A Survey of Consensus Algorithms in Public Blockchain Systems for Crypto-currencies

Md Sadek Ferdous[a,b,*], Mohammad Jabed Morshed Chowdhury[c], Mohammad A. Hoque[d]

[a]*Shahjalal University of Science and Technology, Sylhet 3114, Bangladesh.*
[b]*Imperial College London, London SW7 2AZ, U.K. E-mail:s.ferdous@imperial.ac.uk*
[c]*La Trobe University, Melbourne, Victoria-3086, Australia*
[d]*University of Helsinki, 3835 Helsinki, Helsinki Finland.*

Abstract

In recent years, crypto-currencies (a form of decentralised digital currencies) have been quite popular as an alternative form of payments. They are underpinned by a breakthrough technology called *Blockchain* which extensively use a number of cryptographic mechanisms and other advanced techniques from the domain of distributed computing. This blockchain technology has received unparalleled attention from academia, industry, and governments worldwide and is considered to have the potential to disrupt several application domains, other than currencies, touching all spheres of our lives. The sky-rocket anticipation of its potential has caused a wide-scale exploration of its usage in different application domains. This has resulted in a plethora of blockchain systems for various purposes. However, many of these blockchain systems suffer from serious shortcomings related to their performance and security, which need to be addressed before any wide-scale adoption can be achieved. A crucial component of any blockchain system is its underlying consensus algorithm, which determines its performance and security in many ways. Therefore, to address the limitations of different blockchain systems, several existing as well novel consensus algorithms have been introduced. A systematic analysis of these algorithms will help to understand how and why any particular blockchain performs the way it functions. Towards this aim, there are a number of existing works that have surveyed and reviewed a number of consensus algorithms. However, all these works have some major shortcomings. For example, the factors upon which the consensus algorithms have been analysed are not comprehensive. Importantly, a wide range of consensus algorithms utilised in public blockchain systems supporting mainly crypto-currencies have different variants. Such variants and their internal mechanisms utilised in many existing crypto-currencies have not been considered at all. This article fills these gaps by analysing a wide range of consensus algorithms leveraged in different public blockchain systems using a comprehensive taxonomy of properties. We have also analysed more than a hundred top crypto-currencies belonging to different categories of consensus algorithms to understand their properties and implicate different trends in these crypto-currencies. Finally, we have presented a decision tree of the reviewed algorithms to be used as a tool to test the suitability of consensus algorithms for a particular application under different criteria.

Keywords: Blockchain, Distributed Consensus, Proof of Work, PoW, Proof of Stake, PoS, Delegated Proof of Stake, DPoS.

1. Introduction

Introduced in 2008, Bitcoin [1] has emerged as the first widely-used decentralised digital currency in the world. A decentralised currency, unlike its counterpart centralised currency (or *fiat currency*) does not rely on any central entity such as a central bank for its issuance and circulation. Motivated with the technological breakthrough and financial success of Bitcoin, a plethora of such digital currencies have emerged. A recent estimate suggests that there are currently around $5,583$ digital currencies in the world as of June, 2020 [2]. All such decentralised digital currencies are underpinned by a novel technology called *Blockchain Technology* along with an intelligent combination of cryptography and distributed computing. Because of their extensive utilisation of cryptographic mechanisms, such digital currencies are colloquially known as *crypto-currencies*.

In the last few years, this blockchain technology has received wide-spread attention among the industry, the Government, and academia alike. While crypto-currencies have emerged as the principal and the most popular application of blockchain technology, many enthusiasts from different disciplines have identified and proposed a plethora of other applications of blockchain in a multitude of domains [3, 4]. The possibility of exploiting blockchain in so many areas has created huge anticipation surrounding blockchain systems. Indeed, it is regarded as one of the fundamental technologies to revolutionise the landscapes of the identified application domains.

A blockchain system is, fundamentally, a distributed system that relies on a consensus algorithm that ensures agreement on the states of certain data among distributed nodes. A consensus algorithm is the core component that directly dictates how the system behaves and the performance it can achieve. Distributed consensus has been a widely studied research topic in distributed systems, however, with the advent of blockchain, it has received renewed attention. A wide variety of crypto-currencies

[*] Corresponding author

Email addresses: sadek-cse@sust.edu (Md Sadek Ferdous), m.chowdhury@latrobe.edu.au (Mohammad Jabed Morshed Chowdhury), mohammad.a.hoque@helsinki.fi (Mohammad A. Hoque)

Author version. Published in Elsevier Journal of Network and Computer Applications

targeting different application domains has introduced an array of unique requirements that can only be satisfied by their corresponding consensus mechanisms. This fact has fuelled the need not only to examine the applicability of existing consensus algorithms in newer settings, but also to innovate novel consensus algorithms. Consequently, several consensus algorithms have emerged, each of which possesses interesting properties and unique capabilities.

As the characteristics of various types of blockchain systems are fundamentally dependent on the consensus algorithms they use, a systematic analysis of existing consensus algorithms is required to examine, compare, and contrast these algorithms. Towards this aim, there have been a number of attempts which can be found in [194, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. However, all these works have some major shortcomings. For example, the factors upon which the consensus algorithms have been analysed are not comprehensive. Importantly, a wide range of consensus algorithms utilised in public blockchain systems supporting mainly crypto-currencies have different variants. Such variants and their internal mechanisms utilised in many existing crypto-currencies have not been considered at all. In addition, all of these studies have failed to capture the practical interrelation between such blockchain systems and their corresponding consensus algorithms. All in all, there is a pressing need for a study that analyses a wide range of existing consensus algorithms and their variants in public blockchain systems in an implementation-oriented way and synthesises this analysis into a conceptual framework in a concise yet comprehensive manner. The principal motivation of this article is to fill in this gap. A table illustrating the gap is provided in Table 18 which clearly highlights the need for a review work as the one presented in this article.

Scope & Target audience. The principal scope of this survey article is the consensus algorithms and their variants, which have been exclusively used in different public blockchain systems supporting crypto-currencies. There are other consensus algorithms utilised in non-blockchain systems, systems which do not utilise a blockchain, e.g. IOTA [165], Nano [166] or even in private blockchain systems, e.g. Hyperledger platforms [163], Quorum [164]. Such consensus algorithms are excluded from consideration in this article. The target audience for this article is anyone, academicians from any discipline, industrial practitioners or even general public, who would like to familiarise themselves with the inner-workings as well as a comparative analysis of different consensus algorithms available in public blockchain systems with crypto-currencies. The survey is also useful for anyone who would like to choose a particular consensus algorithm for a new blockchain system.

Contributions. The main contributions of the article are presented below:

- A taxonomy of consensus properties, capturing different aspects of a consensus algorithm, has been compiled.
- Three major types of consensus algorithms, namely PoW (Prof of Work), Proof of Stake (PoS) and others (representing algorithms beyond PoW and PoS), covering a wide variety of public blockchain systems have been presented, compared and analysed against the properties of the consensus taxonomy.
- The major issues in each category of consensus algorithm have been examined in detail, and their implications have been further analysed.
- Over hundred crypto-currencies, that utilise different consensus algorithms, have been examined to understand their different properties. These properties then have been utilised to analyse and identify different future trends among these crypto-currencies.
- Finally, a decision tree of consensus algorithms has been presented. This tree can test a consensus algorithm's suitability for a particular application.

In short, with these contributions, this article represents one of the most comprehensive studies of consensus algorithms for public blockchain systems and crypto-currencies as of now.

Structure. In Section 2, we present a brief background on distributed consensus and blockchain highlighting their different aspects, components and properties. A taxonomy of consensus algorithms and their underlying properties is presented in Section 3. Different PoW, PoS, and other types of consensus algorithms have been analysed in Section 4, Section 5 and Section 6 respectively. In Section 7, a detailed discussion on different issues involving the analysed consensus algorithms and the corresponding crypto-currencies is presented along with the decision tree. We compare and contrast our review with the existing related surveys and highlight the gaps in the previous surveys that our work is trying to fulfil in Section 8. Then, we present a few existing challenges and possible future research directions in Section 9. Finally, we conclude in Section 10.

2. Background

In this section, we present a brief background of distributed consensus as well as blockchain and its related terminologies in Section 2.1 and Section 2.2 respectively.

2.1. Distributed Consensus

Consensus mechanisms in distributed systems have been a well studied research problem for nearly three decades [169]. Such mechanisms enable consensus to be achieved regarding a shared state/data among a set of distributed nodes. The need for a shared state created the notion of replicated database systems in order to ensure resiliency against node failures within a network. Such database systems ensure that data is not lost when one or more nodes fail to function in an expected fashion. To ensure consensus among the distributed nodes, it must be ensured that these nodes receive the same set of messages in the exact same order (the phenomenon known as *atomic broadcast*). It is imperative that a protocol is defined to ensure the timely dissemination and atomic broadcast of messages among

Properties	Note
Safety/ Consistency	A consensus protocol is considered safe (or consistent) only when all nodes produce the same valid output, according to the protocol rules, for the same atomic broadcast.
Liveness/ availability	If all non-faulty participating nodes produce an output (indicating the termination of the protocol), the protocol is considered live.
Fault Tolerance	It exhibits the network's capability to perform as intended in the midst of node failures.

Table 1: Properties of Distributed Consensus Protocols.

the nodes and, in many ways, dictates how a distributed consensus is achieved and maintained. Hence, such a protocol is aptly called a consensus protocol.

Designing and deploying a consensus protocol is a challenging task as it needs to consider several crucial issues such as resiliency against node failures, node behaviour, network partitioning, network latency, corrupt or out-of-order inputs, and so on. According to [8], a consensus protocol should have the following three properties; namely consistency, availability, and fault tolerance. These properties are elaborated in Table 1.

There are two major fault-tolerance models within distributed systems: crash failure (or tolerance) and Byzantine failure [6, 8, 194]. The crash failure model deals with nodes that simply fail to respond due to some hardware or software failures. The byzantine failure model, on the other hand, deals with nodes that misbehave due to some software bugs or because of the nodes being compromised by an adversary. A Byzantine node, first identified and formalised by Leslie Lamport in [17], can behave maliciously by arbitrarily sending deceptive messages to others, which might affect the security of distributed systems. Hence, such nodes are mostly relevant in application with security implications.

To address these two failure models, there are two corresponding major types of consensus mechanisms: Crash-tolerant consensus and Byzantine consensus [194]. Next, we briefly discuss each of them, along with their associated properties.

1. **Crash-tolerant consensus:** Algorithms belonging to this class aim to guarantee the atomic broadcast (total order) of messages within the participating nodes in the presence of a certain number of node failures. These algorithms utilise the notion of views or epochs, which imply a certain duration of time or events. A leader is selected for each epoch who takes decisions regarding the atomic broadcast, and all other nodes comply with its decisions. In case a leader fails due to a crash failure, the protocol elects a new leader to function. The best known algorithms belonging to this class can continue to function if the following condition holds: $t < n/2$ where t is the number of faulty nodes and n is the total number of participating nodes [194]. Examples of some well-known crash-tolerant consensus protocol are: Paxos [18, 19], Viewstamped Replication [20], ZooKeeper [21], and Raft [22].

2. **Byzantine consensus:** This class of algorithms aims to reach consensus in the presence of certain nodes exhibiting Byzantine behaviour. Such Byzantine nodes are assumed to be under the control of an adversary and behave unpredictably with malicious intents. Similar to any crash-tolerant consensus protocol, these protocols also utilise the concept of a leader election in a view/epoch for atomic broadcast, and other honest nodes are assumed to follow the instructions from the leader. Consensus algorithms belonging to this class can achieve consensus in the presence of a certain number of Byzantine nodes and are aptly called Byzantine Fault Tolerant (BFT) consensus algorithms with Practical Byzantine Fault Tolerant (PBFT) being one of the most well-known algorithms in this category [23]. The tolerance level of PBFT is $f < n/3$, where f the number of Byzantine nodes and n denotes the number of total nodes participating in the network [194].

2.2. *Blockchain*

At the centre of the blockchain technology is the blockchain itself stored by the nodes of a P2P network [24]. A blockchain is a distributed ledger consisting of consecutive blocks chained together following a strict set of rules. Here, each block is created at a predefined interval, or after an event occurs, in a decentralised fashion by means of a consensus algorithm. Within each block, there are transactions by which a value is transferred in case of crypto-currencies or a data is stored for other blockchain systems.

Even though the terms blockchain and DLT (Distributed Ledger Technology) are used inter-changeably in the literature, there is a subtle difference between them which is worth highlighting. A blockchain is just an example of a particular type of ledgers, there are other types as well. When a ledger (including a blockchain) is distributed across a network, it can be regarded as a Distributed Ledger.

Smart-contract. Advancing from the original concept of blockchain where only data is distributed, a new breed of blockchain systems have emerged. Such new systems support the notion of distributed VM (Virtual Machine) [167] facilitating the deployment and execution of computer programs, known as *smart-contracts*, on top of the corresponding blockchain. A smart-contract is deployed and subsequently executed using transactions which ultimately change the states of the VM. These transactions and the changes states are recorded in the blockchain. The atomic broadcast and the corresponding consensus algorithm ensure that the replicated VMs evolve independently in a similar way as if every peer has access to the one single form of VM, thereby, emulating the notion of a single computer in the whole world. Being part of the blockchain makes smart contracts and their executions autonomous, immutable and irreversible, which are sought after properties

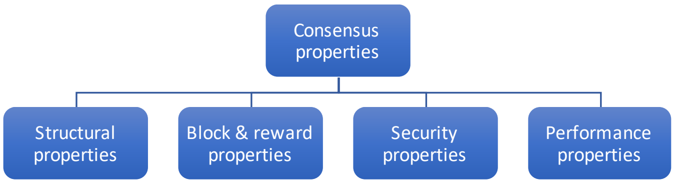

Figure 1: Taxonomy of consensus properties.

having a wide range of applications in different application domains. Ethereum is the first-ever platform which realised the notion of smart-contract supporting blockchain [72]).

Blockchain type. Depending on the application domains, different blockchain deployment strategies can be pursued. Based on these strategies, there are predominantly two types of blockchains, namely public and private blockchain, as discussed below:

- **Public blockchain**: A public blockchain, also known as the *Unpermissioned or permissionless Blockchain*, allows anyone to participate in the blockchain to create and validate blocks as well as to modify the chain state by storing and updating data through transactions among participating entities. This means that the blockchain state and its transactions, along with the data stored is transparent and accessible to everyone. This raises privacy concerns for particular scenarios where the privacy of such data needs to be preserved.

- **Private blockchain**: A private blockchain, also known as the *Permissioned Blockchain*, has a restrictive notion in comparison to its public counterpart in the sense that only authorised and trusted entities can participate in the activities within the blockchain. In this way, a private blockchain can keep the chain data only known to the trusted entities instead of the generic public, which might be desirable in some use-cases.

3. Consensus taxonomy & properties

With the introduction and advancement of different blockchain systems, a number of consensus algorithms have been introduced where each algorithm has different characteristics and serves different purposes. To compare these disparate groups of consensus algorithms, we need to define evaluation criteria. In this section, we present these evaluation criteria in the form of taxonomies of consensus properties. These properties have been collected from existing researches, such as [6, 194], and compiled as a taxonomy in this work.

The taxonomy is presented in Figure 1. According to this taxonomy, a consensus mechanism has four major groups of properties: *Structural*, *Block & reward* , *Security* and *Performance* properties. Each of these properties is briefly discussed below.

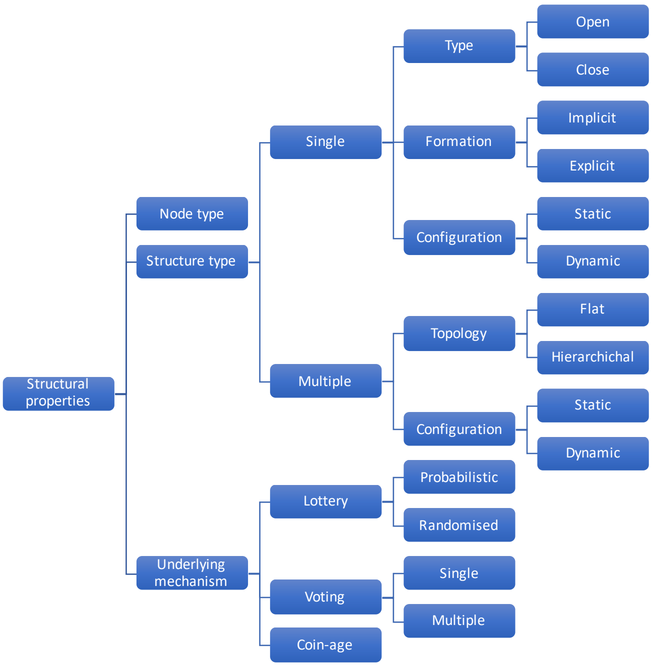

Figure 2: Taxonomy of structural properties.

3.1. Structural properties

Structural properties define how different nodes within a blockchain network are structured to participate in a consensus algorithm. These properties can be sub-divided into different categories as illustrated in Figure 2. We briefly describe each of these categories below.

- **Node types:** It refers to *different types of nodes* that a consensus algorithm is required to engage with to achieve its consensus. The types will depend on the specific consensus algorithm which will be presented in the subsequent section.

- **Structure type:** It refers to *the ways different nodes are structured* within the consensus algorithm using the concept of a committee. The committee itself can be of two types: single and multiple committees. Each of these committees is described below.

- **Underlying mechanism:** It refers to the *specific mechanism that a consensus algorithm deploys to select a particular node*. The mechanism can utilise lottery, the age of a particular coin (known as *coin-age*) or a voting mechanism. A lottery can utilise either a cryptography based probabilistic mechanism or other randomised mechanisms. In a voting mechanism, voting can be carried out either in a single or multiple rounds. On the other hand, the coin-age utilises a special property, which depends on how long a particular coin has been owned by its owner.

Next, we explore different types of voting committees for existing consensus algorithms.

Single committee. A single committee refers to a special group of nodes among the participating nodes which actively participate in the consensus process by producing blocks and extending the blockchain. Each single committee can have different properties which are briefly explored in the following.

- **Committee type:** A committee can be open or close. A committee is open if it is *open* to any participating nodes or closed if it is restricted to a specific group of nodes.

- **Committee formation:** A committee can be formed either implicitly or explicitly. An implicit formation does not require the participating nodes to follow any additional protocol rules to be in the committee, whereas an explicit formation requires a node to follow additional protocol steps to be a part of the committee.

- **Committee configuration:** A committee can be configured in a static or a dynamic fashion.

  - **Static:** In a static configuration, the members of the committee are pre-selected and fixed. No new members can join and participate in the consensus process.

  - **Dynamic:** In a dynamic configuration, the committee members are defined for a time-frame (known as epoch), after which new members are added, and old members are removed based on certain sets of criteria. In such a committee, nodes are selected using a voting mechanism where voting is carried either in a single or multiple rounds. Some consensus algorithms, however, do not specify any specific time-frame, and hence, members can join or leave any time at will. Nodes in such configurations are selected using a lottery mechanism which utilises either a cryptography based probabilistic mechanism or other randomised mechanisms.

Multiple committee. It has been observed that the time it takes to achieve consensus in a single committee tends to increase as the number of the member starts to increase [6], thereby reducing performance. To alleviate this problem, the concept of multiple committee has been introduced, where each committee consists of different validators (special nodes with the responsibilities to create blocks in a specific type of blockchain system, explained later) [6]. A multiple committee can have different properties. Next, we explore two properties.

- **Topology:** It refers to the way different committees are organised. For example, the topology can be *flat* to indicate that different committees are at the same level or can be *hierarchical* where the committees can be considered in multiple layered levels.

- **Committee configuration:** In addition, like a single committee, the multiple committees can be configured in a static or dynamic way.

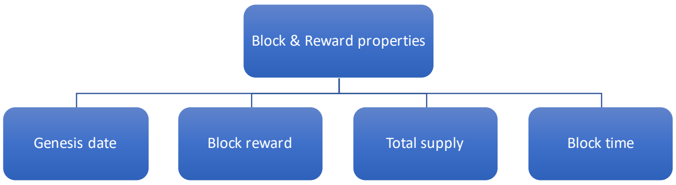

Figure 3: Taxonomy of block & reward properties.

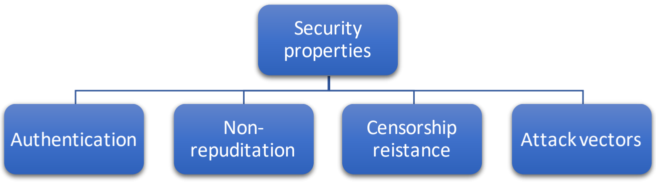

Figure 4: Taxonomy of consensus security properties.

3.2. Block & reward properties

Properties under this category can be utilised as quantitative metrics to differentiate different crypto-currencies. The properties are (Figure 3): genesis date, block reward, total supply, and block creation time. These properties do not necessarily characterise different consensus algorithms directly, however, most of them (except the genesis date) have a direct and indirect impact on how consensus is achieved in a particular crypto-currency based blockchain system. For example, block reward incentivises miners to act accordingly by solving a cryptographic puzzle, which is then ultimately used to achieve consensus. The properties are described below:

- **Genesis date** represents the timestamp when the very first block was created for a particular blockchain system.

- **Block reward** represents the reward a node receives for creating a new block.

- **Total supply** represents the total supply of a crypto-currency.

- **Block time** represents the average block creation time of a blockchain system.

3.3. Security properties

A consensus algorithm must satisfy a number of security properties as presented in as shown in Figure 4 and Figure 5. In Figure 4, different security properties are presented while in Figure 5 different attack vectors are presented. The outlined security properties and attack vectors are described below:

- **Authentication:** This implies if nodes participating in a consensus protocol need to be properly verified/authenticated.

- **Non-repudiation:** This signifies if a consensus protocol satisfies non-repudiation.

- **Censorship resistance:** This implies if the corresponding algorithm can withstand against any censorship resistance.

