BRSU Communication Report Nr. 1
ISBN: 978-3-96043-081-0, Digital Object Identifier: 10.18418/978-3-96043-081-0

Technical Fundamentals of Blockchain Systems

Oliver Kattwinkel[1], Michael Rademacher[2]

Abstract
This work provides a short but technical introduction to the main building blocks of a blockchain. It argues that a blockchain is not a revolutionary technology but rather a clever combination of three fields: cryptography, decentralization and game theory. In addition, it summaries the differences between a public, private and federate blockchain model and the two prominent consensus mechanism Proof-of-Work (POW) and Proof-of-Stake (POS).

Keywords
Blockchain — Cryptography — Proof-of-Work — Proof-of-Stake

[1] *Kleenecode GmbH, Sankt Augustin, Germany*
[2] *Department of Computer Science, University of Applied Sciences Bonn-Rhein-Sieg, Sankt Augustin, Germany*
oliver.kattwinkel@kleenecode.com, michael.rademacher@inf.h-brs.de

Contents

1 Introduction 1
2 Evolution 1
3 Building Blocks 2
3.1 Cryptography . . . 2
3.2 Decentralization . . . 3
3.3 Game Theory . . . 3
4 Classified Models 3
4.1 Public Blockchains . . . 4
4.2 Private Blockchains . . . 4
4.3 Federated Blockchains . . . 4
5 Cryptographic Fundamentals 5
5.1 Hash Functions . . . 5
5.2 Cryptographic Hash Functions . . . 5
5.3 Digital Signatures . . . 5
6 Consensus Mechanisms 6
6.1 Proof-of-Work . . . 6
6.2 Proof-of-Stake . . . 7
Acknowledgments 8
References 9

1. Introduction

The purpose of this publication is to elucidate the most important technical fundamentals of blockchain systems. To comprehend the subject as a whole, this work follows a top-down approach and begins in Section 2 with a summary of the early development steps of distributed currencies and the rationale for their failure. The underlying challenges are exemplified by outlining the design of Bitcoin [1] — the first feasible distributed currency and blockchain system. Having derived the key ideas and the addressed challenges, three building blocks of a blockchain system are described in Section 3: cryptography, decentralization and game theory. The breakdown in these independent research areas reveals that a blockchain is an innovative and clever combination of those rather than a revolutionary new field. In Section 4, this work continues with the classification of three different blockchain models: public, private and federated. Each model represents varying properties to achieve the aspired goal of a distributed ledger. Some of them will be discussed controversially. For each one, the main characteristics and differences will be defined. Subsequently, in Section 5, the topic is further subdivided to delve into the low-level cryptographic fundamentals of a blockchain system: hash functions and digital signatures. In Section 6, this work deals with different consensus algorithms.

2. Evolution

The concept of a blockchain was first announced in November 2008 [2]. The person or the organization hidden by the pseudonym *Satoshi Nakamoto* published a whitepaper about a digital payment system called Bitcoin [2]. The system got deployed and launched in 2009 and firstly introduced a functional and fully distributed ledger. Bitcoin is based on a decentralized Peer-to-Peer (P2P) network [2], which synchronizes all transactions on a common public ledger. Thus, every participant in the network is able to read the entire history of all transactions. However, by utilizing secure concepts of cryptography, transactions can only be written or modified by authorized participants. Bitcoin cleverly combined existing contributions from decades of research and most importantly, it solved several fundamental problems in a highly sophisticated and feasible way [3]. Blockchain technology is still a relatively new approach in the field of computer science. It is an emerging technology, which is currently studied and tested for many applications and several use cases.

The idea of fully distributed money has been around from the early 1980s [3]. Distributed money is not controlled and operated by a single organization. It should com-

pletely eliminate intermediaries like a bank and enable the transfer of ownership rights only between a payment sender and receiver. Early attempts to create distributed currencies always failed because all previous systems depend on a trust model that has a central authority, providing a clearinghouse service for transaction verification and ownership record organization [4]. Consequently, such authorities retain full control over the data stored on centralized ledgers. The idea to solve this problem was to introduce the concept of a fully distributed ledger. No single or exclusively designated group of authorities should have the power to control the stored data. The term Distributed Ledger Technology (DLT) summarizes the approach of distributed storage of transaction data in redundant ledger copies. The distribution of the data is a known and solved problem. The challenge and goal is to reach a consensus for all distributed data copies.

A blockchain is an innovative approach to implement a distributed ledger (see Section 4). However, until the launch of Bitcoin all attempts to implement a fully distributed currency led to one fundamental unsolved problem. Distributed currencies suffered from the risk to double spend coins. Since digital copies are trivial, the same coin could be transferred in parallel by one sender to two or more differing recipients. This so-called *double-spending problem* is a major challenge for distributed currencies [5]. With the publication of Bitcoin in the year of 2008, Satoshi Nakamoto presented a solution to this problem [2]. The essential approach of this solution is the concept summarized by the term blockchain. A blockchain solves the double-spending problem by defining a chronological order of all transactions. If two or more conflicting transactions are identified, only the first one is accepted and all the other transactions are discarded. Accordingly, a blockchain system can be understood as a distributed timestamp server [2]. This concept enables one ledger as a single source of truth. The challenge is to find a consensus on the state of the ledger between all participants in a decentralized P2P environment.

