Chapter 17
Revisiting Practical Byzantine Fault Tolerance Through Blockchain Technologies

Nicholas Stifter, Aljosha Judmayer, and Edgar Weippl

Abstract The connection between Byzantine fault tolerance and cryptocurrencies, such as Bitcoin, may not be apparent immediately. Byzantine fault tolerance is intimately linked to engineering and design challenges of developing long-running and safety-critical technical systems. Its origins can be traced back to the question of how to deal with faulty sensors in distributed systems and the fundamental insight that majority voting schemes may be insufficient to guarantee correctness if arbitrary, or so-called Byzantine failures, can occur. However, achieving resilience against Byzantine failures has its price, both in terms of the redundancy required within a system and the incurred communication overhead. Together with the complexity of correctly implementing Byzantine fault-tolerant (BFT) protocols, it may help to explain why BFT systems have not yet been widely deployed in practice, even though practical designs exist for almost 20 years. On the other hand, asking anyone about Bitcoin or blockchain 10 years ago would have only raised quizzical looks. Since then, the ecosphere surrounding blockchain technologies has grown from the pseudonymously published proposal for a peer-to-peer electronic cash system into a multi-billion-dollar industry. At the heart of this success story lies not only the technical innovations presented by Bitcoin but a colorful and diverse community that has succeeded in bridging gaps and bringing together various disciplines from academia and industry alike. Bitcoin reinvigorated interest in the topic of BFT as it was arguably the first system that achieved a practical form of Byzantine fault tolerance with a large and changing number of participants. Research into the fundamental principles and mechanisms behind the underlying

N. Stifter (✉) · E. Weippl
Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), Institute of Information Systems Engineering, Technische Universität Wien, Vienna, Austria

SBA Research, Vienna, Austria
e-mail: nstifter@sba-research.org; eweippl@sba-research.org

A. Judmayer
SBA Research, Vienna, Austria
e-mail: ajudmayer@sba-research.org

S. Biffl et al. (eds.), *Security and Quality in Cyber-Physical Systems Engineering*,
https://doi.org/10.1007/978-3-030-25312-7_17

blockchain technology of Bitcoin has since helped advance the field and state of the art regarding BFT protocols. This chapter will outline how these modern blockchain technologies relate to the field of Byzantine fault tolerance and outline advantages and disadvantages in their design decisions and fundamental assumptions. Thereby, we highlight that Byzantine fault tolerance should be considered a practical and fundamental building block for modern long-running and safety critical systems and that the principles, mechanisms, and blockchain technologies themselves could help improve the security and quality of such systems.

Keywords Blockchain · Byzantine fault tolerance · Distributed ledger technologies · Bitcoin · Distributed systems

17.1 Introduction

Currently, the term “blockchain” is hardly associated with long-running, software-intensive, and production- and other, so-called, cyber-physical systems. Instead, many people will likely recall news and articles that cover topics such as the high volatility and speculative nature of cryptocurrencies, security breaches, and technical failures that have led to large financial losses[1] or promises of potential applications of blockchains that are reminiscent of the “peak of inflated expectations” found in the Gartner hype cycle for emerging technologies.[2]

However, beyond this hype, academia and industry alike have started to take a closer look at the technical foundations and possible applications of blockchain and distributed ledger technologies (DLT). Many of their fundamental concepts and building blocks are actually well established and researched technologies, such as cryptographic hash functions, Merkle trees, elliptic-curve cryptography, or moderately hard proof-of-work puzzles (Dwork and Naor 1992; Back 2002; Jakobsson and Juels 1999). The novel and particularly effective interplay between these components within the Bitcoin protocol, as well as the addition of game-theoretic incentives, facilitated the breakthrough which established Bitcoin (Nakamoto 2008) as the first viable cryptographic currency that could operate in a peer-to-peer environment without having to rely on a trusted third party.

It is precisely this seeming ability to avoid any (single) trusted third parties that renders blockchain protocols highly interesting for a variety of use-cases that reach well beyond the realm of virtual currencies. Hereby, the system as a whole exhibits a certain resilience against malicious activities from participants as long as their number and capabilities are reasonably bounded. Essentially, Bitcoin addresses the decades-old problem of *Byzantine fault tolerance* from a different and mostly practically oriented angle.

[1] See https://www.nytimes.com/2016/06/18/business/dealbook/hacker-may-have-removed-more-than-50-million-from-experimental-cybercurrency-project.html.

[2] cf. https://www.gartner.com/en/research/methodologies/gartner-hype-cycle.

Byzantine fault tolerance is of particular importance in the context of critical information infrastructures and other systems where both availability and resilience against faults is essential (Veronese et al. 2009; Esteves-Verissimo et al. 2017). However, widespread adoption of BFT has, at least in part, been hampered by the high resource requirements of early solutions, which may have contributed to the stigma that such protocols are largely impractical, although this issue has been addressed almost 20 years ago (Castro and Liskov 1999). Advancements in both the capacity and cost of technology, as well as new and efficient BFT protocols themselves, have rendered this overhead small enough and that Byzantine fault tolerance should not only be considered for an application in the most critical infrastructure but as a general design philosophy for any system with multiple distinct components that form complex interactions.

Through the current interest and research on blockchain and distributed ledger technologies, the topic of Byzantine fault tolerance is again being drawn into focus. Especially in the context of private or restricted environments, where not every participant should be able to partake in the consensus protocol and be allowed to propose updates to the underlying ledger or shared data structure, classical BFT protocols offer inherent advantages in both security and performance over proof-of-work-based blockchain designs such as Bitcoin.

The rest of this chapter is organized as follows: First, we give an introduction to the research field of Byzantine fault tolerance by outlining its history (Sect. 17.2). Second, we address the topic of blockchain technologies and how they originate from the development of the Bitcoin peer-to-peer cryptocurrency system (Sects 17.3–17.6). An outlook on future challenges and opportunities in this research field is given in (Sect. 17.7). We then outline potential use-cases for blockchain technologies that reach beyond cryptographic currencies (Sect. 17.8). In (Sect. 17.9) the potential application of BFT and blockchain technologies to production system engineering is discussed in more detail before the chapter is concluded (Sect. 17.9).

17.2 Byzantine Fault Tolerance

The origin of the term *Byzantine* failure traces back to the seminal work of Lamport et al. that introduces and addresses an agreement problem called the *Byzantine generals problem* (Lamport et al. 1982). In prior work, the same set of authors had first identified that ensuring consistency in the presence of *arbitrary* failures within distributed systems is more difficult than one would intuitively expect (Pease et al. 1980). Generally speaking, the terms Byzantine and arbitrary failure are used interchangeably, even though the former more explicitly considers the possibility of adversarial behavior. If a system is allowed to exhibit arbitrary failures, it follows that there can also exist execution traces where the sequence and type of failures are indiscernible to that of any adversarial strategy. A clear distinction between the two failure models is usually not made.

Initial research on the Byzantine generals problem, or more generally how to reach agreement, i.e., *consensus*, among a set of processes in the presence of faults, was spawned from practical engineering challenges at the time. Improvements in both microprocessors and networking capabilities had led to a consideration for their application in safety critical systems such as the SIFT fault-tolerant aircraft control system (Wensley et al. 1978). However, thorough analysis of a concrete design problem, namely, clock synchronization among multiple clocks, revealed that synchronization algorithms become impossible for three clocks if one of them is faulty and can drift arbitrarily. The generalization of this problem, that is, reaching agreement upon a vector of values where each value is the private input of a participant in the agreement protocol and the agreed-upon vector must either contain the private input of each participant or that the particular participant was faulty, is referred to as *interactive consistency*.

Pease et al. (1980) were able to show that even in a synchronous system model, i.e., where there is an a priori known upper bound Δ on computation and message transmission times, and a fully connected graph of reliable, point-to-point communication channels without message authentication, interactive consistency requires $3f + 1$ participants to arrive at a solution, where f denotes the maximum number of faulty participants that can exhibit arbitrary failures.

A few years later it was proven in (Fischer et al. 1985), what is now referred to as the *FLP impossibility result*, that deterministic consensus becomes impossible in an asynchronous system if only a single process is allowed to fail in the crash-stop model, even if communication between processes is reliable. The result, however, does not extend to consensus protocols exhibiting only *probabilistic* guarantees for liveness or correctness. Hence, so-called *randomized* Byzantine consensus algorithms, first presented by Ben-Or (1983) and Rabin (1983), which instead eventually terminate with probability $P(1)$ or have a non-zero probability for disagreement, are hereby not affected.