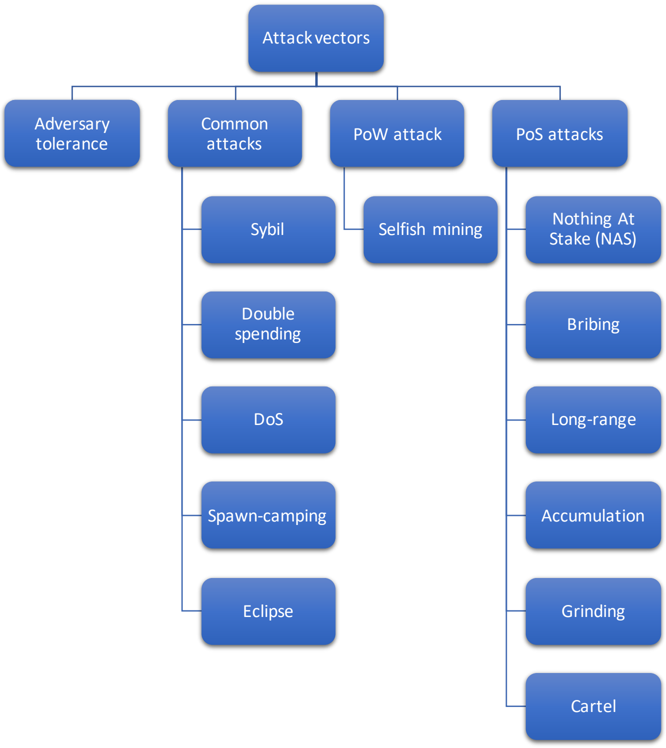

Figure 5: Taxonomy of attack vectors.

- **Attack vectors**: This property implies the attack vectors applicable to a consensus mechanism. There attack vectors are categorised in three groups: common, PoW attacks and PoS attacks. Here, we present the common attack vectors as these attacks are relevant to any consensus algorithm. On the other hand, PoW and PoS attacks, as presented in Figure 5, are applicable to that specific class of consensus algorithms respectively. Therefore, we will discuss them in the respective sections, when we explore such algorithms.
  - **Adversary tolerance:** This signifies the maximum byzantine nodes supported/tolerated by the respective protocol.
  - **Sybil:** In a Sybil attack [28], an attacker can duplicate his identity as required to achieve illicit advantages. Within a blockchain system, a sybil attack implicates the scenario when an adversary can create/-control as many nodes as required within the underlying P2P network to exert influence on the distributed consensus algorithm and to taint its outcome in her favour.
  - **Double spending:** A double spending attack is a critical attack in any financial system, including crypto-currencies. This attack implicates that a user of the respective crypto-currency can spend their crypto-currency twice [113].
  - **DoS (Denial of Service) & Spawn-camping:** An attacker can launch DoS attacks targeting a blockchain system. Particularly, adversaries can join forces to participate in a *spawn-camping* attack, in which they launch DoS attacks simultaneously over and over again to render the network useless for the corresponding blockchain system [112].
  - **Eclipse:** In an eclipse attack an adversary targets a blockchain (victim) node and aims to obscure its view of the blockchain [168]. Being unaware of how the blockchain is evolving, the victim can be targeted for other attacks such as double spending attacks and selfish mining (explained in subsequent sections) [168].

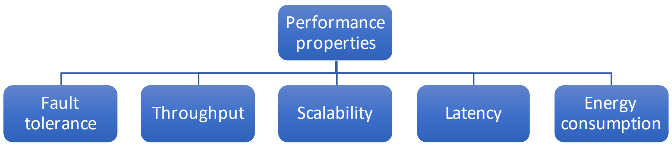

Figure 6: Consensus performance features or properties.

3.4. *Performance properties*

The properties belonging to this group can be utilised to measure the quantitative performance of a consensus protocol. A brief description of each property is presented below with its illustration in Figure 6.

- **Fault tolerance** signifies the maximum faulty nodes the respective consensus protocol can tolerate.

- **Throughput** implies the number of transactions the protocol can process in one second.

- **Scalability** refers to the ability to grow in size and functionalities without degrading the original system's performance [25].

- **Latency (Finality)** refers to "*the time it takes from when a transaction is proposed until consensus has been reached on it*" [6]. It is also known as finality.

- **Energy consumption** indicates if the algorithm (or the utilising system) consumes a significant amount of energy.

4. Proof of Work (PoW)

A Proof of Work (PoW) mechanism involves two different parties (nodes): prover (requestor) and verifier (provider). The prover performs a resource-intensive computational task intending to achieve a goal and presents the task to a verifier or a set of verifiers for validation that requires significantly less resources. The core idea is that this asymmetry, in terms of resource required, between the proof generation and validation acts intrinsically as a deterrent measure against any system abuse.

Within this aim, the idea of PoW was first presented by Dwork and Naor in their seminal article in 1993 [27]. They put forward the idea of using PoW to combat email spamming. According to their proposal, an email sender would require solving a resource-intensive mathematical puzzle and attach the solution within the email as a proof that the task has been performed. The email receiver would accept an email only if the solution can be successfully verified. Hashcash by Back et al. [30] is the earliest example to leverage a PoW mechanism in practical systems. Similar to the proposal of Dwork and Naor, Hashcash is also designed to combat spams. A similar concept has been adopted by many PoW blockchain systems where each PoW mechanism is bound to a threshold, known as the *difficulty parameter*. The prover would carry out the computational task in several rounds until a PoW is generated that matches the required threshold, and every single round is known as a single proof attempt.

PoW has been the most widely-used mechanism to achieve a distributed consensus among the participants regarding the block order and the chain state. In particular, a PoW mechanism in a blockchain serves two critical purposes:

- A deterrent mechanism against the *Sybil Attack*. In PoW, every mining node would require a significant monetary investment to engage in a resource-intensive PoW mechanism during the block creation process. To launch a Sybil attack, an attacker's monetary investment will be proportional to the number of Sybil identities, which might outweigh any advantage gained from launching a Sybil attack.

- The PoW mechanism is used as an input to a function which ultimately is used to achieve the required distributed consensus when a fork happens in a blockchain [29].

We differentiate between three major classes of PoW consensus mechanisms: *Compute-bound* PoW, *Memory-bound* PoW and *Chained* PoW. Each of these is explored in the following sections.

4.1. Compute-bound PoW

A *Compute-bound PoW*, also known as *CPU-bound PoW*, employs a CPU-intensive function that carries out the required computational task by leveraging the capabilities of the processing units (e.g., CPU/GPU), without relying on the main memory of the system. Nakamoto consensus is the compute-bound PoW consensus algorithm leveraged in Bitcoin. It is based on the approach of Hashcash, modified to be applied within the blockchain setting.

To understand the Nakamoto consensus, it is be useful to understand how Bitcoin functions. The Bitcoin network consists of two types of nodes within a P2P (Peer-to-Peer) network: miners (acting as provers) and general nodes (acting as verifiers). A general node is mostly used by users to transfer bitcoin in the network, whereas a miner node is a special node used for generating (mining) bitcoins.

Each node needs to download the Bitcoin software to connect to the network. Each user utilises a special software, called *wallet*, to create identities where an identity consists of a private/public key pair, and a bitcoin address is derived from the corresponding public key. A sender needs to know such an address of the receiver to transfer any bitcoin. Bitcoin is transferred between two entities using the notion of a transaction utilising the wallet software. This transaction is propagated to the network, which is collected by all miner nodes. Each miner node combines these transactions into a block and then engages in solving a cryptographic puzzle, with other miners, in several proof attempts, until the solution is found. In each of these proof attempts, a miner tries to generate a random number, applying a hash function over transactions and other data, which satisfies the required condition: the random number must be less than a target value called the *difficulty target*. When a miner successfully solves the puzzle, that miner is said to have generated a valid block which is then propagated in the network. In addition, before the maximum number of Bitcoins are reached, the Bitcoin protocol generates a certain amount of new Bitcoins for each new valid block and rewards the miner for its effort. Other miners validate this newly mined block and then add it to the blockchain. Each new block refers to the last block in the chain, which in turn refers to its previous block, and so on. The very first block in the chain, known as the *genesis block*, however, has no such reference.

The decentralised nature of this mining process might result in multiple valid blocks generated by different miners and resulting in multiple branches emerging from the same blockchain. This phenomenon in blockchain is known as *fork*. The fundamental goal of the corresponding consensus protocol is to resolve this fork so that only one branch remains and other branches are discarded. The Nakamoto consensus algorithm utilised in Bitcoin follows a simple rule: it lets the branches grow. As soon as one branch grows longer than the others (more specifically, the total cumulative computational effort of one branch exceeds the others), all miners select the longest branch (or the branch with the highest computational effort), discarding all other branches. Such a branch is known as the main branch and other branches are known as orphan branches. Only the miners in the main branch are entitled to receive their Bitcoin rewards. When a fork is resolved across the network, a distributed consensus emerges in the network.

The utilisation of CPU/GPU in the Nakamoto consensus facilitates the scenario in which the computation can be massively optimised for faster calculation using Application-specific Integrated Circuit (ASIC) rigs. This has drawn criticisms among the crypto-currency enthusiasts as general purpose computers cannot be utilised to participate in the mining process and hence, the mining process is mostly centralised among a group of mining nodes.

4.2. Memory-bound PoW

One major criticism of any compute-bound PoW is that it facilitates the utilisation of ASIC-based rigs for the mining purpose (see Section 4.1). To counteract this criticism, memory-bound PoWs have been proposed. A memory-bound PoW requires the algorithm to access the main memory several times. Thus it ultimately binds the performance of the algorithm within the limit of access latency and/or bandwidth as well as the size of memory. This restricts ASIC rigs based on a memory-bound PoW to have the manifold performance advantage over their CPU/GPU based counterparts. In addition, the profit margin of developing ASIC with memory and then building mining rigs with them is not viable as of now for these classes of PoWs. Because of these, memory-bound PoWs are advocated as a superior replacement for compute-bound PoWs in de-monopolising mining concentrations around some central mining nodes (see Section 4.5).

A memory-bound PoW algorithm has many variants such as: Cryptonight, Scrypt and its variants, Equihash, Ethash/Dagger and NeoScrypt. Next, we briefly describe each of these different variants.

1) Cryptonight. Cryptonight [26] utilises internally a memory-hard hash function called *Keccak* [31] and relies on a 2MB scratchpad residing on the memory of a computer. The scratchpad is extensively used to perform numerous read/write operations at pseudo-random addresses within that scratchpad. In the final step, the desired hash is generated by hashing the entire scratchpad.

Its reliance on a large scratchpad on the memory of a system makes it resistant towards FPGA and ASIC mining as the economic incentive to create FPGA, and ASIC mining hardware might be too low for the time being. As such, Cryptonight introduces the notion of so called *Egalitarian proof of work [26]* or proof of equality, which enables anyone to join in the mining process using any modern CPU and GPU.

2) Scrypt and its variants. Scrypt is a password based key deriving function (KDF) that is currently used in many crypto-currencies [32]. A KDF is primarily used to generate one or more secret values from another secret key and is widely used in password hashing. Previous key deriving functions such as DES-based UNIX Crypt-function, FreeBSD MD5 crypt, Public-Key Cryptography Standards#5 (PKCS#5), and PBKDF2 do not impose any specific hardware requirements. This enables any attacker to launch attacks against those functions using specific FPGA or ASIC enabled hardware, the so-called *custom hardware attacks* [33]. Scrypt has been designed to counteract this threat.

Towards this aim, one of the core characteristics of Scrypt is its reliance on the vast memory of a system, making it difficult for FPGA and ASIC enabled custom hardware. In the underneath, Scrypt utilises Salsa20/8 Core [34] as its internal hash function. A simplified version of Scrypt is used in different blockchain systems, which is much faster and easier to implement, and can be performed using any modern CPU and GPU, thereby enabling anyone to join in the mining process. However, the ever-increasing price of crypto-currencies has incentivised miners to produce custom ASIC hardware for some blockchain systems utilising Scrypt in recent times. An example of such hardware that can be used to mine different Scrypt crypto-currencies is Antminer L3+ [35].

To tackle this issue of exploiting ASIC for mining, several Scrypt variants have been proposed: Scrypt-N/Scrypt Jane/Scrypt Chacha and Scrypt-OG, each providing particular advantages over others. Scrypt-N and Scrypt Chacha rely on SHA256 and ChaCha [37] as their internal hash functions, respectively, whereas Scrypt Jane utilises a combination of different hash functions. All of them support progressive and tunable memory requirements, which can be adjusted after a certain period. This is to ensure that custom ASIC hardware is rendered obsolete once the memory requirement is changed. Finally, Scrypt-OG (Optimised for GPU) is optimised to be eight times less memory intensive than Scrypt [36].

NeoScrypt, an extension of Scrypt, is a key derivation function that aims to increase the security and performance on CPUs and GPUs while being strong ASIC resistant [104]. Internally it utilises a combination of algorithms such as Salsa 20/20 [34] and ChaCha 20/20 [37] along with Blake2s [55]. Its constructions impose larger temporal buffer requirements with a larger memory segment size. This makes it 1.25 times more memory intensive than Scrypt, thereby acting as a deterrent towards building ASICs for NeoScrypt.

3) Equihash. Equihash is one of the recent PoW algorithms that has been well received in the blockchain community [39]. It is a memory-bound PoW that requires to find a solution for the Generalised Birthday problem using Wagner's algorithm [40]. Equihash has been designed to decentralise the mining procedure itself, similar to other memory-bound approaches. However, so far, very few of such algorithms have succeeded. One of the crucial reasons for this is that their underlying time-memory complexity trade-off is largely constant. This means that reducing memory requirement in these algorithms have little effect on their corresponding time complexity.

Wagner's solution has a steep time-memory complexity trade-off, reducing memory increases time complexity substantially. This premise has been exploited by Equihash to ensure that mining is exclusively proportional to the amount of memory a miner has. Thus, it is more suitable for a general purpose computer than any ASIC-enabled hardware which can only have relatively small memory to make their production profitable for the mining process. Due to this reason, it has been claimed that Equihash can support ASIC resistance, at least for the foreseeable future.

4) Ethash (Dagger-Hashimoto)/Dagger. Ethash is a memory-bound PoW algorithm introduced for Ethereum with the goal to be ASIC-resistant for a long time [41]. It was previously known as Dagger-Hashimoto algorithm [42] because of its utilisation of two different algorithms: Dagger [43] and Hashimoto [43]. Dagger is one of the earliest proposed memory-bound PoW algorithms which utilises the Directed Acyclic Graph (DAG) for memory-hard puzzle solving. On the other hand, the Hashimoto algorithm relies on the delay incurred for reading data from memory as the limiting factor and is known as an I-O bound algorithm.

Ethash combines these two algorithms to be ASIC-resistant and functions as follows. Ethash utilises pseudo-random DAG recomputed during each epoch. Each epoch is determined by the time it takes to generate 30,000 blocks in approximately five days. During the DAG generation process, a seed is generated at first, which relies on the length of the chain. The seed is then used to compute a 16 MB pseudo-random cache. Then, each item of the DAG is generated by utilising a certain number of items from the pseudo-random cache. This entire process enables the DAG to grow linearly with the growth of the chain. Then, the latest block header and the current candidate nonce are hashed using Keccak (SHA-3) hash function, and the resultant hash is mixed and hashed several times with data from the DAG. The final hashed digest is compared to the difficulty target and accepted or discarded accordingly. This functionality of Ethash ensures that the optimisation is limited to the memory access delay, rendering ASIC hardware less useful. That is why it is thought to be suitable for commodity computing hardware with good powerful GPUs.

4.3. Chained PoW

A chained PoW utilises several hashing functions chained together in a series of consecutive steps to ensure ASIC resistance. In addition to this, it also aims to address one particular weakness of any compute-bound and memory-bound PoW algorithm: their reliance on a single hashing function. With the advent of quantum computing, the security of a respective hashing algorithm might be adversely affected, which might undermine the security of the corresponding blockchain system. If this happens, the old algorithm needs to be discarded, and a new quantum resistant hashing algorithm needs to be incorporated to the respective blockchain using a mechanism called *hard-fork*, a mechanism to enforce a major update in a blockchain system. This is a complex disruptive procedure that might harm any blockchain system. In such scenarios, a chained PoW algorithm would continue to function until all its hashing functions are compromised.

Several chained PoW algorithms are currently available, which are discussed next.

1) X11/X13/X15. X11 is a widely-used hashing algorithm in many blockchain systems. In X11, eleven hashing algorithms are consecutively executed one after another. The hashing algorithms are *blake, bmw, groestl, jh, keccak, skein, luffa, cubehash, shavite, simd, and echo*.

One advantage of X11 is that it is highly energy efficient: GPUs computing X11 algorithm requires approximately 30% less wattage and remains 30 – 50% cooler in comparison to Scrypt [38]. Even though X11 was designed to be ASIC resistant, the economic incentives have allowed the creation of ASIC hardware for X11 blockchain systems. X13 and X15 are its different variants in which 13 and 15 hashing functions are deployed respectively.

2) Quark. Quark PoW algorithm relies on six different hashing functions: BLAKE [55], Blue Midnight Wish [44], Grøstl [45, 101], JH [46], Keccak and Skein [47]. These functions are implemented in a mixed series with nine steps [100]. Within these nine steps, three functions are randomly applied in three steps depending on the value of a bit. The main motivations of mixing these six functions is to ensure ASIC resistance and security against one hashing algorithm being compromised. Even so, there are currently available hardware to enable Quark mining using GPU and ASIC [48].

3) Lyra2RE. Lyra2RE is a chained PoW which utilises five hash functions: BLAKE, Keccak, Lyra2, Skein, and Grøstl. It was designed to be CPU friendly, however, it was discovered in 2015 that the majority of the hashing power utilised for mining a Lyra2RE crypto-currency in its network was facilitated by a botnet stealing CPU cycles from a large number of infected computers. This motivated the developers to release Lyra2REv2, which utilises six hash functions, BLAKE, Keccak, CubeHash, Lyra2, Skein, and Blue Midnight Wish with GPU only PoW.

4) Magnificent 7. *Magnificent 7* (M7) is a chained PoW algorithm which utilises seven hash functions to generate the candidate hash during the mining process of Cryptonite blockchain system (not to be confused with the Cryptonight PoW algorithm) [102]. The utilised hash functions are SHA-256, SHA-512, Keccak, RIPEMD, HAVAL, Tiger and Whirlpool. Internally, the candidate block's header is sequentially hashed by the corresponding functions and then multiplied to generate the final hash, which is then compared against the difficulty threshold. Even though it a not memory-bound PoW, it has been claimed that the multiplication operation enables it to run on a general purpose CPU easily, however, makes it difficult to run on GPUs and ASICs [102]. Even so, there are is at least one GPU miner available for M7 [103].

4.4. PoW Crypto-currencies

Currently, there are many public blockchain systems generating different crypto-currencies that utilise different variants of PoW consensus algorithms. Table 2 shows the top five (if available) currencies for each variant of PoW consensus algorithms according to their market capitalisation as rated by CoinGecko [1] (a website which tracks different activities related to crypto-currencies) during the writing of this article. The table also presents their block and reward properties as presented in Figure 3. It is to be noted that information regarding the properties in Table 2 for these (and other subsequent) currencies has been collected by consulting their corresponding whitepapers, websites and introductory announcements on Reddit website [2].

In Table 2, we have used the notation “NA” to denote if the corresponding data for a crypto-currency is not found. For the *Block reward* property, the corresponding reward for each crypto-currency has been provided. For Cryptonite, though, “Dynamic” has been used to signify that the block reward is dynamically generated for each block. Similarly, for *Total supply*, the corresponding supply has been provided. The term “*Infinite supply*” has been used to indicate an infinite supply for a particular crypto-currency. Finally, we have block time for each currency has been in provided in minutes under the “Block time” category.

Apart from the side-by-side comparative analyse of different PoW crypto-currencies, Table 2 outlines an interesting phenomenon. Different crypto-currencies even belonging to a single variant of a consensus algorithm (e.g. Bitcoin, Syscoin,

[1] https://www.coingecko.com/

[2] https://www.reddit.com/

Peercoin, Counterparty and Emercoin, all belonging to Nakamoto variant) differ considerably with respect to their block reward, total supply and block time. This pattern is visible in other variants as well. The underlying motivation is difficult to guess and has not been analysed yet. These properties can be further analysed to identify different other phenomenons among different crypto-currencies. A few of such analyses have been presented in Section 7.

4.5. PoW Limitations

PoW (Nakamoto) consensus algorithm has been widely accoladed for its breakthrough in the distributed consensus paradigm, starting with Bitcoin. It has laid down the foundation for the subsequent advancement, which resulted in different PoW algorithms and blockchain systems supporting crypto-currencies as discussed in the earlier sections. Even so, there are some significant limitations. Next, we briefly discuss these limitations:

- **Energy consumption:** Each PoW algorithm needs to consume electricity to compute the hash. As the difficulty of the network starts to increase, so does the energy consumption. The amount of consumed energy is significant when calculated over the whole network consisting of ASIC/GPU mining rigs worldwide. According do Digiconomist [3], a website that tracks the electricity consumption of Bitcoin and Ethereum, the energy consumption of Bitcoin and Ethereum is around 40 TWh (Tera-Watt Hour) and 10 TWh, respectively [105]. To put this into perspective, it has been estimated that the electricity consumed by Bitcoin in a year could power up $6,770,506$ American households and is much more than what Czech Republic consumes in a year [105]. The utilisation of this tremendous amount of electricity has raised the question of the sustainability of PoW-based blockchain systems.

- **Mining centralisation:** With the ever-increasing difficulty rate, miners within a PoW-based blockchain system need to upgrade the capability of their ASIC/GPU mining rigs to increase their chance of creating a new block. Even so, it becomes increasingly difficult for a single miner to join in the mining process without a substantial investment in the mining rigs. This phenomenon implies that the *economies of scale* phenomenon strongly impacts PoW algorithms. In economic theory, the economies of scale is the advantage a producer can gain by increasing its output [106]. This happens because the producer can spread the cost of per-unit production over a larger number of goods, which increases the profit margin. This analogy also applies to PoW mining as explained next. A mining pool can be created where the mining resources of different miners are aggregated to increase the chance of creating a new block. Once a mining pool receives a reward for creating the next block, the reward is then proportionally divided among the participating miners. Unfortunately, this has led to the centralisation problems where the probability of block creations is increasingly being limited to a handful of miners. According to [108] (a website which tracks hashrate distribution of mining pools), only five mining pools control the 75% of hashrate of the whole network and there is a risk of collusion which could lead to a 51% attack.

