Finance research, a promising field of financial activity on blockchains.

Other than Ethereum, smart contracts are expected to play a significant role in the Internet of Things [12]. Once used by billions of connected devices, smart contracts will create large amounts of graphs data.

4.3 Decentralized Autonomous Organizations

> When this happens, machines (hardware and software) become peers in the economy rather than mere tools. Imagine a world where drones that own themselves make deliveries, where autonomous software applications engage in virtual business such as buying and selling server time or even buying and selling stocks and bonds. One day, you may just be hired by a machine, as a one-time gig, or perhaps even for full-time employment.
>
> Adam Hayes [18]

Ethereum's smart contracts made it possible to create a system where actors get into contracts with each other to record matters, decisions, and results of decisions for everyone to see. The contracts encouraged the community to advocate for a future where Ethereum can be used to create an online democracy.

This utopic future is discussed under the term **Decentralized Autonomous Organizations (DAO)**.

The first attempt to create such a future was *the DAO project*, which aimed to create an online investor fund. Joining members could vote on decisions to invest and earn money through their investments. Due to a hacking incident, the DAO became a traumatic experience that lost $50M and led Ethereum to split into two. We will mention this story in Section 4.4.

The DAO project left a bitter taste in the community and hindered the development of new DAOs. Although successful examples, such as the Maker/Dao stablecoin on Ethereum, exist, there is not yet a consensus about what a DAO is and how it should use humans, robots, contracts, and incentives. Some argue that humans should not be involved in anything other than creating contracts, putting them online, and occasionally providing services to DAO (and getting paid). Others use a few humans as *curators* that will manually filter incoming proposals to the DAO before a vote. We may add that even the term DAO is used in different meanings. However, a consensus on DAO terms may be asking for too much.

Instead, a few influential developers have opined on what should be the common points of all DAOs. For example, the creator of Ethereum mentions three points [11].

1. DAO is an entity that lives on the internet and exists autonomously.
2. DAO heavily relies on hiring individuals to perform certain tasks that the automaton itself cannot do.
3. DAO contains some internal property that is valuable in some way, and it can use that property as a mechanism for rewarding certain activities

Opinions differ on how DAOs should function, but not on how influential they will be. Consider this example from Hayes [18]. A DAO acquires a car and puts its contract on Ethereum. The DAO investors can vote (without management) to send the car to work at the Dallas Fort Worth Airport. Riders pay the car on the blockchain, and the vehicle pays its investors dividends from what is left after gas, tax, maintenance, and insurance fees. This DAO can buy new cars, decide on their working sites and even connect them to plan routes together. All of this is already possible through Ethereum.

4.4 Fork Issues

A short reading quickly reveals that most Blockchain technologies are forks of each other (See mapofcoins.com for a map). Blockchain uses **soft forks** to continue on the same main chain while changing a few aspects of the underlying technology. These are considered improvements or extensions. A soft fork is backward compatible and reflects community consensus on how the network should evolve. **Hard forks**, on the other hand, creates a split in the main chain: two versions of the main chain are maintained by different groups of people. In a sense, it builds a new currency, technology, or community. The most famous hard fork happened in the Ethereum project in 2016 due to the DAO hacking.

A hacker stole $50M from *the DAO project* which had raised $150M from the community for a proof-of-concept investor fund. The DAO was hacked because of coding mistakes that allowed multiple refunds for the same investment. Hackers drained DAO while the community was watching the theft in real-time, helplessly. Many developers wanted to roll back the transactions to forfeit the stolen amount. Disagreeing users rejected the rollback and stuck to the existing blockchain. The fork created two versions: Ethereum and Ethereum Classic. This fork started a discussion on how Ethereum should be governed. The **Code is law** proponents claim that any behavior that conforms to the Blockchain protocols should be accepted; a theft, once happened, cannot be punished by rolling back transactions. For legal aspects, see the article by Abegg [1].

In August 2017, Bitcoin faced a hard fork of its own due to the Segregated Witness extension.