Nevertheless, at the time, the takeaway from these first results was that systems for reaching consensus, in particular in the presence of *Byzantine failures* while in principle feasible, were largely impractical for most real-world scenarios (Castro and Liskov 2002). For instance, the papers presenting the Byzantine generals problem and interactive consistency contain accompanying solutions where the distributed algorithms have an exponential message complexity in the number of participating processes. Together with the additional computational overhead, as well as large number of additional replicas that are required to tolerate Byzantine failures over the more benign crash failures, early BFT consensus protocols were simply too prohibitive for most use-cases.

It would take over a decade until publications such as *Practical Byzantine Fault Tolerance* (PBFT) by Castro and Liskov (Castro and Liskov 1999) showed that Byzantine fault-tolerant consensus algorithms could indeed be rendered practicable under realistic system assumptions. Nevertheless, while research on the topic of BFT consensus was ongoing (Cachin et al. 2000; Clement et al. 2009; Guerraoui et al. 2010; Veronese et al. 2013), it remained a comparatively isolated topic area, given the broad range of potential applications. In part, this may be attributed to the fact that consensus protocols are often discussed in the context of *state machine replication* (Lamport 1984; Schneider 1990) and achieving active replication for services such as databases. For these scenarios, all replicas, i.e., participants, may be under the control of a single entity and achieving only the more benign crash-fault tolerance can often be a tenable system model. In particular, Lamport's crash-fault-tolerant *Paxos* consensus algorithm (Lamport 1998) and derivations thereof have found their way into practical applications (Chandra et al. 2007).

However, even in such scenarios where crash-fault tolerance may have previously been considered acceptable, it can still be advantageous to gain the additional resilience of BFT. In particular, because these previously isolated systems are increasingly becoming interconnected, e.g., by operating in cloud environments (Vukolić 2010), it appears sensible to be able to tolerate Byzantine failures. Even before the advent of Bitcoin and blockchain technologies, calls from the scientific community had become louder that Byzantine fault-tolerant protocols could increasingly meet a wide range of practical demands and should hence be adopted (Clement et al. 2008; Liu et al. 2016). The recent hype surrounding blockchain and distributed ledger technologies has seemingly provided a crucial stepping stone in this regard and could help achieve more widespread adoption of BFT protocols. Demand for private or consortium blockchains, as well as a quest for achieving better scalability in terms of transaction throughput and resource consumption, has put modern BFT consensus protocols at the heart of many new ledger designs (Vukolić 2015). Further, research and newly found insights into the fundamental principles and mechanisms of Bitcoin and similar proof-of-work-based blockchains have resulted in hybrid system models and promising new approaches for BFT protocols (e.g., Miller et al. 2016; Abraham et al. 2018; Gilad et al. 2017; Pass and Shi 2018). Together, these advancements may facilitate the deployment of BFT protocols in various systems as part of the process of exploring and familiarizing oneself how blockchain technologies could be meaningfully integrated.

17.3 What is Blockchain?

Nowadays, blockchain is all too often encountered as a marketing buzzword or fuzzy umbrella term whose intended meaning is best translated to *"technologies that are loosely related to Bitcoin"*. *Bitcoin* is a proposal and subsequent implementation of a *"peer-to-peer electronic cash system"*, whose novel approach promises to solve the distributed *double spending problem* (Jarecki and Odlyzko 1997; Hoepman 2007) without having to rely on a trusted third party.

However, beyond the hype the underlying concepts and technologies have managed to spark the interest of the scientific community and led to a plethora of research efforts and publications from various different disciplines, such as cryptography, IT security, distributed and fault-tolerant computing, formal methods, game theory, economics, and legal sciences. This serves to highlight the interdisciplinary nature of cryptocurrencies and blockchain technologies, as a new area of research is beginning to take shape.

Interestingly, the term *blockchain* itself was not directly introduced by the pseudonymous author or authors going by the name *Satoshi Nakamoto* in the original Bitcoin white paper (Nakamoto 2008); instead only the words *blocks* and *chains* are mentioned. As part of Bitcoin's underlying data structure, transactions are grouped into blocks which are linked or *chained* together using hash pointers (Narayanan et al. 2016). The combination of these words was subsequently used early on within the Bitcoin community when referring to certain concepts of this so-called cryptographic currency or simply *cryptocurrency*.

As a result, two common spellings can be encountered throughout the literature, namely, *blockchain* and *block chain*. Although the latter variant was actually used by Satoshi Nakamoto in a comment within the original source code,[3] the former, i.e., blockchain, has established itself as the de facto standard in both the community and academic literature.

Generally speaking, blockchain or blockchain technologies may be used to refer to the mechanisms and principles by which Bitcoin and similar systems are able to achieve some form of *decentralized agreement* upon a shared ledger. On the other hand, blockchain may also specifically refer to the underlying data structure of such systems. Currently, there is no broad agreement on the exact meaning of the term, and definitions are evolving as research in this field is ongoing.

17.4 The Early Days of Cryptocurrencies

In the early 1980s, around the same time early research on BFT consensus was being established, David Chaum presented the cryptographic concept of *blind signatures* (Chaum 1983) together with a use-case in the form of an untraceable (electronic) payment system. It was arguably the first step toward the development of research on (anonymous) electronic cash systems, and the heavy reliance on cryptography to instill upon such systems new desirable properties would eventually lead to the term cryptographic currency or simply *cryptocurrency*. However, while Chaum's proposal presented a significant improvement toward preserving the privacy of users, it still suffered from the drawback that a single (trusted) authority was necessary to issue currency units and prevent their *double spending*. Unfortunately, despite various commercialization efforts (Pitta 1999), this early concept failed to reach a broad audience. Nevertheless, the seed had been planted that would inspire further research toward electronic cash systems that could better satisfy desirable properties of traditional physical money.

What followed was a new generation of cryptocurrencies such as Wei Dai's *b-money* (Dai 1998), Nick Szabo's *bit gold* (Szabo 2005), Hal Finney's reusable

[3] https://github.com/trottier/original-bitcoin/blob/master/src/main.h#L795-L803.

proofs-of-work (RPOW) (Finney 2004), and Adam Back's *Hashcash* (Back 2002). While these second-generation systems still could not entirely avoid the necessity for a trusted third party, they started to incorporate an interesting cryptographic primitive as a new approach for controlling the issuance of new currency units, referred to as proof-of-work (PoW). The underlying concept of proof-of-work, namely, to require the solution to a *moderately hard* but easy to verify computation as some form of pricing mechanism, was originally devised as a means for combating junk mail by Dwork and Naor (1992).

In the context of the presented research for both BFT fault tolerance and cryptocurrencies, Bitcoin was able to provide an interesting and novel approach that appeared to be practical.

17.5 The Decentralization of Trust

Bitcoin is the first cryptographic currency that does not have to rely on a trusted third party to solve the double-spending problem. It achieves this by combining clever incentive engineering and well-studied cryptographic primitives in a novel way, such that participants are able to establish (eventual) agreement on the state changes of the underlying transaction ledger (Bonneau et al. 2015).

When Bitcoin was first presented by Satoshi Nakamoto, both the publication and subsequent release of a prototype implementation (Nakamoto 2008) garnered relatively little attention, in particular from academia.

Interestingly, the original Bitcoin white paper did not relate its proposed solution to the distributed double-spending problem to previous research on Byzantine fault tolerance or consensus,[4] thereby rendering it less likely for readers to immediately make a connection to this field of research. Similarly, from the perspective of a cryptographer, at a first glance Bitcoin did not introduce any fundamentally new concept beyond a novel application of proof-of-work.

Furthermore, it can be argued that despite its reliance on well-discussed primitives such as cryptographic hash functions, (Elliptic curve) digital signatures, and Merkle trees, the presented concept behind Bitcoin nevertheless left room for skepticism, in particular because the author(s) did not provide formalizations of the claimed properties and security guarantees of the system.

Irrespective of this initial obscurity to much of the scientific community, Bitcoin as a system continuously gained real-world adoption and quickly outgrew its hobbyist cradle, both in valuation and ability to be effectively mined on consumer hardware (Taylor 2013). In retrospect, one may argue that Bitcoin was able to

[4] Satoshi Nakamoto did claim that Bitcoin's fundamental mechanism is a solution to the Byzantine generals problem in the cryptography mailing list; see http://www.metzdowd.com/pipermail/cryptography/2008-November/014849.html.

effectively bridge various research fields precisely because it avoided placing itself into a single category early on.

The first peer-reviewed publications related to Bitcoin were published in 2011 (e.g., Reid 2011), and most of the early works covering this topic area had a focus on double-spending attacks, network properties, and the privacy guarantees that could be achieved in such systems (Androulaki et al. 2012; Ron and Shamir 2013; Meiklejohn et al. 2013).