The challenges of the double-spending problem are those of the Byzantine Generals Problem (BGP) [6]. The BGP concerns the problem of mutual agreement on a consistent state for distributed information [6]. The famous analogy in [6] describes the problem of different spatially separated generals besieging a city and trying to agree on the best time for an attack. It is a communication, coordination and synchronization problem. Especially in the presence of selfish or malicious participants, i.e. a general acting as a traitor, a practical solution to this problem is far from trivial. One possibility to solve this problem in a decentralized environment, is to introduce the concept of a voting scheme. In theory, a voting network of peers can achieve a correct and consistent network state as long as the majority of the peers is honest. Accordingly, a correct ledger state, i.e. a system-wide consensus, can be achieved by election. If the participants trust each other and can communicate directly, no difficulties are present. However, a remote voting scheme induces vulnerabilities and is prone to diverse attack vectors. With the assumption of synchronous and reliable communication, the BGP reaches consensus as long as the term $n \geq 3f + 1$ is satisfied [6]. The original problem description defines n physically separated generals trying to mutually agree via messengers on a battle plan [6]. However, f traitors, with $f \subseteq n$, try to thwart the agreement [6]. A decentralized system can tolerate failures (or traitors) as long as the number of malicious participants is less than one third of all participants. Such systems with a resilience to byzantine failures are considered Byzantine Fault Tolerant (BFT) [6].

Nakamoto's blockchain architecture masters the BGP from a practical perspective and balances feasibility and security [3]. The ambitious design of Bitcoin enables an increased BFT of $n \geq 2f + 1$ [7]. Accordingly, the number of malicious participants is increased from less than one third to less than one half of all participants. Thus, Bitcoin is a practical demonstration of the theoretical assumption of majority based consensus networks. As long as 51% of all transaction validators are honest, the network will have a quorum and eventually reach consensus. Bitcoin or more specifically, the Nakamoto consensus mechanism operating inside, represented a new milestone of practical decentralization.

3. Building Blocks

The design of Bitcoin and hence the concept of a blockchain system consists of a network containing a distributed and chronological ordered transaction database, which is utilized for the manipulation-proof storage of linear data records. It enables a distributed consensus between untrusted participants and represents a continuously expanding open ledger. Any of the stored data is protected against unauthorized manipulation mainly by aspects of the building blocks summarized by the research areas of cryptography, decentralization and game theory. In the following sections, each area will be explained in the context of blockchain systems. The innovative characteristics of blockchain systems are a result of the clever combination of these research areas. They enable the elimination of intermediaries, central authorities and data monopolies, and thus provide an unprecedented protection against data manipulation with robust censorship resistance.

3.1 Cryptography

The most important building block of a blockchain system is cryptography (see Section 5 for more details). Cryptography is used to store data in units of immutable blocks with a fixed chronological order. Each block is linked with the former one by a backward pointer to the previous block header as shown in Figure 1. This concatenation forms a hash linked list and is facilitated by a unique and unforgeable hash value of each block. Any changes of the data stored in former blocks would invalidate all following newer blocks. An attacker would have to re-calculate all the blocks from that given point. Practically, this is infeasible because the creation involves significant computational work (see Section 6). Accordingly, previous blocks can not be manipulated unnoticed and hence the hash linked list provides a secure data history with a high degree of integrity. As long as the linked hashes remain valid from start to finish, all previous transaction data is secured by the

contents of the current block and hence reduce the need of trust for each block individually. With every further block this structure is strengthened.

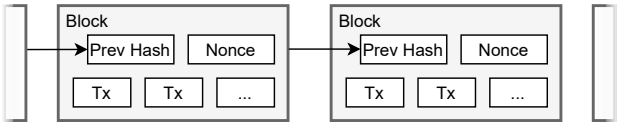

Figure 1. Hash linked blocks. Redrawn from [2].

Furthermore, cryptography enables the concept of ownership rights. Unforgeable digital signatures contribute to the security aspects of authenticity, non-repudiation and integrity for each published transaction. They provide a layer of validation and security to transactions sent through a insecure communication channel (Internet). Secure transfer of unforgeable ownership rights is realized by having to digitally sign all generated transactions. Subsequent signature verifications is easily accomplished.

3.2 Decentralization

Decentralized system architectures support unauthorized manipulation. Since every state change implies agreement by the majority of the network, unauthorized manipulation is identified more easily. The complete history of the data stored on a blockchain is transparent throughout the network. There is no hierarchically organized structure and no central storage location (cf. Figure 2). A blockchain is not created or managed by a central authority, but is assembled independently by every contributing node in the network. All data is stored redundant by a flat end-to-end architecture on the basis of a P2P network. Bitcoin for example utilizes an unstructured and decentralized P2P network based on persistent TCP connections as its communication structure [3].

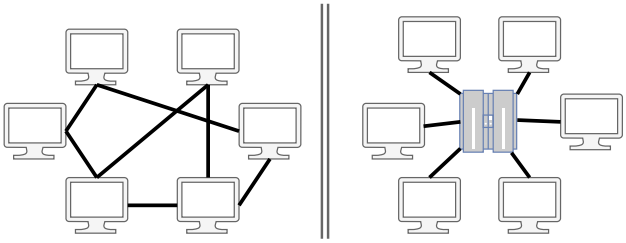

Figure 2. P2P and server-based network.

All the peers, i.e. individual nodes, form a decentralized environment, which contributes to a distributed consensus between all participants without a superordinate party. Consequently, any data modification must be agreed and executed by the majority of the network. As long as this majority is honest, the network is consistent and features a system-wide consensus. In addition to the objective of immutable data storage, redundancy increases availability and failure tolerance of the stored data and eliminates single point of failures regarding data loss. Regarding currency or payment systems, this results in promising and unprecedented properties.