- **Tragedy of commons:** Many PoW algorithms suffer an economic problem called the *Tragedy of the commons*. In economic theory, the tragedy of the commons occurs when each entity rushes to maximise its profit from a depleting resource without considering the well-being of all that share the same resource [107]. This situation occurs in a blockchain system if its reward mechanism is deflationary in nature with limited supply, e.g. Bitcoin. It has been argued when the reward of creating a new block in Bitcoin will reach nearly zero; the miners will have to rely on the transaction fees to cover their expenses solely. This might create an unhealthy competition among the miners to include as many transactions as possible, just to maximise one's profit. Consequently, the transaction fees will keep decreasing, which might lead to a situation where miners cannot make enough profit to continue the mining process. Eventually, more and more miners will leave the mining process, which might lead towards a 51% attack or other scenarios that could destabilise the Bitcoin network.

- **Absence of penalty & attack vectors:** All PoW algorithms (both compute and memory bound) are altruistic in the sense that they reward behaving miners, however, do not penalise a misbehaving one. One example is that a miner can collude with a group of miners (a phenomenon known as the *selfish mining* attack) to increase its profitability in an illegitimate way [109]. A miner can also engage in a Denial-of-Service attack by just not forwarding any transaction or block within the network or even the *spawn-camping* attack. A penalty mechanism would disincentivise any miner to engage in any such malicious misbehaviour.

4.6. Analysis

In this section, we summarise the properties of different PoW algorithms in Table 3, Table 4 and Table 5 utilising the taxonomies presented in Section 3. In these tables, the symbol "●" is utilised to indicate if the corresponding algorithm supports a certain property, whereas the symbol "○" is used to denote that the respective property is not supported. For other properties, explanatory texts have been used.

As presented in Table 3, different types of PoW algorithms share exactly similar characteristics. In these algorithms, they are mainly two types of nodes: clients and miners. Miners are responsible for creating a block using a randomised lottery mechanism. Conversely, clients are the nodes responsible for validating each block and creating transactions between different users. Committees in these algorithms represent the set

[3] https://digiconomist.net/

Table 2: Top crypto-currencies utilising different PoW consensus algorithms and the properties.

Currency	Variant	Genesis date (dd.mm.yyyy)	Block reward	Total supply	Block Time
Bitcoin/Bitcoin Cash/Bitcoin SV [49] [50] [51]	Nakamoto	03.01.2009	6.25	21.00 millions	10.0 minutes
Syscoin [52]	Nakamoto	16.08.2014	80.05	888.00 millions	1.0 minute
Peercoin [53]	Nakamoto	19.08.2012	55.17	2000.00 millions	10.0 minutes
Counterparty [54]	Nakamoto	01.02.2014	NA	2.60 millions	NA
Emercoin [56]	Nakamoto	11.12.2013	5020.00	41.00 millions	10.0 minutes
Monero [57]	Cryptonight	18.04.2014	4.87	18.40 millions	2.0 minutes
Bytecoin [58]	Cryptonight	04.07.2012	666.76	184460.00 millions	2.0 minutes
AEON [59]	Cryptonight	06.06.2014	5.48	18.40 millions	4.0 minutes
Boolberry [60]	Cryptonight	17.05.2014	4.85	18.50 millions	2.0 minutes
Karbowanec [61]	Cryptonight	30.05.2016.	8.83	Infinite supply	4.0 minutes
Litecoin [62]	Scrypt	13.10.2011	25.00	84.00 millions	2.5 minutes
Verge [63]	Scrypt	15.02.2016	730.00	16500.00 millions	0.5 minutes
Bitmark [64]	Scrypt	13.07.2014	20.00	27.58 millions	2.0 minutes
Dogecoin [65]	Scrypt	06.12.2013	10000.00	Infinite supply	NA
GameCredits [66]	Scrypt	01.06.2015	12.50	84.00 millions	1.5 minutes
Zcash [67]	Equihash	28.10.2016	10.00	21.00 millions	2.5 minutes
Bitcoin Gold [68]	Equihash	24.10.2017	12.50	21.00 millions	10.0 minutes
Komodo [69]	Equihash	15.10.2016	3.00	200.00 millions	1.0 minute
Zclassic [70]	Equihash	6.11.2016	12.50	21.00 millions	2.5 minutes
ZenCash [71]	Equihash	30.05.2017	7.50	21.00 millions	2.5 minutes
Ethereum [72]	Ethash	30.07.2015	2.00	infinite supply	10-20 seconds
Ethereum Classic [73]	Ethash	30.07.2015	4.00	3880.00 millions	10-20 seconds
Ubiq [74]	Ethash	28.01.2017	6.00	NA	88 seconds
Shift [75]	Ethash	01.08.2015	1.00	Infinite supply	27 seconds
Expanse [76]	Ethash	13.09.2015	4.00	31.40 millions	1.0 minute
Red Pulse [77]	NeoScrypt	17.10.2017	NA	1360.00 millions	NA
Feathercoin [78]	NeoScrypt	16.04.2013	40.00	336.00 millions	1.0 minute
GoByte [79]	NeoScrypt	17.11.2017	3.71	31800.00 millions	2.5 minutes
UFO Coin [80]	NeoScrypt	03.01.2014	625	4000.00 millions	1.5 minutes
Innova [81]	NeoScrypt	19.10.2017	2.64	1.29 millions	2.0 minutes
Dash [82]	X11	19.01.2014	1.55	22.00 millions	2.5 minutes
Regalcoin [88]	X11	28.09.2017	NA	7.20 millions	NA
Memetic [89]	X11	05.03.2016	NA	NA	NA
ExclusiveCoin [90]	X11	12.06.2016	NA	NA	NA
Creditbit [91]	X11	02.11.2015	NA	100.00 millions	1.0 minute
Stratis [83]	X13	09.08.2016	NA	NA	NA
Cloakcoin [84]	X13	03.06.2014	496.00	4.5 millions	1.0 minute
Stealthcoin [85]	X13	04.07.2014	NA	20.70 millions	1.0 minute
DeepOnion [86]	X13	13.07.2017	4.00	18.90 millions	4.0 minutes
HTMLcoin [87]	X15	12.09.2014	NA	90000 millions	1.0 minutes
Quark [47]	Quark	21.07.2013	1.00	247.00 millions	0.5 seconds
PIVX [92]	Quark	NA	5.00	NA	1.0 minute
MonetaryUnit [93]	Quark	26.07.2014	18.00	1000000.00 millions	0.67 minute
ALQO [94]	Quark	30.10.2017	.003	NA	1.0 minute
Bitcloud [95]	Quark	15.08.2017	22.50	200.00 millions	6.5 minutes
Vertcoin [96]	Lyra2RE	10.01.2014	25.00	84.00 millions	2.5 minutes
Monacoin [97]	Lyra2RE	01.01.2014	25.00	105.00 millions	1.5 minutes
Crypto [98]	Lyra2RE	30.03.2015	NA	65.80 millions	0.5 minute
Cryptonite [99]	Magnificent 7	28.07.2014	Dynamic	1840.00 millions	1.0 minute

of miners, exhibiting the property of a single open committee structure, which is formed implicitly in a dynamic fashion, indicating any miner can join or leave whenever they wish.

As per Table 4, none of the algorithms requires any node to be authenticated to participate in the algorithm. All of them have strong support for non-repudiation in the form of a digital

Table 3: Structural properties of PoW consensus algorithms.

Node type	Single committee			Mechanism
	Type	Formation	Configuration	
Clients & Miners	Open	Implicit	Dynamic	Lottery, Randomised

Table 4: Security properties of PoW consensus algorithms.

Authn	Non repud.	Censorship resistance	Attack Vectors						
			Adversary tolerance	Sybil	Double spend	DoS	Spawn camp.	Eclipse	Self. mining
○	●	High	$2f + 1$	●	●	●	●	○	○

Table 5: Performance properties of PoW consensus algorithms.

Fault tolerance	Throughput	Scalability	Latency	Energy consumption
$2f + 1$	Low	Low	Medium-High	High

signature as part of every single transaction. These algorithms have a high level of censorship resistance, which means that it will be difficult for any regulatory agency to impose any censorship on these algorithms. As for the attack vector, each PoW algorithm requires every miner node to invest substantially in mining hardware to participate in these consensus algorithms. This feature, thus, acts as a deterrent against any Sybil or DoS (as well as spawn-camping) attack in any PoW algorithm. The adversary tolerance is based on the assumption that PoW suffers from 51% attacks, and thus, adversary nodes need to have less than 50% of the total hashing power of the network. Moreover, the consensus algorithm ensures resiliency against any double spending attacks. However, a PoW algorithm is not resilient against an eclipse or a selfish mining attack.

According to Table 5, these algorithms have low throughput, and unfortunately, do not scale properly. Furthermore, most of the algorithms require a considerable time to reach finality, and their energy consumption is considerably high, as explained in Section 4.5. The fault tolerance in these algorithms is $2f + 1$ like any BFT algorithm, implying they can achieve consensus as long as more than 50% of nodes function correctly.

5. Proof of Stake

To counteract the limitations of any PoW algorithm, another type of consensus algorithm, called Proof of Stake (PoS), has been proposed. The earliest proposal of a PoS algorithm can be found on the *bitcointalk* forum in 2011 [110]. Soon after, several projects started experimenting with the idea. Peercoin [53], released in 2012, was the first blockchain crypto-currency to utilise the PoS consensus algorithm.

The core idea of PoS evolves around the concept that the nodes who would like to participate in the block creation process must prove that they own a certain number of coins at first. Besides, they must lock a certain amount of its currencies, called *stake*, into an escrow account in order to participate in the block creation process. The stake acts as a guarantee that it will behave as per the protocol rules. The node that escrows its stake in this manner is known as the stakeholder, leader, forger, or minter in the PoS terminology. The minter can lose the stake, in case it misbehaves.

In essence, when a stakeholder escrows its stake, it implicitly becomes a member of an exclusive group. Only a member of this exclusive group can participate in the block creation process. In case the stakeholder gets the chance to create a new block, the stakeholder will be rewarded in specific ways.

5.1. Different Aspects of PoS

In this section, we explore different aspects of a PoS consensus algorithm.

Advantages: It has been argued that the incentive method, coupled with any punitive mechanism, can provide a similar security level of any PoW algorithm. Moreover, PoS can offer several other advantages. Next, we discuss a few benefits of a PoS mechanism [112].

- **Energy Efficiency:** A PoS algorithm does not require any node to solve a resource-intensive cryptographic puzzle. Consequently, such an algorithm is extremely energy efficient compared to their PoW counterpart. Therefore, a crypto-currency leveraging any PoS algorithm is likely to be more sustainable in the long run.

- **Mitigation of Centralisation:** A PoS algorithm is less impacted by the economies of scale phenomenon. Since it does not require to build up a mining rig to solve any resource-intensive cryptographic puzzle, there is no way to maximise gains by increasing any output. Therefore, it is less susceptible to the centralisation problem created by the mining pool.

- **Explicit Economic Security:** A carefully designed penalty scheme in a PoS algorithm can deter any misbehaving attack, including spawn-camping. Anyone engaging in such attacks will lose their stake and might be banned from any block creation process in the future, depending on the protocol. This eventually can strengthen the security of the system.

Bootstrapping: One of the major barriers in a PoS algorithm is how to generate the initial coins (crypto-currencies) and fairly distribute them among the stakeholders so that they can be used as stakes. We term this barrier as the *bootstrapping* problem. There are two approaches to address the bootstrapping problem:

- Pre-mining: A set of coins is pre-mined, sold before the launch of the system in an IPO (Initial Public Offering), or ICO (Initial Coin Offering).
- PoW-PoS transition: The system starts with a PoW system to fairly distribute the coins among the stakeholders. Then, it slowly transitions towards a PoS system.

Reward process: Another important aspect is the rewarding process to incentivise the stakeholder to take part in the minting process. Unlike any PoW, where a miner is rewarded with new coins for creating a valid block, there is no reward for creating a valid block. Instead, to incentivise a minter, two types of reward mechanisms are available within a PoS algorithm:

- Transaction Fee: The minter can collect fees from the transactions included within the minted block.
- Interest rate: A lower interest rate is configured, which allows the currency to inflate over time. This interest is paid to the minter as a reward for creating a valid block.

Selection process: A crucial factor in any PoS algorithm is how to select the stakeholder who can mint the next block. In a PoW algorithm, a miner is selected based on who can find the resource-intensive desired hash. Since PoS does not rely on hind such a hash as the mechanism to find the next block, there must be a mechanism to select the next stakeholder.

PoS Types: Currently, there are three different approaches to Proof of Stake: Chained, BFT and Delegated. Next, we explore these approaches.

5.2. Chained PoS

The general idea of a chained PoS is to deploy a combination of PoW and PoS algorithms chained together to achieve any consensus. Because of this, there can be two types of blocks, PoW and PoS blocks, within the same blockchain system. To accomplish this, the corresponding algorithm relies on different approaches to select/assign a particular miner for creating a PoW block or select a set of validators for creating a PoS block in different epochs or after a certain number of blocks created. In general, a chain based PoS can employ any of the following three different approaches to select a miner/stakeholder:

- *Randomised PoW Mining:* A miner who can solve the corresponding cryptographic PoW puzzle is selected randomly.
- *Randomised Stakeholder Selection:* A randomised PoS utilises a probabilistic formula that takes into account the staked currencies and other parameters to select the next stakeholder. The other parameters ensure that a stakeholder is not selected based only on the number of their staked coins and acts as a pseudo-random seed for the probabilistic formula.
- *Coin-age based selection.* A coin-age is defined as the holding period of a coin by its owner. For example, if an owner receives a coin from a sender and holds it for five days then the coin-age of the coin can be defined as five coin-days. Formally,

  $$coin - age = coin * holdingperiod$$

  A stakeholder can be selected using the staked coins of the stakeholders and their corresponding coin-age.

Next, we present two examples of a chained PoS algorithm to illustrate how this approach works in practice.

1) PeerCoin (PPCoin). Peercoin is the first crypto-currency (blockchain system) to formalise the notion of PoS by utilising a hybrid PoW-PoS protocol [131] which utilises *coin-age* for a PoS algorithm while minimising the disadvantages associated with a PoW algorithm.

Peercoin protocol recognises two different kinds of blocks: PoW blocks and PoS blocks, within the same blockchain. These blocks are created by two separate entities: miners and minters. Miners are responsible for creating PoW blocks, similar to Bitcoin, whereas minters are responsible for creating PoS blocks. Irrespective of the last block type, the next block either can be a PoW block or a PoS block [132]. Miners compete with other miners to find a valid PoW block that matches the PoW difficulty target and minters compete among themselves to find a valid PoS block that matches the PoS difficulty target (similar to a PoW algorithm but requires much less computation). As soon as any PoW or PoS block is found, it is broadcast to the network, and other nodes validate it.

Within a PoS block, a minter utilises their holding coins as a stake, and the minter is rewarded proportionally to the coin-age of the staked coins. Once a PoS block is added to the chain, the coin-age of the staked coins is reset to zero. This indicates that all the stacked coins are consumed and cannot be used over and over again to create a PoS block within a short period of time. The block reward for a PoW block in Peercoin decreases and will cease to be significant after a certain period of time. It is currently used for the coin generation and distribution purpose and will be completely phased out in the future [162].Peercoin is highly regarded for formalising the first alternative mechanism to PoW, however, it suffers from all the attack vectors of PoS, as presented in Section 5.5.

2) CASPER FFG. Casper the Friendly Finality Gadget (CFFG), also known as Ethereum 2.0, is a PoW-PoS hybrid consensus algorithm proposed to replace the Ethereum's PoW consensus algorithm [137]. In fact, CFFG provides an intermediate PoS overlay on top of its current PoW algorithm so that Ethereum can be transformed to a pure PoS protocol called Casper the Friendly Ghost (CTFG) described below (Section 5.3).

The PoS layer requires the participation of validators. Any node can become a validator by depositing some Ethereum's native crypto-currency called *Ether* to a designated smart-contract, which acts as a security bond. The network itself will mostly consist of PoW miners who will mine blocks according to its current PoW algorithm. However, the finalisation/check-pointing of blocks will be carried out by PoS validators. The check-pointing/finalisation is the process to ensure that the chain becomes irreversible up to a certain block and thus, short and low range attacks (particular types of PoS only attacks presented in Section 5.5) as well as the 51% attack cannot be launched beyond the check-pointing block.

The check-pointing occurs every 50 blocks, and this interval of 50 blocks is called an *epoch* [115]. The finalisation process requires two rounds of voting in two successive epochs. The process is as follows. In an epoch, the validators vote on a certain checkpoint c (a block). A super-majority (denoted as +2/3) occurs when more than 2/3 of the validators vote for the checkpoint c. In such a case, the checkpoint is regarded as *justified*. If in the next epoch, (+2/3) of the validators vote on the next checkpoint c' (a block which is a child of the block belonging to c), c' is considered justified whereas c is considered finalised. A checkpoint created in this manner for each epoch is assumed to create a checkpoint tree where c' is a direct child of c. The process can be summarised in the following way: +2/3 Vote c → Justify c → +2/3 Vote c' → Finalise c and Justify c'

Once a checkpoint is finalised, the validators are paid proportionally to the number of ethers deposited. If a fork occurs, indicating that a validator has deviated from the protocol, the validator is penalised by destroying the validator's deposit. CGGF ensures that the block finalisation occurs quickly, however, how it performs in reality is yet to be seen as it has not been implemented yet. The protocol is mostly secure against all PoS attacks except the cartel formation attack (a particular type of PoS only attack presented in Section 5.5).

5.3. BFT PoS

BFT PoS is a multi-round PoS algorithm. In the first step, a set of validators are pseudo-randomly selected to propose a block. However, the consensus regarding committing this block to the chain depends on the +2/3 quorum of the super-majority among the validators on several rounds. It inherits any BFT consensus properties, and it tolerates up to 1/3 of byzantine behaviour among the nodes. Next, we describe four notable BFT PoS algorithms: Tendermint, CTFG, Ouroboros, Harmony and Algorand.

1) Tendermint. Tendermint is the first to showcase how a BFT consensus can be achieved within the PoS setting of blockchain systems [134, 135, 136]. It consists of two major components: a consensus engine known as Tendermint Core and its underlying application interface, called the *Application BlockChain Interface* (ABCI). The Tendermint core is responsible for deploying the consensus algorithm, whereas the ABCI is utilised to deploy any blockchain application using any programming language.

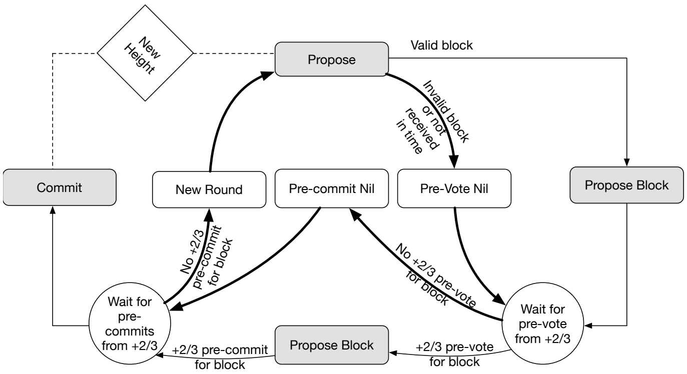
New Height
Propose
Valid block
Invalid block or not received in time
Commit
New Round
Pre-commit Nil
Pre-Vote Nil
Propose Block
No +2/3 pre-commit for block
No +2/3 pre-vote for block
Wait for pre-commits from +2/3
+2/3 pre-commit for block
Propose Block
+2/3 pre-vote for block
Wait for pre-vote from +2/3

Figure 7: Tendermint consensus steps.

Tendermint is a round-based algorithm that relies on a set of validators. The consensus algorithm consists of three steps (propose, pre-vote, and pre-commit) in each round bound by a timer equally divided among the three steps. These steps signify the transition of states in each validator. Figure 7 illustrates the state transition diagram for each validator. At the beginning of each round, an entity called a *proposer* is chosen from the validator set to propose a new block at the latest height. The proposer is selected using a deterministic round-robin algorithm by utilising the voting power of the validators. The voting power, on the other hand, is proportional to the security deposit of the validators. The proposed block needs to go through a two-stage voting mechanism before it is committed to the blockchain.

When a validator receives the proposed block, it validates the block at first, and if okay, it pre-votes for the proposed block. If the block is not received within the *propose* timer or the block is invalid, the validator submits a special vote called *Prevote nil*. Then, the validator waits for the *pre-vote* interval to receive pre-votes from the super-majority (denoted as +2/3) of the validators. A +2/3 pre-votes signifies that the super-majority validators have voted for the proposed block, implying their confidence on the proposed block and is denoted as a *Polka* in Tendermint terminology. At this stage, the validator pre-commits the block. If the validator does not receive enough pre-votes for the proposed block, it submits another special vote called *Pre-commit nil*. Then, the validator waits for the *pre-commit* time-period to receive +2/3 pre-commits from the super-majority of the validators. Once received, it commits the block to the blockchain. Otherwise, the next round is initiated where a new proposer is selected, and the steps are repeated.

To ensure the safety guarantee of the algorithm, Tendermint is also coupled with locking rules. Once a validator pre-commits a block after a polka is achieved, it must lock itself onto that block. Then, it must obey the following two rules:

- it must pre-vote for the same block in the next round for the same blockchain height,

- the unlocking is possible only when a newer block receives a polka in a later round for the same blockchain height.

With these rules, Tendermint guarantees that the consensus is secure when less than one-third of validators exhibit byzantine behaviour, meaning conflicting blocks will never be committed at the same blockchain height. Hence, no fork will occur. Since Tendermint favours safety over availability, it has one particular weakness. It requires 100% uptime of its +2/3 (super-majority) validators. If more than one-third (+1/3) are validators are off-line or partitioned, the system will stop functioning [134]. In such cases, out-of-protocol steps are required to tackle this situation. Tendermint is mostly a consensus plugin which can be retro-fit to other blockchain systems. Consequently, there is no reward/punitive mechanism. However, a consensus mechanism can be easily introduced in the application layer via the ABCI. Together with carefully designed reward and punishment mechanisms, all PoS attacks can be effectively handled.