In 2014 Miller and LaViola made a first step toward the formalization of Bitcoin's consensus mechanism in a synchronous system model by considering its applicability for solving a single instance of (eventual) binary consensus (Miller and LaViola 2014). The following year, Garay et al. presented the first formal analysis and description of Bitcoin's underlying protocol and consensus approach, referring to it as the ”Bitcoin Backbone Protocol" (Garay et al. 2015). Their initial system model assumes a static set of participants, i.e., nodes, where the ratio of computational power among them remains the same, a fully connected network that supports synchronous message communication and constant mining difficulty. Formalization efforts of the Bitcoin protocol and its underlying consensus mechanism, generally referred to as *Nakamoto consensus*, are ongoing (Stifter et al. 2018; Garay and Kiayias 2018), extending, for instance, to models of weaker (partial) synchrony (Pass et al. 2017) and chains of variable difficulty (Garay et al. 2017).

This novel (Byzantine fault tolerant) consensus approach and the practical demonstration of its feasibility are significant scientific contributions of Bitcoin. Nakamoto consensus allows for so-called *permissionless* participation (Vukolić 2015) because it only requires a very weak form of identity in the shape of computational resources. Arguably, decentralization poses the requirement that (consensus) participants are readily able to join and leave the system at will. Classical BFT consensus assumes a *static* set of participants that are *a priori determined*, because allowing for so-called dynamic group membership has proven to be difficult to solve for the Byzantine failure case, demanding strong system assumptions (Kihlstrom et al. 1998) that are unrealistic to achieve in a peer-to-peer electronic cash system over the Internet.

Part of the problem lies in preventing an adversary from simply generating multiple identities to perform a so-called Sybil attack. The concept of Sybil attacks is first introduced in Douceur (2002) and addresses the problem that an adversary in a peer-to-peer environment can cheaply (in terms of utilized resources) generate multiple identities with which to participate and thereby undermine any redundancy requirements that are employed to mitigate faulty or malicious behavior. Interestingly, a few years before the Bitcoin white paper was released, Aspnes et al. proposed the utilization of *moderately hard puzzles* as a way to expose Byzantine impostors (Aspnes et al. 2005) and address the Sybil attack. In such a model an adversary that is bounded in its computational resources also becomes bounded in the number of identities it can generate over a given period of time, thereby rendering Byzantine consensus solvable as long as a sufficiently large fraction of the overall computational power used to create identities is controlled by honest participants. Individual participants are able to join and leave the consensus process by either commencing or seizing the generation of puzzle solutions; however certain assumptions may still need to hold to provide meaningful guarantees.

Analogously, Bitcoin also leverages on proof-of-work as a core component of its consensus mechanism which acts as a weak form of identity by designating a round leader eligible for proposing the next state updates to the underlying ledger. In a sense, the group membership problem is hereby avoided and a system where participants can potentially anonymously partake in becomes possible.

Another important contribution Nakamoto consensus makes is its scalability in terms of the possible number of consensus participants (Vukolić 2015). Traditional BFT consensus protocols have so far achieved a message communication complexity that is at best quadratic in the number of participants, i.e., $O(n^2)$ (Miller et al. 2016). This generally limits the number of consensus nodes to less than one hundred active participants if the protocol is to remain practicable. Given an efficient peer-to-peer gossip mechanism and initial setup, Nakamoto consensus is able to achieve a communication complexity that lies in $O(n)$ (Garay et al. 2015).

It is these aspects of Nakamoto consensus that facilitate decentralization and permit a permissionless, peer-to-peer consensus setting.

17.6 From Bitcoin to Blockchain

With a continuous influx of new users and developers interested in Bitcoin, questions about its design and also what other applications could potentially benefit from the underlying technology were increasingly being discussed and explored.

In 2011, *Namecoin* was developed as the first successful fork and extension of the (open source) Bitcoin protocol code. Namecoin extends the concept of a cryptocurrency by adding a decentralized key-value store to allow for such use-cases as providing a decentralized domain name service (Schwarz 2011). It was followed by a growing number of alternative implementations with a variety of different goals in mind.

Alternative cryptocurrencies, in short *altcoins*, is a broad term encompassing the nowadays hundreds of cryptocurrency designs[5] that loosely follow Bitcoin's principles or its backbone protocol. Needless to say, not all altcoins have been successful, some of which only existed for a short period of time. Many of these projects are variations on the parametrization of the Bitcoin protocol with few actual modifications to the underlying code (Palmer et al. n.d; Litecoin.org n.d.). Some, however, have incorporated more profound changes and even provide entirely new code bases (King and Nadal 2012; Ben Sasson et al. 2014; Schwartz et al. 2014), where their applicability as a *cryptocurrency* may only play a secondary role, i.e., as part of a decentralized smart contract and application platform such as Ethereum (Buterin 2014a).

[5] See https://coinmarketcap.com/.

The difficulty in drawing a clear distinction between a cryptocurrency and an alternative application based on blockchain technology becomes apparent when we consider the core principles behind Nakamoto consensus: Participants compete to solve a proof-of-work of certain difficulty over their proposed state changes to the underlying ledger, referred to as *mining*. Furthermore, each puzzle input also explicitly includes the reference to a previous solution, in order to establish a causal relationship between them. As an agreement mechanism, the longest consecutive chain[6] of such puzzle solutions, starting from a pre-agreed-upon *genesis block*, is considered valid, and its current head will be referenced by honest participants when searching for new puzzle solutions.

The security of this approach also depends on the game theoretic aspect that consensus participants, so-called miners, are incentivized by being rewarded cryptocurrency units for finding valid puzzle solutions and extending the longest chain. Since this property is an intrinsic and natural byproduct of a cryptocurrency system such as Bitcoin, it is not easily replaceable in other application scenarios without potentially affecting the security guarantees of the underlying system. Therefore, as a prudent approach, many projects resort to adopting all properties of a blockchain-based cryptocurrency and add their additional application-specific components on top of them.

If we recall the previously outlined *decentralization* properties that Bitcoin's Nakamoto consensus provides, an interesting question that arises is how modifications to the underlying consensus affect the resulting system. In a distributed environment, the utilization of an authenticated data structure such as a blockchain can have its merits beyond an immediate application as part of a permissionless cryptocurrency. Depending on the application scenario, it may not actually be necessary, or even desirable, to allow such permissionless access to the underlying consensus mechanism. In particular, the required continuous resource consumption of proof-of-work renders Nakamoto consensus both impractical and insecure for small-scale deployment, as the provided security guarantees only hold under the assumption that the majority of computational power is controlled by honest participants.

Furthermore, Nakamoto consensus achieves decentralization at the cost of rendering transaction scalability seemingly more difficult to achieve than what traditional BFT consensus approaches are able to offer (Vukolić 2015). Therefore, so-called *permissioned* blockchains with alternative Byzantine fault-tolerant consensus mechanisms are increasingly being considered for corporate application scenarios (Dinh et al. 2017; Vukolić 2015). However, applying those technologies to a different use-case or system, while at the same time preserving desirable characteristics of blockchain technologies, has turned out to be a nontrivial task (Cachin and Vukolić 2017).

More recently, *hybrid* consensus models have emerged that aim toward bringing together properties from both permissioned and permissionless systems (Pass and

[6] More precisely, it is the chain with the most cumulative proof-of-work.

Shi 2017b; Luu et al. 2016; Pass and Shi 2018). The quest for addressing resource consumption in Nakamoto consensus has furthermore led to promising research and results on the topic of so-called *proof-of-stake* (PoS) consensus protocols (Kiayias et al. 2016; Bentov et al. 2016; Micali 2016). In such PoS systems, *virtual resources* in the form of cryptocurrency units are *staked* instead of requiring actual computational effort while retaining most of the desirable properties of PoW-based Nakamoto consensus.

Overall, we can conclude that the rise in popularity of Bitcoin and its derivatives has also led to an increased and renewed interest in the underlying technologies and core components behind blockchain and DLT, e.g., BFT consensus, that render such systems possible.

17.7 Future Challenges and Opportunities in Blockchain Research

Albeit academia's initial slow reaction to Bitcoin and blockchain technologies, the pace of new publications, and research has continuously increased over the last few years. With a growing understanding of the fundamental principles behind blockchain technologies, the focus is now shifted toward both new application domains and potential improvements. State-of-the-art findings and insights are increasingly being adopted and considered in new system proposals and improvements, such as Ethereum's incorporation of a variant of GHOST (Sompolinsky and Zohar 2013) as part of its design.