4.5 Tokens

Blockchain platforms, such as, Ethereum, encourage developers to create smart contracts that use the blockchain to offer services/goods. These contracts are allowed to develop their own named assets, called tokens. Tokens are created by the contract and bought by investors using blockchain currency such as ether. These tokens are similar to arcade tokens we used to buy at high school and insert into that helicopter game where you could shoot at ships directly or bounce bombs of the ground to do so. [7] So why tokens instead of just using the system currency? Mainly because we can use tokens in creative ways. For example, we can digitize real assets such as houses and represent them with unique tokens (NFT) to be sold on the blockchain.

4.6 Scalability Issues on Blockchain

Bitcoin expects a block to be mined every 10 minutes, and it also imposes a block size of 1 MB, thereby limiting how many transactions Bitcoin can process in 24 hours. As a result, Bitcoin processes 7-15 transactions per second only. The payment method VISA, on the other hand, processes 2000 transactions per second, on average (See the bitcoin wiki for more details [9].).

Size and speed limitations have been eased in some other coins; Litecoin, for example, mines blocks every 2.5 minutes. The Bitcoin community has been discussing ways to increase the number of transactions. Bitcoin adopted a solution in 2017 with the Segregated Witness (SegWit) extension. Segregated Witness removes transaction signatures from the block, which reduces block size by 60%. Further improvements would require increasing the block size itself, which the community still debates. See the bitcoin wiki for details [8].

4.7 First or Second Layer Technologies

Over time, blockchains started to run into scalability issues due to limited block sizes and increased user participation. Initial solutions, such as SegregatedWitness,[8] were developed to leave some of the encryption signatures and other non-transactional data out of blocks to make space for more transactions. Scalability efforts have culminated in second layer solutions, such as the Lightning Network [32], where most of the transactions are executed off the blockchain. The first layer (i.e., the blockchain itself) only stores a summary of transactions that occur on the second layer. Second layer networks, such as the Lightning Network, are a hot research area in network analysis that can be the topic of an extensive review article on its own. Due to space limitations, this

[7] Regretfully, the author could never remember the game's name.

[8] https://en.bitcoin.it/wiki/Segregated_Witness

manuscript will only teach blockchain networks that can be extracted from the first layer, i.e., the blockchain itself.

4.8 Private Blockchains

By definition, any user can join the Bitcoin blockchain, and all transactions are immutable and public. For corporate settings, this transparency means that rivals can learn company finances and buy/sale relationships. The **Hyperledger Project** was created to use blockchain in industrial settings. Supported by many organizations, Hyperledger offers membership services to choose blockchain participants and uses permissioned and even private blockchains. The project is hosted by the Linux Foundation and focuses on the blockchain's storage, capacity, and availability aspects. Hyperledger users are allowed to choose their consensus and mining approaches. For more on Hyperledger, see [2].

4.9 Proof-of-X

A major criticism of Bitcoin is that while miners are racing to find the next block, no thought goes into how much power and resource is wasted. There can be thousands of other miners whose computations are wasted for a single winner at every block. In 2014, it was estimated that one bitcoin cost 15.9 gallons of gasoline. [9] It will only get worse as more miners join the race. As Swanson notes, Bitcoin is, in reality, a "peer-to-peer heat engine" [38]. However, as Eric Jennings writes "the cost for having no central authority is the cost of that wasted energy".

Alternatives to POW aim to reduce computational inefficiency by relaxing a few assumptions made by Bitcoin. Proof-of-X is an umbrella term that covers POW alternatives in block mining. Each alternative expects miners to prove that they have done enough work or spent enough wealth before creating the block. In POW, the work is the mining computations, and the proof is the hash value. Instead of doing some work, some wealth can be destroyed (as a kind of sacrifice) to mine a block. *Proof-of-Stake* considers coin-age as wealth; 10 2-year-old coins create a wealth of $10 \times 2 = 20$ stakes. The block whose miner has the highest stake becomes the next block in the chain. Once coins are used as a sacrifice, their age becomes zero. Coins will have to accumulate their wealth again. *Proof-of-Burn* goes a step further and indeed sacrifices coins. In the scheme, a miner first creates a transaction and sends some coins to a "verifiably unspendable" address. These coins are called burned, but other than the sender, no one in the network is yet aware that the receiving address is an invalid/unusable one. Afterward, the miner creates a block and shows the proof of burnt coins. The proof is a script that shows how the address was created through erroneous computations. The miner who has burnt the highest number of coins can mine the next block. In this scheme, burning coins to collect transaction fees is only viable if transaction fees/rewards are high enough. Furthermore, the total coin supply will decrease in time. In *Proof-of-Disk* nodes that commit computer memory to maintain the network and perform network functions are chosen as miners.