2) Casper the Friendly Ghost (CTFG). CTFG is a pure BFT PoS algorithm that aims to transform Ethereum to a PoS-only blockchain system [138]. As described above, CFFG is geared towards a gentle transition from a PoW to a PoS model for Ethereum, where CTFG will take control of the consensus mechanism.

CTFG is based upon a rigorous formal model called Correction by Construction (CBC) that utilises the GHOST (Greedy Heaviest-Observed Subtree) primitive as its consensus rule during fork [139]. The idea is that the CTFG protocol will be partially specified at the initial stage along with a set of desired properties. Then, the rest of the protocol is dynamically derived to satisfy the desired properties - hence the name correction by construction. This is in contrast to the traditional approach for designing a protocol where a protocol is fully defined at first, and then it is tested to check if it satisfies the desired properties [112].

To achieve this, CTFG introduces a safety oracle, acting as an ideal adversary, which raises exceptions when a fault occurs and also approximates the probability of any future failure. Based on this, the oracle can dynamically fine-tune the protocol as required to evolve it towards its completion.

Like CFFG, CTFG also requires a set of bonded validators that will bond ethers as a security deposit in a smart contract. However, unlike any other PoS mechanisms, the validators will bet on the block, which has the highest probability to be included in the main chain according to their own perspective. If that particular block is included in the main chain, the validators receive rewards for voting in favour of the block. Otherwise, the validators receive certain penalties.

Like any PoW algorithm, CTFG favours availability over consistency. This means that blocks are not finalised instantly, like Tendermint. Instead, as the chain grows and more blocks are added, a previous block is considered implicitly final. A major advantage of CTFG over Tendermint is that it can accommodate dynamic validators. This is because the finality condition in Tendermint requires that its block interval is short, which in turn demands a relatively small number of pre-determined validators. Since CTFG does not rely on any instant finality, it can theoretically accommodate a higher number of dynamic validators.

CTFG is currently the most comprehensive proposal which addresses all PoS attack vectors. However, it is to be noted that this is just a proposal at the current stage. Therefore, its performance in real settings is yet to be analysed.

3) Ouroboros. Ouroboros is a provably secure PoS algorithm [146, 147] utilised in the Cardano platform [148]. Cardano is regarded as a third-generation blockchain system supporting smart-contracts without relying on any PoW consensus algorithm.

In Ouroboros, only a stakeholder can participate in the block minting process. A stakeholder is any node that holds the underlying crypto-currency of the Cardano platform called **Ada**. Ouroboros is based on the concept of *epoch*, a predefined time period. Each epoch consists of several slots. A stakeholder is elected for each slot to create a single block. The selected stakeholder is called a slot leader and is elected by a set of *electors*. An elector is a specific type of stakeholders which has a certain amount of Ada in its disposal.

In each epoch, the electors select the set of stakeholders for the next epoch using an algorithm called *Follow the Satoshi* (FTS). The FTS algorithm relies on a random seed to introduce a certain amount of randomness in the election process. A share of the random seed is individually generated by all electors who participate in a multiparty computation protocol. Once the protocol is executed, all electors posses the random seed, constructed with all of their shares. The FTS algorithm utilises the random seed to select a coin for a particular slot. The owner of the coin is then elected as the slot leader. Intuitively, the more coins a stakeholder possesses, the higher is its probability of being selected as the slot leader.

Ouroboros is expected to provide a transaction fee based reward to incentivise stakeholders to participate in the minting process. However, the details are in the process of being finalised. It has been mathematically proven to be secure against almost all PoS attack vectors except the cartel formation [146]. Nevertheless, how it will perform once deployed is yet to be seen.

4) Harmony. Harmony is a new breed of sharded blockchain which utilises a BFT PoS called FBFT (Fast BFT) [140, 141]. Sharding [186] is a phenomenon well-known in the database domain in which data within a database are horizontally partitioned into different segments, or shards, in order to provide fault tolerance and improve performance [187]. A similar concept is being investigated in blockchain domain where a blockchain is partitioned in different shards [143]. The main motivation is that sharding a blockchain in this way will facilitate scalability and storage optimisation in blockchain systems [143, 186].

In Harmony, nodes, which act as validators, are assigned to different shards using a random function which provides provable security as there is no way for a malicious node to influence which shard they will belong to. Harmony utilises the notion of an *epoch*, a pre-determined time interval during which the set of validators remains fixed for a shard. Within a shard, a leader is selected at each consensus round for proposing block and other validators validates the proposal. The selection of leaders is dependant on the number of stakes of each validator. That is, a validator with larger stakes has a higher probability of being selected as a leader.

The consensus mechanism for a single round in Harmony has three stages: announce, prepare and commit. In the announce phase, the leader constructs a new block and then broadcast its header and contents to all validators. Each validator validates the header and creates a BLS (Boneh–Lynn–Shacham) signature [144], which is a multi-signature scheme, using the header data and returns it back to the leader, thereby concluding the announce phase. In the prepare phase, the leader waits for valid signatures from $2f+1$ validators (including the leader) and creates an aggregated BLS mulit-signature. The leader then broadcasts this multi-signature along with a list of signing validators, thereby concluding the prepare step.

Finally, in the commit phase, each validator validates the BLS mulit-signature from $2f+1$ validators, verifies the transactions of the proposed block and creates another BLS signature and sends it back to the leader. The leader waits for valid signatures from $2f + 1$ validators (including the leader and can be a different setp of validators from the prepare phase) and creates another aggregated BLS mulit-signature. All these multi-signatures and the list of signers are then combined wihin the proposed block which is then committed to the validators. Each validator then checks the committed block and adds it to the blockchain.

Harmony consensus algorithm has been designed in such a way that it offers better performance than PBFT algorithm with respect to communication complexity. The sharding mechanism ensures scalability and storage optimisation. Furthermore, the algorithm tackles a number of attacks as explored in Section 5.6.

5) Algorand. Algorand uses a pure proof-of-stake (PPoS) consensus protocol built on an extension of Byzantine agreement, denoted $BA*$ [207]. This means the system can resist malicious users as long as a super majority of the stake is in non-malicious hands. The users' influence on the choice of a new block is proportional to their stake in the system (number of algos). Users are randomly and secretly selected to both propose blocks and vote on block proposals. All online users have the chance to be selected to propose and to vote. The likelihood that a user will be chosen is directly proportional to their stake.

In this consensus, accounts propose new blocks to the network and then the committee votes on proposals and filters down to one, it is called *soft vote*. Then, a different committee votes to certify the block. Each node receives a certificate for the block and writes it to the ledger. A new round is initiated and the process starts over with new block proposers and voters.

This protocol can tolerate an arbitrary number of malicious users as long as honest users hold a super majority of the total stake in the system. It uses two protection mechanisms to safeguard the protocols from adversaries.

Firstly, the adversary is unaware of the node which will certify the block as the node is selected secretly and individually. Secondly, when the adversary knows the chosen node, it is too late to influence the outcome. Because in that time, the selected node has already performed its responsibility. For the next round, a new set of nodes (users) will be selected again privately and individually.

In terms of the security of the consensus, it requires $3f + 1$ honest users. It will ensure "sufficiently honest" committee for Byzantine Agreements (BA). To prevent an adversary from targeting committee members, BA selects committee members in a private and non-interactive way. This means that every user in the system can independently determine if they are chosen to be on the committee, by computing a variable random function (VRF) of their private key and public information from the blockchain. To achieve liveness, Algorand makes a "strong synchrony" assumption that most honest users (e.g., 95%) can send messages that will be received by most other honest users (e.g.,95%) within a known time bound. This assumption allows the adversary to control the network of a few honest users, but does not allow the adversary to manipulate the network at a large scale, and does not allow network partitions.

5.4. DPoS

Delegated Proof of Stake (or DPoS in short) is a form of consensus algorithm in which reputation scores or other mechanisms are used to select the set of validators [145]. Even though it has the name Proof of Stake associated with it, it is quite different from other PoS algorithms in terms of how validators are selected within a system and the number of validators deployed.

In DPoS, users of the network vote to select a group of delegates (or witnesses) who are responsible for creating blocks. Users utilise reputations scores or other mechanisms to choose their delegates. Delegates are the only entities who can propose new blocks. For each round, a leader is selected from the set of delegates who can propose a block. How such a leader is chosen depends on the respective system. The leader gets rewards for creating a new block and is penalised and de-listed from the set of validators if it is found to misbehave.

The delegates themselves compete with each other to get included in the validator list. In such, each validator might offer different levels of incentives for the voters who vote for it. For example, if a delegate is selected to propose a block, it might distribute a certain fraction of its reward among the users who have voted for it. Since the number of validators is small, the consensus finality can be fast.

There are several mechanisms deployed by different blockchain systems which deploy DPoS. Next, we present a few prominent approaches of some well-known DPoS based blockchain systems.

1) EOS. EOS is the first and the most widely known DPoS blockchain and smart-contract platform [149, 150]. With the promise of greater scalability and higher transactions per second than Ethereum, EOS raised 4 billion USD in the highest ever ICO event to date [151]. Initially, EOS crypto-currency was created on the Ethereum platform, and then migrated to its own blockchain network. Blocks in EOS are produced in rounds where each round consists of 21 blocks [152]. At the beginning of each round, 21 validators, also known as *Block Producers* (BPs), are selected with votes from EOS token (currency) holders. The number of times a particular BP is selected to produce a block is proportional to the total votes received from the token holders. Next, each of selected BPs gets a chance to create a block in a pseduo-random fashion within that particular round. Once a BP produces a block, other BPs must validate the block. A block is confirmed only when (+2/3) majority of the BPs reach the consensus regarding the validity of the block. Then, the block and the associated transactions are regarded as confirmed or final, so no fork can happen. Currently, an EOS block is created in 0.5s.

2) Tron. Tron is another popular DPoS based smart-contract supported blockchain platform, very similar to Ethereum and EOS in functionality. [153]. Its consensus mechanism utilises 27 validators, known as Super Representatives (SRs) [154]. The SRs are selected in every six hours with votes by TRX holders who must freeze a certain amount of TRX to vote for an SR. The deposits amount is returned after three days once the voting is cast [155]. A block in Tron is created in every 3s and the corresponding SR receives a reward of 32 TRX. Another important feature of Tron is that there is no in-built inflation mechanism in the protocol, which implies that the total supply will remain constant throughout its lifespan.

3) Tezos. Tezos is, like EOS and Tron, a smart-contract platform which utilises a variant of DPoS consensus algorithm [156]. With a block reward of 16 XTZ (Tezos crypto-currency) and block creation time of 60s, Tezos does not require any predefined number of stakeholders (or *Bakers* as defined in Tezos) [157]. This differentiates Tezos from other DPoS platforms. Instead, the consensus mechanism utilises a dynamic range of stakeholders where anyone holding a substantial amount of XTZ can be a stakeholder. This limits general users to participate in the consensus mechanism. To rectify this problem, Tezos provides a mechanism by which anyone can delegate their XTZ to someone so that it can accumulate the required number of XTZ to be a baker. In return, the baker would return a certain proportion of their received block reward to the delegating party. Tezos started with an initial supply of 765 Million XTZ tokens. It relies on an annual inflation of 5.51% and the inflated currencies are used to reward the bakers.

4) Lisk. Lisk is a unique DPoS blockchain platform which can accommodate and operate with multiple blockchains, known as *sidechains* along with a central blockchain called *mainchain* [158]. Each sidechain can be deployed and maintained by a particular application provider, which needs to be synced with the mainchain as per the Lisk's protocol rule. In this way, different applications can leverage different sidechains simultaneously without burdening off the mainchain. Even though the responsibility of maintaining a sidechain relies on the particular application provider, the mainchain must be maintained with the Lisk DPoS consensus protocol, which utilises 101 delegates [159] which can only produce a block. These delegates are selected using votes from Lisk currency (denoted with LSK) owners, where each holder has 101 votes. The weight of each vote is proportional to the amount of LSK owned by the respective owner. The selection of delegates happens before a round, where each round consists of 101 block generation cycle. Thus, in a round, each delegate is randomly selected to create a block. It has a block creation time of 10 seconds and block reward of 5 LSK. Started with an initial supply of 100 million LSK, Lisk has an annual inflation of 5.65%.

5) Ark. Ark is yet another DPoS based blockchain platform [160]. It utilises 51 delegates to create 51 blocks in each round [161]. With a block creation time of 8s, each round lasts for 408s. Each delegate receives 2 ARK (the native currency of the ARK platform) for creating a block. It had an initial supply of 125 million. With an annual inflation of 5.55%, the supply was around 150 million (as of June 2020). Like other DPoS blockchains, the delegates in Ark are also selected with votes by Ark currency owner, where the weight of each is proportional to the amount of ARK owned by the voter.

5.5. Limitations of PoS

Even though the variants of different PoS algorithms offer several significant advantages, there are still a few disadvantages in these classes of algorithms. We explore these disadvantages below.

- **Collusion:** If the number of validators is not large enough, it might be easier to launch a 51% attack on the corresponding consensus algorithm by colluding with other validators.
- **Wealth effect:** The sole reliance on coin-wealth in a consensus algorithm or for the selection of validators creates an environment where people with a large portion of coins can exert greater influences.

In addition to these disadvantages, there have been a few other attack vectors identified for the PoS algorithms:

- **Nothing-at-stake (NAS) attack [114]:** During a blockchain fork, an attacker might attempt to add its newly created block in all forked branches to increase their probability to add their block as the valid block. Such scenario is unlikely to occur in any PoW algorithm. This is because a miner would need to share their resources in order to mine at different branches. This would eventually decrease their chance of finding a new block because of the resources shared in multiple branches. Since it does not cost anything significant for a minter in a PoS algorithm to add blocks in multiple parallel branches, the attacker is motivated to do so. Applying a penalty for such misbehaviour could effectively tackle this problem.
- **Bribing (short-range, SR) attack [114, 133]:** In this attack, an attacker tries to double spend by creating a fork. An example of this attack would be as follows. The attacker pays to a seller to buy a good. The seller waits for a

certain number of blocks (e.g., six blocks) before the good is delivered to the attacker. Once delivered, the attacker forks the main chain at the block (e.g., six blocks back, which is relatively short and hence the name) in which the payment was made. Then, the attacker bribes other minters to mint on top of the forked branch. As long as the bribed amount is lower than the price of the delivered good, it is always profitable for the attacker. The colluding minter has nothing to lose if it is coupled with the nothing-at-stake attack on their part but can gain from the bribery. Again, it can be tackled by introducing a penalty mechanism for all misbehaving parties.

- **Long-range (LR) attack [114]:** In this attack, an attacker attempts to build an alternative blockchain starting from the earliest blocks after colluding with the majority of the stakeholders. The motivation might be similar to double spending or related issues that provide advantages to the attacker and the colluded stakeholders. As explained above, the colluded stakeholder has nothing to lose if it can be coupled with the nothing-at-stake attacks. Check-pointing is one of the methods by which it can be tackled. The check-pointing codifies a certain length of the blockchain to make it non-forkable up to that point, and thereby undermining the attack.
- **Coin-age accumulation (CAC) attack [114, 133]:** The PoS algorithms that rely on the uncapped coin-age parameter are susceptible to this attack. In this attack, the attacker waits for their coins to accumulate enough coin-age to exploit the algorithm for launching double spends by initiating a fork. This attack can be tackled by introducing a cap on the coin-age which minimises the attack vector.
- **Pre-computing (PreCom) attack [114, 111]:** A pre-computing attack, also known as Stake-grinding attack, would allow an attacker to increase the probability of generating subsequent blocks based on the information of the current block. If there is not enough randomness in the PoS algorithm, the attacker can attempt to pre-compute subsequent blocks by fine-tuning the current block's information. For a particular set of information (e.g., a set of transactions), if the attacker finds that the probability of minting a few subsequent blocks is less than desired, the attacker can update the set of transactions to increase their probability of determining the next few blocks. It can be effectively tackled by introducing a secure source of randomness in the algorithm.
- **Cartel formation (CAF) attack [115]:** In economic theory, an oligopoly market is dominated by a small set of entities having a greater influence or wealth than other entity. They can collude with one another by forming a cartel to control price or reduce competition. It has been argued that ”*Blockchain architecture is mechanism design for oligopolistic markets.*” [116] which affects both PoW and PoS algorithms. Such a cartel can launch 51% attacks on the PoS algorithm or exploit the stakes to monopolise the PoS algorithm.

5.6. *Analysis*

In this section, we summarise the properties of different PoS algorithms utilising the taxonomies and PoS attack vectors in Table 6, Table 7, Table 8 and Table 9. We have used the symbol “●” to denote a certain property is satisfied by the respective protocol, whereas the symbol “○” denotes that the protocol does not fulfil the respective property. We use the symbol “⊠” to indicate that a certain property is not applicable for the respective protocol while “?” indicates that no information has been found for that particular feature. For other properties, explanatory texts have been used as well.

From Table 6, chained algorithms as well as FBFT and BA* are based on a multiple committee utilising a flat topology with a dynamic configuration. Among them, chained algorithms use a probabilistic lottery to select a minter whereas FBFT and BA* use voting mechanisms. Conversely, other PoS algorithms, except Tendermint, are based on a single committee having an open type and explicit formation with a dynamic configuration and mostly rely on voting mechanisms. Tendermint uses a closed committee with a static configuration.

As per Table 7, no algorithm, except Tendermint, requires any node to be authenticated to participate in the algorithm. All of them have strong support for non-repudiation in the form of a digital signature as part of every single transaction. These algorithms have a high level of censorship resistance, as do all PoW algorithms. As for the attack vector, each PoS algorithm requires every minter node to invest substantially to participate in this algorithm. This feature, thus, acts as a deterrent against any Sybil or DoS (as well as a spawn-camping) attack in any PoS algorithm. The adversary tolerance for chained systems can be calculated using this formula: $min(2f + 1, 3f + 1) = 3f + 1$. This is because a chained algorithm utilises both PoW and PoS algorithms and thus needs to consider the adversary tolerance for both of them. We consider the minimum of these two ($3f + 1$). The supported adversary tolerance for other algorithms is $3f + 1$ except BFT Ouroboros whose adversary tolerance is $2f + 1$. Furthermore, all PoS algorithms are resilient against any double spending attack. However, like any PoW algorithm, each PoS algorithm is susceptible to an eclipse attack. The Peer-coin's chained algorithm is susceptible to the selfish mining attack as it utilises PoW, however, this attack is not applicable to other PoS algorithms.

Table 8 outlines a comparison of additional attack vectors. CTFG, Tentermint, Ouroboros and Algorand have mitigation mechanisms against these attack vectors. However, Casper FFG, and any DPoS algorithms cannot successfully defend against the cartel formation attack. Peercoin, on the other hand, has a mechanism against this cartel formation attack; unfortunately, it suffers from all other attack vectors.

According to Table 9, all BFT, FBFT, BA* and DPoS algorithms have considerably high throughput, low latency, and high scalability. Their energy consumption is negligible. However, the chained algorithms have a comparatively lower throughput, lower scalability, and higher latency with respect to their BFT and DPoS counterparts. The fault tolerance of chained and BFT algorithms is $2f + 1$ like any BFT algorithm, implying they can achieve consensus as long as more than 50% of nodes function properly. However, DPoS algorithms require a $3f + 1$ fault tolerance.

While analysing these tables, it is important to keep in mind the following points:

- The higher the adversary and fault tolerance the better a consensus system is, assuming other factors are same. For example, in between $2f + 1$ and $3f + 1$ for both of these tolerance factors, $2f + 1$ is better as this implies that a particular system will continue to function until more than 50% of nodes behave maliciously (adversary tolerance) or become faulty (fault tolerance). Conversely, a $3f+1$ model implies that a particular system will continue to function until more than 33% of nodes behave maliciously or become faulty.

- When comparing DPoS with another class of algorithms having the same tolerance level ($3f + 1$), it is important to understand that the number of validators (f) in DPoS will be significantly lower than those used in other algorithms e.g. CFFG or CTFG. This has security implications as it might be easier to theoretically collude with the low number of validators used in DPoS algorithms in comparison to the hundreds of validators in CFFG or CTFG.

- It might look that the analysed PoS consensus algorithms do not have that much of differences among each other when evaluated against different properties of the taxonomy. However, a careful investigation would reveal that, in between any PoS algorithms, there are differences at least in one property among the consensus algorithms.

Finally, comparisons of the reviewed BFT and DPoS crypto-currencies are presented in Table 10 and Table 11 respectively. In addition to the block properties, the DPoS table (Table 11) also a validator nos column which presents the number of validators for each DPoS system.

6. Beyond PoW and PoS Consensus Algorithms

Some consensus algorithms take a different approach in which they do not solely rely on any PoW or PoS mechanism. Instead, they use an approach in which a PoW/PoS mechanism is combined with another approach. We consider such algorithms as hybrid algorithms that are presented in Section 6.1. Other approaches adopt a more drastic approach in which they do not leverage any type PoW/PoS algorithm whatsoever. Such algorithms are tagged as *N-POS/POW* (to symbolise Non-PoS/PoW) algorithms and discussed in Section 6.2.

6.1. Hybrid Consensus

In this section, we outline several hybrid consensus algorithms.

1) Proof of Research (PoR). The Proof-of-research is a hybrid consensus algorithm leveraged by Gridcoin [117, 118], a blockchain system that enables any to earn Gridcoin crypto-currency by sharing their personal computing resources with the BOINC (Berkeley Open Infrastructure for Network Computing) project [119]. BOINC is a grid computing platform widely used by the scientific community to harness idle personal computing resources for research in different domains.

The PoR consensus merges the concept of PoS with the Proof-of-BOINC and is mostly dominated by the PoS mechanism with Proof-of-BOINC acting as a reward mechanism for the contributors, known as *Researchers* in Gridcoin terminology. The PoS mechanism is similar to the traditional PoS algorithm. A minter, known as *Investor* in Gridcoin terminology, needs to own a certain amount of Gridcoin and participate in the minting process. A researcher installs the BOINC software and registers a project with his email address from **BOINC**. Then, a unique cross project identifier (CPID) is assigned to download the work share. Once the computation is completed, the researcher returns the result with a credit recommendation for the completed workload. The recommendation is compared with that of another researcher, and the minimum credit is rewarded. This workload credit data is stored in each block's header, which enables a contributor to prove his contribution to the BOINC project. Consequently, the researcher is rewarded with the corresponding amount of Gridcoin.