Blockchain and distributed ledger technologies have many different aspects and can therefore be viewed from various angles, including the *financial* and *economic* perspective, *legal* perspective, *political* and *sociological* perspective, as well as *technical* and *socio-technical* perspectives. These very different viewpoints can be separated even further; for example, the technical aspects can be divided into the following non-exhaustive list of fields: *cryptography*, *distributed computing*, *game theory*, *data science*, and *software and language security*. Because of these many different viewpoints and the broad potential applicability of these technologies, it is not only helpful but necessary to strive for interdisciplinary collaboration.

As the adoption and use of DLTs is steadily increasing, new challenges and limitations of the underlying technologies are increasingly becoming apparent (Croman et al. 2016; Vukolić 2015). In particular, concerns about future scalability and performance are a current driving force behind new research and discussions. Furthermore, many open questions on governance, the handling of human and technological failures, and other life cycle events of blockchain technologies are no longer just hypothetical (Buterin 2016) but have been rendered current and pressing issues by real-world events (De Filippi and Loveluck 2016). We outline some of these open questions in more detail.

17.7.1 Scalability

Bitcoin-like cryptocurrencies that are based on proof-of-work blockchains have certain drawbacks when it comes to scalability. Due to network latencies and structure and the very nature of the computationally expensive proof-of-work, there are certain performance limitations. The Bitcoin network is currently capable of handling around 7–10 transactions per second (Vukolić 2015; Decker and Wattenhofer 2013; Croman et al. 2016). Compared to traditional payment networks or BFT protocols, this is a relatively small number. For example, PayPal is capable of handling a few hundred transactions per second (Kiayias and Panagiotakos 2015), whereas VISA can process up to several thousand transactions per second (Kiayias and Panagiotakos 2015; Croman et al. 2016). It is well known that there are certain tradeoffs between the security and performance of PoW-based cryptocurrencies (Bamert et al. 2013; Kiayias and Panagiotakos 2015; Sompolinsky and Zohar 2013; Gervais et al. 2016). Optimizing the performance of decentralized blockchains while still being able to provide accurate estimates and formal proofs on the security impact of any changes is an ongoing topic of research. Several different approaches have been proposed that aim to minimize intrusive changes to existing protocols, such as *Bitcoin-NG* (Eyal et al. 2016). Others propose switching to entirely different underlying consensus mechanisms (Vukolić 2015; Vukolic 2016). Hybrid system models (Pass and Shi 2017a) that aim to consolidate advantages of both approaches are also being discussed. So-called layer two scaling solutions are another possibility to increase scalability by shifting some of the transaction load off-chain, i.e., in direct payment and state channels (Poon and Dryja 2016; Dziembowski et al. 2017; Coleman et al. 2018). For a general summary of possible directions, see (Croman et al. 2016).

17.7.2 Resource Consumption

All proof-of-work-based schemes rely on the existence of a limited resource that nodes are required to draw upon if they want to generate PoWs. In Bitcoin, this resource is a combination of energy, hardware, and network capacity. If there were a proof-of-work that did not rely on a limited resource, and instead could be claimed in unbounded quantities by anybody, Sibyl attacks would again become possible. It is actually the PoW that allows mining participants to remain "anonymous" and not have to reveal any previous information about themselves when participating in Nakamoto consensus. In a non-anonymous setting, this problem can be partially addressed by determining a set of nodes that are responsible for maintaining consensus on the blockchain's state; however in this case, a certain degree of trust needs to be placed in those nodes. The question of how to solve Byzantine fault tolerance in a dynamic membership setting is however still part of ongoing research.

The question that arises is whether there are provably secure yet practical and scalable schemes that permit a virtualization of the required PoW resources while still providing protection against Sibyl attacks in the permissionless model. Such a scheme would mean that instead of being forced to waste physical resources such as energy and computing hardware, one could only simply rely on virtual counterparts. One of the first approaches toward virtualizing such PoW resources, namely, *proof-of-stake* (PoS), was first introduced in cryptocurrencies such as *Peercoin* (King and Nadal 2012). The general idea behind proof-of-stake is to allow participants to lock up or *stake* part of their cryptocurrency units, which, in relation to the number of units staked by other miners, gives them a certain probability at which they can mine, or *mint*, a new block. Several difficulties and attacks with regard to proof-of-stake cryptocurrencies have been initially pointed out (Bentov et al. 2014) and until recently, concepts and presented protocols often lacked formal models and security proofs. This situation however has been amended by recent works such as *Ouroboros* (Kiayias et al. 2017) and *Snow White* (Bentov et al. 2016), which both present provably secure proof-of-stake blockchain protocols.

Another approach that could help improve the security of proof-of-stake protocols which is, for instance, being pursued by the Ethereum foundation is to integrate or leverage economic incentives and game theory in the PoS consensus process. The proposed protocol is designed to render (certain types of) malicious behavior detectable and consequently *punishes* such behavior by destroying locked-up funds or potential block rewards of the perpetrator (Buterin 2014b; Buterin and Griffith 2017). Research toward understanding and leveraging on game theoretic incentives that influence behavior of protocol participants in the realm of cryptocurrencies has been dubbed *cryptoeconomics*. In the context of traditional BFT protocols, this concept has also been explored in, e.g., the BAR (Byzantine, altruistic, rational) model (Aiyer et al. 2015; Li et al. 2006) or by reevaluating known possibility and impossibility results of distributed protocols, such as consensus, when a subset of participant is modeled as rational actors that follow certain optimization strategies (Groce et al. 2012).

17.7.3 Centralization vs. Decentralization

Studies on the mining landscape of Bitcoin, as well as other cryptocurrencies, show that there is a potential trend toward mining pool centralization in PoW-based systems (Judmayer et al. 2017). The question is, how decentralized should a cryptographic currency ecosystem be, and what methods can be used to enforce certain levels of decentralization? Which single points of failure are acceptable and which are not—for example, powerful exchanges, mining pools, and influential developers?

In the case of blockchain technologies that are based on Byzantine fault-tolerant systems, the question is how to compose and maintain a set of trusted nodes for consensus and who decides which nodes are allowed to participate. If the set of consensus nodes is small and static, resilience against Byzantine failures is more readily achievable; however the system is strongly centralized. The question of how to achieve Byzantine fault tolerance in a dynamic group membership setting which could potentially allow for more decentralization remains part of ongoing research.

17.8 Blockchain Use-Cases Beyond Cryptocurrencies

So far, we have primarily outlined the characteristics and technical challenges of blockchain technologies without expanding upon the potential use-cases that reach beyond the realm of cryptographic currencies. The following examples showcase problem domains and scenarios where an application of blockchain and distributed ledger technologies can be both promising and warranted, given that their engineering goals, challenges, desirable properties of the resulting systems, and also threat models have various overlaps and similarities to those encountered in the cryptocurrency space.

17.8.1 Trusted Timestamping and Data Provenance

The concept of trusted timestamping is not new and it has a wide range of useful applications such as providing tamper-resistant proofs of existence, for instance, for intellectual properties such as patent applications, or to document and commit to a particular state or information (e.g., a Merkle tree root which was derived from the relevant system data or a Git commit hash). In case of a system breach where the adversary may have tampered with data, such cryptographic commitments can later serve as vital references to determine data integrity.

However, this scheme of course requires that the commitment itself is safe from manipulation and ideally spread across multiple systems and media. Public proof-of-work (PoW)-based blockchains, such as Bitcoin, present an ideal platform to record these commitments as part of regular transaction data (requiring the committing party to only pay the appropriate transaction fee). The security and manipulation resistance of such blockchains stems from the sequential chaining of moderately hard puzzles which renders it (exponentially) increasingly unlikely for an adversary to be able to change any recorded transactions with respect to the length of newly mined blocks.

The advantage of PoW-based constructions over basic signature schemes with one or multiple trusted third parties is that, unless a severe flaw is found within the cryptographic hash function, no private keys or trapdoors exist that efficiently allow for equivocation. That is, if an adversary were to gain access to the private keys used in a signature-based timestamping scheme, they could readily forge backdated commitments with very little resource requirements, whereas in a PoW-based model they would have to recompute sequential PoWs which impose a highly prohibitive constraint both in terms of available time and computational resources. Blockchain-based timestamping has been described both in the scientific community (e.g., Gipp et al. 2015; Szalachowski 2018) and is employed in commercial products (cf., https://guardtime.com/ which is partnered with Lockheed Martin to secure systems engineering processes).

Permissioned blockchain systems that are based on classical BFT protocols can also offer advantages in combination with signature schemes for timestamping, especially if they employ the use of append only authenticated data structures such as hash chains. As long as the signing keys for timestamping are not reused in the BFT consensus mechanism and are therefore independent, an adversary would have to compromise both systems to effectively and fully conceal its malicious activity.