3.3 Game Theory

Finally, game theory is a another significant aspect regarding a secure mode of operation for blockchain systems. Game theory is the study of mathematical modeling for decision-making situations [8]. It is based on a strategic interaction between rational decision-makers [8]. The success or failure for each individual strategy does not only depend on one's own actions. For blockchain decision-makers, success and failure is defined by individual behavior based on the network protocol. For valuable digital assets such as cryptocurrencies, the concept of game theory defines significant parts of the trust model of the system. The need for blind trust in other network participants is reduced by rewarding honest participants with economic incentives (see Section 6). Consequently, dishonest participants get penalized by missing any of these rewards. This relationship between reward and penalty enables a system based on logical reasoning driven economic rationality. If, on the basis of the same effort, an honestly acquired reward consists of more value than the output of a successful attack, a malicious participant is best advised to choose a strategy that follows the defined network protocol. A rational self-interested decision-maker will rather contribute to common interest and hence foster the consensus and correct state of the network. On this basis, blockchain applications such as Bitcoin are commonly referred to as *trustless* consensus mechanisms [4, 9].

4. Classified Models

The distributed storage of transaction data in redundant ledger copies can be summarized by the term DLT. A blockchain is an innovative approach to implement a distributed ledger and thus blockchains are classified as DLTs, but not every DLT is a blockchain. A transaction based Directed Acyclic Graph (DAG) [10, p. 118] for example, is another subtype of a DLT. In addition to this broad classification, blockchain systems themselves can be classified as well. Basically, blockchains can be categorized into three different models. These models are named according to the system's degree of decentralization and by the possibility for participants of accessing transaction validation rights. The latter depends on the process in which eligible network participants have the authority to validate transaction data by participating in the consensus mechanism (see Section 6). Unfortunately, not all blockchain models allow public access to these rights and hence tend to hierarchical structures, which are contrary to the fundamental objective as defined in [2].

As illustrated Figure 3, a blockchain is either a public, private or federated network of transactions validators. The models vary significantly in their degree of decentralization and hence in their ability to protect stored data from unauthorized manipulation. Blockchain systems tending towards a small degree of decentralization lose their ability to effectively provide true protection against manipulation. However, this aspired security goal leads to more or less significant performance constraints, due to high consensus complexity. The term blockchain is discussed controversially in relation to public vs. private / federated ledgers.

The following sections will define the main characteristics and differences of each model.

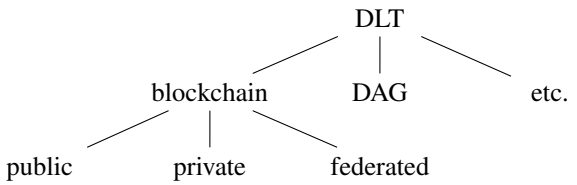

Figure 3. Classification of different blockchain models.

4.1 Public Blockchains

Public blockchains consist of an open and publicly accessible network. Transactions are transparent and may be read and written by any network participant. The ability to validate transactions is the privilege of every network participant. It can be accomplished under a predefined set of well-known conditions (see Section 6). But no participant is excluded from the network by design. There is no hierarchically organized authority. Accordingly, network participants do not have to trust any involved superordinate intermediaries and can join and leave the network at will. The participants do not have to know or trust each other in order to securely transfer assets. In this trustless environment, transaction validators get rewarded with economic incentives for honest behavior, as defined by game theoretic strategies [2]. As long as the majority of the participants are honest, the network will have a quorum, eventually achieve system-wide consensus and result in a correct and consistent state.

Since individual control over the entire network is very hard to accomplish, public blockchain systems provide the highest protection against unauthorized data manipulation. Due to their high degree of decentralization, a public blockchain facilitates true censorship resistance, which may be the most promising key feature of DLTs. However, to effectively provide these properties, fully decentralized blockchain systems suffer from performance constraints. Despite their potential, due to the significant amount of consensus complexity, public blockchains are subject of poor scalability with low transaction processing rates [11, 12].

4.2 Private Blockchains

In contrast to public blockchains, private blockchain systems are closed networks (invitation-only), for which each participant needs special permissions to read, write and validate transactions. The network of transaction validators is a closed group of preselected authorities, having the power to fully control all stored data. The degree of decentralization is low and all transaction data is hardly distributed. As a result, the data structure of private blockchain systems is not immutable and loses the ability to effectively provide true protection against manipulation. Additionally, it decreases availability and failure tolerance of the stored data, enabling single point of failures.

In private blockchain systems, network participants have to trust a centralized group of validators. This concerns the transfer of assets as well as the store of value in one's account. There is no need for economic incentives because all validators know and trust each other. If malicious participants are identified, they simply get excluded by superordinate authorities. There is no need to reward correct behavior. Private blockchains are discussed controversially because they lack most of the necessary key properties as defined in [2]. Due to the small degree of decentralization, the realization of a system-wide consensus is trivial[1]. One remaining advantage of private blockchains is that they are highly scalable and achieve high transaction processing rates.