2) Slimcoin's Proof-of-Burn (PoB). The Proof-of-Burn is a consensus algorithm proposed as an alternative to PoW [120]. In PoW, miners need to invest in building a mining rig in order to participate in the mining process. In PoB, miners need to burn their coins in order to participate in the mining process. Burning coins mean that sending coins to an address without any private key and thus never usable. Hence, it is analogous to the investment for building a mining rig. The amount of burning has a positive correlation with the possibility of being selected for mining the next block. This is similar to a PoW system, where the miners increasingly invest in modern equipment to maintain the hash power.

Slimcoin is a crypto-currency which utilises the idea of PoB in combination with PoW and PoS [121, 122], thus creating a hybrid consensus mechanism. This idea is similar to the chained PoS algorithm of Peercoin as presented in Section 5.2 with additional PoB mechanism sandwiched in between PoW and PoS algorithms. The PoW is used to generate the initial coin supply using the mechanism of Bitcoin. When the system has a sufficient supply of coins, it plans to switch to a hybrid of PoW and PoS mechanism similar to Peercoin where PoB will be used to select the miner. As this happens, the minters will need to burn their accumulated coins in order to be eligible to participate in the PoS minting process. Since the PoB algorithm is mostly used for minter selection, it has hardly any effect on the system's security. Hence, its security and other properties are mostly similar to that of Peercoin.

3) Proof of Stake-Velocity (PoSV). One of the major limitations of a coin-age based PoS is that there is no incentive for the minters to be online for the staking process, as the coin-age increases linearly over time even without participating in the staking process. The lack of participants may facilitate attacks at a certain time.

Table 6: Comparing structural properties of PoS Consensus Algorithms.

Consensus /System	Node type	Single committee			Multiple committee		Mechanism
		Type	Formation	Configuration	Topology	Configuration	
Chained (PeerCoin)	Clients, Miners & Minters	⊠	⊠	⊠	Flat	Dynamic	Probabilistic lottery
Chained (CFFG)	Clients, Miners & Validators	⊠	⊠	⊠	Flat	Dynamic	Probabilistic lottery
BFT (Tendermint)	Clients & Validators	Close	Explicit	Static	⊠	⊠	Voting
BFT (CTFG)	Clients & Validators	Open	Explicit	Dynamic	⊠	⊠	?
BFT (Ouroboros)	Clients, Electors & Stakeholders	Open	Explicit	Dynamic	⊠	⊠	Voting
FBFT (Harmony)	Clients & Validators	⊠	⊠	⊠	Flat	Dynamic	Voting
BA* (Algorand)	Clients & Electors	⊠	⊠	⊠	Flat	Dynamic	Voting
DPoS	Clients & Validators	Open	Explicit	Dynamic	⊠	⊠	Voting

To counteract this problem, a crypto-currency called Reddcoin proposed a novel hybrid algorithm called Proof of Stake-Velocity (PoSV) [123, 125]. The central idea in PoSV is the idea of a mechanism called the *velocity of stakes* coupled with any traditional PoS algorithm. Conceptually, the velocity of stake mirrors the notion of the velocity of money, a terminology from Economics implying the frequency of money flow within the society [126]. Indeed, the velocity of stakes evolves around the idea of increasing the flow of stakes during the PoS consensus mechanism [124]. This (the flow of stakes) can be achieved if the minters are encouraged to actively participate in the consensus mechanism by staking their crypto-currency, instead of holding their coins offline. This process in a way will also increase the overall security of the system and counteract the lack of participant issue in PoS.

To facilitate this, PoSV introduces a non-linear coin-ageing function in which the coin-age of a particular coin is gained much faster in the first few days and weeks than the gain in later weeks. For example, it has been estimated that minters who stake their coins every two weeks or less, can earn up to 20% more than people who do not participate in the staking process [124]. Such incentives encourage the minters to increase the velocity of stakes in the whole network.

6.2. *N-POS/POW*

In this section, we present several prominent N-PoS/PoW approaches.

1) Proof-of-Cooperation (PoC). The Proof-of-Cooperation is a consensus algorithm introduced by the FairCoin blockchain system [127, 128]. This consensus algorithm relies on several special nodes known as Certified Validating Nodes (CVNs) which are responsible for creating blocks. Each CVN node is authenticated by its corresponding Faircoin identifier as well as trusted following a set of community-based rules and technical requirements [128]. The community rules state that a candidate node willing to be a CVN must participate in Faircoin community activities by performing some tasks. Examples of these tasks are running a local node or contributing to any technical or management issue related to Faircoin, which must be confirmed by at least two active members of the community. Besides, the candidate node must follow a set of technical requirements such as 24/7 network availability and special cryptographic hardware used for signature generation.

Blocks in Faircoin are created by one of the CVNs in a round-robin fashion in every three minutes of the epoch. To create a new block, a CVN needs to be selected using a deterministic voting mechanism individually carried out by every single CVN in the network. The steps of this mechanism are:

- Each CVN finds the CVN, which has created a block furthest in the chain by traversing backward through the chain and validates its technical requirements.
- Then, each node creates a data set consisting of the hash of the last block, the ID of the selected CVN for the next block, and its CVN ID, which is then signed by the specified cryptographic hardware. The created dataset, along with the signature, is then propagated through the network.
- The selected CVN receives this dataset along with its signature from multiple CVNs and verifies each signature. As soon as the selected CVN finds that more than 50% CVNs have selected it to be the next block creator, it can be cer-

Table 7: Comparing security properties of PoS Consensus Algorithms.

Consensus /System	**Authn**	**Non repud.**	**Censorship resistance**	**Attack Vectors**						
				Adversary tolerance	**Sybil**	**Double spend**	**DoS**	**Spawn camp.**	**Eclipse**	**Selfish mining**
Chained (PeerCoin)	○	●	High	$3f + 1$	●	●	●	●	○	○
Chained (CFFG)	○	●	High	$3f + 1$	●	●	●	●	○	⊠
BFT (Tendermint)	● (In closed type), ○ (In open type)	●	High	$3f + 1$	●	●	●	●	○	⊠
BFT (CTFG)	○	●	High	$3f + 1$	●	●	●	●	○	⊠
BFT (Ouroboros)	○	●	High	$2f + 1$	●	●	●	●	○	⊠
FBFT (Harmony)	○	●	High	$3f + 1$	●	●	●	●	○	⊠
BA* (Algorand)	○	●	High	$3f + 1$	●	●	●	●	○	⊠
DPoS	○	●	High	$3f + 1$	●	●	●	●	○	⊠

Table 8: Comparison of additional attack vectors protection among PoS Consensus Algorithms

Consensus\System	**Nothing-at-Stake**	**Bribing**	**Long-range**	**Coin-age**	**Pre-computing**	**Cartel formation**
Chained (PeerCoin)	○	○	○	○	○	●
Chained (CFFG)	●	●	●	●	●	○
BFT (Tendermint)	●	●	●	●	●	●
BFT (CTFG)	●	●	●	●	●	●
BFT (Ouroboros)	●	●	●	●	●	●
FBFT (Harmony)	●	●	●	●	●	⊠
BA* (Algorand)	●	●	●	●	●	●
DPoS	●	●	●	●	●	○

Table 9: Comparing performance properties of PoS Consensus Algorithms.

Consensus\System	**Fault tolerance**	**Throughput**	**Scalability**	**Latency**	**Energy consumption**
Chained (PeerCoin, CFFG)	$2f + 1$	Medium	Medium	Medium	Medium
BFT (Tendermint, CTFG, Ouroboros)	$2f + 1$	High	High	Low	Low
FBFT (Harmony)	$2f + 1$	High	High	Low	Low
BA* (Algorand)	$3f + 1$	High	High	Low	Low
DPoS	$3f + 1$	High	High	Low	Low

tain that its turn is next at the end of the current epoch, i.e., three minutes.

- The selected CVN adds all pending transactions into a new block, along with all the received signatures before it propagates the block in the network.

- Upon receiving the block, other CVNs verify the block by

Table 10: Comparison of BFT Currencies

Currency	Genesis date (dd.mm.yyyy)	Total supply	Inflation	Block reward	Block Time
Cardano [148]	27.09.2017	45 billion	7%	Dynamically calculated during each epoch	20s
Harmony [140]	02.06.2019	12.6 billion	6%	Dynamically calculated	2s
Algorand [207]	11.06.2019	10 billion	3.7%	Dynamically calculated during each epoch	5s

Table 11: Comparison of DPoS Currencies

Currency	Genesis date (dd.mm.yyyy)	Initial supply	Inflation	Block reward	Block Time	Validator nos
EOS [149]	01.07.2017	1 billion	5%	1% of inflated currency divided among 21 validators	0.5s	21
Tron [153]	28.08.2017	99 billions	⊠	32 TRX	3s	27
Tezos [156]	30.06.2018	765 millions	5.51%	16 XTZ	60s	Not pre-defined
Lisk [158]	24.05.2016	100 millions	5.65%	5 LSK	10s	101
Ark [160]	21.03.2017	125 millions	5.55%	2 ARK	8s	51

checking the if the CVN who created the block is actually the one selected as the block creator and validate all signatures and transactions. If the verification is successful, the block is added to the blockchain, and the same mechanism continues.

2) Proof of Importance (PoI). PoS exhibits an unfair advantage: the rich get richer by coin hoarding, and hence, everyone holds onto their coins instead of spending them. To solve this unfairness, NEM blockchain system has introduced “Proof of Importance (PoI)” [129]. It functions similarly to PoS: nodes need to ’vest’ an amount of currency to be eligible for creating blocks and are selected for creating a block roughly in proportion to some score. In Proof-of-stake, this ’score’ is one’s total vested amount, but in PoI, this score includes more variables. All the nodes that have more than 10000 XEM (the corresponding crypto-currency of NEM) are theoretically given equal positive importance and with 9B XEM coins there can be a maximum 900K such nodes. However, the actual number of nodes and their significance may vary with time and their amount of transaction in NEM.

In NEM, an account’s importance depends only on the net transfers of XEMs from that account. To be considered for the importance estimation at a certain block height, h , a node must have transferred at least 100 XEMs during the last 30 days or 43, 200 blocks. The “importance score” addresses the hoarding criticism of PoS. This is because hoarding will result in a lower score while spreading XEM around will increase it. Being a merchant pays better than being a hoarder.

6.3. Analysis

In this section, we summarise the properties of different Hybrid and N-Pow/PoS algorithms utilising the taxonomies in Table 12, Table 13, Table 14 and Table 15. Like before, “⊠” signifies that the corresponding property is not applicable for the respective consensus algorithm, “?” indicates that the information the property has not been found, a “●” is used to indicate an algorithm satisfies a particular property and “○” is used to imply the reverse (not satisfied). For other properties, explanatory texts have been used as well.

Table 12 presents the comparison of structural properties for the corresponding consensus algorithms. Among them, PoR and PoB depend on a multiple committee formation with a flat topology and dynamic configuration. Conversely, PoSV and PoI use a single open committee with a dynamic configuration and probabilistic lottery as their underlying mechanism. PoC has an implicit, open, and dynamic single committee, which relies on a voting mechanism.

All these algorithms have an adversary tolerance of $3f + 1$ with the support of non-repudiation, Sybil protection, DoS resistance, and high censorship resistance as reported in Table 13. Entities in PoB, PoSV, and PoI do not require authentication, while PoC entities must be authenticated, and researchers in PoR need to be authenticated. However, other entities in PoW can remain non-authenticated. All of them except PoC and PoI have $3f+1$ adversary tolerance because of their usage of PoS algorithms. We have not found anything regarding the adversary tolerance for PoC and PoI. Furthermore, algorithms under this category are resilient against any double spending attack. However, like any PoW or PoS algorithm, these algorithms are susceptible to an eclipse at-tack. Because of the utilisation of PoW, PoB is susceptible to the selfish mining attack, however, this attack is not applicable to other algorithms in this category.

Table 14 presents the comparison of some additional attack vectors for the Hybrid algorithms. As evident from the table, since these algorithms utilise PoS as one of their consensus algorithms, they suffer from the similar limitations of any PoS algorithm. For example, none of them has any guard against most of these additional attack vectors. The only exception is PoB which is because of its use of Peercoin like functionality, can resist the cartel formation attack.

The comparison of the performance properties for these algorithms is presented in Table 15. All of them have $2f + 1$ fault tolerance except PoC and PoI, as no information regarding their fault tolerance has been found. In terms of Scalability, Latency and Energy, every algorithm except PoB exhibits some similar characteristics: consume low energy, and have low latency, meaning they reach finality quickly. Because it relies on PoW, PoB has low scalability, low latency, and also consume medium energy. Finally, PoR, PoSV and PoI have high throughput, whereas PoC has a low throughput and PoB has a medium throughput.

Finally, a comparison of the selected Hybrid and N-PoW/PoS crypto-currencies is presented in Table 16.

7. Discussion

As per our analysis in different sections, it is clear that PoW consensus algorithms have major limitations, specifically power consumption and scalability. Many regard PoS, and it is variant DPoS, to be the most suitable alternatives. To understand the applications of these algorithms in public blockchain systems, we have analysed the top 100 crypto-currencies, as reported on CoinMarketCap [4] during the writing of this article.

Consensus algorithms in top 100 crypto-currencies. In the first analysis, we have calculated the number of consensus algorithms used in these (top 100) crypto-currencies. The distribution of consensus algorithms is presented in Figure 8. According to our analysis, PoW is still the most widely used (57%) consensus algorithms to date, whereas DPoS is the second most with 11%, and PoS is the third most used algorithm with 6%. All other consensus algorithms represent the remaining 26%. This means that, even though many consider that PoS and DPoS are the best alternatives to PoW, their adoption is still far behind PoW. As seen in Figure 8, there are few other consensus algorithms, not discussed previously, which are utilised by many top 100 crypto-currencies. Next, we present a very brief discussion of these consensus algorithms:

- **XRP Ledger Protocol:** XRP ledger protocol [196, 197] is a BFT consensus protocol utilised by the Ripple platform which is targeted for real-time gross settlement (RTGS) within a payment network for cross-border payments. The major differences with other BFT consensus algorithms is that XRP protocol deploys a small set of trusted validators, whereas, other BFT algorithms deploy a large number of validators with the assumption that some of validators might be malicious. Because of its usage of a small number of validators, it achieves a low latency which is essential for payment settlement.

- **Stellar Consensus Protocol (SCP):** SCP is a consensus algorithm employed by the Stellar blockchain platform [199]. Stellar, like Ripple, is also a decentralised cross-border payment system. It utilises a modified Byzantine consensus algorithm called Federated Byzantine Agreement [198]. Anyone can be a validator within the network. The protocol ensures that its latency is low and trust is flexible.

- **Proof of Authority:** This is a consensus algorithm employed by VeChainThor blockchain [200, 201]. In this algorithm, only known and verified entities can be a validator, known as *Authority Masternodes*. The motivation is that being verified and known, a validator will behave honestly so as to ensure their reputation is not jeopardised. From the set of validators, each validator is randomly chosen with equal probability and receives reward for producing a block. The algorithm ensures a high throughput and low latency.

- **Proof of Believability (PoB):** PoB is a BFT consensus algorithm supporting sharding developed by IOST blockchain platform [202, 203]. In this algorithm, a validator is chosen with a 'believability' data, a reputation score, which itself is calculated using a number of criteria such as the number of corresponding crypto-currency owned by the validator, positive reviews from other validators and previous transaction history. The use of PoB algorithm enables IOST platform to achieve high throughput and low latency.

- **Proof of Activity:** This algorithm is a hybrid algorithm consisting of PoW and PoS [204]. A miner/validator at first utilises a PoW algorithm to solve a puzzle for a template block which consists of some meta-information. This information is then utilised to choose a set of validators by a PoS algorithm. A leader is chosen from these validators based on their stake which proposes a valid block, consisting of transactions. This block is signed by all validators and the block reward is then split between the leader and other signing validators. This algorithm is predominantly used in Decred blockchain platform [205].

- **Loop Fault Tolerance (LFT):** LFT is a BFT consensus algorithm developed by ICON blockchain platform [206**?**]. To create a block, a leader is selected from a set of validators using a voting mechanism. The creates block is then verified by the other validators. The algorithm has been designed in such a way which ensures a fast throughput.

[4] https://coinmarketcap.com/

Table 12: Comparing structural properties of Hybrid and N-POS/POW Consensus Algorithms.

Consensus /System	Node type	Single committee			Multiple committee		Mechanism
		Type	Formation	Configuration	Topology	Configuration	
PoR	Clients (Researchers) & Minters	☒	☒	☒	Flat	Dynamic	Probabilistic lottery
PoB	Clients, Miners & Minters	☒	☒	☒	Flat	Dynamic	Probabilistic lottery
PoSV	Clients & Minters	Open	Implicit	Dynamic	☒	☒	Probabilistic lottery
PoC	Clients & CVNs	Open	Explicit	Dynamic	☒	☒	Voting
PoI	Clients & transaction partners	Open	Implicit	Dynamic	☒	☒	Probabilistic lottery

Table 13: Comparing security properties of Hybrid and N-POS/POW Consensus Algorithms.

Consensus	Authn	Non repud.	Censorship resistance	Attack Vectors						
				Adversary tolerance	Sybil	Double spend	DoS	Spawn camp.	Eclipse	Selfish mining
PoR	●	●	High	$3f+1$	●	●	●	●	○	☒
PoB	○	●	High	$3f+1$	●	●	●	●	○	○
PoSV	○	●	High	$3f+1$	●	●	●	●	○	☒
PoC	●	●	High	?	●	●	●	●	○	☒
PoI	○	●	High	?	●	●	●	●	○	☒

Table 14: Comparison of additional attack vectors protection for Hybrid and N-POS/POW Consensus Algorithms

Consensus /System	Nothing-at-Stake	Bribing	Long-range	Coin-age	Pre-computing	Cartel formation
PoR	○	○	○	○	○	○
PoB	○	○	○	○	○	●
PoSV	○	○	○	○	○	○

Table 15: Comparing performance properties of Consensus Algorithms of Hybrid and N-POS/POW.

Consensus	Fault tolerance	Throughput	Scalability	Latency	Energy
PoR	$2f+1$	High	Medium	Low	Low
PoB	$2f+1$	Medium	Low	Medium	Medium
PoSV	$2f+1$	High	Medium	Low	Low
PoC	?	LoW (10.6 TPS [130])	Medium	Low	LoW
PoI	?	High	Medium	Low	Low

- **Proof of Service (PoServ):** This is a hybrid consensus algorithm which combines PoS and PoW mechanisms and automatically checks and guarantees valid service providers [214]. It uses a match making algorithm to link a resource request to a resource provider based on their description. This algorithm introduces the idea about punishing cheating service providers or even kicking them out from the system.

- **Proof of Retrievability (PoR):** Using PoR [208], a

Table 16: Hybrid & Non-PoW/PoS currencies

Currency	Genesis date (dd.mm.yyyy)	Block reward	Total supply	Consensus	Block Time
Gridcoin [117]	24 Mar 2016	Minting	42 Million	PoR, PoS	1 minute
Slimcoin [121]	May 2014	50-250 coins	133 Million	PoB, PoW, PoS	1.5 minutes
Reddcoin [123]	January 20, 2014	Block reward	2.8 Billion	PoSV	1 minute
Faircoin [127]	6th of March, 2014.	Block reward	5.3 Million	PoC	Depends on Time-weight Parameter
NEM [129]	March 31st, 2015	transaction fees only + node rewards	899 Million	PoI	1 minute

filesystem (prover) proves that a file is intact to a client (verifier). This idea was leveraged to create a consensus algorithm by Permacoin [209] which can be used to store large files in chunks distributively provided by a file dealer. Thus, it can be an attractive solution for Cloud-based storage systems.

- **Delegated Byzantine Fault Tolerance (dBFT):** dBFT was introduced by NEO blockchain [211, 210]. It is a PBFT based consensus algorithm combined with the characteristics of DPoS that aims to improve the network performance drops as the number of participants increases in pure PoS. Towards this aim, the NEO holders elect some validators through a voting mechanism. Those validators use the BFT to reach the consensus and produce blocks.

- Verifiable Byzantine Fault Tolerance (VBFT): VBFT is another variation of BFT [212]. It is a hybrid consensus algorithm which combines Proof of Stake (PoS) with BFT that utilises a Verifiable Random Function (VRF). The VRF ensures the randomness and fairness during the consensus process. It is developed by Ontology blockchain [213], a platform for Self-sovereign Identity [167].

Year-wise distributions of crypto-currencies. To investigate it further, we have analysed a year-wise distribution of the genesis dates of different crypto-currencies. It is to understand if there is any inclination towards an alternative consensus algorithm over PoW in recent years. The distribution is illustrated in Figure 9, which represents a surprising observation: PoW is still the most widely used algorithms for crypto-currencies which have been created in recent years. For example, the numbers of crypto-currencies created with PoW algorithms in last three years (2017, 2018 & 2019) are 11, 19 and 4 respectively, in comparison to 4, 2 and 2 for PoS and DPoS combinedly. This implies that PoW is still the most popular consensus algorithm among the crypto-currency community. A deeper investigation reveals another insight though. The top 100 list retrieved from Coinmarketcap also contains crypto-tokens generated on the top of any smart-contract platform such as Ethereum, EOS, and Tron built on top of Ethereum. Most of these tokens have emerged after 2016 with Ethereum utilising PoW. This could be the reason why the most recent crypto-currencies have been found to utilise PoW. This will mostly likely change in future when Ethereum transitions its consensus algorithm towards PoS.

Market capitalisation of top 100 crypto-currencies. Another indication of PoW domination over other algorithms is the market-cap distribution of their corresponding crypto-currencies. The distribution is presented in Table 17. Not surprisingly, PoW currencies with a market cap of around 221 Billion USD have a massive 93% dominance over other currencies. DPoS and PoS currencies are the nearest rivals with a market cap of around 6 Billion USD and dominance of only 3% for each group.