There is an essential shortcoming in POW alternatives. Maliciousness does not have repercussions. If miners decide to game the system by supporting every competing fork, Proof-of-Work becomes too expensive due to the required computations. Other schemes are not as effective against miner malice.

Consider the case with two competing forks in the blockchain. In Stake or Burn-based proofs, miners can create multiple blocks built on different forks using the same (burned or aged) coins. Eventually, one of the forks will be the longest and become the main chain. Regardless of which one is chosen, the miner's block will have taken its place in the chain, and the miner will receive block rewards. This is known as the *nothing as stake problem*; the miner does not lose anything by gaming the system.

[9] http://www.coindesk.com/carbon-footprint-bitcoin/

5 BLOCKCHAIN NETWORK ANALYSIS

We can categorize blockchains into two classes in terms of data models in their transaction networks: **account based** (e.g., Ethereum), **unspent transaction output (UTXO) based** (e.g., Bitcoin, Litecoin) blockchains, and DAG (directed acyclic graph) based blockchains.

Traditionally cryptocurrencies are UTXO based, whereas platforms are account-based. The difference between UTXO and account-based models has a profound impact on blockchain networks.

In the first type of account-based blockchains, an account (i.e., address) can spend a fraction of its coins and keep the remaining balance. An analogy to account-based blockchains is a bank account that makes payments and keeps the remaining balance in the account. In these blockchains, a transaction has exactly one input and one output address. We may use an address to receive and send coins multiple times. The resulting network is similar to traditional social networks, implying that we can directly apply social network analysis tools to account networks.

On UTXO based blockchains, nodes (i.e., addresses) are dynamic, which complicates network analysis. As Bitcoin became popular, research works analyzed the network for coin price predictions. Various features, such as mean account balance, the number of new edges, and clustering coefficients, have been used in research works [14, 36]. Other than features, network flows [39] and temporal behavior of the network [20] have also been used in predictions.

Table 1. Blockchain types. The private/public columns indicate block mining permissions. IOTA and Ripple are maintained by consortiums that own the mining rights. Many projects (such as Ripple and IOTA) developed smart contract functionality many years after their launch. Although IOTA uses a directed acyclic graph instead of a canonical blockchain, the IOTA transaction model is UTXO based. Privacy coins Monero and Zcash use UTXO models with cryptographic security.

	UTXO	Account	DAG	Platform	Cryptocurrency	Private	Public
Bitcoin	✓				✓		✓
ZCash	✓				✓		✓
Monero	✓				✓		✓
Ripple		✓		✓		✓	
Ethereum		✓		✓			✓
IOTA	✓		✓	✓		✓	

Table 1 shows a list of blockchains and their types. 2017 version of this primer had only discussed Bitcoin networks. In 2021, Bitcoin is still the main blockchain of discussion, and this primer again focuses on Bitcoin to teach graph models. However, influential blockchains that do not use the UTXO model have appeared. For the graph models of account and DAG-based blockchains, we refer the reader to our recent manuscript [16].

Studies in network features show that since 2010 the Bitcoin network can be considered a scale-free network [24]. In and out-degree distributions of the transactions graphs are highly heterogeneous and show disassortative behavior [21]. Active entities on the network change frequently, but there are consistently active entities [30]. The most central nodes in the network are coin exchange sites [5].

Each Blockchain application has two layers: application and network. In the application layer, content graphs reflect transfers of assets on the Bitcoin protocol. In the network layer, communication graphs reflect the underlying P2P communication network among Bitcoin addresses. P2P networks have a limited impact on how transactions and blocks are created (except for malicious cases where nodes are isolated in Eclipse or Network Partition attacks). For this reason, we focus our analysis on content graphs created from blockchain transaction networks.