4.3 Federated Blockchains

At last, the third model is a tradeoff between the former two. Federated blockchain systems are a hybrid solution, enabling a semi open and publicly accessible network. Transactions are mostly transparent and may be read and written by any network participant. However, the ability to validate propagated transactions is a privilege that only preselected and/or subsequently statically elected validators have. This group of authorities forms a consortium. Each member of this consortium is selected carefully and might get excluded for incorrect behavior. Similar to private blockchain systems, there is no economic incentive, since there is no need to reward correct behavior. All members of the consortium know and trust each other. If no malicious members are present, consensus is established easily.

Since the complexity of finding consensus is rather low, federated blockchain systems are easily scalable and allow high transaction processing rates. Depending on the size of the consortium and on the architecture of the system, the degree of decentralization as well as the distribution of all transaction data might be high. As a result, the data structure of federated blockchains might be immutable and may provide the ability to effectively provide protection against manipulation, but none of this can be guaranteed by system design. Despite the protocol rules, if the consortium agrees on arbitrary system states, it may enforce them at will. Network participants have to trust the consortium to operate honestly. This concerns the transfer of assets as well as the store of value in one's account. For similar reasons as for private blockchain systems, federated blockchains are also a controversial topic. Even if the probability of abuse of power is lowered by distributing it between public entities on the basis of a transparent consortium, the ability of misuse still remains. If the consortium is large enough, open and publicly accessible by all network participants, the constraints of a federated blockchain would be fixed. However, this would render it as a public blockchain system.

All blockchain models are summarized and compared in Table 1.

[1] Or less complex when using a replication algorithms like Practical Byzantine Fault Tolerance (PBFT) [13].

Table 1. Comparison of the classified blockchain models.

	Public	Private	Federated
Access	open, public	closed	semi open
Validators	any participant	closed group	consortium
Decentralization	high	low	high/low
Consensus	complex	trivial	trivial
Algorithm	POW, POS, etc.	FBA, etc.	PBFT, RPCA, etc.
Processing Rate	low	high	high
Scalability	low	high	high
Latency	high	low	low
Game Theory	econ. incentive	n/a	n/a
Trust	low	high	medium/high
Application	Bitcoin [1], Litecoin [14], Ethereum [15], etc.	Hyperledger [16], Quorum [17], etc.	Ripple (XRP) [18], Stellar [19] etc.

5. Cryptographic Fundamentals

The fundamentals of a blockchain system can be categorized by three main security objectives, namely integrity, authenticity and non-repudiation. To facilitate a fully distributed ledger, certain aspects of cryptography have to be utilized. These are essentially cryptographic hash functions as well as digital signatures on the basis of asymmetric public key cryptography. They enable many promising properties of blockchain applications, including decentralized trust and control, ownership certification, and the cryptographic-proof security model [4]. Both techniques will be explained in the following sections.

5.1 Hash Functions

A hash function is simply any mathematical function that maps input data of arbitrary size x to an output of fixed size y, as defined by $f_{hash}(x) = y$. The input of a hash function is often referred to as message and the output is usually called hash value, message digest, checksum or fingerprint. These terms stand for more general functions with rather different properties and purposes, such as efficiently searching for data entries in large databases using hash tables, protection against manipulation by ensuring the integrity of any hashed input values, or identification of arbitrary transmission errors. Important properties of hash functions are efficient calculation and determinism. The latter implies identical input values must always generate identical hash values, given the same hash function.

Furthermore, different input values should generate completely different hash values. However, this relationship only applies by a very high probability. Accordingly, so-called collisions are possible and render their application for blockchain systems nearly useless. A hash collision occurs if two unequal input values x and z generate the same output value, as defined by $f_{hash}(x) = f_{hash}(z)$ with $x \neq z$. Consequently, a linear concatenated hash linked list of transaction records, would allow modifications in former transactions and hence loose the feature of immutability. All transactions as well as the chronological order of the blockchain would be invalidated. To prevent such critical conditions, a secure blockchain implementation demands additional hash function security features.

5.2 Cryptographic Hash Functions

As a subset of general hash functions, cryptographic hash functions facilitate additional security features. They provide the following five properties. Again, they must be efficiently calculable and deterministic. Furthermore, cryptographic hash functions must be practically irreversible, as defined by $f_{hash}(x) \Rightarrow y$. It must be practically infeasible to find an input value x to a given output value y, meaning to generate a message that yields a given hash value. Additionally, they should provide collision resistance, so that $f_{hash}(x) \neq f_{hash}(z)$ with $x \neq z$. It must be practically infeasible to find an identical output value for unequal input values x and z. And finally, even a small change to an input value should radically change the output value. There should be no correlation of input values between similar output values. For all properties, the only useful strategy to break the function is to try different input values until a match is found. Similar to brute forcing a password, this can be an exhausting task.

A well known representative of the cryptographic hash functions is the Secure Hash Algorithm 256-bit (SHA-256), as standardized by NIST [20]. It generates collision resistant one-way 256-bit hash values. These specified security features allow the concepts of a manipulation-proof block of transactions with high degree of integrity and the concatenation of those as a hash linked list with a fixed chronological order. Furthermore, they enable a secure transaction address generation mechanism, the basis for the consensus algorithm of [2] (see Section 6), as well as the basis for public key based authentication with digital signatures.