Table 17: Market capitalisation of major consensus algorithms in top 100 Crypto-currencies

Consensus Algorithms	Market-cap (USD)
PoW	$221,238,526,412$
DPoS	$6,483,606,020$
PoS	$6,287,224,485$
PoW+PoS	$2,436,683,929$
Proof of Authority	$572,188,935$
Proof of Activity	$274,066,240$

Economic analysis. There are two kinds of mechanisms used to maintain the value of the currencies and keep the miners motivated. One mechanism is a limited supply. It creates scarcity, drives up the demand and hence, the price is increased. For example, Bitcoin has a total supply of only 21 million, and the block reward is gradually decreasing. It creates a scarcity in the market and helps to increase the value of Bitcoin. Though the relationship between the limited supply and the price of the currency is not always linear, it can be used as a ”rule of thumb” (see Table 2 for the total supply of different currencies). Secondly, inflation is used to keep the value of the coin stable. It is also often used to distribute coins to block producers from

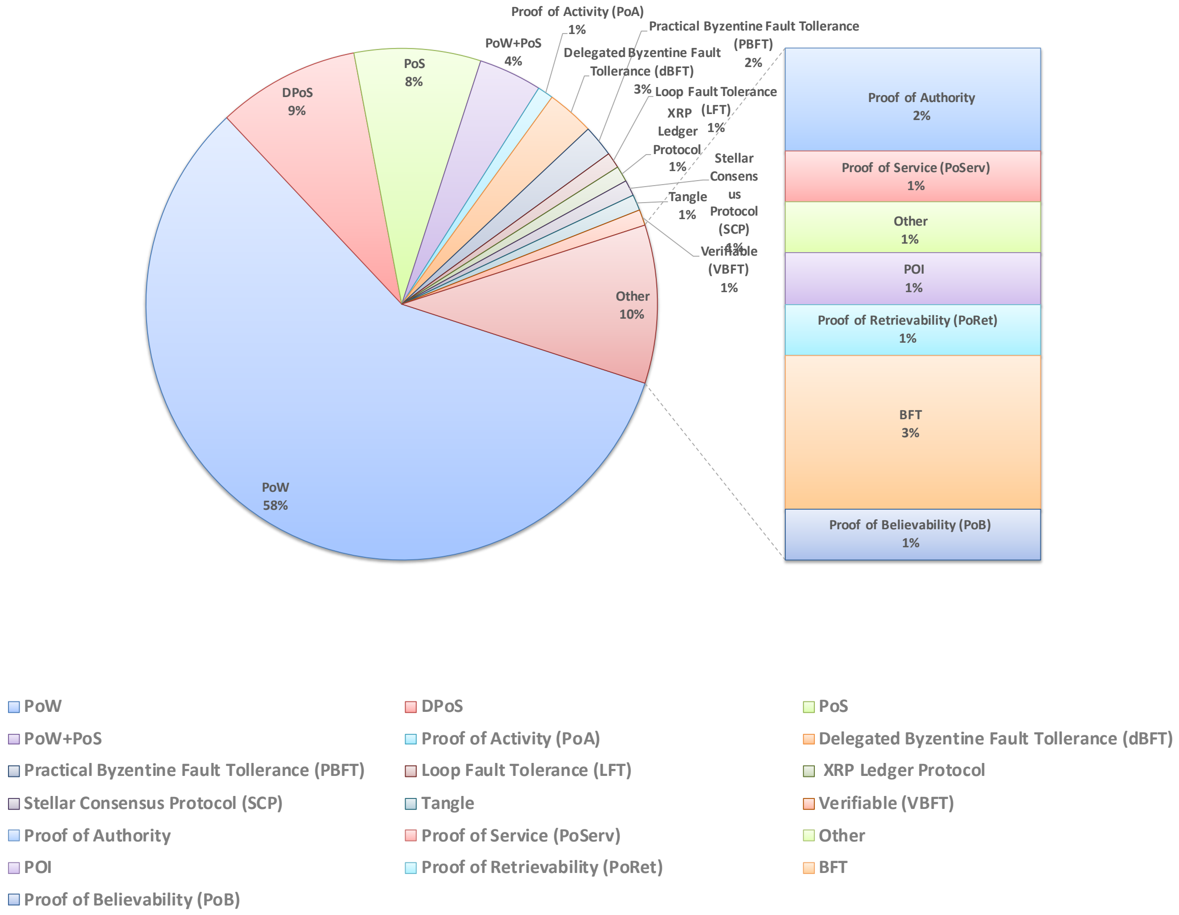

Figure 8: Consensus algorithms in Top 100 Crypto-currencies

running and securing the network. For example, EOS has distributed 1% of the inflated currency among its 21 validations (see Table 11).

Observation. From our investigation, it is evident that PoW algorithm, even with its major limitations, is still the most popular consensus algorithm to be utilised in different crypto-currencies. Currencies which utilise PoW algorithms consume a significant amount of energy, as illustrated in Section 4.5. Besides, they have a reduced throughput (in terms of transaction number) compared to PoS and DPoS currencies. For example, the reported TPS (Transactions Per Second) for Bitcoin and Ethereum are 7 and 15 – 25, respectively [177], while DPoS currencies such as EOS has a reported and estimated TPS of 50 and 4000 respectively [177] and Tron has a claimed TPS of 2000 [178]. DPoS currencies have better performance, at least in terms of TPS, over any PoW currency. Therefore, one might ask the underlying reason behind this counter-intuitive trend of PoW being the most popular consensus algorithm. We have identified a few reasons behind this which are presented below:

- Bitcoin is the most dominant crypto-currency in terms of market cap. As of 18 July, it has a market cap of around 171 Billion USD. In addition to this, its different forked variants (Bitcoin Cash [50] and Bitcoin Satoshi Vision [51]) also have a combined market cap of 8 Billion USD. If we exclude Bitcoin and its variants, we have a slightly different distribution of market-cap: the market-cap percentage of PoW algorithm is reduced from 93% to 71% percent, which is still significant in comparison to DPoS and PoS, its nearest rivals.

- PoW has the first-mover advantage because of Bitcoin and Ethereum, both being the pioneer in their respective domain. Bitcoin has been the first successful crypto-currency, while Ethereum is the first blockchain-based smart-contract platform. Other crypto-currencies, being motivated by their success, might have adopted the approach of utilising PoW as their corresponding consensus algorithm.

- Another strong argument in favour of PoW is its underlying security. The number of miners is far greater in Bitcoin than the number of validators in PoS and DPoS. This implies a better decentralisation in Bitcoin than PoS or DPoS. For example, EOS has only 21 validators, while Tron has 27 validators. The probability of collusion among these validators is far greater than that of any popular PoW currency. A recent study has shown that selfish mining is possible with DPoS Tezos [175], however, such act may outweigh the benefit as the penalty would slash the bon-

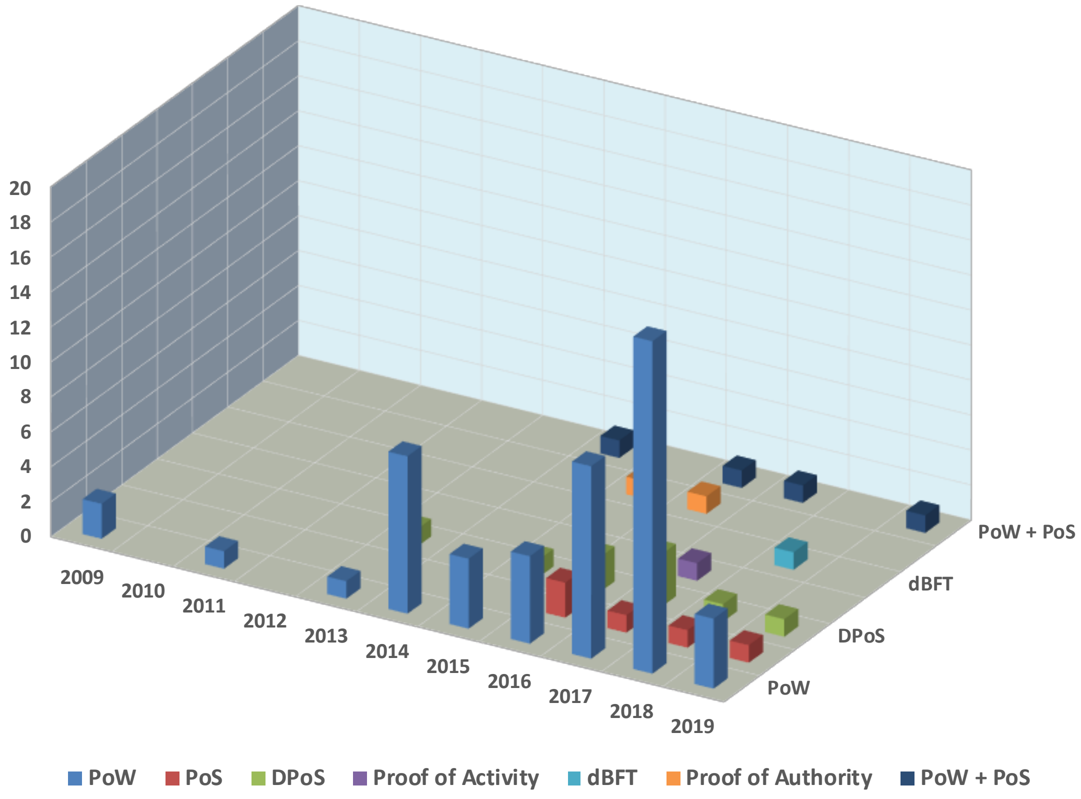

Figure 9: Year-wise distribution of consensus algorithms in Top 100 Crypto-currencies

ded stake significantly once caught. For these reasons, the security of any PoS/DPoS currency is regarded to be lower than a popular PoW currency. However, there is a counter argument against this. The mining centralisation is an issue as highlighted in Section 4.5. Therefore, a PoW currency might also suffer from a collusion attack. However, they also may suffer from the 51% attack due to small available hashpower for a particular algorithm family. For example, Verge, Ethereum Classic, Bitcoin Gold, Feathercoin, and Vertcoin suffered from such attacks in 2019 [176].

Decision Tree for Consensus Algorithms. The existence of numerous algorithms and wide variations in their properties impose a major challenge to comprehend them properly. In particular, it is difficult to test the suitability of a particular algorithm under certain criteria. A visual tool would be a great help in this regard. Towards this aim, we present a decision tree in Figure 10, which can be used to determine suitable consensus algorithms under certain criteria in different scenarios. For example, such a decision tree diagram can be leveraged to select a particular consensus algorithm while designing/developing a new blockchain system.

The tree utilises four critical criteria to achieve its goal: energy consumption, scalability, security (with respect to adversary tolerance) and ASIC-resistance. Energy consumption is a crucial determining factor in choosing appropriate consensus algorithms. PoW-type algorithms consume high energy, whereas PoS algorithms and their derivatives consume a moderate amount of energy. PoW-types algorithms are very slow as of now and can process only a limited number of transactions. However, compromising a popular PoW-based blockchain network is very difficult, and therefore, they are more secure than their counterparts. PoW-based algorithms can also be differentiates based on ASIC-resistance. As discussed earlier, ASIC is a specialised hardware, designed and used to solve hash-based computational problems. ASIC is expensive and hinders common people from participating in the blockchain network. Therefore, memory-based PoW has been designed. Now it is widely used in different crypto-currencies. On the other hand, DPoS algorithms are highly scalable whereas other hybrid and N-PoW/PoW algorithms have medium scalability. In terms of security, both DPoS and hybrid and N-PoW/PoW algorithms are less secure than any PoW algorithm.

For clarity, we provide a few examples to utilise the decision tree diagram presented in Figure 10. If a highly scalable blockchain system with low energy consumption is required, DPoS and BFT derivatives such as Tendermint, CTFG, and Ouroboros are the preferred options. However, they will have moderate security as described earlier. On the other hand, if security is of the highest priority, PoW algorithms are more suitable. In this scenario, there are two options: memory-bound or CPU bound. If ASIC resistance is desired, one should opt for memory-bound PoW algorithms. However, in such a case, one has to sacrifice scalability, and such algorithms will consume high energy.

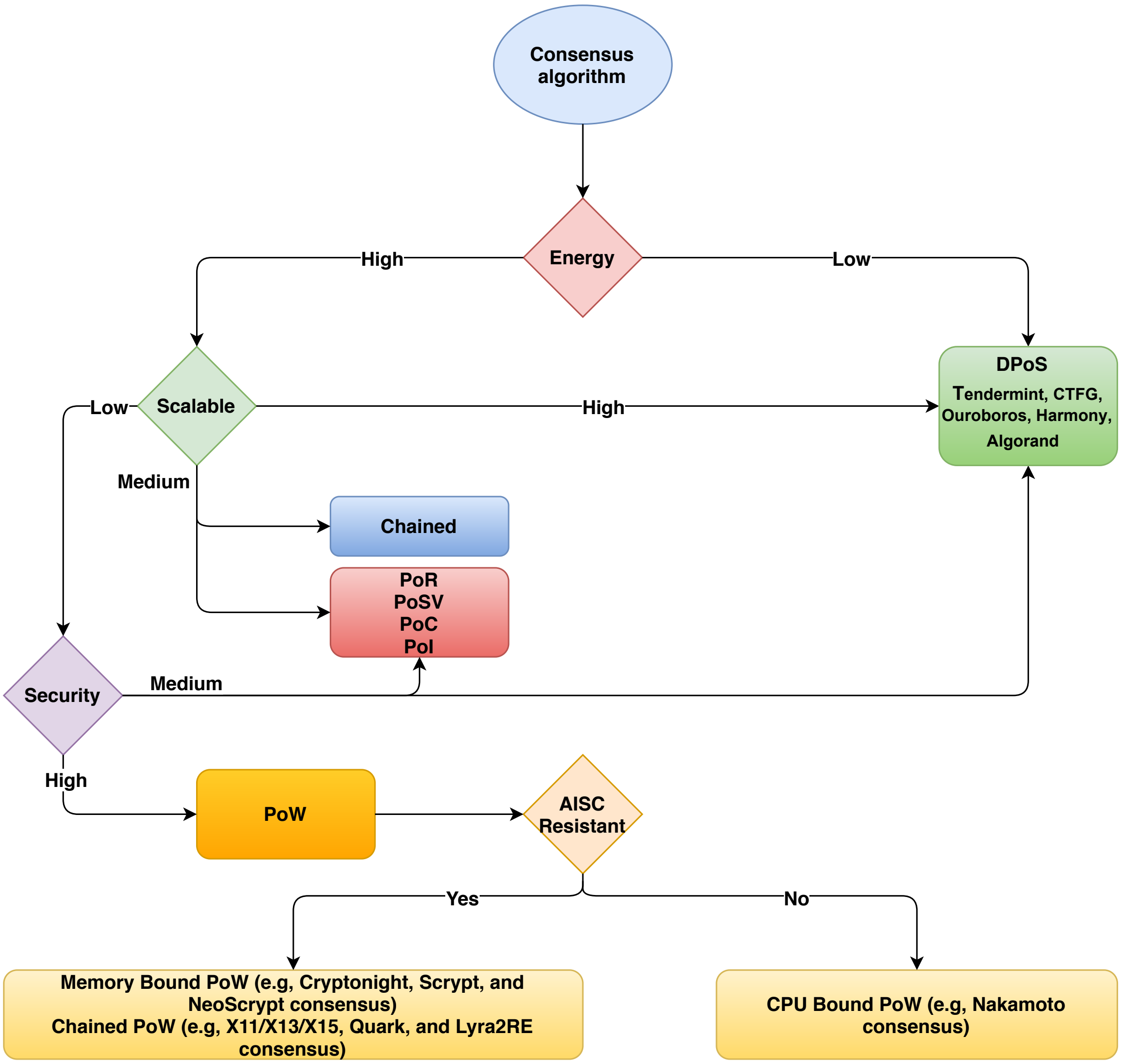
Consensus algorithm
Energy
High
Low
Scalable
Low
High
Medium
DPoS
Tendermint, CTFG, Ouroboros, Harmony, Algorand
Chained
PoR
PoSV
PoC
PoI
Security
Medium
High
PoW
AISC Resistant
Yes
No
Memory Bound PoW (e.g, Cryptonight, Scrypt, and NeoScrypt consensus)
Chained PoW (e.g, X11/X13/X15, Quark, and Lyra2RE consensus)
CPU Bound PoW (e.g, Nakamoto consensus)

Figure 10: Decision tree to choose appropriate consensus algorithms

Note that this is just an example of how such a decision tree can be developed using our selected four criteria. Other criteria can be utilised to generate a different diagram which might be suitable for other specific scenarios. Whenever such a diagram is to be developed, Table 3, Table 4, Table 5, Table 6, Table 7, Table 8, Table 9, Table 12, Table 13, Table 14 and Table 15 will be crucial as the these tables provide the required templates.

8. Related Surveys

There have been several works comparing and analysing different consensus algorithms, such as [194, 6, 7, 8, 9, 10, 11, 12, 13, 14] and [15]. In this section, we compare and contrast these works with the current work against a set of properties.

The properties that have been used are: taxonomy, number of structural, security, performance, the block reward and other features, variation, crypto-currency, protocol comparison and decision tree. The taxonomy property underlines if the corresponding work has utilised a taxonomy of criteria to compare different consensus algorithms. A taxonomy is often useful to classify different criteria in a meaningful way and hence, its usage will ensure that the comparison is carried out against closely related properties so as to increase the effectiveness of the comparison. The number of different properties highlights how many different properties are considered during the comparison. The variation and the crypto-currency properties signify if the corresponding work has considered different variations of a consensus algorithm compared to different crypto-currencies, respectively. The protocol comparison property indicates if there is a comparison of the consensus protocols in the respective work, whereas the decision tree property implies if there is a decision tree used in work in order to select a particular algorithm under certain criteria.

With these properties, the result of the comparative analysis between existing surveys and our one is summarised in Table 18. Like before, we have used the symbol “●” to denote a certain property is satisfied by the respective protocol whereas the symbol “○” denotes that the protocol does not fulfil the respective property.

Among the analysed works, the works carried out by Cachin et al. [194] and Bano et al. [6] are noteworthy as they represent the pioneer works in this scope. Cachin et al., in their work, explored different aspects of distributed systems and consensus algorithms and focused on consensus algorithms deployed in private blockchain systems. They also compared these algorithms against three properties. On the other hand, Bano et al. 's focus is more general in the sense that they explored consensus algorithms used both in public and private blockchain systems and compared them against 16 different criteria. Another thorough work in this scope is by Wang et al. [7] in which the authors presented a comprehensive survey of different aspects of consensus, mining, and blockchains in a detailed fashion with a comparison of different consensus algorithms against 7 criteria. Similarly, in [8], the authors analysed five different consensus algorithms against 6 criteria. None of these four works utilised any taxonomy, did not consider variations among the consensus algorithms and crypto-currency, and had not provided any decision tree.

Sanker et al. [9] analysed just two consensus protocols without utilising any property and satisfied none of our selected criteria. On the other hand, the authors in [10] compared a few crypto-currencies with a very brief analysis of their underlying consensus algorithms using just two criteria. Both of these two works did not utilise any taxonomy for their analysis with no comparison of the protocols and no decision tree.

A mathematics oriented analysis of different consensus algorithms, both for public and private blockchain systems, has been presented in [15]. The authors compared these algorithms using just five criteria. Another similar work is by Nguyen et el. ([13]), in which the authors analysed and compared a few consensus algorithms and their variations using 13 criteria. However, they also did not use any taxonomy, compare any crypto-currency, and consider any decision tree. Bach et al. analysed a few consensus algorithms and then compared a few crypto-currencies against four criteria [14]. However, they also did not consider any taxonomy, variations of consensus algorithms and a decision tree.

Finally, we have reviewed a couple of recent works. In [11] Garay et al. analysed a number of consensus algorithms in a detailed fashion against a set of six criteria with a taxonomy. Any variation of the consensus algorithms, the associated crypto-currencies, and any decision tree were absent in their study. In a similar note, Xiao et al. reviewed a number of consensus algorithms along with their algorithmic representations [12]. They compared these algorithms against a set of 15 criteria and also considered their different variations. However, they did not classify the consensus algorithms and did not present any method to select a consensus algorithm in their work. Besides, they did not compare the associated crypto-currencies.

In comparison to these works, our work presented in this article has utilised a taxonomy with a comparative analysis of the algorithms using 32 criteria, making it the most comprehensive one in this regard. We have also reviewed different variants of the selected consensus algorithms and analysed different crypto-currencies. Moreover, we have provided a decision tree to aid any researcher in choosing a particular algorithm for any future research.

9. Challenges and Future Research Directions

In this section we highlight some of the remaining challenges of the current generation blockchain systems (Section 9.1) and outline some possible future research directions (Section 9.2) in this domain.

9.1. Challenges

Researchers from both the industry and academia have proposed different new consensus algorithms and variants of the exiting consensus to overcome the limitations. However, there are still a few challenges.

Blockchain bloating: Blockchain bloating is one of the major challenges that the current generation of public blockchain systems is facing. Blockchain bloating refers to the phenomenon of ever-increasing size of any blockchain due to its append-only nature. For example, the size of Bitcoin and Ethereum blockchain is around 322GB (Gigabyte) [182] and 196GB [183] respectively as of January, 2021 which will continue to grow in future. A full node, a blockchain node which stores a full blockchain, needs to continuously expand their storage to store an ever-increasing blockchain. A light node, on the other hand, stores only the block headers [184]. Therefore, it can significantly reduce the blockchain storage requirements and thus, even a mobile device can be act as a light node. However, such nodes rely on other full nodes for blockchain operations and that is why light nodes are not fully trusted.

Blockchain trilemma - Scalability, Security & Decentralisation: The blockchain scalability trilemma is a well-known phenomenon in the blockchain community. The trilemma essentially outlines the trade-off required when a blockchain aims to achieve three core properties: Scalability, Security & Decentralisation [185]. In short, the trilemma states that a blockchain can only achieve two out of these three properties. The phenomenon is often observed in real systems as well: Bitcoin and Ethereum provides a strong security and decentralisation, however, they do not scale, as indicated by their low TPS. Similarly, even though PoS and its DPoS derivatives improve the scalability and decentralisation of a public blockchain system, this gain is achieved with the utilisation of a considerably low number of validators which might lower the security of the system. There has been a lot of research involved to solve the trilemma, however, an effective solution which can satisfy all three properties is still at large.