Transaction networks are used to track and locate transactions and addresses for many use cases such as taint analysis, price prediction, and transaction type classification. In this section, we will look at these practices.

Graph Models

Transaction networks are modeled in three forms: transaction, address, and entity/user graphs.

Transaction graphs omit address nodes from the transaction network and create edges among transactions only. Figure 4 shows the transaction graph of the network shown in Figure 1. The most important aspect of the transaction graph is that a node can appear only once. There will be no future edges that reuse a transaction node.

The transaction graph contains far fewer nodes than the network it models. We can immediately observe a few drawbacks from Figure 4. Unspent transaction outputs are not visible; we cannot know how many outputs are there in t_1, t_3 and t_4. In Bitcoin, many outputs stay unspent for years; the transaction graph will ignore all of them. Second, the transaction graph cannot model address reuse; when a past address receives coins again from a transaction, we cannot create the edge because that would introduce a cycle in the graph.

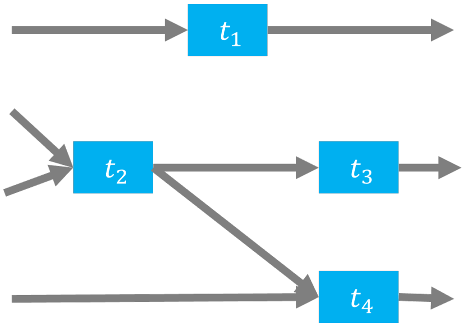


Fig. 4. A graph of the 8 transactions from Figure 1a. Transactions are ordered on the horizontal axis by their appearance in blocks (see Figure 2).

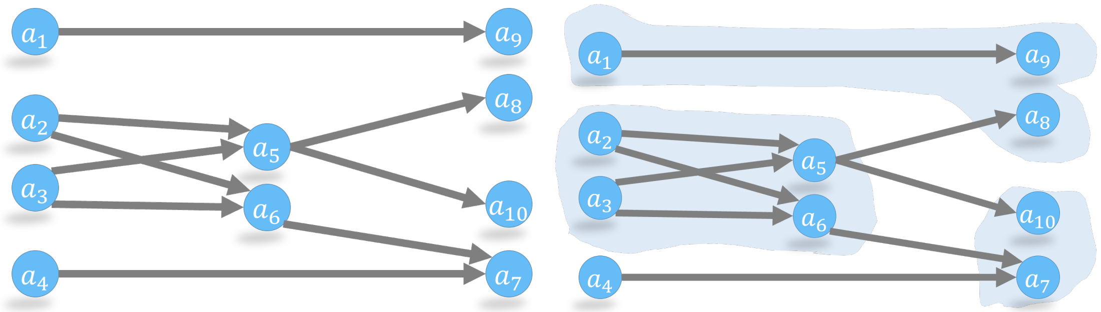
The advantages of the transaction graph are multiple. First, we may be more interested in analyzing transactions than addresses. For example, anti-money laundering tools aim at detecting mixing transactions, and once they are found, we can analyze the involved addresses next. Many chain analysis companies focus their efforts on identifying transactions that are used in e-crime. Second, the graph order (node count) and size (edge count), useful for large-scale network analysis, are reduced from the blockchain network. In UTXO networks, transaction nodes are typically less than half the number of address nodes. For example, Bitcoin contains 400K-800K unique daily addresses but 200K-400K transactions only. However, the real advantage of the transaction graph is its reduced size. As we will explain in the next section, the address graph contains many more edges than the transaction graph.

The address graph (Figure 5a) is the most commonly used graph model for UTXO networks. The address graph omits transactions and creates edges between addresses only. Address nodes may appear multiple times, which implies that addresses may create new transactions or receive coins from new transactions in the future.Figure 5a shows the address graph of our recurring example from Figure 1. As in the transaction graph, each edge carries a weight (i.e., bitcoin amount), but multiple edges can exist between addresses.

Furthermore, if the change is sent back to the input address, the graph contains self-loops. In practice, however, an address is used to send assets only once. As a community rule, **address reuse** is generally avoided to prevent transaction privacy attacks. Due to this, loops and multi edges are rare in the address graph.