5.3 Digital Signatures

Blockchain systems establish the concept of asset ownership through public / private key pairs, transaction destination addresses (a repeated hash value of a public key or a transaction script), as well as through digital signatures. Public

and private keys have a fundamental mathematical relationship with each other. The relationship makes it possible to encrypt or sign data with one key and subsequently decrypt or verify the output with the corresponding other key. The decryption of an asymmetric encrypted message can only be accomplished by the private key holder. The verification of a signature however, can be accomplished by anyone with access to the public key. Ironically, encryption is not an important part for most blockchain applications, but unforgeable digital signatures are. For example, the Bitcoin protocol utilizes elliptic curve multiplication [2], to generate a public key K given a randomly generated private key k. Bitcoin and other blockchain implementations make widely use of such elliptic curve cryptography, to facilitate secure transactions [21, 22]. For Bitcoin, digital signatures are generated with the Elliptic Curve Digital Signature Algorithm (ECDSA), as standardized by NIST [23, pp. 26-30]. Other signature algorithms are for example Digital Signature Algorithm (DSA) [23, pp. 15-21] or Rivest Shamir Adleman (RSA) [23, pp. 22-25].[2]

Given a public / private key pair, a digital signature serves three generic purposes. First, it facilitates the verification of authenticity of published transactions. It guarantees a cryptographic proof, i.e. a strong reason to believe, that the signature was generated only by the private key holder, who is by implication also the owner of the assets, and thus the transfer of ownership was validly authorized. Subsequently, the signature can be verified using the public key that corresponds to the private key. Second, the cryptographic proof of authorization is undeniable, fostering non-repudiation. Last, a signature proofs that the content of a published transaction has not been modified in transit. Accordingly, a signature serves integrity for the content of every transaction, such as for example the transferred amount and/or the receiver of the assets.

Digital signatures are equivalent to traditional handwritten signatures or stamped seals, but properly implemented, digital signatures provide an increased protection against forgery. A digital signature is utilized on the basis of a mathematical signature scheme, which consists of three parts. The first part is to generate the underlying public / private key pair as described before. The second part is the application of the signature algorithm to sign an arbitrary message. A private key k is applied to a hash value, a digital fingerprint of the aspired transaction TX, to produce a numerical signature Sig, as defined by $Sig = f_{sign}(f_{hash}(TX), k)$. The third part is the application of an algorithm that enables the verification of the signature. The signature Sig, the transaction or the hash value of it, and the public key K are required for verification. As defined by $f_{hash}(TX) = f_{verify}(Sig, K)$, the verification process results with success, if the algorithm, given the signature and the public key, calculates the corresponding transaction fingerprint. A match confirms that only the owner of the private key has authorized the transaction. This useful property of asymmetric cryptography makes it possible for anyone to verify every valid signature on every transaction, while ensuring that only the owner of private key can produce them [4].

[2] RSA may also be applied for asymmetric encryption [24, 25].

6. Consensus Mechanisms

The main idea of a blockchain system can be determined by analyzing the utilized consensus algorithm operating inside. The consensus algorithm is one of the key components of a blockchain system and enables the mechanism to eventually find a system-wide consensus in a decentralized network of untrusted participants. As defined in Section 4, the utilization of a consensus mechanism depends on the degree of decentralization and on the possibility for participants of accessing transaction validation rights. Unfortunately, the blockchain models with restriction to these rights, by not being an open and publicly accessible network, are not the core innovation of blockchain systems. Especially the implemented reward scheme incentivizing validators to be honest and foster a valid consensus is of significant importance. As a result, the following sections describe consensus algorithms utilized by public blockchain systems. At first, the basis of all blockchain implementations, the Proof-of-Work consensus, will be defined. Afterwards, a promising alternative approach tackling identified weaknesses and challenges will finish this section.

6.1 Proof-of-Work

The consensus mechanism implemented in the design of Bitcoin is based on a POW algorithm and commonly referred to as the Nakamoto consensus [2]. A POW scheme is any task suitable to be difficult to solve, but trivial to verify. Such a task may be a random process like trying to find a solution to a cryptographic puzzle, e.g. trying to adapt the input to a hash function in such a way that the hash value has a specific form. Depending on the format of the aspired solution, such tasks can be computationally difficult. Before Bitcoin, one of the best known examples of a POW scheme is Hashcash [26], providing spam protection. Other distributed currency projects like B-Money [27], RPOW [28] or Bit Gold [29] (a direct precursor of Bitcoin) implemented the POW algorithm with similar motivation as to Bitcoin. Namely, to consider the solution of the cryptographic puzzle as a scarce and valuable digital good [3]. However, none of them solved the fundamental double-spending problem [5] in decentralized environments, which arises if transactions can not be chronologically ordered, i.e. synchronized on a distributed ledger.

Algorithm Bitcoin solves the previous problems of the POW algorithms by using a pseudo distributed timestamp server [2]. If transactions are ordered by time, only the first transaction is valid. Transactions are stored in block entities alongside with a hash value of the former block. This concatenation forms a chronological ordered hash linked list and solves the problem of conflicting transactions by adding a pseudo timestamp to the transactions. To strengthen this structure and finalize valid transactions, each block additionally contains a nonce with the following purpose. The nonce is continuously incremented to alter the content of the block in order to produce a solution to the POW cryptographic puzzle. The puzzle consists of calculating a hash value of the generated block, by adjusting the nonce in such a way, that the hash value of the block header is lower than a certain target value. Similar to brute forcing a password,

the only useful strategy to find a valid hash, is to try different nonces until a match is found [3]. The difficulty of the puzzle depends exclusively on the target value.