17.8.2 PKI and Digital Identities

An interesting proposition is the utilization of blockchain technologies to record identity information or serve as the basis for public key infrastructures (PKI). A general problem with most identity systems is the establishment of trusted infrastructure that secures and links cryptographic public keys to identities. Blockchain technologies could help augment traditional approaches such as certificate authorities (CAs). In particular, more recent developments in this area, such as certificate transparency, already embrace authenticated data structures as a means of identifying manipulation attempts. In the context of production systems, blockchain-based public key infrastructure could, for instance, help provide more robust mechanisms for establishing (and revoking) digital identities that are used for aspects such as access control or rights management, both in the development process and the operational design of the system. Another interesting application lies in the area of supply chain management where blockchain-based identity systems may help improve provenance. Proposals from the scientific community that leverage blockchain technologies, for instance, regarding certificate transparency, already exist (Wang et al. 2019) and there are currently concerted development efforts under way for establishing both standards and working systems for blockchain-based identity systems (e.g., https://identity.foundation/). However, many of these use-cases raise several important questions regarding user privacy and compliance with legislation and regulations, such as the EU General Data Protection Regulation (GDPR), and leave many open research questions and challenges.

17.8.3 Smart Contracts and Trusted Execution Environments

The currently established term “smart contract” is an unfortunate misnomer when it comes to succinctly describing its principal purpose or functionality, as it easily draws upon associations to legal contracts. A smart contract can be best thought of as program code that is executed in some distributed trusted execution environment. More specifically, the execution environment is generally a distributed or decentralized platform that offers both replication and, more importantly, Byzantine fault tolerance to ensure the correct execution and integrity of the smart contract code and its data storage. This is in contrast to the prevalent approach of implementing *Trusted Execution Environments* (TEEs) within computer hardware, such as Intel's SGX platform (McKeen et al. 2016; Costan and Devadas 2016), where the hardware manufacturer still acts as a single trusted third party to ensure the correctness and integrity of code execution. Permissionless blockchain-based smart contract platforms such as Ethereum (Buterin 2014a), but also permissioned counterparts such as the various incarnations of the Hyperledger platform (Cachin 2016), offer a unique trusted execution environment for program code, where the correctness and agreement upon the result of computations can be publicly verified and is secured by Byzantine agreement. As generalized computing platforms,[7] the previously mentioned use-cases can readily be implemented within smart contracts, thereby allowing the contract owner to leverage the security and availability of the base platform to provide such services without having to deploy another blockchain where the desired functionality then has to be integrated.

17.9 BFT and Blockchain Technologies in Production System Engineering

In this section we address how Byzantine fault tolerance and blockchain technologies can contribute to address the fundamental challenges and research questions that have been outlined in Chap. 1 regarding the engineering process of software-intensive technical systems.

Recapitulating the general system assumptions and challenges, this engineering process is often conducted by multiple teams and possibly subcontractors which have to collaborate and exchange engineering artifacts and other data. These collaborating actors may not extend mutual trust toward each other and there may even be incentives for participants to act dishonestly, such as attempting to gain access to inside knowledge of competitors, manipulate data and engineering artifacts, or otherwise disrupt the engineering process. This challenging collaborative environment calls for novel security methods where confidentiality, integrity, and availability can be guaranteed, as well as establishing traceability and accountability for engineering artifacts and the different actors within this environment. In the following, we address aspects among the research questions posed within Chap. 1, in particular regarding questions *RQ3a: Which security concepts mitigate cyber threats targeting the engineering process of complex cyber-physical systems?* and *RQ3b: How can the security of complex cyber-physical systems be enhanced by considering security*

[7] In principle such platforms offer Turing completeness for code; however executions are generally bounded in their complexity by requiring users to pay a certain price for each operation to prevent trivial denial of service attacks.

aspects during the engineering phase?, through domain-related research on block chain and BFT technologies.

17.9.1 Byzantine Fault Tolerance in PSE

The challenge of securing collaborative environments and shared data-stores against adversarial behavior is a long-standing research topic that has been addressed and informed by various research fields such as cryptography, (Byzantine) fault tolerance, and distributed systems. For instance Herlihy and Tygar (1987), address the question of how replicated data can be made more secure, where the notion of security encompasses the two properties of *secrecy* and *integrity*. It is argued that there seemingly is a trade-off between easier security measures for centralized services and a resilience to faults, which can be improved through redundancy. A solution employing threshold cryptography is therein presented where an adversary cannot ascertain or alter the state of a (shared) data object, if it can only compromise fewer than a threshold of repositories. Subbiah and Blough (2005) improve upon this approach by utilizing a more efficient secret sharing scheme and considers collaborative work environments. The topic of intrusion tolerance in collaborative environments is considered in Dutertre et al. (2002) where techniques for Byzantine fault tolerance and secret sharing are applied to *group communication* primitives (Chockler et al. 2001) to render the design more robust against adversaries. Kallahalla et al. (2003) present a protocol for scalable and secure file sharing using untrusted storage. Hereby, the novelty stems from a practical approach of an encrypt-on-disk system where key management and distribution is handled by the participants' clients rather than the storage provider or administrator. The approach helps to protect against data leakage attacks, e.g., by an untrusted administrator or compromised server; allows users to set arbitrary policies for key distribution; and improves scalability by shifting computationally demanding cryptographic operations to the client. In Zhao and Babi (2013) Byzantine fault tolerance in the context of real-time collaborative editing systems is addressed and a comprehensive threat analysis is performed. An interesting insight the paper presents, namely, that the detection of malicious updates to a document can only be done by its publisher or participants because it is application-specific, can be related to the more general result of Doudou et al. (2002) that the detection of Byzantine behavior by a failure detector cannot be entirely independent of the algorithm in which the failure detector is used.

17.9.2 Blockchain Technologies in PSE

We have previously outlined possible use-cases for blockchain and DLT technologies in Sect. 17.8 that reach beyond the realm of cryptocurrencies. In this regard it is not always clear if a scenario will stand to substantially benefit by its adoption. Wang et al. (2017) explores possible application scenarios of blockchain and DLT for construction engineering management and attempts to envision how the technologies may be employed in these settings. Technical details and their feasibility are presented primarily at a conceptual level, as the intention of the publication is to present a possible outlook what these technologies may offer to the problem domain. The desire or necessity to reduce reliance on *trusted third parties* and *tamper-proof documentation of interaction between parties and modification of shared data* are, among others, identified as key aspects where DLT could provide advantages.

A promising research topic related to the challenges of production system engineering is the implementation or improvement of access control mechanisms through blockchain technologies. For instance, Maesa et al. (2017) outlines, based on the example of Bitcoin, how attribute-based access control (ABAC) can be integrated into existing blockchain-based systems. In Paillisse et al. (2019) it is shown how access/policy-based networking for multi-administrative domains can be implemented and managed using blockchain technologies by presenting a prototype implementation that expands upon Group-Based Policy (GBP) and is based on the Hyperledger Fabric (Cachin 2016) blockchain framework.

17.9.3 Discussion

So far, core aspects of BFT and blockchain technologies were outlined, and research in these fields was presented that also addresses challenges which are encountered in production system engineering. Hereby we show that many of the fundamental problems, assumed system models, and security threats (e.g., the existence of multiple distrusting parties that may act maliciously while at the same time have to collaborate to produce some common result, the necessity to provide resistance against manipulation of shared data) between these different fields are closely related. Decades of insights and improvements in developing practical Byzantine fault-tolerant consensus protocols, as well as the hype and subsequent explosion of research in blockchain technologies, can be leveraged when seeking to provide solutions or improvements to the security and quality of production system engineering as well as the long-running software-intensive technical systems that are hereby created.

As a concrete example, consider Chap. 12 of this book, which covers the topic of securing information manipulation in PSE. The therein assumed system architecture, as depicted in Fig. 12.3, may be augmented with concepts and techniques described within this chapter. Instead of relying on a single centralized PSE platform to exchange data, for example, the secure and scalable file sharing approach outlined by Kallahalla et al. (2003), could prove both beneficial and practical. Its design considers key exchange and access management on client devices rather than by a centralized provider. This aspect could further be strengthened by employing blockchain-based access control management where said clients act as nodes in either a public or private DLT network and commit all relevant updates to the access rights to the ledger. The advantage of such an approach is that a tamper-resistant, (eventually) ordered log of events is distributed among the participants such that manipulation by a subset of them is rendered difficult and readily detectable.

Further research in this regard is both necessary and warranted to determine if such an application of modern BFT protocols and blockchain technologies can indeed lead to practical designs and techniques for production system engineering and, more importantly, contribute toward an improvement in their quality and security.