Address graphs are larger than transaction graphs in node and edge counts. As the mapping rule states, a UTXO transaction does not explicitly create an edge between input and output addresses in the blockchain transaction network. When omitting the intermediate transaction node, we cannot know how to connect input-output address pairs. As a result, data scientists create an edge between every pair. For example, as shown in edges from a_2 and a_3 to nodes a_5 and a_6, all input addresses contained within a transaction are deemed to have edges to each output address. If there are few addresses in the transaction, this may not be a big problem. However, large transactions can easily end up creating millions of edges. For example, the highest number of inputs in a Bitcoin transaction was 20000 (821 on Litecoin), whereas the highest number of outputs in Bitcoin was 13107 (5094 on

Litecoin). The address graph approach will have to create one million edges for a transaction with one thousand inputs and one thousand outputs.



(a) Address graph creates edges between all input-output address pairs. The resulting graph may have millions of edges for a single transaction.

(b) Entity graph with owners of addresses shown as super nodes. Nodes in a transaction graph cannot create new edges in the future.

Fig. 5. Blockchain address and entity graphs for the transactions of Figure 1

Graph size is not the only problem. The address graph loses the association of input or output addresses. For example, the address graph in Figure 5a loses the information that edges a_2 and a_3 were used in a single transaction; address graph edges would be identical if the addresses had used two separate transactions to transfer coins to a_5 and a_6. This issue can be solved by adding an attribute (e.g., transaction id) to the edge.

Differing from others, the **entity graph** (Figure 5b) tries to find which addresses are owned by the same entity. These efforts are also known as user/entity clustering. Some entities, such as Wikileaks, publicly advertise its address. Some other addresses can be found on mailing lists, forums, etc. publicizing a user's address might provide benefits; if the user is known (i.e., trusted), transactions from her address can be accepted by vendors without waiting for six blocks. These known and trusted addresses are called **Green/Marked addresses**.

In general, there is not a clear-cut method to link addresses together. Some works have proposed graph algorithms to cluster users [33]. Despite having false positives, heuristics have been widely used by the community and in research works. These heuristics are Peeling Chain [28], Change Closure [3], Idioms of Use, Transitive Closure and Ip Clustering [31].

The **Peeling Chain** is frequently used in criminal activities to divide funds and hide their origins. In a Peeling Chain, an address contains an initial amount of bitcoins. A small amount is peeled from the initial amount, and the remainder is sent to a change address. This peeling can continue thousands of times until the initial sum is divided and transferred to the peeled addresses. All the peeling addresses are expected to belong to the same entity. **Change Closure** follows the community practice of sending whatever change remains from a transaction to a new address owned by the spender. For this heuristic, the change address should never appear before and never be reused to receive payments. In Figure 5a after paying $a_1$0, a_5 sends the change to a_8. Both a_5 and a_8 may belong to the same entity. **Idioms of Use** posits that all input addresses in a transaction should belong to the same entity because only the owner could have signed the inputs with associated private keys. By this heuristic, a_2 and a_3 from Figure 1 belong to the same user. **Transitive Closure** extends Idioms of Use: if a transaction has inputs from a_x and a_y, whereas another transaction has from a_x and a_z, a_y and a_z belong to the same entity. **Ip Clustering** proposes using the network layer to find entity clusters [7]. Bitcoin addresses that use the same IP address can be clustered together.

By nature, all user clustering heuristics are error-prone. Some community practices further complicate the issue. For example, online wallets such as coinbase.com buy/sell coins among their customers without using transactions; ownership of an address is changed by transferring the associated private keys to another user. Although the user associated with the address changes, nothing gets recorded in the blockchain.

Another measure to prevent matching addresses to users is known as **CoinJoin** [26]. The key idea is to use a central server that mixes inputs from multiple users. Only the server knows the mapping between inputs and outputs. Mistakenly, the Idioms of Use heuristic would mark all these input addresses to belong to the same user. Relying on a central server is a major weakness in CoinJoin, but there are attempts to develop viable alternatives [34]. Specifically, privacy coins ZCash, Monero, and Dash have seen increasing adoption since 2019. In theory, Monero, Dash, and ZCash privacy coins are UTXO blockchains; their Blockchain networks are similar to Bitcoin's. In practice, privacy coins employ cryptographic techniques to hide node and edge attributes in the blockchain network. For example, ZCash hides information in its shielded pool, whereas Monero adds decoy UTXOs to the input UTXO set.