If the SHA-256 hash value (64 digits in hexadecimal notation) of a generated block must be a number beginning with n zeros, the chance of finding a correct nonce for a given block data at each try is $\frac{16^{64-n}}{16^{64}}$. At the time of writing, a hashed Bitcoin block header must be a binary number beginning with $n = 19$ zeros in hexadecimal notation [30]. Currently, the most powerful and efficient hardware for this process, so-called ASIC miners, can produce SHA-256 hash values with a speed of approx. 100TH/s [31]. Given a single ASIC device, it would still take approx. 8745 days[3] to find a valid solution for the afore defined difficulty of the puzzle. The average work required is exponential to the number of zero bits required [2]. However, verification of the solution is realized by a single hash value in no-time. Given that all network participants aim to solve the puzzle in competition with each other, the probability of being first is proportional to the fraction of the total computing power of the network [3]. Currently, the overall hashing power of the bitcoin network is 124.22EH/s [32]. This leads to an average time to find a solution with $n = 19$ zeros of 10 minutes[4]. Once one participant has found a valid nonce, the block containing it is propagated throughout the network. Subsequently, all participants will verify the received block and update their local copy of the blockchain. To proceed with the processing of further transactions, these so-called miners continue with the validation process of the next block. The term *mining* refers to unforgeable scarcity due to the costliness of creation (cf. precious metals and collectibles) [29]. With every further block the structure of a POW based blockchain system is physically strengthened by the means of real-world computation energy.

Attack Vectors The mapping of real-world resources into digital systems, is an important aspect in untrusted P2P networks. Peers are usually identified by forgeable IP addresses. However, a majority of IP addresses could be subverted by anyone able to allocate a sufficient amount of addresses. Accordingly, the network is vulnerable to a so-called sybil attacks. The concept of one-IP-address-one-vote is thus not applicable in an public P2P network. To limit the number of votes by each peer, the Nakamoto consensus fixes this fragile majority decision by the concept of one-CPU-one-vote and enables a quorum, as long as 51% of all transaction validators are honest. The POW forces validators to prove that they are real identities by consuming real-world resources (computation time ergo electricity ergo money). The reason for this effort is an economic incentive, i.e. the possibility to obtain a reward for every generated and subsequently accepted block. The published block must comply with the underlying protocol. Consequently, only honest participants are rewarded and the need for trust in other network participants is reduced. Dishonest participants get penalized by missing any of the rewards. In addition to the reward scheme incentivizing honest participants, the block reward system regulates the supply and distribution of new assets and prevents arbitrary inflation. These characteristics contribute to a distributed consensus between untrusted participants and provide an unprecedented protection against data manipulation in open and decentralized P2P networks.

Given a balanced P2P network, the total amount of decentralized computing power, i.e. the hash rate, affects the feasibility of different attacks on the consensus mechanism. As described in Section 3, any manipulation of the transaction data stored in former blocks would invalidate all of the following newer blocks. An attacker would have to re-calculate all the blocks from that given point. On the basis of the principle of the highest amount of computational effort is preferred, the so-called *longest/heaviest chain* rule ensures that the re-calculation of a chain is practically infeasible. An attacker would have to generate a chain (fork), which must eventually outpace any competing chains. The probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added [2]. Accordingly, the attacker would have to gain more than 50% of the total computing power and perform a so-called 51% attack to eventually win the race. As mentioned previously, at the time of writing, the total hash rate of the Bitcoin network produces approx. up to 125EH/s [32]. An attacker would have to control and operated approx. 637,500 of the high-end ASIC miners defined earlier, to perform such an attack.[5] This would consume approx. 2.5MWh or 500,000USD per hour[6]. Given a self published transaction with an age of one day, an attacker would have to spent approx. 12 million USD to double-spend or alter it subsequently[7] Thus, in relation to the transferred value, it should be well considered whether such an attack is worth the effort.

For reasons of network stability and low latency in block propagations, the network tries to produce one block per specified time frame. The greater the block time in relation to the block propagation time, the less likely is the probability of creating network forks. Consequently, these parameters mitigate inconsistent network states allowing double-spending attacks. Bitcoin miners find a solution to the POW cryptographic puzzle approx. every 10 minutes. To guarantee this block time in a dynamic network having variable computing power, the difficulty of the cryptographic puzzle is retargeted every 2016 blocks, i.e. approx. every 14 days [33]. These block parameters in addition to the longest chain rule facilitate independent network reorganizations, due to network forks on the basis of propagation delays.

6.2 Proof-of-Stake

Apart from POW, several alternative consensus algorithms have been proposed [34]. Many of them are simple software forks of the original Bitcoin protocol. Others tried to implement a new innovative consensus mechanism. For both kinds, many of them have not been tested in a real-

[3] Approximation: $\frac{16^{64}}{16^{64-n}} \cdot \frac{1}{100*10^{12}} \cdot \frac{1}{60*60*24}$

[4] Approximation: $\frac{16^{64}}{16^{64-n}} \cdot \frac{1}{125*10^{18}}$

[5] 125EH/s · 0,51 ÷ 100TH/s = 637,500

[6] 637,500 · 3.5kWh · ≈ 2.5MWh · 0.20USD/kWh = 500,000USD/h

[7] 24h · 500,000USD/h = 12,000,000USD. On the basis of an average block time of 10 minutes (cf. Bitcoin), this would correspond to a backlog of 144 blocks.