Privacy: The transparency of blockchain data is a big advantage, however, such visibility of data often raises privacy concerns. This aspect of privacy has not been considered at any consensus algorithm yet. There are a few privacy-friendly crypto-currencies, such as Monero [57] and Zcash [67] which have implemented privacy protection mechanisms at the crypto-currency level. There is another privacy issue which is getting

Table 18: Comparison of existing surveys with our work

Survey	Taxonomy	Struct. prop. (#)	Sec. prop. (#)	Perform. prop. (#)	Block reward prop. (#)	Other prop. (#)	Variation	Crypto-currency	Protocol comparison	Decision tree
Cachin et al. [194]	○	1	1	1	0	0	○	○	●	○
Bano et al. [6]	○	8	3	5	0	0	○	○	●	○
Wang et al. [7]	○	0	0	0	0	7	○	○	●	○
Baliga et al. [8]	○	2	2	2	0	2	○	○	●	○
Sanker et al. [9]	○	0	0	0	0	0	○	○	○	○
Mukhopadhyay et al. [10]	○	0	0	0	0	2	●	●	○	○
Mingxiao et al. [15]	○	0	1	4	0	0	○	○	●	○
Nguyen et al. [13]	○	1	4	2	1	5	●	○	●	○
Bach et al. [14]	○	0	1	3	0	○	○	●	○	
Garay et al. [11]	●	2	1	3	0	0	○	○	○	○
Xiao et al. [12]	○	1	0	3	4	7	●	○	●	○
Our work	●	11	13	5	4	0	●	●	●	●

traction in the blockchain community: the *right to be forgotten* introduced in the European GDPR (General Data Protection Regulation) [190]. The right enables any European to request an organisation to remove their personal data and the organisation is bound to comply with the request under certain condition within a stipulated time. This introduces an inevitable challenge for an organisation which uses a blockchain system to store personal data [191] as blockchain is inherently immutable. Even if a certain data is removed/updated within a blockchain at a point of time using a smart-contract, the corresponding previous transaction which was used to store the data at the first place will remain to exist. There, organisations must take pre-cautions before storing any personal data, even in encrypted format, within a blockchain. New research is required to tackle this challenge.

Initial barrier to join the mining/staking process: The mining process in of the popular crypto-currencies which utilise PoW algorithms is very competitive and requires lots of capital investment. This prohibits the general people from participating as the miner and has fuelled the mining pools' rise. Even in many PoS/DPoS mechanisms, one has to deposit a significant amount of corresponding crypto-currency, limiting general people to act as miners/validators. This is the ideologically exact opposite to what the blockchain was proposed in the first place.

Slow adoption of new consensus algorithms: There is a common problem in the blockchain domain. The adoption of new protocols is sometimes prohibitively slow. This jeopardises the implementation of any novel and better consensus mechanism.

Susceptibility towards Eclipse attacks: As per our analysis, all consensus algorithms are susceptible towards eclipse attacks. An eclipse attack is a network level attack and hence, it is difficult to prevent on the application level in which a consensus algorithm operates. According to [168], there are a few ways by which we can try to minimise the impact of an eclipse attack: accept blockchain data only from white-listed (trusted) sources, introduce measures (e.g. new connections are selected randomly) which would make attacks more costly and so on. However, it is to be noted, none of such measures can guarantee resiliency against an eclipse attack.

Centralisation of mining pools: As highlighted in Section 4.5, the mining centralisation is a major challenge to democratise the mining process. Even though PoS algorithms have been advocated to solve the problem, as our analysis suggests, currencies with PoW consensus algorithms are still the dominant ones. Therefore, it would be important to find a solution to this issue. However, an optimal solution is not on the horizon yet.

The other side of blockchain immutability: Immutability is a cornerstone of blockchain security. Unfortunately, code and data immutability might cause serious security concerns. If there is a bug within a deployed smart-contract there is no way to rectify the error [192]. The solution is to fix the bug, deploy another smart-contract and abandon the buggy contract. Such a contract keeps existing in the blockchain even though there is no utility whatsoever and thus wasting valuable storage. One infamous example of the implication of this issue is the Ethereum DAO (Decentralised Autonomous Organisation) attack in which an attacker was able to steal 60Million USD worth of Ether (Ethereum cryptocurrency) in their address by exploiting a bug in the DAO contract [192]. The immutability of blockchain prohibited the recovery of the stolen ethers, thereby, forcing a hard-fork of Ethereum. It is a challenging problem to address and will require significant research in this domain in future.

9.2. Future Research Directions

In this section, we explore a few possible future research directions.

From PoW to PoS/DPoS/BFT PoS: With the dominance of PoW over other consensus algorithms, one might wonder what lies ahead. We believe that there will be most definitely a shift of balance among the consensus algorithms: from PoW to PoS/DPoS algorithms. In this regard, the PoS transformation process of Ethereum will be a crucial factor. The proposed Ethereum PoS consensus mechanisms, both CFFG and CTFG, are highly regarded by the academics and industrial enthusiasts for their strong guarantee of security. In addition, with their strong focus on economic incentive and game-theoretic based approach, it is believed that their security will be as close as PoW and much better than any current PoS/DPoS algorithm can provide. In particular, the number of validators will be much higher than any number leveraged in the current PoS/DPoS algorithms.

Game theory and crypto-economics: Game theory and crypto-economics have become important tools to analyse the security of any consensus algorithm. There have been numerous works in this scope for the core PoW algorithm, which can be found in [179, 180, 181]. However, there have been minimal works exploring the game theoretic and crypto-economic analysis for the variants of PoW algorithms or other consensus algorithms analysed in this article. Such analysis would be critical to increase the trustworthiness and adoption of the respective consensus protocol.

Blockchain Sharding: As mentioned earlier, the notion of blockchain sharding is extensively explored as solutions to blockchain bloating, fault tolerance and scalability issues in public blockchain systems. However, such sharding will require a different level of consensus algorithm so as to enable the interactions among different shards and ensure their security. In addition, there are different types of sharding such as network sharding which targets structural configurations of blockchain validator nodes, transaction sharding which dictates how transactions are divided and processed in different shards and state sharding which deals with dividing and managing the whole blockchain in different shards [141]. Deciding which sharding mechanism achieves better performance and security will be an exciting avenue for blockchain research in the future.

Public and Private Blockchain Collaboration: It is clear that there will be a number of different public blockchains that will co-exist in the future. In addition, there will be many private blockchain systems serving purposes for application domains for which public blockchains will not be suitable. With these two different types of blockchains side-by-side, it is envisioned that some applications will require the interaction and linkage between multiple public and private blockchain systems. We have already noticed the emergence of blockchain systems that can interact with public and private blockchain systems. Examples of such “multi-chain” systems are Polkadot [5], Cosmos [6], MultiChain [7] and others. The traditional consensus algorithms might not be suitable for such systems. Therefore, it will be required either to modify the existing consensus algorithms to make them fit for the purpose or to introduce a novel consensus algorithm. We believe that this will be an exciting research avenue in the future domain of consensus algorithms.

Application specific consensus algorithms: We have noticed a recent trend within the academia which evolves around the idea that new consensus algorithms are required for some particular application domains as the existing consensus algorithms are not fit-for-purpose. Towards this aim, there have been proposals for new consensus algorithms in multiple application domains such as IoT [170, 171], electricity trading [172], vehicular network [173, 174] and so on. We expect this trend to grow more in future.

Redactable blockchain: To address the issues of fixing buggy smart-contracts and GDPR compliance within blockchain, the notion of redactable blockchain has been proposed [193, 195]. A redactable blockchain is essentially a blockchain that supports mutability, facilitating ‘edits’ of transactions/blocks under strict conditions. However, this goes against the immutability philosophy of blockchain and the idea has not gained enough traction yet. Despite this, it might be an engrossing avenue for future research.

10. Conclusion

There is a high anticipation among the blockchain enthusiasts that blockchain technology will disrupt many existing application domains. Unfortunately, most of the existing blockchain systems struggle to properly satisfy the need for any wide-scale real-life deployment as they have serious limitations such as scalability. Many of these limitations are due to the issues in the underlying consensus algorithms used in a particular system. This is because a consensus algorithm is the core component of any blockchain system, and it dictates how a system behaves and performs. In the quest to create more practical blockchain systems, the principal focus has been on consensus algorithms. This has led to the explorations: either existing consensus algorithms have been exploited or novel consensus mechanisms have been introduced. The ultimate consequence of this phenomenon is a wide-range of consensus algorithms currently in existence. To advance this the state of

[5] https://polkadot.network/

[6] https://cosmos.network/

[7] https://www.multichain.com/

blockchain technology, it is essential to synthesise these consensus algorithms under a systematic study, which is the main motivation of this article. However, as our analysis in this article suggests, an ideal consensus algorithm is still elusive as almost all algorithms have significant disadvantages in one way or another with respect to their security and performance. Until a consensus algorithm finds the correct balance between these crucial factors, we might not see the wide-scale adoption as many crypto-currency enthusiasts are hoping.

There is one issue that must be highlighted before we conclude this article. This article explores and synthesises the consensus algorithms available in different public blockchain systems with crypto-currencies. However, there are other distributed ledger systems, which do not rely on any blockchain-type structure. Instead, they utilise other structures to represent their respective ledgers. Examples of two such prominent blockchain systems are IoTA [165] and NANO [166]. Both of their ledgers are based on DAG (Directed Acyclic Graph), a specific type of directed graph with no cycle. However, IoTA uses a novel consensus algorithm called Tangle [189], while NANO utilises a representative based consensus mechanism [188]. These two systems have received significant attention because of their feeless structure and fast transaction rates. However, we do not consider these systems any further as they are out of scope for this article. We plan to investigate such novel systems in a different exploration in the future.

Acknowledgement

The authors would to like extend their heartfelt gratitude to Dr Alan Colman, Senior Lecturer of Swinburne University of Technology for his comments and suggestions to improve this manuscript.

References

[1] Nakamoto, S. “Bitcoin: A peer-to-peer electronic cash system”. 2008. [Online] Available: https://bitcoin.org/bitcoin.pdf. Accessed on March 1, 2019.

[2] Wanguba, J. “How Many Cryptocurrencies Are There In 2020?”. Jun 25, 2020. [Online] Available: https://bitcoin.org/bitcoin.pdf. Accessed on July 7, 2020.

[3] Pilkington, M. “11 Blockchain technology: principles and applications”. Research handbook on digital transformations, 225, 2016.

[4] Crosby, M., Pattanayak, P., Verma, S. and Kalyanaraman, V. “Blockchain technology: Beyond bitcoin”. Applied Innovation, 2(6-10), p. 71, 2016.

[5] Cachin, C. and Vukolić, M. “Blockchains Consensus Protocols in the Wild”. arXiv preprint arXiv:1707.01873, 2017.

[6] Bano, S., Sonnino, A., Al-Bassam, M., Azouvi, S., McCorry, P., Meiklejohn, S., and Danezis, G. “Consensus in the Age of Blockchains.”. arXiv preprint arXiv:1711.03936, 2017.

[7] Wang, W., Hoang, D.T., Hu, P., Xiong, Z., Niyato, D., Wang, P., Wen, Y. and Kim, D.I. “A survey on consensus mechanisms and mining strategy management in blockchain networks”. IEEE Access, 7, pp.22328-22370, 2019.

[8] Baliga, A. “Understanding Blockchain Consensus Models”. April, 2017. [Online] Available: https://www.persistent.com/wp-content/uploads/2017/04/WP-Understanding-Blockchain-Consensus-Models.pdf. Accessed on August 5, 2019.

[9] Sankar, L. S. Siva, Sindhu, M. and Sethumadhavan, M. “Survey of consensus protocols on blockchain applications” 4th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE, 1–5, 2017.

[10] Mukhopadhyay, U., Skjellum, A., Hambolu, O., Oakley, J., Yu, L., and Brooks, R. “A brief survey of cryptocurrency systems.” In Proceedings of the 14th annual conference on privacy, security and trust (PST). IEEE, 745–752, 2016.

[11] Garay, J. and Kiayias, A. “Sok: A consensus taxonomy in the blockchain era” Cryptographers’ Track at the RSA Conference. Springer, 284–318, 2020.

[12] Xiao, Y., Zhang, N., Lou, W. and Hou, Y. T. “A survey of distributed consensus protocols for blockchain networks” IEEE Communications Surveys & Tutorials. IEEE, 2020.

[13] Nguyen, G. and Kim, K. “A Survey about Consensus Algorithms Used in Blockchain” Journal of Information processing systems. 14(1), 2018.

[14] Bach, L.M., Mihaljevic, B. and Zagar, M. “Comparative analysis of blockchain consensus algorithms” In Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, 1545–1550, 2018.

[15] Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W. and Qijun, C. “A review on consensus algorithm of blockchain” In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2567–2572, 2017.

[16] Schneider, F. B. “Implementing fault-tolerant services using the state machine approach: A tutorial”. ACM Computing Surveys (CSUR), 22(4), 299-319, 1990.

[17] Lamport, L., Shostak, R. and Pease, M. “The Byzantine generals problem”. ACM Transactions on Programming Languages and Systems (TOPLAS), 4(3), 382-401, 1982.

[18] Lamport, L. “The part-time parliament”. ACM Transactions on Computer Systems, 16(2), 133–169, May 1998.

[19] Lamport, L. “Paxos made simple”. SIGACT News, 32(4), 51–58, 2001.

[20] Oki, B. M. and Liskov, B. “Viewstamped replication: A new primary copy method to support highly- available distributed systems”. In Proc. 7th ACM Symposium on Principles of Distributed Computing (PODC), ACM, 8–17, 1988.

[21] Hunt, P., Konar, M., Junqueira, F. P. and Reed, B. “ZooKeeper: Wait-free coordination for internet- scale systems”. In Proc. USENIX Annual Technical Conference, 8(9), 2010.

[22] Ongaro, D. and Ousterhout, J. K. “In search of an understandable consensus algorithm”. In Proc. USENIX Annual Technical Conference, pp. 305–319, 2014.

[23] Castro, M. and Liskov, B. “Practical Byzantine fault tolerance and proactive recovery”. ACM Transactions on Computer Systems, 20(4), 398–461, Nov. 2002.

[24] Chowdhury, M. J. M., Ferdous, M. S., Biswas, K., Chowdhury, N., Kayes, A. S. M., Alazab, M. and Watters, P. “A Comparative Analysis of Distributed Ledger Technology Platforms”. IEEE Access, 7:167930-167943, 2019.

[25] Ferdous, M. S., Biswas, K., Chowdhury, M. J. M., Chowdhury, N. and Muthukkumarasamy, V. ”Integrated platforms for blockchain enablement”. Advances in Computers, Elsevier, 2019.

[26] Seigen, M. J., Nieminen, N. T. and Juarez, A. M. “CryptoNight Hash Function”. March, 2013. [Online] Available: https://cryptonote.org/cns/cns008.txt. Accessed on May 5, 2019.

[27] Dwork, C. and Naor, M. “Pricing via processing or combatting junk mail”. Annual International Cryptology Conference, 139–147, 1992.

[28] Douceur, J. R. “The sybil attack”. In International workshop on peer-to-peer systems, pp. 251-260, 2002.

[29] Finlow-Bates, K. “A Lightweight Blockchain Consensus Protocol”. August, 2017. [Online] Available: http://www.chainfrog.com/wp-content/uploads/2017/08/consensus.pdf. Accessed on September 11, 2019.

[30] Back, A. Hashcash-a denial of service counter-measure. 2002.

[31] Bertoni, G., Daemen, J., Peeters, M. and Van Assche, G. “Keccak” In Annual International Conference on the Theory and Applications of Cryptographic Techniques pp. 313–314, 2013.

[32] Percival, C. and Josefsson, S. “The scrypt Password-Based Key Derivation Function”. August, 2016. [Online] Available: https://tools.ietf.org/html/rfc7914 Accessed on September 11, 2019.

[33] “Wikipedia entry on Custom Hardware Attack”. [Online] Available: https://en.wikipedia.org/wiki/Custom_hardware_attack.

Accessed on 21 May, 2019.

[34] Bernstein, D. J. "The Salsa20 family of stream ciphers". New stream cipher designs. pp. 84–97, Springer, 2008.

[35] "SThe Scrypt Mining Algorithm: Everything You Need to Know". [Online] Available: https://www.easypc.io/crypto-mining/scrypt-hardware/. Accessed on May 21, 2019.

[36] Buntinx, JP "Scrypt vs X11 vs SHA-256". March 23, 2017. [Online] Available: https://themerkle.com/scrypt-vs-x11-vs-sha-256/. Accessed on May 25, 2019.

[37] Bernstein, D. J. ChaCha, a variant of Salsa20. "New stream cipher designs". Vol. 8, pp. 3–5, 2008.

[38] "Cryptocurrency Mining Hash Algorithms". July 24, 2019. [Online] Available: http://www.bitcoinlion.com/cryptocurrency-mining-hash-algorithms/. Accessed on July 24, 2019.

[39] Biryukov, A. and Khovratovich, D. "Equihash: Asymmetric proof-of-work based on the generalized birthday problem". Ledger(2). 2017.

[40] Wagner, D. "A generalized birthday problem". Cryptor(2442). pp. 288–303. 2002.

[41] "Ethash". [Online] Available: https://github.com/ethereum/wiki/wiki/Ethash. Accessed on July 24, 2019.

[42] "Dagger-Hashimoto". [Online] Available: https://github.com/ethereum/wiki/blob/master/Dagger-Hashimoto.md. Accessed on July 24, 2019.

[43] Buterin, V. "Dagger: A Memory-Hard to Compute, Memory-Easy to Verify Scrypt Alternative". [Online] Available: http://www.hashcash.org/papers/dagger.html. Accessed on July 24, 2019.

[44] Gligoroski, D., Klima, V., Knapskog, S. J., El-Hadedy, M. and Amundsen, J. "Cryptographic hash function blue midnight wish". In Proceedings of the 1st International Workshop on Security and Communication Networks pp. 1–8, 2009

[45] Gauravaram, P., Knudsen, L. R. and Matusiewicz, K., Mendel, F. and Rechberger, C., Schläffer, M. and Thomsen, S. "Grøstl-a SHA-3 candidate". Dagstuhl Seminar Proceedings. 2009.

[46] Wu, H. "The hash function JH". Submission to NIST (round 3). Vol. 6, 2011.

[47] "Quark Coin". [Online] Available: http://www.quarkcoins.com/. Accessed on July 24, 2019.

[48] "First Impressions from the Baikal Mini Miner ASIC". [Online] Available: https://cryptomining-blog.com/tag/quark-asic-miner/. Accessed on July 24, 2019.

[49] "Bitcoin". [Online] Available: https://www.bitcoin.org/. Accessed on April 24, 2019.

[50] "Bitcoin Cash". [Online] Available: https://www.bitcoincash.org/. Accessed on July 24, 2019.

[51] "Bitcoin SV". [Online] Available: https://bitcoinsv.io/. Accessed on July 24, 2019.

[52] "Syscoin". [Online] Available: http://syscoin.org/. Accessed on July 24, 2019.

[53] "Peer Coin". [Online] Available: https://peercoin.net. Accessed on July 24, 2019.

[54] "Counterparty". [Online] Available: https://counterparty.io. Accessed on July 24, 2019.

[55] Aumasson, J., Neves, S., Wilcox-O'Hearn, Z., Winnerlein, C. "BLAKE2: simpler, smaller, fast as MD5". International Conference on Applied Cryptography and Network Security. pp. 119–135, 2013.

[56] "Emercoin. [Online] Available: https://emercoin.com/. Accessed on July 24, 2019.

[57] "Monero". [Online] Available: https://www.getmonero.org/. Accessed on July 10, 2019.

[58] "Bytecoin". [Online] Available: https://bytecoin.org/. Accessed on July 10, 2019.

[59] "AEON". [Online] Available: https://www.aeon.cash/. Accessed on July 10, 2019.

[60] "Boolberry". [Online] Available: http://boolberry.com/. Accessed on July 10, 2019.

[61] "Karbowanec". [Online] Available: https://karbo.io/. Accessed on July 10, 2019.

[62] "Litecoin". [Online] Available: https://litecoin.org/. Accessed on July 24, 2019.

[63] "Verge". [Online] Available: https://vergecurrency.com. Accessed on July 20, 2019.

[64] "Bitmark". [Online] Available: https://bitmark.com/. Accessed on July 20, 2019.

[65] "Dogecoin". [Online] Available: http://dogecoin.com/ Accessed on July 20, 2019.

[66] "Gamecredit)". [Online] Available: https://gamecredits.com/. Accessed on July 20, 2019.

[67] "Zcash". [Online] Available: https://z.cash/. Accessed on July 10, 2019.

[68] "Bitcoin Gold". [Online] Available: https://bitcoingold.org/. Accessed on July 10, 2019.

[69] "Komodo". [Online] Available: https://komodoplatform.com/en. Accessed on July 10, 2019.

[70] "Zclassic". [Online] Available: http://zclassic.org/. Accessed on July 10, 2019.

[71] "ZenCash". [Online] Available: https://zensystem.io/. Accessed on July 10, 2019.

[72] "Ethereum". [Online] Available: https://www.ethereum.org/. Accessed on July 10, 2019.

[73] "Ethereum Classic". [Online] Available: https://ethereumclassic.github.io/. Accessed on July 10, 2019.

[74] "Ubiq". [Online] Available: https://ubiqsmart.com/. Accessed on July 10, 2019.

[75] "Shif". [Online] Available: http://www.shiftnrg.org/. Accessed on July 10, 2019.

[76] "Expanse". [Online] Available: https://www.expanse.tech/. Accessed on July 10, 2019.

[77] "Red Pulse". [Online] Available: https://www.red-pulse.com/landing. Accessed on July 10, 2019.

[78] "Feathercoin". [Online] Available: https://www.feathercoin.com/. Accessed on July 10, 2019.

[79] "GoByte". [Online] Available: https://gobyte.network/. Accessed on July 10, 2019.

[80] "UFO Coin". [Online] Available: https://ufocoin.net/. Accessed on July 10, 2019.

[81] "Innova". [Online] Available: https://innovacoin.info/. Accessed on July 10, 2019.

[82] "Dash". [Online] Available: https://www.dash.org/. Accessed on July 10, 2019.

[83] "Stratis. [Online] Available: https://stratisplatform.com/. Accessed on July 10, 2019.