The increasing usage of coins in crime has necessitated finding users/entities behind addresses. For example, ransom software that encrypts hard drives uses Bitcoin as a medium for ransom payments. Coins have also been stolen (i.e., transferred to addresses of thieves) from online wallets such as Mt.Gox. Companies such as Elliptic and Numisight develop Blockchain graph analysis tools to track these crime-related coins. In research works, BitIodine [37] and McGinn [27] offer visualization tools. GraphSense [17] is an analytics platform that also provides path search on transaction graphs.

With so many works on identifying entities behind addresses, privacy research on Blockchains has also become an exciting topic. See [22] for a recent study on user privacy on Bitcoin.

6 CONCLUSION

We have presented a holistic view on blockchains and described aspects that affect graph mining on blockchain transaction networks. In this 2021 version of the manuscript, we have updated address and transaction details of blockchains and discussed the emergence of blockchain platforms with their asset networks.

REFERENCES

[1] Lukas Abegg. 2016. Code is Law? Not Quite Yet. Online: https://www.coindesk.com/code-is-law-not-quite-yet/. https://www.coindesk.com/code-is-law-not-quite-yet/

[2] Shubhani Aggarwal and Neeraj Kumar. 2021. Hyperledger. In *Advances in Computers*. Vol. 121. Elsevier, 323–343.

[3] Elli Androulaki, Ghassan O Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. 2013. Evaluating user privacy in bitcoin. In *IFCA*. Springer, 34–51.

[4] Adam Back et al. 2002. Hashcash-a denial of service counter-measure.

[5] A. Baumann, B. Fabian, and M. Lischke. 2014. Exploring the Bitcoin Network.. In *WEBIST (1)*. 369–374.

[6] Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cédric Fournet, Anitha Gollamudi, Georges Gonthier, Nadim Kobeissi, Natalia Kulatova, Aseem Rastogi, Thomas Sibut-Pinote, Nikhil Swamy, et al. 2016. Formal verification of smart contracts: Short paper. In *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security*. ACM, 91–96.

[7] Alex Biryukov and Sergei Tikhomirov. 2019. Deanonymization and linkability of cryptocurrency transactions based on network analysis. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 172–184.

[8] Bitcoin. 2016. Block size limit controversy: https://en.bitcoin.it/wiki/Block_size_limit_controversy. Online. https://en.bitcoin.it/wiki/Block_size_limit_controversy

[9] Bitcoin. 2016. Scalability: https://en.bitcoin.it/wiki/Scalability. Online. https://en.bitcoin.it/wiki/Scalability

[10] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. 2014. The structure and dynamics of multilayer networks. *Physics Reports* 544, 1 (2014), 1–122.

[11] Vitalik Buterin. 2014. DAOs, DACs, DAs and More: An Incomplete Terminology Guide: https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide/. Online. https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide/

[12] K. Christidis and M. Devetsikiotis. 2016. Blockchains and smart contracts for the internet of things. *IEEE Access* 4 (2016), 2292–2303.

[13] Cynthia Dwork and Moni Naor. 1992. Pricing via processing or combatting junk mail. In *Annual International Cryptology Conference*. Springer, 139–147.

[14] A. Greaves and B. Au. 2015. Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin. *No Data* (2015).

[15] Ken Griffith. 2014. A quick history of cryptocurrencies BBTC-Before Bitcoin. *Bitcoin Magazine. April* 16 (2014).

[16] Cuneyt Gurcan Akcora, Murat Kantarcioglu, and Yulia R Gel. 2021. Blockchain Networks: Data Structures of Bitcoin, Monero, Zcash, Ethereum, Ripple and Iota. *Wiley WIREs* (2021), 1–40.

[17] B. Haslhofer, R. Karl, and E. Filtz. 2016. O Bitcoin Where Art Thou? Insight into Large-Scale Transaction Graphs.. In *SEMANTiCS (Posters, Demos, SuCCESS)*.

[18] Adam Hayes. 2016. Decentralized Autonomous Organizations: IoT Today. Online. http://www.investopedia.com/articles/investing/022916/decentralized-autonomous-organizations-iot-today.asp

[19] IBM. 2017. Hyperledger. Online: https://www.ibm.com/blockchain/hyperledger.html. https://www.ibm.com/blockchain/hyperledger.html

[20] D. Kondor, I. Csabai, J. Szüle, and G. Pósfai, M.and Vattay. 2014. Inferring the interplay between network structure and market effects in Bitcoin. *New J. of Phys.* 16, 12 (2014), 125003.

[21] D.l Kondor, M. Pósfai, I. Csabai, and G. Vattay. 2014. Do the rich get richer? An empirical analysis of the Bitcoin transaction network. *PloS one* 9, 2 (2014), e86197.

[22] Le Lai, Tongqing Zhou, Zhiping Cai, Zhiyao Liang, and Hao Bai. 2021. A Survey on Security Threats and Solutions of Bitcoin. *Journal of Cybersecurity* 3, 1 (2021), 29.

[23] Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 4, 3 (1982), 382–401.

[24] Matthias Lischke and Benjamin Fabian. 2016. Analyzing the bitcoin network: The first four years. *Future Internet* 8, 1 (2016), 7.

[25] Juri Mattila et al. 2016. *The Blockchain Phenomenon–The Disruptive Potential of Distributed Consensus Architectures*. Technical Report. The Research Institute of the Finnish Economy.

[26] Greg Maxwell. 2013. CoinJoin: Bitcoin privacy for the real world. In *Post on Bitcoin Forum*.

[27] D. McGinn, D.and Birch, D. Akroyd, M. Molina-Solana, Y. Guo, and W. J. Knottenbelt. 2016. Visualizing dynamic Bitcoin transaction patterns. *Big data* 4, 2 (2016), 109–119.

[28] S. Meiklejohn, M. Pomarole, G. Jordan, D. Levchenko, K.and McCoy, G. M Voelker, and S. Savage. 2013. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 127–140.

[29] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system.

[30] M. Ober, S. Katzenbeisser, and K. Hamacher. 2013. Structure and anonymity of the bitcoin transaction graph. *Future internet* 5, 2 (2013), 237–250.

[31] M. S. Ortega. 2013. The bitcoin transaction graph anonymity. *Master's thesis, Universitat Oberta de Catalunya* (2013).

[32] Joseph Poon and Thaddeus Dryja. 2016. The bitcoin lightning network: Scalable off-chain instant payments.

[33] D. Ron and A. Shamir. 2013. Quantitative analysis of the full bitcoin transaction graph. In *International Conference on Financial Cryptography and Data Security*. Springer, 6–24.

[34] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. 2014. CoinShuffle: Practical decentralized coin mixing for Bitcoin. In *European Symposium on Research in Computer Security*. Springer, 345–364.

[35] Muhammad Saad, Jeffrey Spaulding, Laurent Njilla, Charles Kamhoua, Sachin Shetty, DaeHun Nyang, and David Mohaisen. 2020. Exploring the attack surface of blockchain: A comprehensive survey. *IEEE Communications Surveys & Tutorials* 22, 3 (2020), 1977–2008.

[36] M. Sorgente and C. Cibils. 2014. The Reaction of a Network: Exploring the Relationship between the Bitcoin Network Structure and the Bitcoin Price. *No Data* (2014).

[37] M. Spagnuolo, F. Maggi, and S. Zanero. 2014. Bitiodine: Extracting intelligence from the bitcoin network. In *International Conference on Financial Cryptography and Data Security*. Springer, 457–468.

[38] T. Swanson. 2014. Learning from Bitcoin's past to improve its future.

[39] S. Y Yang and J. Kim. 2015. Bitcoin Market Return and Volatility Forecasting Using Transaction Network Flow Properties. In *IEEE SSCI*. 1778–1785.