world scenario. Thus, their resilience and robustness against various attack vectors is not practically tested and fully understood yet. However, a promising alternative approach tackling identified weaknesses and challenges of the viable Nakamoto consensus has been developed, e.g. [35, 36, 37]. The democratic concept of one-CPU-one-vote, to mitigate sybil attacks, is threatened by the utilization of specialized mining hardware. As discussed before, such Application-Specific Integrated Circuit (ASIC) miner hashing equipment outperforms the efficiency of any traditional CPU with ease. Currently, there is no working implementation of an ASIC-resistant consensus algorithm [38]. A POW system tending towards centralization of mining power, enables the ability to suppress participants with less computational energy [3]. As result, solvent individuals or organizations gain more voting power by constantly applying more efficient hardware equipment. The hash rate of the network is thus not distributed evenly. Consequently, these groups have more control over the transaction validation process of the network and hence obtain a higher probability of successfully performing double-spending attacks. It is assumed the cost for a 51% attack is lower on POW based blockchain systems as for certain alternative consensus mechanisms [3]. However, objections to this claim exist [39].

To replace the consensus algorithm of a blockchain system proven to work (Bitcoin), an alternative with the following fundamental required characteristics has to be found. First, the ability to generate new blocks has to be an exhausting process. In terms of the sybil attack in a P2P network, the number of votes for each peer has to be limited and forgery proof. Second, there must be a common rule to resolve both intended (51% attack) and unintended (propagation delays) network forks. Network reorganization must eventually achieve system-wide consensus and result in a correct and consistent network state.

The consensus algorithm POS facilitates both of these requirements with the concept of a stake, either simply by amount or more effectively by *coin age* [3, 35]. A coin age can simply be defined as the amount of coins multiplied by their holding period. The coin age in POS systems operates similar to the computing power in POW systems. For example, a large amount of long unused coins is equivalent in power to a powerful ASIC miner [3]. However, for POS the term *power* is independent of computing power. With the concept of one-stake-one-vote, voting power depends on a staked deposit. With no exorbitant artificially increased difficulty of the cryptographic puzzle, POS provides an answer to the legitimate criticism that POW based blockchains waste energy [3]. Similar to POW based blockchains, transactions are stored in block entities alongside with a hash value of the former block. This concatenation forms a chronological ordered hash linked list and solves the problem of conflicting transactions. To generate a valid block, the hash value of the block header must be lower or equal than a certain target value. In contrast however, no nonce is used to to alter the content of the block in order to produce a valid solution. Instead, every predefined time frame, e.g. every second, a timestamp is updated and all miners have a new chance of finding a valid low-energy consuming solution, to obtain the block reward. By claiming a reward, miners destroy the coin age of their staked deposit. Furthermore, the difficulty does not depend exclusively on the target value. It is individually determined and inversely proportional to the staked coin age [3]. This process shifts the competitive tournament (POW) to a more randomized lottery scheme and mitigates the risk of centralization of mining power. The greater the staked in-band deposit (coin age), the higher the probability of being selected as a transaction validator for the next block.

However, these POS specific properties also encourage to hoard coins, making them more like collectible as of transferable assets. Since the coins themselves are the basis of receiving more of them by obtaining block rewards, the idiom *rich get richer* fully applies. Consequently, rich individuals gain more and more voting rights, which leads to the same attack vectors mentioned before. Another major challenge of the POS algorithm is the drawback of inconsequential penalization for dishonest/malicious behavior on the basis of network forks. Unlike the longest chain rule, a POS algorithm favors the chain which was built using the highest amount of coin age. Again, this mitigates 51% attacks and facilitates network reorganizations. However, unlike the POW algorithm, miners are not forced to select one single chain at one time. Although the number of votes for each peer are limited by the concept of coin age, a miner can still exploit the consensus mechanism by not having to provide real-world resources, e.g. computational power. If a network fork is identified, miners can effortless continue the process of transaction validation on multiple subchains. As a result, miners are incentivized to extend every potential fork to increase the probability of getting block rewards [34]. They put *nothing a stake* while working on multiple chains simultaneously. This process restrains a system-wide consensus by not penalizing intended inconsistencies and is correspondingly summarized as the so-called *nothing-at-stake problem* [40].

Acknowledgments

The majority of content in this publication has been extracted from the masters-thesis “Enhancing the Usability of Blockchain based Digital Asset Transfer” from Oliver Kattwinkel. Thanks to Andrei Ionita for the helpful comments.

List of Acronyms

ASIC Application-Specific Integrated Circuit
BFT Byzantine Fault Tolerant
BGP Byzantine Generals Problem
CPU Central Processing Unit
DAG Directed Acyclic Graph
DLT Distributed Ledger Technology
DSA Digital Signature Algorithm
ECDSA Elliptic Curve Digital Signature Algorithm
FBA Federated Byzantine Agreement
IP Internet Protocol

NIST National Institute of Standards and Technology
P2P Peer-to-Peer
PBFT Practical Byzantine Fault Tolerance
POS Proof-of-Stake
POW Proof-of-Work
RPCA Ripple Protocol Consensus Algorithm
RSA Rivest Shamir Adleman
SHA-256 Secure Hash Algorithm 256-bit
TCP Transmission Control Protocol

References

[1] bitcoin.org. *Bitcoin - Open source P2P money*, 2020 (accessed 2020-07-01). https://bitcoin.org.

[2] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, bitcoin.org, 2008.

[3] Florian Tschorsch and Björn Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys Tutorials*, 18(3):2084–2123, 2016.

[4] Andreas M. Antonopoulos. *Mastering Bitcoin: Programming the Open Blockchain*. O'Reilly Media, 2 edition, 6 2017.