[84] "Cloakcoin". [Online] Available: https://www.cloakcoin.com/. Accessed on July 10, 2019.

[85] "Stealthcoin". [Online] Available: https://www.stealthcoin.com/. Accessed on July 10, 2019.

[86] "DeepOnion". [Online] Available: https://deeponion.org/. Accessed on July 10, 2019.

[87] "HTMLcoin". [Online] Available: https://htmlcoin.com/. Accessed on July 10, 2019.

[88] "Regal Coin". [Online] Available: https://regalcoin.co/. Accessed on July 10, 2019.

[89] "Memetic". [Online] Available: https://memetic.ai/. Accessed on July 10, 2019.

[90] "Exclusive Coin". [Online] Available: https://exclusivecoin.pw/. Accessed on July 10, 2019.

[91] "Creditbit". [Online] Available: https://www.creditbit.org/. Accessed on July 10, 2019.

[92] "PIVX". [Online] Available: https://pivx.org/. Accessed on July 10, 2019.

[93] "MonetaryUnit". [Online] Available: https://www.monetaryunit.org/. Accessed on July 10, 2019.

[94] "ALQO". [Online] Available: https://alqo.org/. Accessed on July 10, 2019.

[95] "Bitcloud". [Online] Available: https://bit-cloud.info/. Accessed on July 10, 2019.

[96] "Vertcoin". [Online] Available: https://vertcoin.org/. Accessed on July 10, 2019.

[97] "Monacoin". [Online] Available: https://monacoin.org/. Accessed on July 10, 2019.

[98] "Crypto". [Online] Available: http://tailflick.wixsite.com/official-crypto. Accessed on July 10, 2019.

[99] "Cryptonite". [Online] Available: https://cryptonite.info/.

Accessed on July 10, 2019.

[100] "Quark Coin Wiki". [Online] Available: http://coinwiki.info/en/Quark. Accessed on July 10, 2019.

[101] "Lyra2RE-A new PoW algorithm for an ASIC-free future". [Online] Available: https://vertcoin.org/wp-content/uploads/2017/10/Vertcoin_Lyra2RE_Paper_11292014.pdf. Accessed on July 10, 2019.

[102] "Wiki entry on M7". [Online] Available: http://cryptonite.info/wiki/index.php?title=M7_PoW. Accessed on July 10, 2019.

[103] "CudaMiner - a multi-threaded GPU miner for Cryptonite". [Online] Available: https://github.com/MiniblockchainProject/CudaMiner. Accessed on July 10, 2019.

[104] Doering, John "NeoScrypt, a Strong Memory Intensive Key Derivation Function". July 26, 2014. [Online] Available: http://phoenixcoin.org/archive/neoscrypt_v1.pdf. Accessed on July 10, 2019.

[105] "Bitcoin energy consumption". [Online] Available: https://digiconomist.net/bitcoin-energy-consumption. Accessed on July 10, 2019.

[106] "Wikipedia entry on Economies of scale". [Online] Available: https://en.wikipedia.org/wiki/Economies_of_scale. Accessed on July 10, 2019.

[107] "Wikipedia entry on Tragedy of the commons". [Online] Available: https://en.wikipedia.org/wiki/Tragedy_of_the_commons. Accessed on July 10, 2019.

[108] "Bitcoin hashrate distribution". [Online] Available: https://blockchain.info/pools. Accessed on July 10, 2019.

[109] Eyal, I. and Sirer, E.G. "Majority is not enough: Bitcoin mining is vulnerable". International Conference on Financial Cryptography and Data Security pages 436–454. March, 2014.

[110] Quantum Mechanic "Proof of stake instead of proof of work". July 11, 2011. [Online] Available: https://bitcointalk.org/index.php?topic=27787.0. Accessed on May 11, 2019.

[111] "Proof of Stake FAQ". [Online] Available: https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ. Accessed on August 5, 2019.

[112] Choi, Jon "Ethereum Casper 101". October 22, 2017. [Online] Available: https://medium.com/@jonchoi/ethereum-casper-101-7a851a4f1eb0. Accessed on July 10, 2019.

[113] Natoli, C. and Gramoli, V. "The blockchain anomaly". IEEE 15th International Symposium on Network Computing and Applications (NCA) pages 310–317. 2016.

[114] "Proof of Stake versus Proof of Work". September 13, 2015. [Online] Available: http://bitfury.com/content/5-white-papers-research/pos-vs-pow-1.0.2.pdf. Accessed on August 5, 2019.

[115] "Consensus Compare: Casper vs. Tendermint". November 16, 2017. [Online] Available: https://blog.cosmos.network/consensus-compare-casper-vs-tendermint-6df154ad56ae. Accessed on July 10, 2019.

[116] Zamfir, Vlad "The History of Casper - Chapter 4". December 12, 2016. [Online] Available: https://medium.com/@Vlad_Zamfir/the-history-of-casper-chapter-4-3855638b5f0e. Accessed on July 10, 2019.

[117] "Gridcoin". [Online] Available: https://github.com/gridcoin-community/Gridcoin-Wiki/wiki. Accessed on June 22, 2019.

[118] "Gridcoin Whitepaper. [Online] Available: https://gridcoin.us/assets/img/whitepaper.pdf. Accessed on 24 June, 2019.

[119] "Wikipedia entry on Berkeley Open Infrastructure for Network Computing (BOINC)". Accessed on 21 June, 2019. [Online] Available: https://en.wikipedia.org/wiki/Berkeley_Open_Infrastructure_for_Network_Computing. Accessed on 21 June, 2019.

[120] "Proof of Burn". [Online] Available: https://en.bitcoin.it/wiki/Proof_of_burn. Accessed on 21 June, 2019.

[121] "Slimcoin". [Online] Available: http://slimco.in/. Accessed on 24 June, 2019.

[122] "Slimcoin Whitepaper". [Online] Available: https://www.doc.ic.ac.uk/~ids/realdotdot/crypto_papers_etc_worth_reading/proof_of_burn/slimcoin_whitepaper.pdf. Accessed on 24 June, 2019.

[123] "Reddcoin". [Online] Available: https://www.reddcoin.com/. Accessed on 24 June, 2019.

[124] "Reddcoin Whitepaper. [Online] Available: https://www.reddcoin.com/papers/PoSV.pdf. Accessed on 24 June, 2019.

[125] "Reddcoin Wiki". [Online] Available: https://wiki.reddcoin.com/Proof_of_Stake_Velocity_(PoSV). Accessed on 24 June, 2019.

[126] "The Velocity of Money for Beginners". [Online] Available: https://www.joshuakennon.com/the-velocity-of-money-for-beginners/. Accessed on 25 June, 2019.

[127] "Faircoin Crypto-currency". [Online] Available: https://fair-coin.org. Accessed on 24 June, 2019.

[128] "Faircoin Whitepaper, Version 1.2". July, 2018. [Online] Available: https://fair-coin.org/sites/default/files/FairCoin2_whitepaper_V1.2.pdf. Accessed on 24 June, 2019.

[129] "NEM Technical Reference, Version 1.2.1". February 23, 2018. [Online] Available: https://nem.io/wp-content/themes/nem/files/NEM_techRef.pdf. Accessed on 24 June, 2019.

[130] "Faircoin FAQ". [Online] Available: https://fair-coin.org/en/faircoin-faqs. Accessed on 24 June, 2019.

[131] King, Sunny and Nadal, Scott "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake". August 19, 2012. [Online] Available: https://peercoin.net/assets/paper/peercoin-paper.pdf. Accessed on May 5, 2019.

[132] "Peercoin discussion forum, discussion#3043". November 4, 2014. [Online] Available: https://talk.peercoin.net/t/ppc-switching-between-pos-and-pow/3043. Accessed on May 5, 2019.

[133] Bentov, I., Gabizon, A., and Mizrahi, A. "Cryptocurrencies without proof of work". International Conference on Financial Cryptography and Data Security pages 142–157. 2016

[134] "Tendermint introduction". [Online] Available: http://tendermint.readthedocs.io/projects/tools/en/master/introduction.html. Accessed on July 30, 2019.

[135] Kwon, J. "Tendermint: Consensus without Mining". 2014. [Online] Available: https://tendermint.com/static/docs/tendermint.pdf. Accessed on July 30, 2019.

[136] "Tendermint wiki". [Online] Available: https://github.com/tendermint/tendermint/wiki/Byzantine-Consensus-Algorithm. Accessed on July 30, 2019.

[137] Buterin, V. and Griffith, V. "Casper the Friendly Finality Gadget". [Online] Available: https://github.com/ethereum/research/blob/master/papers/casper-basics/casper_basics.pdf. Accessed on August 1, 2019.

[138] Zamfir, V. "Casper the Friendly Ghost-A "Correct-by-Construction" Blockchain Consensus Protocol". [Online] Available: https://github.com/ethereum/research/blob/master/papers/CasperTFG/CasperTFG.pdf. Accessed on August 1, 2019.

[139] Sompolinsky, Y. and Zohar, A. "Secure high-rate transaction processing in bitcoin". International Conference on Financial Cryptography and Data Security pp. 507-527. January, 2015.

[140] "Harmony Blockchain". [Online] Available: https://www.harmony.one/. Accessed on January 7, 2021.

[141] Harmony Team "Harmony Technical Whitepaper". [Online] Available: https://harmony.one/whitepaper.pdf. Accessed on January 7, 2021.

[142] Yu, G., Wang, X., Yu, K., Ni, W., Zhang, J A., Liu, R. P. "Survey: Sharding in blockchains". IEEE Access. Vol. 8, pp. 14155–14181, 2020.

[143] Yu, G., Wang, X., Yu, K., Ni, W., Zhang, J A., Liu, R. P. "Survey: Sharding in blockchains". IEEE Access. Vol. 8, pp. 14155–14181, 2020.

[144] Boneh, D., Lynn, B., Shacham, H. "Short signatures from the Weil pairing". In Proceedings of the International conference on the theory and application of cryptology and information security pp. 514–532, 2001

[145] Buntinx, J. P. "What is Delegated Proof-of-Stake?". April 20, 2017. [Online] Available: https://themerkle.com/what-is-delegated-proof-of-stake/. Accessed on August 1, 2019.

[146] Kiayias, A., Russell, A., David, B., and Oliynykov, R. "Ouroboros: A provably secure proof-of-stake blockchain protocol". Annual International Cryptology Conference. pp. 357-388. August, 2017.

[147] "OUROBOROS PROOF OF STAKE ALGORITHM". Accessed on August 2, 2019. [Online] Available: https://cardanodocs.com/cardano/proof-of-stake/

[148] "Cardano Platform". [Online] Available: https://www.cardanohub.org/en/home/. Accessed on August 2, 2019.

[149] “EOS Platform”. [Online] Available: https://eos.io/. Accessed on May 24, 2019.

[150] “EOS Whitepaper”. [Online] Available: https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md. Accessed on 24 May, 2019.

[151] Kim, Jonathan “EOS Raises Record-Breaking $4 Billion from Crowdsale”. May 29, 2018. [Online] Available: https://cryptoslate.com/eos-raises-record-breaking-4-billion-from-crowdsale/. Accessed on May 24, 2019.

[152] Rosic, Ameer “What is EOS Blockchain: Beginners Guide”. [Online] Available: https://blockgeeks.com/guides/eos-blockchain/. Accessed on May 24, 2019.

[153] “Tron Platform”. [Online] Available: https://tron.network/. Accessed on May 24, 2019.

[154] “Tron Whitepaper”. [Online] Available: https://tron.network/static/doc/white_paper_v_2_0.pdf. Accessed on May 24, 2019.

[155] “TRON’s Consensus, “How it works””. November 11, 2018. [Online] Available: https://medium.com/@TRONNews/trons-consensus-how-it-works-6fd231b63715. Accessed on May 24, 2019.

[156] “Tezos Platform”. [Online] Available: https://tezos.com/. Accessed on May 24, 2019.

[157] Goodman, L.M. “Tezos Whitepaper”. September 2, 2014. [Online] Available: https://tezos.com/static/white_paper-2dc8c02267a8fb86bd67a108199441bf.pdf. Accessed on May 24, 2019.

[158] “Lisk Platform”. [Online] Available: https://lisk.io/. Accessed on June 1, 2019.

[159] “Lisk Whitepaper”. [Online] Available: https://github.com/slasheks/lisk-whitepaper/blob/development/LiskWhitepaper.md. Accessed on June 1, 2019.

[160] “Ark Platform”. [Online] Available: https://ark.io/. Accessed on June 1, 2019.

[161] “Ark Whitepaper”. [Online] Available: https://ark.io/Whitepaper.pdf. Accessed on June 1, 2019.

[162] “Peercoin discussion forum, discussion#2524”. June 15, 2014. [Online] Available: https://talk.peercoin.net/t/the-complete-guide-to-minting/2524. Accessed on May 5, 2019.

[163] “Hyperledger”. [Online] Available: https://www.hyperledger.org/. Accessed on July 10, 2019.

[164] “Quorum Blockchain”. [Online] Available: https://www.goquorum.com/. Accessed on July 10, 2019.

[165] “IOTA”. [Online] Available: https://www.iota.org/. Accessed on July 10, 2019.

[166] “Nano”. [Online] Available: https://nano.org/. Accessed on July 10, 2019.

[167] Ferdous, M.S., Chowdhury, F. and Alassafi, M. “In Search of Self-Sovereign Identity Leveraging Blockchain Technology”. IEEE Access, 7, pp.103059-103079, 2019.

[168] Heilman, E., Kendler, A., Zohar, A., and Goldberg, S. “Eclipse attacks on bitcoin’s peer-to-peer network”. 24th {USENIX} Security Symposium ({USENIX} Security 15, pp.129-144, 2015.

[169] Nolan, L. “Distributed Consensus Algorithms for Extreme Reliability”. USENIX Association, 2015. [Online] Available: https://www.goquorum.com/. Accessed on November 20, 2020.

[170] Huang, J., Kong, L., Chen, G., Wu, M., Liu, X. and Zeng, P. “Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism”. IEEE Transactions on Industrial Informatics, 15(6), pp.3680-3689, 2019.

[171] Tsang, Y. P., Choy, K. L., Wu, C. H. and Ho, G. T. S. and Lam, H. Y. “Blockchain-driven IoT for food traceability with an integrated consensus mechanism”. IEEE access, 7, pp.129000–129017, 2019.

[172] Liu, C., Chai, K. K., Zhang, X. and Chen, Y. “Peer-to-peer electricity trading system: smart contracts based proof-of-benefit consensus protocol”. Wireless Networks, 2019.

[173] Kudva, S., Badsha, S., Sengupta, S., Khalil, I. and Zomaya, A. “Towards secure and practical consensus for blockchain based VANET”. Information Sciences, 545, pp.170–187, 2020.

[174] Zheng, Z., Pan, J. and Cai, L. “Lightweight Blockchain Consensus Protocols for Vehicular Social Networks”. IEEE Transactions on Vehicular Technology, 2020.

[175] Neuder, M., Moroz, D. J. and Rao, R. and Parkes, D. C “Selfish Behavior in the Tezos Proof-of-Stake Protocol”. arXiv preprint arXiv:1912.02954, 2019. [Online] Available: https://arxiv.org/pdf/1912.02954.pdf

[176] Attah, E. “Five most prolific 51% attacks in crypto: Verge, Ethereum Classic, Bitcoin Gold, Feathercoin, Vertcoin”. [Online] Available: https://cryptoslate.com/prolific-51-attacks-crypto-verge-ethereum-classic-bitcoin-gold-feathercoin-vertcoin April 24, 2019. Accessed on November 25, 2020.

[177] Dogan, T. “Who Scales It Best? Blockchains’ TPS Analysis”. [Online] Available: https://hackernoon.com/who-scales-it-best-blockchains-tps-analysis-pv39g25mg. Accessed on July 27, 2019.

[178] O’Neal, S. “Who Scales It Best? Inside Blockchains’ Ongoing Transactions-Per-Second Race”. January 22, 2019. [Online] Available: https://cointelegraph.com/news/who-scales-it-best-inside-blockchains-ongoing-transactions-per-second-race. Accessed on July 27, 2019.

[179] Lewenberg, Y. and Bachrach, Y. and Sompolinsky, Y. and Zohar, A. and Rosenschein, J. S. “Bitcoin mining pools: A cooperative game theoretic analysis”. In Proc. International Conference on Autonomous Agents and Multiagent Systems, 919–927, 2015

[180] Beccuti, J. and Jaag, C. “The bitcoin mining game: On the optimality of honesty in proof-of-work consensus mechanism”. Swiss Economics Working Paper 0060, 2017

[181] Johnson, B., Laszka, A., Grossklags, J., Vasek, M. and Moore, T. “Game-theoretic analysis of DDoS attacks against Bitcoin mining pools”. In Proc. International Conference on Financial Cryptography and Data Security, 72–86, 2014

[182] “Bitcoin blockchain size”. [Online] Available: https://ycharts.com/indicators/bitcoin_blockchain_size. Accessed on January 12, 2021.

[183] “Ethereum blockchain size”. [Online] Available: https://blockchair.com/ethereum/charts/blockchain-size. Accessed on January 12, 2021.

[184] “Nodes and Clients”. December 5, 2020. [Online] Available: https://ethereum.org/en/developers/docs/nodes-and-clients/. Accessed on January 10, 2021.

[185] Hafid, A., Hafid, A. S., Samih, M. “Scaling blockchains: A comprehensive survey”. IEEE Access, 8, pp.125244–125262, 2020.

[186] Wang, G., Shi, Z. J., Nixon, M. and Han, S. “Sok: Sharding on blockchain”. In Proc. 1st ACM Conference on Advances in Financial Technologies, 41–61, 2019

[187] Bagui, S. and Nguyen, L. T. “Database sharding: to provide fault tolerance and scalability of big data on the cloud” International Journal of Cloud Applications and Computing, IGI Global, 14(1), 36–52, 2015.

[188] LeMahieu, C. “Nano: A Feeless Distributed Cryptocurrency Network”. [Online] Available: https://nano.org/en/whitepaper. Accessed on July 27, 2019.

[189] Popov, S. “The Tangle”. April 30, 2018. [Online] Available: https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf. Accessed on July 27, 2019.

[190] Politou, E., Alepis, E., Patsakis, C. “Forgetting personal data and revoking consent under the GDPR: Challenges and proposed solutions”. Journal of Cybersecurity, 4(1), 2018.

[191] Bayle, A., Koscina, M., Manset, D., Perez-Kempner, O. “When blockchain meets the right to be forgotten: technology versus law in the healthcare industry” In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI). IEEE, 788–792, 2018.

[192] Atzei, N., Bartoletti, M., Cimoli, T. “A survey of attacks on ethereum smart contracts (sok)” In Proceedings of the International conference on principles of security and trust. Springer, 164–186, 2017.

[193] Politou, E., Casino, F., Alepis, E., Patsakis, C. “Blockchain mutability: Challenges and proposed solutions” IEEE Transactions on Emerging Topics in Computing, 2019.

[194] Cachin, C. and Vukolić, M. “Blockchains Consensus Protocols in the Wild”. arXiv preprint arXiv:1707.01873, 2017.

[195] Ateniese, G., Magri, B., Venturi, D., Andrade, E. “Redactable blockchain–or–rewriting history in bitcoin and friends” In Proceedings of 2017 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 111–126, 2017.

[196] “XRP ”. [Online] Available: https://ripple.com/xrp/. Accessed

on January 11, 2021.

[197] Chase, B., MacBrough, E. “Analysis of the XRP ledger consensus protocol”. arXiv preprint arXiv:1802.07242, 2018.

[198] Mazieres, D. “The stellar consensus protocol: A federated model for internet-level consensus” Stellar Development Foundation, 32, 2015.

[199] “Steller Network”. [Online] Available: https://www.stellar.org/. Accessed on January 11, 2021.

[200] “VeChainThor”. [Online] Available: https://www.vechain.org/. Accessed on January 11, 2021.

[201] “VeChainThor Whitepaper”. [Online] Available: https://www.vechain.org/whitepaper/. Accessed on January 11, 2021.

[202] “IOST Blockchain”. [Online] Available: https://iost.io/. Accessed on January 11, 2021.

[203] “IOST Whitepaper”. [Online] Available: https://iost.io/624/. Accessed on January 11, 2021.

[204] Bentov, I., Lee, C., Mizrahi, A., Rosenfeld, M. “Proof of activity: Extending bitcoin’s proof of work via proof of stake [extended abstract]”. ACM SIGMETRICS Performance Evaluation Review, 42(3), 34–37, 2014.

[205] “Decred Blockchain”. [Online] Available: https://www.decred.org/. Accessed on January 12, 2021.

[206] “ICON Blockchain”. [Online] Available: https://icon.foundation/. Accessed on January 11, 2021.

[207] Ateniese, G., Magri, B., Venturi, D., Andrade, E. “Algorand: Scaling byzantine agreements for cryptocurrencies” Proceedings of the 26th Symposium on Operating Systems Principles. PP, 51–68, 2017.

[208] Bowers, K. D., Juels, A., Oprea, A. “Proofs of retrievability: Theory and implementation” Proceedings of the 2009 ACM workshop on Cloud computing security. PP, 43–54, 2009.

[209] Miller, A., Juels, A., Shi, E., Parno, B., Katz, J, “Permacoin: Repurposing bitcoin work for data preservation” Proceedings of the 2014 IEEE Symposium on Security and Privacy. PP, 475–490, 2014.

[210] “Delegated Byzantine Fault Tolerance (dBFT)”. [Online] Available: https://docs.neo.org/developerguide/en/articles/consensus/consensus_algorithm.html. Accessed on January 11, 2021.

[211] “NEO Blockchain”. [Online] Available: https://neo.org. Accessed on January 11, 2021.

[212] “Verifiable Byzantine Fault Tolerance (VBFT)”. [Online] Available: https://docs.ont.io/ontology-elements/consensus-mechanism. Accessed on January 11, 2021.

[213] “Ontology Blockchain”. [Online] Available: https://ont.io. Accessed on January 11, 2021.

[214] Sharma Pradip Kumar, Chen Mu-Yen, Park Jong Hyuk “A software defined fog node based distributed blockchain cloud architecture for IoT”. IEEE Access (6) (2018), pp. 115-124