17.10 Conclusion

While blockchain technologies are hardly the answer to life, the universe, and everything, as ideologists or advertising sometimes paint it,[8] the fusion of its underlying principles and methods has opened up new pathways and outlined new possibilities in different areas of research. Bitcoin has created a new class of BFT consensus systems and rekindled research in the field of distributed computing and Byzantine fault tolerance, leading to new and interesting permissioned, permissionless, and hybrid blockchain constructs. It furthermore bootstrapped a vivid and diverse community that is driving the development and practical application of this set of technologies further.

The renewed interest in Byzantine fault tolerance, fueled by the hype surrounding blockchain technologies, may also prove to be highly beneficial to a variety of other problem domains, such as the herein discussed topic of production system engineering. Intriguing design proposals, promising system architectures, and even fully functional prototypes that tolerate Byzantine failures are being rediscovered and reconsidered as both practical and desirable approaches when evaluating if blockchain or DLT could benefit a particular use-case. Many of the examples presented in this chapter highlight that these technologies should not be considered as mutually exclusive and can actually stand to benefit from each other, showcasing that interdisciplinary thinking can lead to novel approaches and solutions with practical applications. It remains to be seen what impact blockchain technologies and cryptocurrencies will ultimately have on society and technology; however it is clear that they have the potential to be more disruptive than just a superficial speculative bubble.

Acknowledgements We thank Georg Merzdovnik as well as the participants of Dagstuhl Seminar 18152 “Blockchains, Smart Contracts and Future Applications” for valuable discussions and insights. This research was funded by Bridge 1 858561 SESC, Bridge 1 864738 PR4DLT (all FFG), the Christian Doppler Laboratory for *Security and Quality Improvement in the Production System Lifecycle (CDL-SQI)*, Institute of Information Systems Engineering, TU Wien, and the competence center SBA-K1 funded by COMET. The financial support by the Christian Doppler

[8] Cf., https://www.theguardian.com/world/2016/jul/07/blockchain-answer-life-universe-everything-bitcoin-technology.

Research Association; the Austrian Federal Ministry for Digital and Economic Affairs; and the National Foundation for Research, Technology, and Development is gratefully acknowledged.

References

Abraham, I., Gueta, G., & Malkhi, D. (2018). Hot-stuff the linear, optimal-resilience, one-message bft devil. arXiv:1803.05069. https://arxiv.org/pdf/1803.05069.pdf

Aiyer, A. S., Alvisi, L., Clement, A., Dahlin, M., Martin, J.-P., & Porth, C. (2005). Bar fault tolerance for cooperative services. In *ACM SIGOPS Operating Systems Review* (Vol. 39, pp. 45–58). New York, NY: ACM. http://www.dcc.fc.up.pt/~Ines/aulas/1314/SDM/papers/BAR%20Fault%20Tolerance%20for%20Cooperative%20Services%20-%20UIUC.pdf

Androulaki, E., Capkun, S., & Karame, G. O. (2012). Two bitcoins at the price of one? Double-spending attacks on fast payments in bitcoin. In *CCS*. http://eprint.iacr.org/2012/248.pdf

Aspnes, J., Jackson, C., & Krishnamurthy, A. (2005). *Exposing Computationally-Challenged Byzantine Impostors*. Department of Computer Science, Yale University, New Haven, CT, Tech. Rep. http://www.cs.yale.edu/homes/aspnes/papers/tr1332.pdf

Back, A. (2002). Hashcash-a denial of service counter-measure. Retrieved March 9, 2016, from http://www.hashcash.org/papers/hashcash.pdf

Bamert, T., Decker, C., Elsen, L., Wattenhofer, R., & Welten, S. (2013). Have a snack, pay with bitcoins. In *2013 IEEE Thirteenth International Conference on IEEE Peer-to-Peer Computing (P2P)* (pp. 1–5). Piscataway, NJ: IEEE. http://www.bheesty.com/cracker/1450709524_17035424cb/p2p2013_093.pdf

Ben-Or, M. (1983). Another advantage of free choice (extended abstract): Completely asynchronous agreement protocols. In *Proceedings of the Second Annual ACM Symposium on Principles of Distributed Computing* (pp. 27–30). New York, NY: ACM. http://homepage.cs.uiowa.edu/~ghosh/BenOr.pdf

Ben Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., et al. (2014). Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy (SP)* (pp. 459–474). Piscataway, NJ: IEEE. http://zerocash-project.org/media/pdf/zerocash-extended-20140518.pdf

Bentov, I., Lee, C., Mizrahi, A., & Rosenfeld, M. (2014). Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y. *ACM SIGMETRICS Performance Evaluation Review, 42*(3), 34–37. http://eprint.iacr.org/2014/452.pdf

Bentov, I., Pass, R., & Shi, E. (2016). Snow white: Provably secure proofs of stake. Retrieved November 11, 2016, from https://eprint.iacr.org/2016/919.pdf

Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., & Felten, E. W. (2015). Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *IEEE Symposium on Security and Privacy*. http://www.ieee-security.org/TC/SP2015/papers-archived/6949a104.pdf

Buterin, V. (2014a). Ethereum: A next-generation smart contract and decentralized application platform. Retrieved August 22, 2016, from https://github.com/ethereum/wiki/wiki/White-Paper

Buterin, V. (2014b). Slasher: A punitive proof-of-stake algorithm. Retrieved March 24, 2017, from https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm/

Buterin, V. (2016). Chain interoperability. Retrieved March 25, 2017, from https://static1.squarespace.com/static/55f73743e4b051cfcc0b02cf/t/5886800ecd0f68de303349b1/1485209617040/Chain+Interoperability.pdf

Buterin, V., & Griffith, V. (2017). Casper the friendly finality gadget. arXiv:1710.09437. Retrieved November 6, 2017, from https://arxiv.org/pdf/1710.09437.pdf

Cachin, C. (2016). Architecture of the hyperledger blockchain fabric. Retrieved August 10, 2016, from https://www.zurich.ibm.com/dccl/papers/cachin_dccl.pdf

Cachin, C., Kursawe, K., & Shoup, V. (2000). Random oracles in constantinople: Practical asynchronous byzantine agreement using cryptography. In *Proceedings of the Nineteenth Annual*

ACM Symposium on Principles of Distributed Computing (pp. 123–132). New York, NY: ACM. https://www.zurich.ibm.com/~cca/papers/abba.pdf

Cachin, C., & Vukolić, M. (2017). Blockchain consensus protocols in the wild. In *31 International Symposium on Distributed Computing*. arXiv preprint arXiv:1707.01873

Castro, M., & Liskov, B. (2002). Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems, 20*, 398–461.

Castro, M., Liskov, B. (1999). Practical byzantine fault tolerance. In *OSDI* (Vol. 99, pp. 173–186). http://pmg.csail.mit.edu/papers/osdi99.pdf

Chandra, T. D., Griesemer, R., & Redstone, J. (2007). Paxos made live: An engineering perspective. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Principles of Distributed Computing* (pp. 398–407). New York, NY: ACM. https://www.kth.se/polopoly_fs/1.116933!/Menu/general/column-content/attachment/paxoslive.pdf

Chaum, D. (1983). Blind signatures for untraceable payments. In *Advances in cryptology* (pp. 199–203). Berlin: Springer. http://blog.koehntopp.de/uploads/Chaum.BlindSigForPayment.1982.PDF

Chockler, G. V., Keidar, I., & Vitenberg, R. (2001). Group communication specifications: a comprehensive study. *ACM Computing Surveys, 33*(4), 427–469.

Clement, A., Marchetti, M., Wong, E., Alvisi, L., & Dahlin, M. (2008). BFT: the time is now. In *Proceedings of the 2nd Workshop on Large-Scale Distributed Systems and Middleware* (p. 13). New York, NY: ACM.

Clement, A., Wong, E. L., Alvisi, L., Dahlin, M., & Marchetti, M. (2009). Making byzantine fault tolerant systems tolerate byzantine faults. In *NSDI* (Vol. 9, pp. 153–168). http://static.usenix.org/events/nsdi09/tech/full_papers/clement/clement.pdf

Coleman, J., Horne, L., & Xuanji, L. (2018). Counterfactual: Generalized state channels [online]. Retrieved May 18, 2019, from https://l4.ventures/papers/statechannels.pdf

Costan, V., & Devadas, S. (2016). Intel sgx explained. *IACR Cryptology ePrint Archive, 2016*(86), 1–118.

Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., et al. (2016). On scaling decentralized blockchains. In *3rd Workshop on Bitcoin and Blockchain Research, Financial Cryptography 16*. http://www.tik.ee.ethz.ch/file/74bc987e6ab4a8478c04950616612f69/main.pdf

Dai, W. (1998). bmoney. Retrieved April 4, 2016, from http://www.weidai.com/bmoney.txt

De Filippi, P., & Loveluck, B. (2016). The invisible politics of bitcoin: governance crisis of a decentralised infrastructure. Retrieved October 18, 2017, from https://halshs.archives-ouvertes.fr/halshs-01380617/document

Decker, C., & Wattenhofer, R. (2013). Information propagation in the bitcoin network. In *2013 IEEE Thirteenth International Conference on Peer-to-Peer Computing (P2P)* (pp. 1–10). Piscataway, NJ: IEEE. http://diyhpl.us/~bryan/papers2/bitcoin/Information%20propagation%20in%20the%20Bitcoin%20network.pdf

Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., & Tan, K.-L. (2017). Blockbench: A framework for analyzing private blockchains. In *Proceedings of the 2017 ACM International Conference on Management of Data* (pp. 1085–1100). New York, NY: ACM.

Douceur, J. R. (2002). The sybil attack. In *International Workshop on Peer-to-Peer Systems* (pp. 251–260). Berlin: Springer. http://www.cs.cornell.edu/people/egs/cs6460-spring10/sybil.pdf

Doudou, A., Garbinato, B., & Guerraoui, R. (2002). Encapsulating failure detection: From crash to byzantine failures. In *International Conference on Reliable Software Technologies* (pp. 24–50). Berlin: Springer.

Dutertre, B., Crettaz, V., & Stavridou, V. (2002). Intrusion-tolerant enclaves. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy* (pp. 216–224). Piscataway, NJ: IEEE.

Dwork, C., & Naor, M. (1992). Pricing via processing or combatting junk mail. In *Annual International Cryptology Conference* (pp. 139–147). Berlin: Springer. https://web.cs.dal.ca/~abrodsky/7301/readings/DwNa93.pdf

Dziembowski, S., Eckey, L., Faust, S., & Malinowski, D. (2017). *Perun: Virtual Payment Channels Over Cryptographic Currencies*. Cryptology ePrint Archive, Report 2017/635. Retrieved November 20, 2017, from https://eprint.iacr.org/2017/635.pdf

Esteves-Verissimo, P., Völp, M., Decouchant, J., Rahli, V., & Rocha, F. (2017). Meeting the challenges of critical and extreme dependability and security. In *2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC)* (pp. 92–97). Piscataway, NJ: IEEE.

Eyal, I., Gencer, A. E., Sirer, E. G., & van Renesse, R. (2016). Bitcoin-ng: A scalable blockchain protocol. In *13th USENIX Security Symposium on Networked Systems Design and Implementation (NSDI'16)*. Berkeley, CA: USENIX Association. http://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-eyal.pdf

F. Reid, M. H. (2011). An analysis of anonymity in the bitcoin system. In *2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing*. http://arxiv.org/pdf/1107.4524

Finney, H. (2004). Reusable proofs of work (RPOW). Retrieved April 31, 2016, from http://web.archive.org/web/20071222072154/http://rpow.net/

Fischer, M. J., Lynch, N. A., & Paterson, M. S. (1985). Impossibility of distributed consensus with one faulty process. *Journal of the ACM, 32*, 374–382. http://macs.citadel.edu/rudolphg/csci604/ImpossibilityofConsensus.pdf

Garay, J., & Kiayias, A. (2018). *Sok: A Consensus Taxonomy in the Blockchain Era*. Cryptology ePrint Archive, Report 2018/754. https://eprint.iacr.org/2018/754.pdf

Garay, J., Kiayias, A., & Leonardos, N. (2015). The bitcoin backbone protocol: Analysis and applications. In *Advances in Cryptology-EUROCRYPT 2015* (pp. 281–310). Berlin: Springer. http://courses.cs.washington.edu/courses/cse454/15wi/papers/bitcoin-765.pdf

Garay, J., Kiayias, A., & Leonardos, N. (2017). The bitcoin backbone protocol with chains of variable difficulty. In *Annual International Cryptology Conference* (pp. 291–323). Berlin: Springer.

Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., & Capkun, S. (2016). On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC* (pp. 3–16). New York, NY: ACM.

Gilad, Y., Hemo, R., Micali, S., Vlachos, G., & Zeldovich, N. (2017). Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles* pp. 51–68. New York, NY: ACM.

Gipp, B., Meuschke, N., & Gernandt, A. (2015). Decentralized trusted timestamping using the crypto currency bitcoin. preprint arXiv:1502.04015.

Groce, A., Katz, J., Thiruvengadam, A., & Zikas, V. (2012). *Byzantine agreement with a rational adversary* (pp. 561–572). Berlin: Springer. http://cs.ucla.edu/~vzikas/pubs/GKTZ12.pdf

Guerraoui, R., Knežević, N., Quéma, V., & Vukolić, M. (2010). The next 700 bft protocols. In *Proceedings of the 5th European Conference on Computer Systems* (pp. 363–376). New York, NY: ACM. https://infoscience.epfl.ch/record/121590/files/TR-700-2009.pdf

Herlihy, M. P., & Tygar, J. D. (1987). How to make replicated data secure. In *Conference on the Theory and Application of Cryptographic Techniques* (pp. 379–391). Berlin: Springer.

Hoepman, J.-H. (2007). Distributed double spending prevention. In *Security Protocols Workshop* (pp. 152–165). Berlin: Springer. http://www.cs.kun.nl/~jhh/publications/double-spending.pdf

Jakobsson, M., & Juels, A. (1999). Proofs of work and bread pudding protocols. In *Secure information networks* (pp. 258–272). Berlin: Springer. https://link.springer.com/content/pdf/10.1007/978-0-387-35568-9_18.pdf

Jarecki, S., & Odlyzko, A. (1997). An efficient micropayment system based on probabilistic polling. In *Financial cryptography* (pp. 173–191). Berlin: Springer. https://www.researchgate.net/profile/Stanislaw_Jarecki/publication/220797099_An_Efficient_Micropayment_System_Based_on_Probabilistic_Polling/links/0f31753c7f02552a9d000000.pdf

Judmayer, A., Zamyatin, A., Stifter, N., Voyiatzis, A. G., & Weippl, E. (2017). Merged mining: Curse or cure? In *Proceedings of the International Workshop on Cryptocurrencies and Blockchain Technology, CBT'17*. https://eprint.iacr.org/2017/791.pdf

Kallahalla, M., Riedel, E., Swaminathan, R., Wang, Q., & Fu, K. (2003). Plutus: scalable secure file sharing on untrusted storage. In *Proceedings of the 2nd USENIX Conference on File and Storage Technologies* (pp. 3). Berkeley, CA: USENIX Association

Kiayias, A., Konstantinou, I., Russell, A., David, B., & Oliynykov, R. (2016). A provably secure proof-of-stake blockchain protocol. Retrieved November 9, 2016, from http://eprint.iacr.org/2016/889.pdf

Kiayias, A., & Panagiotakos, G. (2015). Speed-security tradeoff s in blockchain protocols. Retrieved October 17, 2016, from https://eprint.iacr.org/2015/1019.pdf

Kiayias, A., Russell, A., David, B., & Oliynykov, R. (2017). Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference* (pp. 357–388). Berlin: Springer.

Kihlstrom, K. P., Moser, L. E., & Melliar-Smith, P. M. (1998). The securering protocols for securing group communication. In *Proceedings of the Thirty-First Hawaii International Conference on System Sciences* (Vol. 3, pp. 317–326). Piscataway, NJ: IEEE.

King, S., & Nadal, S. (2012). Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. Retrieved January 7, 2017, from https://peercoin.net/assets/paper/peercoin-paper.pdf

Lamport, L. (1984). Using time instead of timeout for fault-tolerant distributed systems. *ACM Transactions on Programming Languages and Systems, 6*, 254–280. http://131.107.65.14/en-us/um/people/lamport/pubs/using-time.pdf

Lamport, L. (1998). The part-time parliament. *ACM Transactions on Computer Systems, 16*, 133–169. https://www.microsoft.com/en-us/research/uploads/prod/2016/12/The-Part-Time-Parliament.pdf

Lamport, L., Shostak, R., & Pease, M. (1982). The byzantine generals problem. *ACM Transactions on Programming Languages and Systems, 4*, 382–401. http://people.cs.uchicago.edu/~shanlu/teaching/33100_wi15/papers/byz.pdf

Li, H. C., Clement, A., Wong, E. L., Napper, J., Roy, I., Alvisi, L., & Dahlin, M. (2006). Bar gossip. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation* (pp. 191–204). Berkeley, CA: USENIX Association. http://www.cs.utexas.edu/users/dahlin/papers/bar-gossip-apr-2006.pdf

Litecoin.org. (n.d.). Retrieved May 18, 2019, from https://litecoin.org/

Liu, S., Viotti, P., Cachin, C., Quéma, V., & Vukolić, M. (2016). XFT: Practical fault tolerance beyond crashes. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* (pp. 485–500).

Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., & Saxena, P. (2016). A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 17–30). New York, NY: ACM. https://www.comp.nus.edu.sg/~prateeks/papers/Elastico.pdf

Maesa, D. D. F., Mori, P., & Ricci, L. (2017). Blockchain based access control. In *IFIP International Conference on Distributed Applications and Interoperable Systems* (pp. 206–220). Berlin: Springer.

McKeen, F., Alexandrovich, I., Anati, I., Caspi, D., Johnson, S., Leslie-Hurd, R., et al. (2016). Intel® software guard extensions (intel® sgx) support for dynamic memory management inside an enclave. In *Proceedings of the Hardware and Architectural Support for Security and Privacy 2016* (p. 10). New York, NY: ACM.

Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., & Savage, S. (2013). A fistful of bitcoins: Characterizing payments among men with no names. In *Proceedings of the 2013 Conference on Internet Measurement Conference* (pp. 127–140). New York, NY: ACM. https://cseweb.ucsd.edu/~smeiklejohn/files/imc13.pdf

Micali, S. (2016). Algorand: The efficient and democratic ledger. Retrieved Febraury 9, 2017, from https://arxiv.org/pdf/1607.01341.pdf

Miller, A., & LaViola, J. J. (2014). Anonymous byzantine consensus from moderately-hard puzzles: A model for bitcoin. Retrieved March 9, 2016, from https://socrates1024.s3.amazonaws.com/consensus.pdf

Miller, A., Xia, Y., Croman, K., Shi, E., & Song, D. (2016). The honey badger of bft protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 31–42). New York, NY: ACM. https://eprint.iacr.org/2016/199.pdf

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Retrieved July 1, 2015, from https://bitcoin.org/bitcoin.pdf

Narayanan, A., Bonneau, J., Felten, E., Miller, A., Miller, A., & Goldfeder, S. (2016). Bitcoin and cryptocurrency technologies. Retrieved March 29, 2016, from https://d28rh4a8wq0iu5.cloudfront.net/bitcointech/readings/princeton_bitcoin_book.pdf

Paillisse, J., Subira, J., Lopez, A., Rodriguez-Natal, A., Ermagan, V., Maino, F., & Cabellos, A. (2019). Distributed access control with blockchain. preprint arXiv:1901.03568.

Palmer, J., Nakamoto S., /u/PowerLemons, Ricks, C. (n.d.). Dogecoin.com [online]. Retrieved May 18, 2019, from https://dogecoin.com/

Pass, R., Seeman, L., & Shelat, A. (2017). Analysis of the blockchain protocol in asynchronous networks. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 643–673). Berlin: Springer.

Pass, R., & Shi, E. (2017a). Hybrid consensus: Efficient consensus in the permissionless model. In *31st International Symposium on Distributed Computing (DISC 2017)* Merzig-Wadern: Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Pass, R., & Shi, E. (2017b). The sleepy model of consensus. In *International Conference on the Theory and Application of Cryptology and Information Security* (pp. 380–409). Berlin: Springer.

Pass, R., & Shi, E. (2018). Thunderella: Blockchains with optimistic instant confirmation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 3–33). Berlin: Springer.

Pease, M., Shostak, R., & Lamport, L. (1980). Reaching agreement in the presence of faults. *Journal of the ACM, 27*, 228–234. https://www.microsoft.com/en-us/research/uploads/prod/2016/12/Reaching-Agreement-in-the-Presence-of-Faults.pdf

Pitta, J. (1999). Requiem of a bright idea [online]. Retrieved May 18, 2019, from http://www.forbes.com/forbes/1999/1101/6411390a.html

Poon, J., & Dryja, T. (2016). The bitcoin lightning network. Retrieved July 7, 2016, from https://lightning.network/lightning-network-paper.pdf

Rabin, M. O. (1983). Randomized byzantine generals. In *24th Annual Symposium on Foundations of Computer Science* (pp. 403–409). Piscataway, NJ: IEEE. https://www.cs.princeton.edu/courses/archive/fall05/cos521/byzantin.pdf

Ron, D., & Shamir, A. (2013). Quantitative analysis of the full bitcoin transaction graph. In *International Conference on Financial Cryptography and Data Security* (pp. 6–24). Berlin: Springer.

Schneider, F. B. (1990). Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys, 22*, 299–319. http://www-users.cselabs.umn.edu/classes/Spring-2014/csci8980-sds/Papers/ProcessReplication/p299-schneider.pdf

Schwartz, D., Youngs, N., & Britto, A. (2014). The ripple protocol consensus algorithm. Retrieved August 8, 2016, from https://ripple.com/files/ripple_consensus_whitepaper.pdf

Schwarz, A. (2011) Squaring the triangle: Secure, decentralized, human-readable names. Retrieved November 12, 2014, from http://www.aaronsw.com/weblog/squarezooko

Sompolinsky, Y., & Zohar, A. (2013). Accelerating bitcoin's transaction processing. Fast money grows on trees, not chains. http://eprint.iacr.org/2013/881.pdf

Stifter, N., Judmayer, A., Schindler, P., Zamyatin, A., & Weippl, E. (2018). *Agreement with Satoshi—on the Formalization of Nakamoto Consensus*. Cryptology ePrint Archive, Report 2018/400. https://eprint.iacr.org/2018/400.pdf

Subbiah, A., & Blough, D. M. (2005). An approach for fault tolerant and secure data storage in collaborative work environments. In *Proceedings of the 2005 ACM workshop on Storage Security and Survivability* (pp. 84–93). New York, NY: ACM.

Szabo, N. (2005). Bit gold. Retrieved April 4, 2016, from http://unenumerated.blogspot.co.at/2005/12/bit-gold.html

Szalachowski, P. (2018). (short paper) towards more reliable bitcoin timestamps. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)* (pp. 101–104). Piscataway, NJ: IEEE.

Taylor, M. B. (2013). Bitcoin and the age of bespoke silicon. In *Proceedings of the 2013 International Conference on Compilers, Architectures and Synthesis for Embedded Systems* (p. 16). Piscataway, NJ: IEEE Press. https://cseweb.ucsd.edu/~mbtaylor/papers/bitcoin_taylor_cases_2013.pdf

Veronese, G. S., Correia, M., Bessani, A. N., & Lung, L. C. (2009). Highly-resilient services for critical infrastructures. In *Proceedings of the Embedded Systems and Communications Security Workshop*.

Veronese, G. S., Correia, M., Bessani, A. N., Lung, L. C., & Verissimo, P. (2013). Efficient byzantine fault-tolerance. *IEEE Transactions on Computers, 62*, 16–30. https://www.researchgate.net/profile/Miguel_Correia3/publication/260585535_Efficient_Byzantine_Fault-Tolerance/links/5419615d0cf25ebee9885215.pdf

Vukolić, M. (2010). The byzantine empire in the intercloud. *ACM Sigact News, 41*(3), 105–111.

Vukolić, M. (2015). The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In *International Workshop on Open Problems in Network Security* (pp. 112–125). Berlin: Springer. http://vukolic.com/iNetSec_2015.pdf

Vukolic, M. (2016). Eventually returning to strong consistency. Retrieved August 10, 2016, from https://pdfs.semanticscholar.org/a6a1/b70305b27c556aac779fb65429db9c2e1ef2.pdf

Wang, J., Wu, P., Wang, X., & Shou, W. (2017). The outlook of blockchain technology for construction engineering management. *Frontiers of Engineering Management, 4*(1), 67–75.

Wang, Z., Lin, J., Cai, Q., Wang, Q., Jing, J., & Zha, D. (2019). Blockchain-based certificate transparency and revocation transparency. In *International Conference on Financial Cryptography and Data Security* (pp 144–162). Berlin: Springer.

Wensley, J. H., Lamport, L., Goldberg, J., Green, M. W., Levitt, K. N., Melliar-Smith, P. M., et al. (1978). Sift: Design and analysis of a fault-tolerant computer for aircraft control. *Proceedings of the IEEE, 66*(10), 1240–1255.

Zhao, W., & Babi, M. (2013). Byzantine fault tolerant collaborative editing. In *IET International Conference on Information and Communications Technologies (IETICT 2013)* (pp. 233–240). IET.