[5] Usman W. Chohan. *The Double Spending Problem and Cryptocurrencies*, 2017 (accessed 2020-07-01). https://ssrn.com/abstract=3090174.

[6] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

[7] A. Miller and J. J. LaViola. Anonymous byzantine consensus from moderately-hard puzzles: A model for bitcoin. Technical report, University Florida, Gainesville, 2014.

[8] Rodger B. Myerson. Game theory: Analysis of conflict. *Harvard University Press*, 1991.

[9] The Economist. *The trust machine: The technology behind bitcoin could transform how the economy works*, 2015 (accessed 2020-07-01). https://www.economist.com/leaders/2015/10/31/the-trust-machine?fsrc=scn/tw/te/img/cover/st/thetrustmachine.

[10] K. Thulasiraman and M. N. S. Swamy. *Graphs: Theory and Algorithms*. John Wiley and Sons, 1992.

[11] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the security and performance of proof of work blockchains. In *ACM Conference on Computer and Communications Security*, 2016.

[12] Ittay Eyal, Adem Efe Gencer, Emin Gun Sirer, and Robbert Van Renesse. Bitcoin-ng: A scalable blockchain protocol. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 45–59. USENIX Association, 2016.

[13] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, OSDI '99, page 173–186, USA, 1999. USENIX Association.

[14] litecoin.org. *The Cryptocurrency for Payments*, 2020 (accessed 2020-07-01). https://litecoin.org/.

[15] ethereum.org. *Ethereum is a global, open-source platform for decentralized applications*, 2020 (accessed 2020-07-01). https://www.ethereum.org.

[16] Hyperlegder. *Open Source Blockchain Technologies*, 2020 (accessed 2020-07-01). https://www.hyperledger.org.

[17] JPMorgan. *ripple with Quorum. The proven blockchain solution for business*, 2020 (accessed 2020-07-01). https://www.goquorum.com.

[18] Ripple. *Instantly Move Money to All Corners of the World*, 2020 (accessed 2020-07-01). https://www.ripple.com.

[19] David Mazieres. The stellar consensus protocol: A federated model for internet-level consensus. Whitepaper, Stellar Development Foundation, 2018.

[20] Quynh H. Dang. Secure Hash Standard. Technical Report August, National Institute of Standards and Technology, Gaithersburg, MD, jul 2015.

[21] Victor S. Miller. Use of elliptic curves in cryptography. In *5th Conference Advances in Cryptology*, pages 417–426, 1985.

[22] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.

[23] Patrick D Gallagher and Charles Romine. FIPS PUB 186-4 Digital Signature Standard (DSS). Technical report, National Institute of Standards and Technology, Gaithersburg, MD, jul 2013.

[24] K. Moriarty, B. Kaliski, J. Jonsson, and A. Rusch. Pkcs #1: Rsa cryptography specifications version 2.2. RFC 8017, RFC Editor, November 2016.

[25] K. Moriarty, B. Kaliski, J. Jonsson, and A. Rusch. Pkcs #1: Rsa cryptography specifications version 2.2. RFC 8017, RFC Editor, November 2016.

[26] Adam Back. Hashcash - a denial of service countermeasure. Technical report, 2014.

[27] Wei Dai. *B-Money*, 1998 (accessed 2020-07-01). http://www.weidai.com/bmoney.txt.

[28] Hal Finney. *RPOW - Reusable Proof-of-Work*, 2004 (accessed 2020-07-01). https://cryptome.org/rpow.htm.

[29] Nick Szabo. *Bit Gold*, 2005 (accessed 2020-07-01). https://unenumerated.blogspot.com/2005/12/bit-gold.html.

[30] BTC.com. *Diffculty*, 2019 (accessed 2020-07-01). https://btc.com/stats/diff.

[31] 99bitcoins.com. *Bitcoin Mining Hardware Reviews and Comparison*, 2019 (accessed 2020-07-01). https://99bitcoins.com/bitcoin-mining/hardware/.

[32] blockchain.com. *Hash Rate*, 2019 (accessed 2020-07-01). https://www.blockchain.com/de/charts/hash-rate.

[33] Bitcoin Wiki. *Target*, 2016 (accessed 2020-07-01). https://en.bitcoin.it/wiki/Target.

[34] Shehar Bano, Alberto Sonnino, Mustafa Al-Bassam, Sarah Azouvi, Patrick McCorry, Sarah Meiklejohn, and George Danezis. Sok: Consensus in the age of blockchains. Technical report, University College London, United Kingdom, 2017.

[35] Sunny King and Scott Nadal. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. Technical report, 2012.

[36] Aggelos Kiayias, Alexander Russell, Bernardo Machado David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *IACR Cryptology ePrint Archive*, 2016.

[37] Iddo Bentov, Rafael Pass, and Elaine Shi. Snow white: Provably secure proofs of stake. *IACR Cryptology ePrint Archive*, 2016:919, 2016.

[38] StopAndDecrypt. *ASIC Resistance is Nothing but a Blockchain Buzzword*, 2018 (accessed 2020-07-01). https://hackernoon.com/asic-resistance-is-nothing-but-a-blockchain-buzzword-b91d3d770366.

[39] Nicolas Houy. It will cost you nothing to ”kill” a proof-of-stake crypto-currency. *Economics Bulletin*, 34(2):1038–1044, 2104.

[40] Vitalik Buterin. *Proof of Stake FAQ*, 2020 (accessed 2020-07-01). https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ.