Exploring the Attack Surface of Blockchain: A Comprehensive Survey

Article *in* IEEE Communications Surveys & Tutorials · March 2020

DOI: 10.1109/COMST.2020.2975999

CITATIONS
163

READS
3,362

7 authors, including:

Muhammad Saad
University of Central Florida
35 PUBLICATIONS **848** CITATIONS
SEE PROFILE

Jeffrey Spaulding
Canisius College
16 PUBLICATIONS **323** CITATIONS
SEE PROFILE

Sachin Shetty
Old Dominion University
299 PUBLICATIONS **4,957** CITATIONS
SEE PROFILE

Daehun Nyang
Ewha Womans University
218 PUBLICATIONS **2,121** CITATIONS
SEE PROFILE

Exploring the Attack Surface of Blockchain: A Comprehensive Survey

Muhammad Saad, Jeffrey Spaulding, Laurent Njilla, Charles Kamhoua, Sachin Shetty, DaeHun Nyang, and David Mohaisen

***Abstract*—In this paper, we systematically explore the attack surface of the Blockchain technology, with an emphasis on public Blockchains. Towards this goal, we attribute attack viability in the attack surface to 1) the Blockchain cryptographic constructs, 2) the distributed architecture of the systems using Blockchain, and 3) the Blockchain application context. To each of those contributing factors, we outline several attacks, including selfish mining, the 51% attack, DNS attacks, distributed denial-of-service (DDoS) attacks, consensus delay (due to selfish behavior or distributed denial-of-service attacks), Blockchain forks, orphaned and stale blocks, block ingestion, wallet thefts, smart contract attacks, and privacy attacks. We also explore the causal relationships between these attacks to demonstrate how various attack vectors are connected to one another. A secondary contribution of this work is outlining effective defense measures taken by the Blockchain technology or proposed by researchers to mitigate the effects of these attacks and patch associated vulnerabilities.**

***Index Terms*—Blockchain, security, attack surface, applications, peer-to-peer systems.**

I. Introduction

BLOCKCHAIN technology is being explored in many innovative applications, such as cryptocurrencies [1]–[3], smart contracts [4], [5], communication systems [6], [7], health care [8], [9], Internet of Things [10]–[12], financial systems [13], [14], censorship resistance [15], electronic voting [16], [17], and distributed provenance [18]–[20], among others. Using Blockchain's transparent and fully distributed peer-to-peer architecture, these applications benefit from an append-only model in which "transactions" accepted in the Blockchain cannot be modified [19], [21], [22]. The transparency of the Blockchain enables storing publicly verifiable and undeniable records [23]. Furthermore, the Blockchain's peer-to-peer system provides verifiable ledger maintenance without a centralized authority, thus addressing the single point-of-failure and single point-of-trust [24]. Taking advantage of these properties, several Blockchain applications have been developed including cryptocurrencies such as Bitcoin and Litecoin, smart contract platforms such as Ethereum, and Decentralized Autonomous Organizations (DAO's) such as Dash and Bitshares [25], [26].

Despite the functional features that Blockchain brings to the design space of these applications [44], recent reports have highlighted the security risks associated with this technology [10], [45]–[47]. For instance, in June 2016 an unknown attacker managed to drain $50 million USD from "The DAO", a decentralized autonomous organization that operates on Blockchain-based smart contracts, or pre-programmed rules that govern the organization [48]. In August 2016, bitcoins worth $72 million USD were stolen from the exchange platform Bitfinex in Hong Kong [49]. In June 2017, Bitfinex also experienced a distributed denial-of-service (DDoS) attack that led to its temporary suspension. Several exchanges of Bitcoin and Ethereum (a Blockchain-based distributed computing platform) have also suffered from DDoS attacks and DNS attacks frequently, hampering the service availability to the users.

Often times these attacks are launched on Blockchain-applications due to their popularity or the capital involved in their system. For instance, with Bitcoin, such attacks can cause devaluation of the cryptocurrency, loss of mining rewards, or even closure of cryptocurrency exchanges [30]. Bitcoin's Blockchain is also targeted with dust or spam transactions to delay the processing of legitimate transactions. In May, August, and November 2017, memory pools of Bitcoin were flooded with dust transactions to create stalls and delays in transaction verification, and to increase Bitcoin mining fee [34]. The transaction stall in November 2017, for example, resulted in a payment delay of $700 million USD worth bitcoins [50]. Often the intent of such attacks is to motivate the Bitcoin users to move to other cryptocurrencies with faster transaction processing time.

Due to a publicly verifiable nature, Blockchain-based cryptocurrencies are vulnerable to several fraudulent activities. Mt. Gox, a Bitcoin currency exchange in Japan, was attacked by two malicious users who stole $460 million USD worth of bitcoins [51]. The attackers gathered useful information from Bitcoin's Blockchain and engineered a fake transaction ripple

Manuscript received April 4, 2019; revised September 16, 2019, November 28, 2019, and January 27, 2020; accepted February 10, 2020. Date of publication March 2, 2020; date of current version August 21, 2020. This work was supported in part by Air Force Material Command under Award FA8750-16-0301, and in part by Global Research Lab Program of the National Research Foundation under Grant NRF-2016K1A1A2912757 and Grant NRF-2017R1A4A1015675. *(Corresponding author: David Mohaisen.)*

Muhammad Saad, Jeffrey Spaulding, and David Mohaisen are with the Department of Computer Science, University of Central Florida, Orlando, FL 32816 USA (e-mail: mohaisen@ieee.org).

Laurent Njilla is with the Department of Computer Science, Air Force Research Laboratory, Rome, NY 13441 USA.

Charles Kamhoua is with the Department of Computer Science, Army Research Laboratory, Adelphi, MD 20783 USA.

Sachin Shetty is with the Department of Modeling and Simulation, Old Dominion University, Norfolk, VA 23529 USA.

DaeHun Nyang is with Ewha Womans University, Seoul 03760, South Korea.

Digital Object Identifier 10.1109/COMST.2020.2975999

to increase the market price. Due to such activities, Mt. Gox suffered a heavy loss and eventually became bankrupt.

In May and June 2018, five Blockchain-based cryptocurrencies; namely, Monacoin, Bitcoin Gold, Zencash, Verge, and Litecoin Cash, were targeted by a 51% attack [52], leading to a loss of $5 million USD. The attackers in each cryptocurrency were able to gain more than 51% of the networks' hash rate which was used to rearrange transactions and prevent other miners from computing blocks. As a result, they were able to gain control over the Blockchain and perform double-spending on valuable transactions [53], [54].

The security of Blockchain systems is important for their acceptability by potential users [55]. Since the use of Blockchains is expanding beyond cryptocurrencies and smart contracts, including Internet of Things (IoT), information-centric networking (ICN), Vehicular Ad-hoc Networks (VANET), and edge computing [10], [56], [57], therefore, it is pertinent to systematically investigate their attack surface for potential threat assessment. These large-scale systems hold sensitive data and information related to users. A potential exploit in the Blockchain system can lead to unforeseen attacks which may compromise the overall security and privacy of the system. Understanding the threats in Blockchain systems in general is a first step towards realizing the potential of applications built on it. To this end, this work is dedicated to an in-depth look at the attack surface of Blockchain.

We envision that Blockchain will be used in many applications, and we report on the attacks that could compromise those applications. Namely, the taxonomy of Blockchain attacks in this paper is classified into three broad categories: 1) attacks associated with the mathematical techniques used for creating the ledger (*e.g.*, Blockchain forks, stale blocks, orphaned blocks, etc.), 2) attacks associated with the peer-to-peer architecture used in the Blockchain system, (*e.g.*, selfish mining, the 51% attack, consensus delay, DDoS attack, DNS attacks, Fork After Withholding (FAW) Attacks, etc.), and 3) attacks associated with the application context that uses the Blockchain technology (*e.g.*, Blockchain ingestion, double-spending, wallet theft [58], etc.). In this paper, we mainly focus on the attack surface of public Blockchains. Public Blockchains are suitable for applications that provide open access to system resources while preserving user anonymity. These attributes are well suited for a system that has a weak trust model and high provenance assurance requirements. The weak trust model results from an application's tolerance for adversaries who can game the system while staying anonymous. On the other hand, high provenance means that anyone can access the publicly available resources to transparently audit data. For instance, in Bitcoin and Ethereum, any user can join the network by running an Ethereum software client on their machine and participating in transaction processing. Since the Blockchain is public, anyone outside the system can validate the authenticity of transactions and blocks. Therefore, public Blockchains remain a dominant component among Blockchain applications as shown by the popularity of Bitcoin and Ethereum. On the other hand, the weak trust model exposes public Blockchains to a wide variety of attacks, allowing adversaries to easily compromise the system [59]. Therefore, while the public Blockchains are useful for an open access system, they are not suitable for closed environments where the weak trust model creates attack opportunities.

To address the shortcomings of public Blockchains and reduce the attack opportunities, private Blockchains are used [60]. In private Blockchains, the access to system resources is restricted to a chosen set of peers [61], [62]. These peers are screened prior to their induction in the application. Since the information about peers is known, their identities can be tied (or attributed) to their behavior in order to prevent attacks. Although private Blockchains still act as agents of trust in permissioned settings, they are not significantly exposed to adversarial attacks due to a stronger trust model. Since the aim of this work is to explore and understand the attack surface of Blockchains, it is natural to focus more on the public Blockchains. However, wherever necessary, we will also discuss the security and performance of private Blockchains.

Contributions: In summary, we make the following contributions in this paper. (1) We survey the possible attacks related to the design constructs of Blockchains, the peer-to-peer architecture, and the application-oriented use of Blockchains. (2) We explore the origins of these attacks and the ways in which they affect Blockchain applications and their users. (3) We also show the relationship between a sequence of attacks to outline how one attack can facilitate the possibility of other attacks. Understanding these links can help devise a common cure that can fix multiple problems at the same time. (4) Building on top of the prior work [45], [46], [59], for each attack class, we also explore the possible defenses to harden the security of Blockchains. Since many attacks related to a specific class have a common defense or remedy, while others remain as open problems, we discuss combined countermeasures for each class. Moreover, by highlighting the lessons learned, we also provide future research directions towards a more systematic treatment of the Blockchain attack surface. (5) In Table I and Table II, we provide an overview of the Blockchain attack surface. We ascribe various attacks to attack classes with their implications. (6) In Table III and Figure 2, we provide the countermeasures of these attacks.

Organization: The rest of the paper is organized as follows. In Section II, we provide the motivation of this work. In Section III, we give an overview of Blockchain and its operations. In Section IV, we review the design constructs of Blockchain that enable various attacks, such as Blockchain forks, stale and orphaned blocks. In Section V, we look into the features of distributed networks that create possibilities for the 51% attack, DNS attacks, DDoS attacks, consensus delays, etc. We further describe the aspects of peer-to-peer architecture that enable the possibility of their potential misuse in Blockchain applications. In Section VI, we outline the application-specific vulnerabilities found in Blockchain and assess the threats that they face. That is followed by discussion and open directions in Section VII, and the concluding remarks in Section VIII.

TABLE I
ATTACK VECTORS RELATED TO THE ATTACK CLASS IN BLOCKCHAIN SYSTEMS. WE ALSO SHOW, BY REFERENCING TO THE PRIOR WORK, HOW EACH ATTACK AFFECTS THE ENTITIES INVOLVED WITH BLOCKCHAIN SYSTEMS. FOR INSTANCE, ORPHANED BLOCKS AFFECT THE BLOCKCHAIN, THE MINERS, AND THE MINING POOLS

	Attacks	Blockchain	Miners	Mining Pools	Exchanges	Application	Users
Blockchain Structure	Forks [27]	✓					
	Orphaned blocks [28]	✓	✓	✓			
Peer-to-Peer System	DNS hijacks [29]		✓	✓	✓		✓
	BGP hijacks [30]		✓	✓			✓
	Eclipse attack [31]		✓				✓
	Majority attack [32]	✓	✓			✓	
	Selfish mining [33]	✓	✓	✓			
	DDoS attacks [34]	✓	✓	✓			
	Consensus Delay [35]		✓	✓			✓
	Block Withholding [35]		✓	✓			
	Timejacking attacks [36]		✓	✓		✓	
	Finney attacks [37]		✓	✓			✓
Blockchain Application	Blockchain Ingestion [38]	✓					
	Wallet theft [39]				✓	✓	✓
	Double-spending [40]	✓					✓
	Cryptojacking [41]					✓	✓
	Smart contract DoS [42]	✓				✓	✓
	≈ Reentracy attacks [43]					✓	✓
	≈ Overflow attacks [43]					✓	✓
	≈ Replay attacks [42]	✓		✓		✓	✓
	≈ Short address attacks [43]					✓	
	≈ Balance attacks [42]					✓	✓

TABLE II
IMPLICATIONS OF EACH ATTACK ON THE BLOCKCHAIN SYSTEM IN THE LIGHT OF THE PRIOR WORK. FOR INSTANCE, FORKS CAN LEAD TO CHAIN SPLITTING AND REVENUE LOSS. AS A RESULT OF A FORK, ONE AMONG THE CANDIDATE CHAINS IS SELECTED BY THE NETWORK WHILE THE OTHERS ARE INVALIDATED. THIS LEADS TO INVALIDATION OF TRANSACTION AND REVENUE LOSS TO MINERS

	Attacks	Chain Splitting	Revenue Loss	Partitioning	Malicious Mining	Delay	Info Loss	Theft
Blockchain Splitting	Forks [27]	✓	✓					
	Orphaned Blocks [28]		✓					
P2P System	DNS hijacks [29]		✓	✓				✓
	BGP hijacks [30]		✓	✓				✓
	Eclipse attacks [31]			✓				
	Majority attacks [32]	✓	✓		✓			
	Selfish mining [33]		✓		✓			
	DDoS attacks [34]				✓			✓
	Consensus Delay [35]					✓	✓	
	Block Withholding [35]		✓		✓			
	Timejacking attacks [36]	✓	✓		✓	✓		
	Finney attacks [37]		✓					
Blockchain Application	Blockchain Ingestion [38]						✓	
	Wallet theft [39]		✓					✓
	Double-spending [40]							
	Cryptojacking [41]	✓			✓			✓
	Smart contract DoS [42]		✓			✓		✓
	≈ Reentracy attacks [43]		✓					✓
	≈ Overflow attacks [43]							✓
	≈ Replay attacks [42]		✓				✓	
	≈ Short address attacks [43]		✓					✓
	≈ Balance attacks [42]		✓					✓

II. MOTIVATION AND TARGET AUDIENCE

The motivation of this work is to derive attention towards the security vulnerabilities of Blockchain systems via a systematic and comprehensive study. Recently, Blockchain technology has gained significant attention and its applications are being explored in various domains [63], [64]. Blockchains are capable of augmenting trust and provide provenance in distributed systems. While acknowledging their merits [65]–[67], we argue that it is important to understand their shortcomings, particularly related to security, as evident by the large security surface. To that end, our work is an effort to highlight potential vulnerabilities in Blockchains, with an emphasis on popular public Blockchain applications. We systematically analyze various attack vectors and study their relationships. Alongside, we also survey countermeasures and defenses to the various attack surface elements, and provide future research directions.

Since various research and technology sections are interested in using Blockchains, it is intuitive to explore a deeper understanding of Blockchains' attack surface to establish foundations for their security. For instance, using public

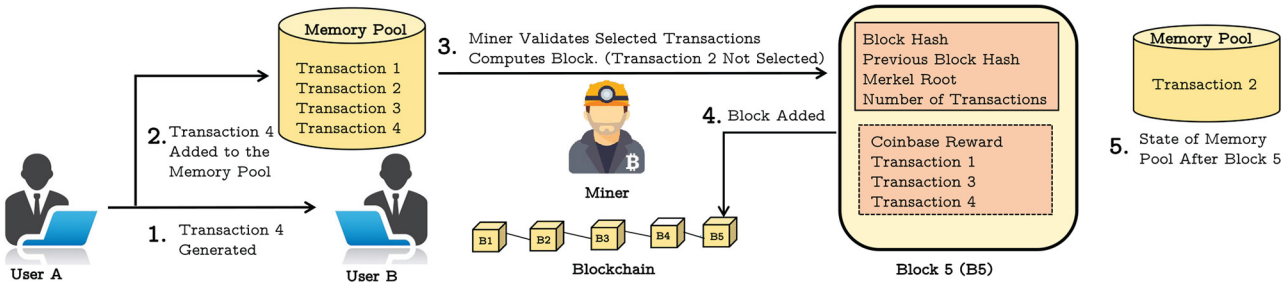


Fig. 1. Transaction life-cycle in a PoW-based cryptocurrency. User A generates a transaction for user B. The transaction is stored in the memory pool along with other unconfirmed transactions. Miner validates transactions from memory pool, and computes a block. A valid block is added to the Blockchain.

TABLE III
COUNTERMEASURES AND THEIR EFFECTIVENESS RELATED TO THE ATTACKS SURFACE OF BLOCKCHAINS. HERE, ○, ●, ◑DENOTE OPEN PROBLEM, FEASIBLE SOLUTIONS, AND INFEASIBLE SOLUTIONS

	ATTACKS	COUNTERMEASURES	EFFECTIVE?
Blockchain Structure	Forks	Joint consensus [48]	○
	Orphans	Increase block time [77]	●
Peer-to-Peer System	DNS hijacks	Routing-awareness [30], [78]	○
	BGP hijacks	Routing-awareness [30], [78]	○
	Eclipse attacks	Peer monitoring [79]	○
	Majority attacks	Two-phased proof-of-work [32]	◑
	Selfish mining	Time-stamping blocks [80]–[83]	○
	DDoS attacks	Increase block size [34]	○
	Consensus Delay	Peer monitoring [79]	○
	Block Withholding	Enforce PoW submission [83]	◑
	Timejacking attacks	Synchronized clocking	○
	Finney attacks	Increase block reward [37]	◑
Blockchain Application	Blockchain Ingestion	Encrypted Blockchains [84]	◑
	Wallet theft	Backups, wallet insurance [39]	●
	Double-spending	OTS schemes [85]	◑
	Cryptojacking	Mineguard [41]	●
	Smart contract DoS	Patch EVM [42]	○
	≈ reentracy attacks	Patch EVM [43]	○
	≈ replay attacks	Secure programming [43]	●
	≈ overflow attacks	Patch EVM [42]	○
	≈ short address attacks	Patch EVM [43]	○
	≈ balance attacks	Secure programming [42]	●

Blockchains in the financial sector may prevent fraud and data tampering, by simply utilizing Blockchain properties, although that also may expose sensitive information of financial transactions to adversaries. Similarly, organizations that are exploring Blockchain-based smart systems [35], [68], while might benefit immensely in addressing functional requirements, need to be aware of the programming languages' constraints and shortcomings, as well as compilation bugs that may lead to data breach and critical assets loss. For this research-driven efforts, we believe our work has the potential to offer future directions toward designing more secure and robust Blockchain solutions that may overcome some of those challenges as outlined in the rest of this survey. Some of these challenges include constructing new consensus algorithms that are secure, scalable, and energy efficient [69]. Additionally, they must also have the capability to prevent race conditions that lead to attacks such as selfish mining, double-spending, majority attacks, and orphaned blocks [70], [71]. To facilitate the process of addressing those challenges, we supplement our work by surveying the existing countermeasures proposed in the literature. These countermeasures can be used as building blocks for more secure and robust solutions. The work surveyed for paper includes the prior research efforts towards the study of Blockchain applications and their security vulnerabilities. In doing so, we also consulted the Comprehensive Academic Bitcoin Research Archive (CABRA) [72], a list of over 900 research papers that keep track of ongoing research in Blockchain systems. CABRA is influenced by a chronological list of Blockchain papers maintained by Scott [73]. From these useful repositories, we curated a list of relevant papers for this study, as starting pointers.

There have been several attempts at understanding the attack surface of Blockchains by various surveys, which we contrast to our work in the following. Towards analyzing the attack surface of Blockchains, Li *et al.* [74] surveyed various security aspects of Blockchains by studying popular Blockchain

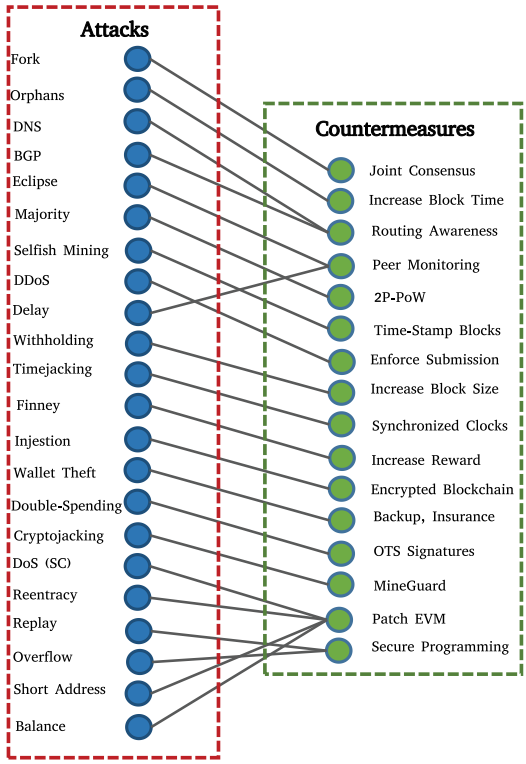


Fig. 2. Relationship among various attacks on blockchains along with their countermeasures. Some attacks have common countermeasures which provides future directions towards a common cure.

applications including Bitcoin, Ethereum, and Monero. They evaluated the robustness of Blockchain applications against popular attacks and the risk factor associated with each attack. Although comprehensive in the survey of attacks, their work, however, does not look into countermeasures. Conti *et al.* [59] surveyed security and privacy of Bitcoin. Although Bitcoin is a motivating example to analyze the attack surface of Blockchains in general, however, Blockchains have evolved beyond Bitcoin and their attack surface has increased accordingly. Furthermore, their work does not cover new attacks related to Blockchain applications such as cryptojacking, among others. Salman *et al.* [75], explored the utilization of the Blockchain technology in providing distributed security services. They mainly focused on the use of Blockchains to provide services including authentication, confidentiality, provenance, and integrity assurance. In contrast, our work is dedicated to the abuse of Blockchains and their applications. Anderson *et al.* [76], looked into the use of new consensus schemes in emerging Blockchain applications such as Namecoin and Peercoin. They also surveyed various security features of these applications with an emphasis on smart contracts. In a similar context, Atzei *et al.* [46] also explored various attacks limited to Ethereum smart contracts. Compared to the existing literature, our work goes beyond the state-of-the-art in outlining new attacks, their implications, their defenses, and relevant case studies.

Kiran and Stanett [86] performed risk analysis on Bitcoin, spanning its vulnerabilities and attack surface. They also explored the risk factors associated with the economics of Bitcoin and cryptocurrency market in general, including deflation, volatility, and complicity. Becker *et al.* [87], outlined challenges and security risks associated with PoW-based Blockchain applications. Moubarak *et al.* [88] explored the security challenges of three major Blockchain applications, namely Bitcoin, Ethereum, and Hyperledger. However, their work is more directed towards the application attacks and did not consider the attacks related to the Blockchain's cryptographic constructs and P2P fabric.

Carlsten *et al.* [89], analyzed the security features of Bitcoin in the absence of Block rewards. Since the number of coins in Bitcoin are deterministic and the coinbase rewards will eventually end when all the coins are mined, the stake of miners in the system will take a paradigm shift which might influence the security properties of Bitcoin. As such, there is an implicit belief that this might not change the attack surface of Bitcoin. However, in [89], the authors outline the limitations of this belief and present new attack avenues and their effects.

In summary, the target audience of this work include both academics, who are interested in understanding the attack landscape of Blockchains, as well as practitioners, who might be interested in understanding the existing solutions to those attacks, to utilize as building blocks, and both benefiting from a systematic analysis of the Blockchain attack surface.

III. Overview of Blockchain and Its Operations

Conceptually, Blockchain can be viewed as a repository of data that is tamper-evident due to its replication over all nodes in a peer-to-peer system. Transactions represent the events that drive the Blockchain application. Blockchain applications use various consensus algorithms for trust among peers over the state of the ledger. Moreover, the consensus algorithms ensure a consistent and transparent view of the Blockchain, thereby resolving conflicts and forks. This is, no block is added to the Blockchain, until it fulfills the conditions outlined by the consensus algorithm. Moreover, each algorithm has unique functional and operational properties that drive the consensus over the Blockchain.

For an accurate characterization of the Blockchain attack surface, it is critical to make a distinction between the operation of various Blockchain applications. Take, for example, four different Blockchain applications, namely Bitcoin, Litecoin, Ethereum, and Peercoin. Bitcoin, Litecoin, and Peercoin are blockchain-based cryptocurrencies in which the ownership of tokens is exchanged as transactions. The scripting languages in these cryptocurrencies provide limited features for users, restricting their usability to transaction exchange only. Addressing these limitations, Ethereum expands the functionality of transaction exchange by pegging a Turing-complete smart contract with the Blockchain. Smart contracts can be programmed with rules to specify conditions under which the transaction exchange can occur. The Blockchain ledger only accepts those transactions which conform to the rules of the smart contract. Common among all these Blockchain applications is the Blockchain data structure, which we later show in Figure 4, which consists of an append-only ledger linked with one-way hash functions. Moreover, all

Blockchain applications have a common underlying network architecture in which the system entities follow a distributed P2P architecture. In terms of differences, Bitcoin, Litecoin, and Ethereum use PoW consensus protocols, while Peercoin uses proof-of-stake (PoS). At the application level, Bitcoin, Litecoin, and Peercoin only support transaction exchange while Ethereum also supports autonomous smart contracts. Keeping such distinctions in mind, we have broadly classified the Blockchain attack surface in terms of the Blockchain data structure, the P2P network architecture, and the application-specific use of Blockchains. While describing various attacks, we use the relevant and well-known Blockchain application as a representative example. For PoW-based cryptocurrencies, we use Bitcoin, for PoS-based cryptocurrencies, we use Peercoin, and for the smart contracts, we use Ethereum as representative examples. Please note that these attacks can be generalized to other applications that follow similar protocols and policies.

Blockchains are suitable in a decentralized and distributed environment, that exhibits a weak trust model. Due to the weak trust model, there is a natural tendency of "conflicts" among the application peers over the *correctness* and the *order* of data in the ledger. To resolve such conflicts, Blockchain applications use various consensus algorithms that ensure a consistent view among peers. We briefly discuss the popular consensus algorithms along with the fundamental cryptographic primitives that are used in Blockchains.

A. Consensus Algorithms

Some of the notable consensus algorithms used in Blockchains include proof-of-work (PoW), proof-of-stake (PoS), proof-of-activity (PoA), proof-of-capacity (PoC), proof-of-burn (PoB), proof-of-knowledge (PoK), and the practical Byzantine fault tolerance (PBFT) [90]–[94]. The most popular consensus algorithm widely used in Blockchains is PoW, followed by PoS and PBFT [95], [96]. We discuss them in the following.

1) Proof-of-Work: In PoW Blockchains, peers in the network try to solve a computationally expensive mathematical challenge. For instance, the challenge in Bitcoin is to come up with a *nonce* that when hashed with block data produces a hash value that is less than a target threshold set by the system. All peers in the system use their computational power to solve the mathematical challenge. The peer who comes up with the solution wins the block race and mines a new block. Once a block is broadcast to the network, each peer verifies the solution and appends the block to his Blockchain. The probability of winning a block race is proportional to the computational power of participants. At the same time, there is a time restriction on the block mining [97], [98]. In Bitcoin, the block time is set to 10 minutes. In other words, the network expects a new solution to the block puzzle after every 10 minutes. However, as the computational power increases, the chance of discovering a new block under 10 minutes increases. To address that, the network dynamically adjusts the difficulty of the challenge according to the change in the computational power of the miners. Oftentimes, more than one miner can come up with a valid solution leading to Blockchain forks and stale and orphaned blocks, which we discuss in Section IV.

In Figure 1, we illustrate the transaction life-cycle in a PoW-based Blockchain application. User A (sender) generates a transaction for user B (receiver). The transaction is broadcast to the entire peer-to-peer network where it is temporarily stored in a transaction repository known as the memory pool (mempool). In a peer-to-peer network, the mempool is a space allocated in the RAM of a full node that stores and relays transactions to other peers. To maintain the state of the Blockchain, there are special nodes in the network known as the miners or verifiers, responsible for verifying transactions and computing a block. The miners query the mempool and select the transactions of their choice to put into blocks. Usually transactions pay a mining fee which can be viewed as an incentive given to the miners to mine the transaction. Naturally, miners give priority to the transactions that pay higher mining fee. Transactions that are not selected by the miners, stay in the mempool until some other miner selects them for a new block. Transactions that do not get mined for a long time, eventually get discarded.

An important aspect of Blockchain is to provide a consistent ordering in the chain space. Informally speaking, when a group of users is presented with two or more chains, they must agree upon one chain and its contents. That way, each user will have a non-conflicting and consistent view of the system's state. A PoW-based Blockchain system ensures this by introducing the *longest prefix* property as a primary bound to select the *best chain* in the chain space. For example consider two chains $\mathcal{C}_1$ and $\mathcal{C}_2$, where $\text{len}(\mathcal{C}_1) > \text{len}(\mathcal{C}_2)$. When a user is presented with two chains, he must give precedence $\mathcal{C}_1$ due to greater length. The length signifies the amount of "work" in the PoW, and longer chain has more work behind it.

In Blockchain systems, it is possible that in a given round, two blocks are produced which extend the same chain tip. Such a situation is called a fork Section IV-A. However, the fork can be resolved by the next block which will extend one of the two chains. As one chain becomes longer, respecting the longer prefix property, the network switches to it and diffuses the other chain. The losing chain creates another problem in Blockchains, known as stale or orphaned blocks, which we later discuss in Section IV-B.

2) Proof-of-Stake: The second most popular consensus algorithm in public Blockchains is PoS [99], [100]. PoS was introduced to address the energy inefficiency of PoW. In PoS, the mining power of a user is determined by the total number of coins he owns. For each new block, an auction is carried out to select the candidate miner. Users place a bid on the block and the one with the highest bid is selected as a miner. Therefore, in contrast to PoW, the hashing power is replaced by the volume of assets owned by the user. The more coins a users owns, the higher his chances of winning the block race. The replace of energy intensive mining with stake-based mining, makes PoS energy efficient and secure against the majority attacks (Section V-B). Unlike PoW, in PoS, all the cryptocurrency tokens are released prior to creation of the genesis block [101]. Therefore, when a new block is mined, it does not introduce new coins in the system. However, miners are rewarded with transaction fee for their contributions.

TABLE IV
AN OVERVIEW OF POPULAR CONSENSUS ALGORITHMS USED IN BLOCKCHAINS. NOTICE THAT PUBLIC AND PERMISSIONLESS BLOCKCHAINS USING POW, POS, AND DPOS HAVE HIGH SCALABILITY, LOW THROUGHPUT, AND HIGH CONFIRMATION TIMES. IN CONTRAST, PERMISSIONED BLOCKCHAINS USING PBFT AND RAFT HAVE LOW SCALABILITY, LOW CONFIRMATION TIME, AND HIGH THROUGHPUT

Properties	**PoW**	**PoS**	**DPoS**	**PBFT**	**RAFT**
Blockchain Type	Permissionless	Permissionless	Permissionless	Permissioned	Permissioned
Participation Cost	Yes	Yes	Yes	No	No
Scalability	High	High	High	Low	Low
Throughput	<10	<1,000	<1,000	<10,000	>10,000
Byzantine Fault Tolerance	50%	50%	50%	33%	–
Crash Fault Tolerance	50%	50%	50%	33%	50%
Confirmation Time	>100s	<100s	<100s	<10s	<10s

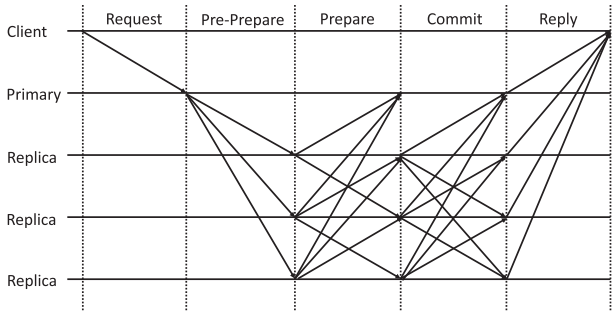


Fig. 3. Overview of PBFT protocol. The client issues a transaction to the primary replica. The primary replica then forwards it to other replicas who jointly execute a four-phased protocol and approve the transaction. Assuming f faulty replicas, the primary would require confirmation from at least $3f + 1$ replicas. PBFT is employed in permissioned and private blockchains.

3) PBFT: The third most popular Blockchain consensus protocol is called the practical byzantine fault tolerance (PBFT) [102], [103] protocol. PBFT is widely used in private and permissioned Blockchains, where the network has a stronger trust model compared to PoS and PoW. In PBFT Blockchains, the system is transposed into a group of active and passive replicas. Among the active replicas, a primary replica is selected who receives transactions from a client and sends them to the active replicas for execution. The process of execution is carried out in four stages, namely, pre-prepare, prepare, commit, and reply stage.

In Figure 3, we show the transaction verification process in a PBFT Blockchain. Notice that compared to PoW and PoS, PBFT has a higher message complexity.

In Table IV, we compare the popular consensus algorithms used in the Blockchain applications. Notice that permissionless Blockchains have low throughput and high confirmation time. Bitcoin has transaction throughput of 3–7 transactions per second. In contrast, permissioned Blockchains have a high throughput and low confirmation time. In terms of security, PBFT has low fault tolerance ($\approx$ 33%) compared to PoW and PoS ($\approx$ 50%). However, since permissioned Blockchains have a stronger trust model, therefore, they are less vulnerable to adversarial attacks. It can also be observed in Table IV, public Blockchains are more scalable than private Blockchains [104], [105]. This can be attributed to the message complexity involved in the transaction verification and the tolerance for Byzantine nodes. Since PBFT has high message complexity and low Byzantine fault tolerance, therefore, it cannot scale well beyond few hundred nodes. Therefore, each consensus scheme has its own benefits and limitations. Therefore, depending upon the application model, a consensus scheme can be selected to meet the requirements.

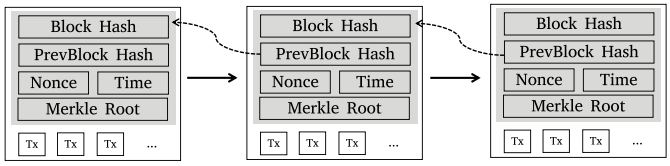


Fig. 4. Data structure of a PoW-based Blockchain. Note that the previous block hash goes into the header of the next block. Therefore, if an attacker tampers with data of a single block, he will be required to change data in all subsequent blocks and find the correct nonce that meets the consensus requirements. Since this is infeasible in practice, therefore Blockchains are considered tamper-proof. Most common Blockchain applications follow a similar data structure with minor changes. For example, in non-PoW-based Blockchains, the nonce is removed.

B. Blockchain Structure

While consensus schemes may vary, the cryptographic constructs are fundamentally the same across applications [106], [107]. Each block in a Blockchain consists of a header and a payload. The header includes primary information, e.g., the hash of the previous block, the merkle root, and the block timestamp. The hash pointer connects each block to the previous block, thereby forming a chain. Since hash functions are one-way and are collision resistant, Blockchain benefits from their properties to become immutable and tamper-proof [108], [109]. In Figure 4, we illustrate this model of Blockchain, where blocks are linked through hash functions.

In a Blockchain application, all nodes are connected in a peer-to-peer architecture. This means that they use the gossip protocol to communicate information, including transactions and blocks. Ideally, each peer is expected to maintain a copy of Blockchain. However, due to the append-only model, the growing size of Blockchain can put space constraint at the node. To address that, various Blockchain applications allow the segmentation of nodes into full nodes and lightweight nodes. The full nodes maintain a complete copy of Blockchain and participate in transaction and block propagation. On the other hand, the lightweight nodes only keep the block header for the verification of a newly published block.

Transaction Generation and Verification: In notable Blockchain applications such as cryptocurrencies, a transaction is a *state update* that reassigns the ownership of a value to a new address. In Bitcoin, for example, if a user transfers 1 bitcoin to another user, he generates a transaction that subtracts 1 bitcoin from the sender's wallet and adds it the recipient's wallet. As such, a transaction must have 1) the sender and receiver's addresses, 2) a verifiable proof that the sender holds 1 bitcoin, 3) an encryption scheme to ensure the secure transfer. Bitcoin uses asymmetric cryptography to provide these features. In particular, Bitcoin uses "Elliptic Curve Digital Signature Algorithm" (ECDSA), to generate private keys and the corresponding public keys for a wallet. These keys are stored in a *wallet.dat* files of Bitcoin Core. The public key is used to create the public address of a sender or a receiver. The private key is used to create digital signatures, which validate the sender's ownership. A typical Bitcoin transaction has an *input*, which is a digital signature that unlocks funds from the previous transaction, and an *output*, which is the hash of the public key of the recipient. Since the private key is known only to the sender, therefore, a verifiable digital signature proves that the sender's ownership.

In cryptocurrencies, the spendable balance is stored in the user's wallet. As such, two models are popularly used to maintain the balance, namely the "Unspent Transaction Output" (UTXO) model and Account/Balance model [110]. The UTXO model is used in Bitcoin in which a new transaction consumes the output of a previous transaction to generate new outputs. Generating two transactions from the same parent transaction is known as double-spending or equivocation which we later discuss in Section VI-B. A user's wallet maintains a list of unspent transactions generated for all the accounts owned by the user. The total balance in the wallet is the sum of all the unspent transactions. The Account/Balance model maintains the balance of each account as a global state. When a user generates a transaction, the account balance is checked to ensure that it is less than or equal to the amount specified in the new transaction. The Account/Balance model is used in Ethereum. For an intuitive analogy, the UTXO model can be perceived as the exchange of paper currency. A user needs to spend previous currency bills (UTXO's) in order to make a purchase. The recipient can later use those currency bills for further purchases. On the other hand, the Account/Balance model is similar to using a Debit card. When a user makes a purchase, the bank checks if the account holds a sufficient balance. After the purchase, the amount is simply deducted from the spender's account and added to the recipient's account.

As stated earlier, several attacks on Blockchain technology are related to the constructs of the Blockchain itself, the behavior of certain miners, and the peer-to-peer architecture it is built upon. In the subsequent sections, we explore the possible attacks associated with the Blockchain structure, attacks associated with the peer-to-peer architecture used in the Blockchain system, and attacks associated with the application services that use Blockchain technology (i.e., Bitcoin or Ethereum). We also supplement each section with possible countermeasures that have been proposed by researchers to address those attacks.

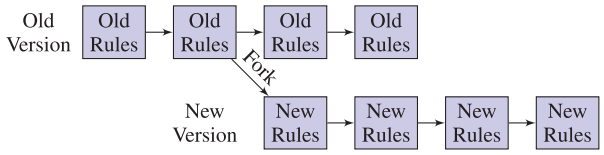


Fig. 5. Hard Fork resulting from set of peers following conflicting rules due to different client software versions. Hard forks can be irreversible at times and may lead to a permanent split in the Blockchain application.

IV. Blockchain Structure Attacks

In this section, we look at the attacks related to the design constructs of the Blockchain. These attacks emerge from the potential vulnerabilities of the Blockchain structures and as such, they can compromise any Blockchain-based application.

A. Blockchain Forks

A fork represents a condition in which nodes in the network have diverging views about that state of the Blockchain persisting over long periods of time or even indefinitely. These forks can be created unintentionally through protocol malfunctions or incompatibilities in client software upgrades. Forks can also be caused by malicious intents such as implanting "Sybil nodes" that follow conflicting validation rules or by carrying out "selfish mining" in race conditions as discussed further in Section V-A. Another form of fork occurs when users of a Blockchain application create a child application from the parent application. For example, in 2017, a group of Bitcoin developers decided to increase the block size limit from 1MB to 8MB by developing a new Bitcoin client that was capable of accepting 8MB blocks. However, their proposal was not accepted by the majority of users, therefore, they created a hard fork on Bitcoin and released a new cryptocurrency called Bitcoin Cash. Bitcoin Cash was the child application of the parent Bitcoin, with new rules and regulations. Therefore, forks can also be created to launch a new application.

Intentional forks can either be soft or hard, the latter of which occurs when new blocks that the network accepts appear invalid to pre-fork nodes. Soft forks, however, occur when some blocks appear invalid to post-fork nodes. In either case, a Blockchain fork represents an inconsistent state that can be exploited by adversaries to cause confusion, fraudulent transactions, and distrust within network [111].

Figure 5 illustrates a hard fork example that results from peers following conflicting rules about the state of Blockchain. Such hard forks may lead to a split in cryptocurrency. A major hard fork on Bitcoin occurred during August 2017, which led to the creation of Bitcoin Cash [112]. Another hard fork on Bitcoin occurred during October 2017, when Bitcoin Gold [113] was created. Some other notable forks in Bitcoin include Bitcoin Classic, Bitcoin XT, and Bitcoin Unlimited. However, due to insufficient user-base and miners, they could not succeed as a separate cryptocurrency.

When hackers stole more than one third of the total digital cash owned by "The DAO" [48], Ethereum used a hard fork to roll back transactions and retrieve millions of dollars' worth of ether (the "fuel" for the Ethereum network). However, this required consensus by the majority of nodes in

TIMELINE 1: *Major Bitcoin Forks*

Jan 3, 2009	Bitcoin genesis block established
Dec 27, 2014	Bitcoin XT
Jan 15, 2016	Bitcoin Unlimited
Feb 10, 2016	Bitcoin Classic
Aug 011, 2017	Bitcoin Cash
Aug 23, 2017	Segregated Witness (Segwit)
Nov 01, 2017	Bitcoin Gold
Nov 15, 2017	Segwit2x
Nov 28, 2017	Protest Fork
Dec 12, 2017	BitcoinX
Dec 31, 2017	Bitcoin Ore
Jan 24, 2018	Bitcoin Atom
Jan 30, 2018	Bitcoin Lite
Sept 30, 2018	Bitcoin Zero
Jan 07, 2019	Bitcoin Stash

Fig. 6. The timeline of major Bitcoin forks (2009-2019).

the network. In such a scenario, if a consensus delay happens due to a majority attack or a DDoS event, fraudulent activities become somewhat difficult to deal with and prolonged delays can ultimately cause devaluation of cryptocurrency. In November 2017, the second version of Segregated Witness (SegWit2x) hard fork was proposed in Bitcoin, which aimed to increase the block size to 2MB. However, due to lack of consensus by the majority, the planned hard fork was canceled. In Figure 6, we provide a list of major forks on Bitcoin. These forks resulted from a group of miners introducing new rules and a faction of peers switching to those rules. All these forks introduced a new version of Bitcoin. This is, we note that a fork may diminish if peers discontinue to follow the new rules and switch back to the old ones. For instance, this has been witnessed in SegWit fork. Initially, a faction of network peers switched to SegWit version of Bitcoin, however, when they moved back to the old version in a protest, the fork ended.

B. Stale Blocks and Orphaned Blocks

Two forms of inconsistencies can occur with the consensus process that can leave valid blocks out of the Blockchain. The first form is a "stale block", which is a block that was successfully mined but is not accepted in the current best Blockchain (i.e., the most-difficult-to-recreate chain). Stale blocks occur mostly in the public Blockchains due to race conditions. In race conditions, the miners actively try to find the next block, and it is possible that two or more miners can come up with

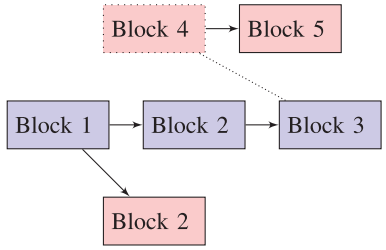


Fig. 7. Stale vs. orphaned blocks. Note that the stale block (block 2, bottom, and block 4) are valid but they are not part of the Blockchain. Orphaned block (block 5) does not have its parent block (block 4) in the Blockchain.

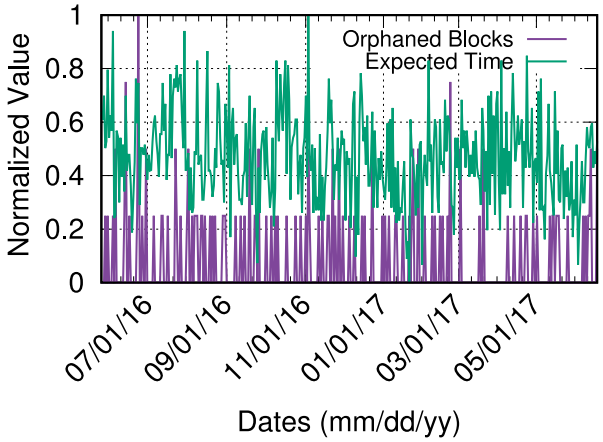
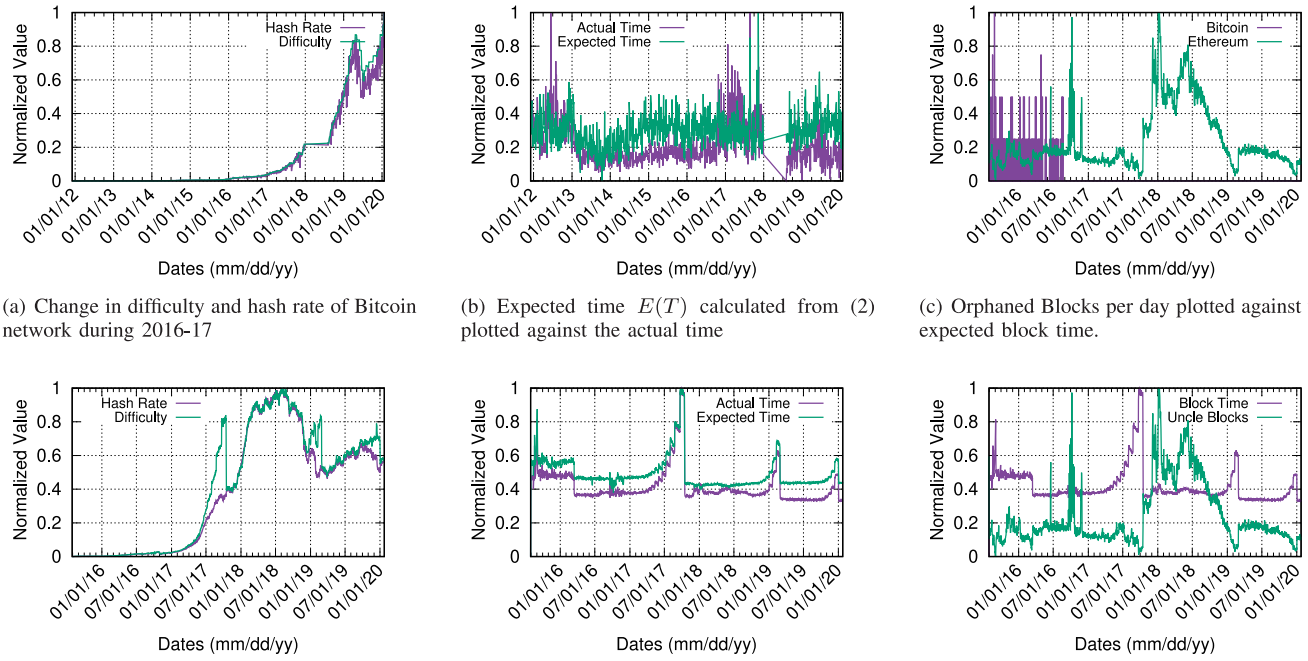


Fig. 8. Orphaned Blocks in Bitcoin and Uncle blocks in Ethereum during 2016–2017. Notice that in Bitcoin, the rate of orphaned blocks has reduced.

a valid solution. The network eventually accepts one of the winning blocks and discards the rest. As a result, the all other valid blocks unaccepted become stale blocks as they do not get attached to the main Blockchain. We will see in Section V-A that a form of Blockchain attack known as "selfish mining" can also lead to the creation of stale blocks in the network, which deprives an honest miner of its reward.

The other form of inconsistency is an "orphaned block": a block whose parent block's hash field points to an unauthentic block that is detached from the Blockchain [114]. These inconsistencies can be introduced by an attacker or caused by race conditions in the work of the miners. Stale blocks may be initially accepted by the majority of the network, but they can be rejected later when proof of a longer Blockchain (i.e., the current best) is received that does not include that block.

Figure 7 illustrates a chain where stale and orphaned blocks can be found. The first orphaned block in Bitcoin was found on March 18, 2015, and that was the beginning of a period in which most orphaned blocks were created. The trend reduced in 2016, and from June 2017 to the date of this paper, no orphaned block has been added to the list [115]. Orphaned blocks are more frequently found in cryptocurrencies where average block computation time is small. In Figure 8, we plot the number of orphaned blocks that occurred in Bitcoin and Ethereum from July 2016 to May 2018. In Ethereum, the orphaned blocks are called Uncle blocks. The data in the figure has been normalized using min-max normalization to scale the data in the range [0, 1]. The min-max scaling is conducted as $x_{scaled} = \frac{x_i - \min(x)}{\max(x) - \min(x)}$. Here x_{scaled} is the normalized



(a) Change in difficulty and hash rate of Bitcoin network during 2016-17

(b) Expected time $E(T)$ calculated from (2) plotted against the actual time

(c) Orphaned Blocks per day plotted against the expected block time.

(d) Change in difficulty and hash rate of Ethereum network during 2015-18.

(e) Expected time $E(T)$ calculated from (2) plotted against the actual time

(f) Uncle Blocks per day plotted against the expected block time.

Fig. 9. Effect of hash rate and difficulty on the rate of orphaned blocks in Bitcoin and uncle blocks in Ethereum. For Ethereum, notice that when the difficulty sharply decreases with constant hash rate around October 2017, the expected and the actual time of block computation decreases sharply. As a result, the number of Uncle blocks increases. The sharp decrease in the difficulty is associated to a byzantium fork that reduced block rewards per block.

value of $x_i, i \in \mathbb{N}$, and min(x) and max(x) are the minimum and maximum values of the entire vector. It can be observed from the figure that as of June 2017, no orphaned block has been found in Bitcoin. On the other hand, in Ethereum, Uncle blocks have increased since November 2017.

In cryptocurrencies such as Ethereum and Bitcoin, the difficulty is a measure of how long it takes to compute a block, which is defined by a target value set by the network [116]. Based on the hashing power, the *target* is adjusted to keep block time under a predefined range (10 minutes for Bitcoin and 12 seconds for Ethereum). The difficulty is recomputed based on the hashing power and the time taken by a series of previous blocks: if hashing power increases, the probability of finding a block under the expected time increases.

To adjust the probability, the difficulty is raised by increasing the target value. In (1), we show how the expected time to compute a block $E(T)$ varies with the difficulty D and hash rate of the network H_r. Here, $E(T)$ is measured in seconds, D is the number of hashes required to solve the current target, and H_r is measured in hashes/second, that a target device can produce over a given string. H_r is the aggregate hashing power of all the miners H_i for $i = 1, 2, \ldots, n$. In (2), we calculate the time T_b (seconds), it takes for a single miner in H_i to compute a block, given a fixed block time set by the network T_n. For Bitcoin and Ethereum, the average block computation time T_n is 600 seconds and 12 seconds, respectively.

$$H_r = \sum_{i=1}^{n} H_i, \quad E(T) = \frac{D}{H_r} \tag{1}$$

$$T_b = \frac{T_n \times H_r}{H_i} \quad T_b = \frac{T_n \times \sum_{i=1}^{n} H_i}{H_i} \tag{2}$$

From (1), it can be observed that when H_r remains constant and the difficulty D is reduced, the expected block time $E(T)$ decreases. Intuitively, lower $E(T)$ means that in a defined network time T_n, more blocks will be produced. However, in the Blockchain, only one block can be accepted. Such a situation will lead to more orphaned blocks in the system. In Figure 9, we plot difficulty, hash rate, block time and orphaned blocks (also called uncle blocks) in Ethereum. It can be noted in Figure 9(f) that as the expected block time (from (2)) decreases, the number of orphaned and uncle blocks increases. In Ethereum, this trend is high due to short block intervals which increase the possibility of block collision. Orphaned blocks may also occur due to unpredictable delays in block propagation. A valid block may not reach majority of the network peers due to network churns and propagation delays. In contrast, a competing block is able to easily propagate through the network and get accepted by the majority.

A recent work by Liu *et al.* [117] explored the effect of block size on the probability of producing orphaned blocks in a Blockchain system. In their analysis, they altered the block size and modeled the block propagation as a Poisson distribution. They noticed that the block propagation had a linear relationship with its propagation patterns and delays. Naturally, bigger blocks experienced more propagation delays and took longer time to reach other nodes. They concluded that larger block sizes increase the *orphaning probability*.

C. Vulnerabilities in Consensus Mechanism

1) Proof-of-Work: The most widely used consensus protocol in cryptocurrencies is PoW which serves as an evidence

TABLE V
EVOLUTION OF MINING HARDWARE. SINCE 2014, ONLY ASIC CHIPS, WITH UPGRADED VERSIONS, ARE BEING USED FOR MINING

Type	Model	Hash Rate (MH/s)	Year
CPU	Xeon E5530	7.14	2009
GPU	Radeon 5890	245	2010
GPU	Radeon 6990	800	2011
FPGA	Xilinx Spartan	245	2012
FPGA	Xilinx Spartan	850	2012
ASIC	ASIC 130nm	12K	2013
ASIC	ASIC 28nm	500k	2014
ASIC	ASIC 20nm	750k	2014
ASIC	ASIC S7	4730k	2017
ASIC	ASIC S9	14000k	2019

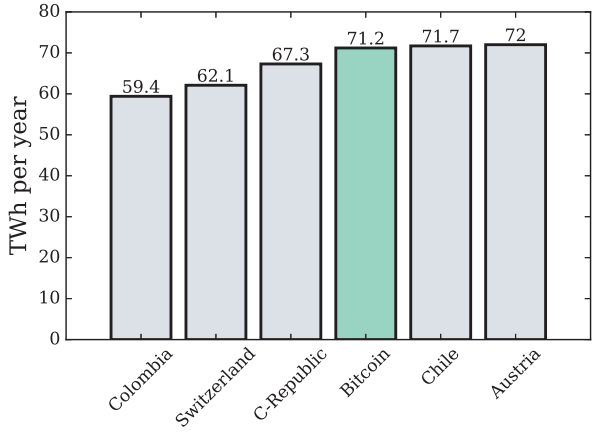


Fig. 10. Energy Profile of Bitcoin and other countries. C-Republic refers to Czech Republic. Note that Bitcoin's consumption is compare able to countries.

for the effort put behind the computation of a valid block. As outlined in (1), the effort for computation of a block can be characterized as the number of hashes required to meet the difficulty parameter D set by the network. As the aggregate hash power of the network H_r increases, the difficulty is raised to keep the standard block time T_n within a defined range (10 minutes for Bitcoin).

In Figure 9(a) and 9(d), we show the increase in difficulty and the aggregate hash rate of Bitcoin and Ethereum, respectively. Since mining in PoW is competitive, therefore, miners use sophisticated hardware with high hash rate to increase their winning chances. Among all PoW-based cryptocurrencies, Bitcoin has the maximum hash rate. In particular, and since 2010, miners in Bitcoin have switched from Central Processing Unit (CPU), to graphics processing unit (GPUs) in 2011, to Field Programmable Gate Array (FPGA) in 2012–13, and finally to Application Specific Integrated Circuit (ASIC) chips since 2014 to date [118]. We show this evolution of Bitcoin hardware, along with the hash rate, in Table V.

One of the key limitations of the PoW consensus protocol is its excessive use of energy [87]. Currently, Bitcoin and Ethereum mining pools use more than 71.12 Terawatt-hours and 4.2 Terawatt-hours (TWh) of electricity per-year, respectively [119]–[121]. Figure 10 provides a comparison of electricity consumption of Bitcoin with other countries, highlighting a massive usage of energy. Apart from the energy inefficiency, there are other issues with PoW including the centralization of hash rate among a few mining pools. This centralization can lead to vulnerabilities such as the majority attack and double-spending (discussed in Section V-B and Section VI-B).

2) PoS: (PoS) [122] allows applications to be more energy-efficient and raise the cost of a majority attack. PoS uses a stake-based deterministic approach to select a validator and to publish a new block [123]. The validator is chosen by a bidding process, and candidate validators make a bid of their stake. The stake is the balance owned by the candidate validator and is used to deter cheating in the system. The candidate with the highest bid is chosen to mine the next block and if he tries to trick the system with bogus transactions he risks losing his committed stake. The process is deterministic since a validator is chosen prior to each bidding. Therefore, blocks are published on their expected time without time deviations or delays. Moreover, to launch a majority attack on a PoS-based cryptocurrency, the attacker is required to acquire more than 50% of the cryptocurrency tokens [124]. While it is relatively easier to acquire 50% hash rate in PoW, it is difficult to obtain 50% coin. Compared to PoW, the cost for launching a majority attack in PoS application is relatively high, which makes the attack less feasible. Although PoS serves as a "green" mining alternative of PoW and raises the attack cost for the majority attacks, it has some major caveats that have prevented its widespread adoption by the Blockchain community. In PoS, a rich validator may keep on winning the bid for the next block to be validated, and accumulate the block reward. As such, the rich validators in the system gets richer for block confirmation, which makes PoS applications centralized around those validators. This challenges the fundamental premise of Blockchain technology as a decentralized system [125]. Moreover, unlike PoW, in which miners with limited resources may still have a chance of winning, small bidders in PoS are certain to lose the bid for each coming block.

3) PBFT: The major limitation of PBFT-based private Blockchains is their low fault tolerance. Each transaction requires approval from $3f + 1$ replicas, where f is the number of faulty replicas or Byzantine nodes. In comparison with PoW and PoS, where the network can withstand up to 50% malicious entities, PBFT can only tolerate 33% malicious replicas. Provided that PBFT already suffers from low scalability [126], a lower fault tolerance increases the opportunity for an adversary to place malicious replicas in the network. In PoW, the fault tolerance relies on the computation power of the adversary. If an adversary acquires 51% hash rate Section V-B, he will be able to produce a private chain with a double-spent transaction Section VI-B. Assuming a system of n nodes in a PoW system, with each node having an equal computational power, then the adversary would need to compromise $n/2$ ($\approx 50\%$) nodes to double-spend. Therefore, PoW provides a higher fault tolerance than PBFT.

Considering the features and shortcomings of existing consensus algorithms, there is a need for new consensus

mechanisms that are secure, scalable, and energy efficient. Currently, this remains an active research area, with some notable recent progress made in this direction [127]–[129].

D. Countering Blockchain Structure Attacks

Resolving soft forks in a Blockchain network is a relatively easy process. All peers in the network can come to a consensus about the true state of the Blockchain and resume activities from there. Resolving hard forks can be challenging because conflicting chains can be lengthy with transaction activities dating back to the time of the conflict. Although the stakes of rolling back from a hard fork are high, they can be resolved by the same principle of consensus as discussed earlier. As was the case with Ethereum, a hard fork was used to retrieve money for the investors after "The DAO" was attacked [48]. Ultimately, the process of solving a fork depends upon the agreement of peers in the network and their stake in the fork.

In Ethereum, uncle blocks are also rewarded and made part of the Blockchain. Recently, the number of orphaned blocks in Bitcoin has decreased due to the shift towards highly centralized mining networks and thus reducing the probability of orphaned blocks prevalent in decentralized mining networks. However centralized mining has other issues such as unfairness in the network and the 51% attack. The other solution to avoid stale or orphaned blocks involves dynamic adjustment of network's difficulty [77]. In Bitcoin, the difficulty is adjusted every two weeks (2016 blocks). In the meantime, if there is a sharp increase in the hash rate of the network or more miners join in, then the expected time of finding new block decreases (2). As a result, there is a higher likelihood of producing stale blocks. Therefore, a dynamic difficulty adjustment helps in reducing the number of stale and orphaned blocks. While there are effective techniques to counter forks and orphaned blocks, the area of consensus remains open. Research efforts need to be dedicated to make PoW more energy efficient, and PoS, more decentralized. In PBFT-based private blockchains, the key issue is limited scalability due to high message complexity. Moreover, PBFT has low fault tolerance which makes it vulnerable to attacks. In Section V-H, we provide more details about making PBFT more scalable and secure.

V. Blockchain's Peer-to-Peer System

The underlying peer-to-peer architecture is the primary reason why certain guarantees are provided by a Blockchain, including security and accessibility. Counter intuitively, this peer-to-peer architecture that the Blockchain resides on actually contributes to several attacks including selfish mining, the 51% attack, DNS attacks, distributed denial-of-service attacks, eclipse attacks, fork after withholding attacks, and consensus delay. In this section, we explore how these attacks can compromise the Blockchain applications.

A. Selfish Mining

The selfish mining attack [130] is a strategy opted by certain miners who attempt to increase their rewards by deliberately keeping their blocks private [33], [79], [131]. Rather than releasing them to the public upon discovery, these selfish miners continue to mine their own private blocks to obtain a longer chain than the public Blockchain. These activities lead to a block race between the public chain of honest miners and the private chain of selfish miners. Once the public Blockchain starts approaching the length of their private chain, selfish miners release their blocks to claim block rewards. Having exceptional mining power may further help selfish miners win the block race. In Figure 11, we illustrate how a selfish mining attack is carried out.

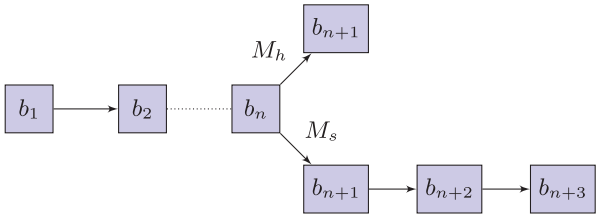


Fig. 11. Illustration of Selfish Mining. Selfish behavior of M_s forks the chain at b_{n+1} and discards M_h's block. M_h's block becomes a stale block.

Consider a Blockchain with blocks $(b_1, b_2, \ldots, b_n)$. Suppose an honest miner M_h has successfully mined the next block b_{n+1} and he publishes it. All the peers in the network validate and accept his block. At the same time, a selfish miner M_s also computes the block b_{n+1}. Instead of publishing his block, M_s chooses to withhold it and successfully mines two more blocks b_{n+2} and b_{n+3}. Despite M_h's block being added to the Blockchain, we show that M_h can still be cheated while having a majority of network's confidence in his block. Let the hash value of M_h's block b_{n+1} be lower than both the target threshold and M_s's block b_{n+1}. If only these two blocks were presented to the network, M_h's block would be chosen (due to its greater computational complexity) over M_s's block and appended to the public Blockchain.

However, after some time, M_s releases all of his blocks b_{n+1}, b_{n+2}, and b_{n+3} and forks the Blockchain at b_{n+1}. Due to the design protocols of Blockchain, the network will invariably shift to the longer chain belonging to M_s and discard the block b_{n+1} of M_h. The effort put forth by M_h in computing his block will be wasted due to selfish behavior of M_s. The incentive in adopting this selfish mining strategy is maximizing block rewards by publishing a longer chain. It should be noted that excluding the M_h's block b_{n+1} from the Blockchain does not destroy the block, rather it leads to another significant problem in the network known as "stale blocks" as shown in Section IV-B.

Selfish mining attacks can produce undesirable results for the rest of the network by invalidating the blocks of honest miners who contribute to the Blockchain. Furthermore, all the transactions in the honest miner's block also get rejected. In a situation where two selfish miners compete to add their chains to the network, the chances of a "Blockchain fork" arise (Section IV-A). These forks can cause a delay of consensus in the network, which can further lead to other potential attacks such as "double-spending" and "fork after withholding", as discussed in Section VI-B. One selfish activity in the network has the potential to disrupt the overall network, and

therefore it is imperative to study their relationship with one another.

B. The Majority Attack

The majority attack also known as the 51% attack is well known vulnerability in Blockchain-based applications that can be exploited when a single attacker, a group of Sybil nodes, or a mining pool in the network attains the majority of the network's hash rate to manipulate the Blockchain. With majority of network's hash rate, the attackers are able to 1) prevent transactions or blocks from being verified (thus making them invalid), 2) reverse transactions during the time they are in control to allow double-spending, 4) fork the main Blockchain and split the network, and 3) prevent other miners (verifiers) from finding any blocks for a short period of time. Under race conditions, the attackers with over 50% hash rate are guaranteed to over take other miners and append their blocks in the Blockchain with high probability [52]. Also, these blocks can possibly have fraudulent or double-spent transactions. For example, if an attacker performs a transaction in exchange for any product with Alice, it can replicate the same transaction with Bob and put it on the block. Transactions on Blockchains are not reversible, and only one transaction can be considered valid. In the following, we elaborate the prospects of double-spending with majority attacks along with the mathematical primitives.

1) Caveats and Realities: Mining pools do not always need 51% of the network's hashing power to carry out the fraudulent activities. As such, even with less hashing power, similar objectives can be achieved with a significant probability of success. To understand this issue, consider the scenario in which a malicious mining pool with significant hash rate carries out a transaction *Tx* with a receiver. At the same time, it generates a fraudulent double-spent transaction *Ty* from the same parent transaction to trick the receiver. The receiver, on the other hand, waits for k confirmations before releasing the product to the miner. The k confirmations mean that k subsequent blocks have been mined by the network after mining the transaction *Tx*. During this process, the malicious miner keeps mining blocks on his end with the double-spent transaction *Ty* and hopes to fork the Blockchain after he receives the product from the recipient. By forking the chain, the malicious miner will be able to invalidate the chain with transaction *Tx*, and will replace it with his own chain with double-spent transaction *Ty*.

To launch a successful attack, the malicious miner needs to publish a longer chain with valid PoW so that the network switches to his forked version. Miner's success depends on his hash rate x as a fraction of the network's hash rate and the number of confirmations k. To find the probability of success $P(s)$ for the attacker, let x be the fraction of miner's hashing power and y be the fraction of remaining hashing power, where $x + y = 1$ [130]. The success probability is shown in (3):

$$P(s) = \begin{cases} 1, & \text{if } x > y \\ \left(\frac{x}{y}\right)^k, & \text{if } x < y. \end{cases} \tag{3}$$

2) Numerical Results: In Figure 12, we show how $P(s)$ changes with varying hash rate. Note that if the miner acquires half of the network's hash rate, he can trick the recipient with 100% success rate. Moreover, an attacker with hash rate less than 50% can still succeed in forking the main chain and cheating the receiver.

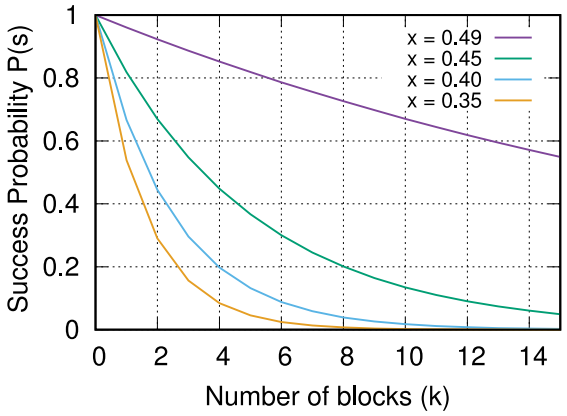


Fig. 12. Change in the success probability $P(s)$ of majority attack with varying hashing power x and number of confirmations k. Notice that with 0 confirmations, an attacker can always double-spend with any magnitude of hash power. In that case we describe the user to be optimistic.

TABLE VI
ATTACK COST REQUIRED TO LAUNCH THE 51%, ON 18TH JUNE 2018, ON THE TOP SIX BLOCKCHAIN-BASED CRYPTOCURRENCIES. HERE CAP DENOTES THE MARKET CAP IN USD, ALGO DENOTES THE ALGORITHM USED FOR BLOCK CONSENSUS, AND COST DENOTES THE ATTACK COST IN USD FOR LAUNCHING THE 51% ATTACK FOR ONE HOUR

SYSTEM	CAP	ALGO	HASH RATE	COST
BITCOIN	112.7B	SHA-256	35,604 PH/s	486K
ETHEREUM	49.5B	Ethash	222 TH/s	347K
B.CASH	14.9B	SHA-256	5,023 PH/s	68K
LITECOIN	5.7B	Scrypt	327 TH/s	60K
DASH	2.1B	X11	2 PH/s	15K
MONERO	2.3B	CryptoNight	365 MH/s	17K

3) Applications and Implications: A Blockchain-based application for Internet of Things (IoT), known as "The Tangle" [132] can be theoretically compromised with one-third of the hash power. Bahack [133] show that the majority attacks are highly feasible with one quarter of the network's hashing power. There are online services such as Nicehash, that rent hashing power to miners on hourly basis [134]. The presence of online services that rent hashing power has been noted by [135]. The authors noticed that by renting hashing power from online services, an adversary can launch short-term majority attacks, which they call "*bribery attacks*". These attacks can provide short-term benefits to a greedy miner while not significantly affecting the long-term health of the system.

A malicious mining pool can rent the computation power for a few hours and launch the majority attack on the targeted cryptocurrency. Since major blockchain systems have a high aggregate hash rate, the renting cost to launch the 51% attack on them is (naturally) high. In Table VI, we outline the top six Blockchain-based cryptocurrencies, and the cost required to successfully launch the 51% attack, based on data obtained from "51crypto" [136]. We notice that Dash with a market cap of 2.3 Billion USD can be compromised for one hour by spending only 17,000 USD (8×10^{-4}% of the market cap).

4) Case Studies: A 51% attack is not beyond the realm of possibilities. In July 2014, a Bitcoin mining pool "GHash.IO" acquired over 51% of the hash rate for one day [32], which raised many concerns in the press and media about Bitcoin and its vulnerabilities, and shed light on the general problem in Bitcoin-based systems. Although no malicious activity was carried out, "GHash.IO" later shrunk in size when miners left its pool and eventually closed in October 2016. In August 2016, a group of attackers, known as "51 crew", hijacked two Ethereum Blockchains, namely Krypton and Shift, and managed to hijack 21,465 Kryptons worth of digital currency by double-spending. In May 2018, a group of malicious miners acquired 51% hash rate in Bitcoin Gold and stole $18 million USD worth of cryptocurrency [137]. In June 2018, four other notable Blockchain-based cryptocurrencies were also attacked; namely Monacoin, Zencash, Verge, and Litecoin Cash.

5) Lowering Bounds on 51% Attack: The bounds on 51% assume a synchronous network in which each miner is concurrently mining on the same block [25]. In theory, this behavior is possible in a completely connected P2P topology, where a block released by a miner reaches all the other miners at the same time with minimum delays. However, in practice, the P2P network of a Blockchain system may not exhibit those properties and block propagation may experience non-uniform delays. As a result, the network may deviate from the theoretical assumption towards an asynchronous setting. This deviation can allow an adversary to delay block propagation to other miners, and gain an extra advantage to extend his own private chain. Moreover, the extra advantage can lower the bounds on the 51% attack.

Prior work by [28] observed the deviation of Bitcoin from the synchronous settings. Using that, they derived a new lower bound of 49.1% for the 51% attack. The formal analysis of Blockchain system in asynchronous networks has been conducted by Paas *et al.* [138] and Zhao [139]. Particularly, Zhao [139] derived a strong lower bound for the consistency property of Nakamoto protocol. Another work in this direction is by Ren [140], which characterizes the synchronous and asynchronous settings of Bitcoin as the *lock-step* and *non lock-step* synchrony model to simplify the analysis. In all these efforts, the key takeaway is that if an adversary can delay block propagation to other miners, it can launch the majority attacks with a lesser hash rate.

6) Majority Attacks in Private Blockchains: In PBFT-based private Blockchains, an adversary can launch a majority attack if he controls ≈33% replicas [141]. In the private Blockchains, the size of the network is known to the participating nodes, which allows the adversary to calculate the number of sybil nodes he needs to introduce in the network for an attack. Assuming that the adversary controls f sybil nodes such that the total network size is $n < 3f + 1$, then the attacker will be able to launch the attack to stop the transaction verification process. For each transaction sent by the primary, the sybil replicas will not reply with their approvals. Since the primary will need approvals from at least $3f + 1$ replicas, it will not be able to process any transaction, and the system activities will be halted.

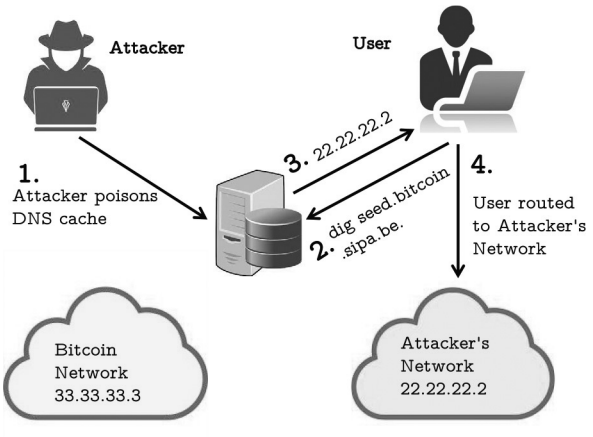


Fig. 13. DNS resolution attack on Bitcoin. The attacker poisons DNS cache and modifies the data. When a user queries the server to obtain IP addresses of peers who are accepting connections, he is routed to attacker's network. The attacker can game the user by feeding him fake blocks and transactions.

C. Network Attacks

Blockchain applications are decentralized and use peer-to-peer network architecture as the medium of communication between the network entities. In this section, we will look into the attacks associated with the peer-to-peer network and we will use Bitcoin network as our example to provide details of these attacks. The attacks associated to the Blockchain network include among others, the DNS attacks, spatial partitioning, and Eclipse attacks. For each of these attacks, the goal of the attacker is to isolate users and miners from the real network, limit their access to the network resources, or create partition in the network and enforce conflicting rules among the peers.

1) DNS Attacks: When a node joins the Bitcoin network for the first time, it is not aware of the active peers in the network. To discover the active peers (identified by their IP addresses) in the network, a bootstrapping mechanism is required. The Domain Name System (DNS) can be used as a bootstrapping mechanism, and DNS seeds are queried by nodes upon joining the network to obtain further information about other active peers. The initial DNS query returns one or more DNS A records along with their corresponding IP addresses of peers that are accepting incoming connections. Once the new node establishes connections to the peers, it can send *addr* command with port numbers to establish connections with other peers.

It has been mentioned in the developer's guide of Bitcoin systems [29] that the DNS opens a wide attack surface to the Bitcoin networks in general. Namely, the DNS resolution is vulnerable to man-in-the-middle attacks (at the resolver side), cache poisoning, and stale records, among many others. For this attack, an adversary can either inject an invalid list of seeder nodes in the open source Blockchain software, or poison DNS cache at the resolver. By default, the Blockchain software client has a list of seeders that allow the network discovery. If the attacker injects a fake list of seeders, the user will be compromised. As a result, the adversary can potentially isolate Blockchain peers and lead them to a counterfeit network. In Figure 13, we illustrate how a DNS attack can be carried out by poisoning DNS cache. A node in Bitcoin network

TABLE VII
LOCATION OF FULL NODES IN THREE MAJOR CRYPTOCURRENCIES. – IN BITCOIN REFERS TO THE NODES THAT USE TOR SERVICES AND THEIR LOCATION CANNOT BE IDENTIFIED

	Bitcoin		**Ethereum**		**Litecoin**	
Rank	Country	Nodes	Country	Nodes	Country	Nodes
1	Unite States	2445 (24.98%)	United States	6549 (37.99%)	United States	79 (24.38%)
2	Germany	2445 (24.98%)	China	2202 (12.77%)	Russia	36 (11.12%)
3	China	675 (6.90%)	Canada	1118 (6.49%)	Germany	19 (6.49%)
4	France	663 (6.77%)	Russia	846 (4.91%)	China	17 (5.21%)
5	Netherlands	475 (4.85%)	Germany	783 (4.54%)	Netherlands	17 (5.21%)
6	Canada	369 (3.77%)	United Kingdom	559 (3.24%)	United Kingdom	16 (4.91%)
7	—	368 (3.76%)	Netherlands	470 (2.73%)	France	11 (3.42%)
8	United Kingdom	307 (3.14%)	South Korea	429 (2.49%)	Brazil	11 (3.42%)
9	Russia	296 (3.02%)	France	399 (2.31%)	Canada	11 (3.42%)
10	Japan	219 (2.24%)	Japan	279 (1.62%)	Hong Kong	11 (3.42%)

has an IP address of 33.33.33.3 (for illustration purpose only) while the attacker's node in a counterfeit network has an IP address 22.22.22.2. The attacker poisons the DNS cache to lure the user into the counterfeit network. The user makes the DNS query *dig seed.bitcoin.sipa.be.* and instead of responding with 33.33.33.3, the DNS resolver returns 22.22.22.2. As a result, the user connects to malicious nodes in the counterfeit network and malicious nodes may feed false blocks to the user. For more on DNS security, we refer to the work in [142].

2) BGP Hijacks and Spatial Partitioning: There are two types of nodes in most Blockchain applications, namely full nodes and lightweight nodes. Full nodes are the actual participants in the network responsible for relaying blocks and transactions and maintaining an updated copy of the Blockchain. Lightweight nodes do not maintain a Blockchain and only use the services of full nodes to get access to the network. Since lightweight nodes draw their view of the Blockchain from the full nodes, when a full node is compromised all of its associated lightweight nodes are also compromised. Full nodes in a Blockchain network are spatially distributed across the Internet. In Table VII, we show the spatial spread of full nodes in three major Bitcoin systems (cryptocurrency). In each system, a majority of the nodes is located in United States, Germany, China, and Russia. The flow of traffic on the Internet is controlled by Internet Service Providers (ISPs), which own one or more Autonomous Systems (ASes), responsible for handling traffic routing [143], [144].

Spatial concentration of nodes within an AS or an ISP makes them vulnerable to routing attacks such as BGP hijacking [145]. An adversarial AS can hijack the traffic for a target AS that hosts a majority of the Blockchain application nodes. This can disrupt the flow of valuable information, including transactions and blocks, to the nodes being hosted by the target AS. When the victim nodes are miners or mining pools, the attacker can substantially reduce the hash rate of the Blockchain application, thereby affecting the system activities. In a mining pool, the miners communicate using stratum overlay protocol. The stratum servers act as a drop-zone where miners submit their PoW. Stratum servers have a public IP address that makes them vulnerable to routing attacks

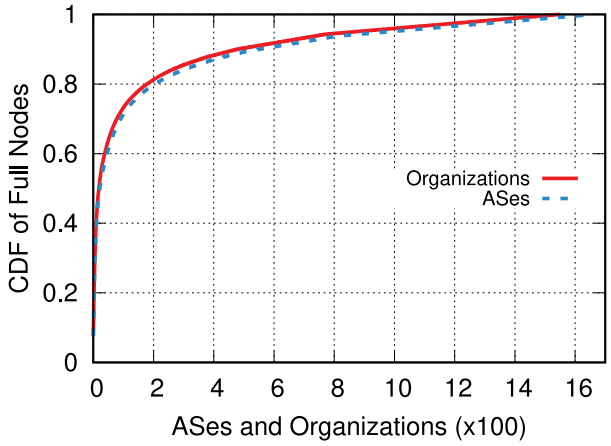


Fig. 14. Distribution of full nodes in Bitcoin across ASes and ISPs (organizations). Notice that less than 50 ASes and ISPs host more than 50% of nodes showing that the network is centralized and vulnerable to BGP attacks.

and flood attacks. Apostolaki *et al.* [30], Saad *et al.* [146] studied that by hijacking fewer than 100 border gateway protocol (BGP) prefixes in Bitcoin, an attacker can isolate up to 50% of the network's hash rate. They further explored that 60% of all Bitcoin traffic traverses only three Internet service providers (ISPs). Every month, over a 100 Bitcoin nodes suffer from routing attacks and BGP hijacks. Furthermore, they estimated that the routing attacks can delay block propagation by up to 20 minutes. As mentioned in Section IV-B, the average block computation time in Bitcoin is 10 minutes. Therefore, the routing attacks can delay the propagation of two or more blocks to a group of nodes. Such delays increase the likelihood of other attacks including Blockchain fork, consensus delay, and double-spending.

To verify their results and further analyze the spatial vulnerability of Bitcoin network, we replicated their study and noticed that Bitcoin network has further centralized with respect to ASes and ISPs. We crawled data from "Bitnodes", an online service that maintains information related to full nodes in Bitcoin [147]. In Figure 14, we plot the CDF of the spatial distribution of full nodes across ASes and ISPs in the world. In Table VIII, we show the distribution of mining

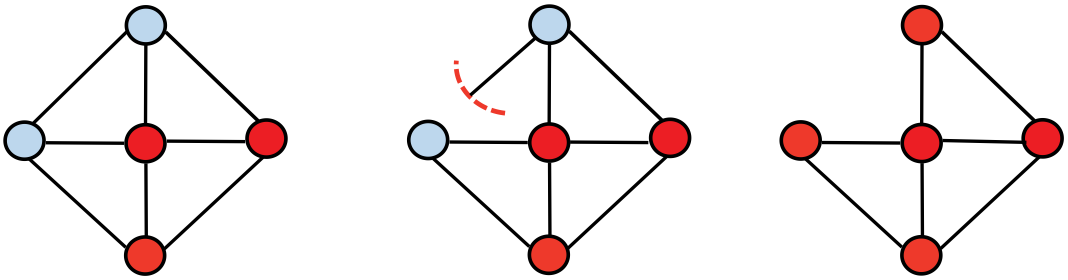


Fig. 15. Eclipse attack on a cryptocurrency network. Here, blue nodes represent the honest nodes following the true state of Blockchain while the red nodes represent the malicious nodes that form a cluster around the blue nodes. If the connection between the honest nodes is compromised, the malicious nodes may feed fake blocks to the honest nodes and partition them from the network. As a result, the honest nodes end up having the wrong view of the Blockchain.

TABLE VIII
TOP 5 MINING POOLS PER HASH RATE, ASES, AND ORGANIZATIONS. 65.7% OF MINING DATA GOES THROUGH ONLY THREE ORGANIZATIONS. ALIBABA ALONE HAS A VIEW OF AT LEAST 60% OF THE MINING DATA. WE EXCLUDE THE REMAINING 12 MINING POOLS FROM THE STUDY AS THEIR TOTAL CONTRIBUTION TO HASH RATE IS MINIMAL

Mining Pool	H. Rate %	ASes	ISP
BTC.com	25%	AS37963	Alibaba
		AS45102	AliBaba
Antpool	12.4%	AS45102	AliBaba
ViaBTC	11.7%	AS45102	AliBaba
BTC.TOP	10.3%	AS45102	AliBaba
F2Pool	6.3%	AS45102	AliBaba
		AS58563	Chinanet
12 others	34.3%	—	—

pools across ASes and ISPs in Bitcoin. Notice that 60% of the hash rate is solely intercepted by *AliBaba*. Our results show that compared to the prior work by Apostolaki *et al.* [30], the Bitcoin network has further centralized and become more vulnerable to routing attacks.

Case studies: Over the last few years, a number of BGP attacks have been launched against ASes that host mining pools or cryptocurrency exchanges. In 2014, a malicious ISP in Canada announced BGP prefixes belonging to major ISPs including *Amazon, OVH, Digital Ocean, LeaseWeb,* and *Alibaba*, and intercepted the traffic routed to mining pools. As a result, the attacker made a fortune of 83,000 USD. In April 2018, BGP attacks were launched against *MyEtherWallet.com*, an open source Web application used for exchanging Ethereum tokens online. Attackers managed to steal 152,000 USD from the Web application [148].

3) Eclipse Attacks: Blockchain's peer-to-peer system is also vulnerable to a form of attack known as the eclipse attack [31], [79], [149], in which a group of malicious nodes isolates its neighboring nodes using IP addresses, thereby compromising their incoming and outgoing traffic. For example in Bitcoin, a node can actively connect to all the other nodes in the network, forming a node cluster. In the node cluster, every peer is aware of the IP address of all other peers. With sufficient compromised nodes in a cluster, the attacker can isolate honest nodes and change their Blockchain view. He can control their incoming and outgoing traffic and feed them with fake information regarding Blockchain and transactions.

In Figure 15, we illustrate this attack procedure. As long as the honest node maintains a connection with one other honest node, it is likely to receive the correct information to maintain the true state of the Blockchain. However, when the connection between the honest nodes is compromised, they will get surrounded by malicious nodes and become vulnerable to the eclipse attack. When such nodes are fed with fake transactions and blocks, they eventually develop the wrong view of the state of Blockchain and become part of the malicious node cluster. Furthermore, if another honest node establishes a connection with the malicious node cluster, it is also exposed to the same vulnerability which leads to the cascade effect of propagation of fake transactions and blocks.

D. Distributed Denial of Service Attacks

One of the most common attacks on online services is the distributed denial-of-service (DDoS) attack [150]. Blockchain technology, and despite being a peer-to-peer system, is still prone to DDoS attacks. Blockchain-based applications, such as Bitcoin and Ethereum, have repeatedly suffered from these attacks [34], [151], [152]. DDoS attacks manifest themselves in a number of ways, depending upon the application nature, network architecture, and peers behavior. For example, in the Bitcoin network, the 51% attack can lead to denial-of-service. Specifically, if a group of miners acquire a significant hashing power, they can prevent other miners from adding their mined blocks to the Blockchain, invalidate ongoing transactions, and cause service failure in the network. Intentional forks; forks that are the result of malicious behavior; can turn into hard forks, resulting in similar outcomes of denial-of-service.

1) Stress Testing: Another possibility for the attack is due to the limited number of transactions per block a Blockchain application can process in a given time. For example, on average, it takes the Bitcoin network 10 minutes to mine a block, which has a maximum size of 1MB. Although the size of transactions in Bitcoin varies, the average size of a transaction in Bitcoin is approximately 500 bytes, allowing approximately 2,000 transactions per block on average—the maximum number of transactions added to a block in Bitcoin is reported to be 2,210 [115]. Furthermore, the average time needed to mine a block, based on the predefined difficulty, is approximately 10 minutes. As such, for all current transactions in the network to be successfully included in the Blockchain, their number may not exceed 200 transactions per minute. Taking that into

account, and the fact that each transaction requires a minimum of two peers (identified by two different public identifiers) to be involved in a transaction, the total active peers served by the network per minute (i.e., where a block containing their transaction will be mined) will not exceed 200 peers. Given these constraints, the throughput of Bitcoin is 3-7 transactions per second. Throughput of Bitcoin is low compared to mainstream payment processors such as Visa Credit that can verify up to 2,000 transactions per second.

An adversary may exploit the aforementioned operational reality of the Bitcoin system by introducing Sybil identities; the same adversary also may control multiple wallets. Furthermore, using those identities, the adversary may issue several dust transactions (*e.g.*, 0.001 BTC per transaction) between the various Sybil identities under his control. By introducing a large number of transactions of small value over a short period of time, the network will be congested by creating blocks containing those transactions, and service to legitimate users in the network will be denied. As a result of this congestion the adversary may as well launch other attacks; *e.g.*, double-spending of tokens not mined due to the congestion.

To counter this attack one can increase the mining fee limit for each transaction. However, this would: 1) require a consensus of all miners on a minimum fee, 2) give an unfair advantage to miners assuming an agreement happens, and 3) create problems for normal users who cannot afford high mining fee. Therefore, increasing the mining fee in Bitcoin or the gas fee in Ethereum only partially solve the problem. Moreover, assuming that a blockchain system implements a fee or a gas limit, an attacker may still exploit other vulnerabilities to DDoS the system. One such example is to flood the memory pool with unconfirmed transactions to artificially spike the fee. We provide the details of this attack below.

2) Mempool Flooding: Another form of DDoS attack is carried out at the memory pools (mempools) of the cryptocurrencies to increase the mining fee. As outlined in Figure 1, mempools act as a cache of unconfirmed transactions. Although the block size is limited in the cryptocurrencies, the mempool size has no size limit. However, users estimate the size of mempools to prioritize their transactions. If there are more transactions in the mempool, then the competition for mining becomes high. To prioritize their transactions, users start paying more mining fees as an incentive for miners. Saad *et al.* [34], identified a low cost DDoS attack on Blockchain applications in which the adversary along with Sybil nodes may flood the mempools with unconfirmed transactions. Such an attack creates panic among the legitimate users who are tempted to pay higher mining fee to prioritize their transactions while the attacker's transactions do not get mined. As a result, the attacker launches a DDoS attack.

3) Case Studies: In Bitcoin, malicious users have been flooding the mempool with dust transactions to make legitimate users pay higher mining fees. On November 11, 2017, the Bitcoin mempool size exceeded 115k unconfirmed transactions, resulting in $700 million USD worth of transaction stall [50]. In June 2018, again the mempool was attacked with 4,500 unconfirmed spam transactions which increased the mempool size by 45MB. The increased size led to a spike in the mining fee and legitimate users were propelled to pay higher fee to get their transactions mined [153]. In Figure 16, we plot the mempool size and the mining fee of Bitcoin over from 2016 to 2017. We use min-max normalization to scale the data points.

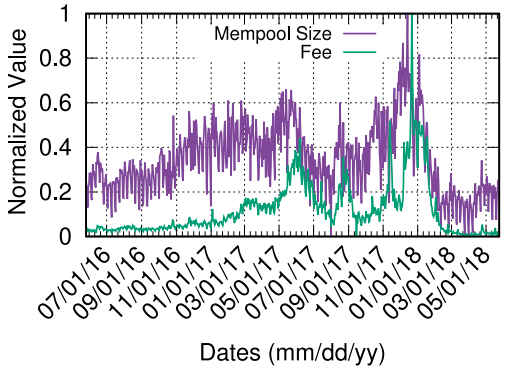


Fig. 16. Bitcoin mempool size and average fee paid to the miner. It can be observed from the figure that as the mempool size is increases, the fee paid to the miners also increases.

E. Block Withholding Attacks

The peer-to-peer network of cryptocurrencies can be exploited to create conflicting views about the Blockchain. Malicious nodes can intentionally mask, forge, or withhold important information that needs to be relayed across the network. Some of the known attacks of this nature are "The Finney Attack" and "Block Withholding Attack" [154].

1) The Finney Attack: The Finney attack is a variant of the double-spending attack in which a miner delays block propagation to double-spend his transaction [155], [156]. The miner generates a transaction, computes a block, and chooses not to relay the block. In the meantime, he generates a duplicate of his previous transaction and sends it to a recipient. After the recipient accepts the transaction and delivers the product, the miner publishes his previous block with the original transaction in it. Therefore, the previous transaction sent to the recipient becomes invalid and the miner successfully double-spends transaction.

The Finney attack has low success probability due to short block intervals and time sensitive attack procedure. The block time in Bitcoin and Ethereum are 10 minutes and 15 seconds. If an attacker attempts to launch this attack on Ethereum, it is unlikely that he will be able to 1) generate a double spent transaction, 2) trick an optimistic receiver, 3) receive product before confirmation, and 4) publish a block before any other miner, within 15 seconds. Since the attack procedure is more time consuming than the block interval time, Finney attack is highly infeasible and as such, no case of Finney attack has yet been reported on any cryptocurrency.

2) Classical Block Withholding Attack: The block withholding attack is launched against decentralized mining pools with intent to harm the pool operator by withholding a valid PoW [157], [158]. In decentralized mining pools, all participants consume electricity and CPU power to find a nonce whose value of a hash with the block is less than the target threshold. Once the valid solution is found, all participants

are rewarded based on their aggregate effort put towards the computation of the solution. Since nonce finding is a probability, therefore, miners with less hash power may come up with a valid solution before other miners with a higher hash rate. In the block withholding attack, a compromised miner in the pool finds the PoW and chooses not to disclose it to the pool operator. Unaware of the compromised miner, the rest of the miners in the pool waste their resources to find the nonce and eventually lose the race. The malicious miner then can collude with other mining pools and share the PoW with them for a higher reward, or even publish the block independently with a different identity. Due to this unfair behavior of one miner in the pool, the entire pool is deprived of block rewards.

Another form of withholding attack is possible when two mining pools intentionally try to fork the Blockchain to create a network partition [111]. For instance let there be two mining pools in a cryptocurrency, namely Mp_A and Mp_B, and Mp_A computes a valid block but decides not to publish it. Mp_A waits for Mp_B to compute and publish the block. As soon as Mp_B releases its block, Mp_A also releases its block and resulting in two valid blocks in the network. This will fork the Blockchain and nodes in the network will have a consensus disagreement upon receiving two valid blocks. Although this attack may partition the network, it may also cause loss to both mining pools. Therefore, no such attack has been reported in any Blockchain application so far.

3) Fork After Withholding Attack: Another form of withholding attack is known as the fork after withholding (FAW) attack. Introduced by Kwon *et al.* [111], FAW is always more rewarding than block withholding attacks. In the following, we outline the attack procedure of FAW.

- A malicious miner joins two mining pools Mp_A and Mp_B respectively.
- The miner computes a valid PoW in mining pool Mp_A.
- He withholds the solution and only publishes the block once Mp_B also publishes the block.
- The network selects one block among the two.
- The malicious miner gets rewarded either way.

Kwon *et al.* [111] also show that if the FAW attack is launched by two or more mining pools against each other, then the bigger mining pool will always win in the race condition. Therefore, the FAW attack is always more profitable than selfish mining and block withholding.

4) Block Withholding in Private Blockchains: In private Blockchains, the primary replica can launch a block withholding attack after receiving a confirmations from other replicas. Private Blockchains work under the assumption that the primary will faithfully execute the protocol. Moreover, the adversarial model assumes that the attacker controls a subset of faulty replicas among all the other replicas. However, if the adversary also controls the primary replica, he can withhold blocks and transactions from all other replicas. As shown in Figure 3, the primary receives a transaction request from the client and sends the transaction to other replicas to obtain their signatures. Finally, it computes the block when a sufficient number of transactions are processed. However, if the primary gets compromised, he may: 1) withhold a transaction issued by the client and abort the verification process, 2) delay the verification process by sending the transaction to fewer replicas, 3) receive the signatures and discard them, 4) compute a block and withhold it from the rest of the network. In each case, the primary can launch a withholding attack to compromise the system and delay the transaction processing.

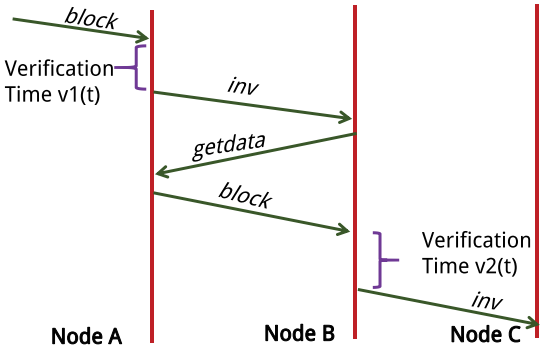


Fig. 17. Block propagation between nodes A, B, and C. Notice that the maximum time is consumed in block verification, $v1(t)$ and $v2(t)$. Consensus delay can be caused by propagating false or stale blocks in the network.

F. Consensus Delay

Another attack associated with the peer-to-peer nature architecture is the consensus delay, noticed by Geravis *et al.* [159]. In this attack, an attacker may inject false blocks to add latency or prevent peers from reaching consensus about the state of the Blockchain. In Figure 17, we illustrate the delays incurred during block propagation in Bitcoin. When node A receives a block, it authenticates the block and sends an *inv* message to its neighbors including node B. If node B does not have the block, it sends *getdata* message back to A. Upon receiving the *getdada* message from B, A sends the block to B. Once B has the block, it also authenticates the block and sends an *inv* message to its neighbors. As Figure 17 shows, the maximum delay is incurred during authenticity check ($v1(t)$ and $v2(t)$). The other delays include transmission delays and propagation delays of messages and block. Transmission delays are subject to the size of block and messages while the propagation delays depend upon the bandwidth of the link between the nodes.

In such conditions, intentional delays can be introduced in the network by propagating stale blocks or double-spent transactions. The nodes which are not aware of stale blocks will respond with *getdata* messages and upon receiving the block, they will waste time in its verification. If an attacker controls a set of sybil nodes in the node cluster Section V-C, it can add significant delays among legitimate nodes in that cluster. The problem is further exacerbated for time-critical applications such as Blockchain-based peer-to-peer gaming, where resolution needs to be achieved within short time.

In PBFT-based private Blockchains, an adversary can also cause consensus delay by using sybil nodes. As shown in Figure 3, a major component of transaction processing is the exchange of messages and signatures among participating replicas. Especially, in the *prepare* and *commit* phase, each replica sends its signatures to every other replica. As outlined in Section V-D, if the adversary controls more than 33% of

replicas, he can launch a DoS attack. On the other hand, if the adversary controls fewer replicas, he may still be able to cause consensus delay in transaction processing [160], [161]. The sybil nodes can also send bogus signatures to the other replicas during the prepare phase and the commit phase. Since each replica is then required to verify signatures, therefore, bogus signatures will cause additional verification overhead. If the sybils continue to send such signatures, they can stall the completion of the commit phase and eventually cause a delay in the reply phase. As a result, the primary will not receive the required number of approvals for the transaction verification. This will cause consensus delay and reduce the throughput of the application.

G. Timejacking Attacks

In Bitcoin systems, such as Bitcoin, full nodes maintain an internal counter that denotes the network time. The network time is obtained by receiving a version message *ver* from neighboring peers and calculating its median during the bootstraping phase. If the median time of all the neighboring peers exceeds 70 minutes, the network time counter is automatically reverted to the system time of the node. This creates an attack opportunity for malicious nodes which may connect to the target node as shown in Figure 15. In such case, the attacker can feed varying timestamps with median value exceeding 70 minutes. Furthermore, in Bitcoin, for example, a node rejects a block if its timestamp exceeds the network time by 120 minutes. An attacker can compute a new block and set its timestamp ahead of network's timestamp by 50 minutes. The attacker then, along with Sybil nodes, can slow the network time of a target node by launching a timejacking attack against it. As a result, the difference between the block time and the target node's counter will exceed 120 minutes. As a result, if the target node is presented with the block, it will reject it and all the subsequent blocks. The target node eventually gets isolated from the activities of the main network.

H. Countering Peer-to-Peer Attacks

Prior research has been conducted to address the problem of selfish mining, and researchers have suggested several possible solutions [80]–[82], [130]. Solat and Potop-Butucaru [83] proposed a "lifetime" for blocks that prevents *block withholding* by selfish miners. If the expected lifetime of a block expires (calculated by the honest miners), it is rejected by the network. Heilman [81] impedes the profitability of selfish miners by introducing a defense scheme called "Freshness Preferred." Heilman [81] builds on top of the previous work by Eyal and Sirer [130] by adding unforgeable timestamps to blocks and prefers blocks with more recent timestamps compared to older ones. His work reduces the incentive for selfish miners to withhold their blocks for long periods of time. Eyal [27] modeled a game between two mining pools carrying out *block withholding* and discovered *miner's dilemma*, where both mining pools suffer a loss in equilibrium.

Majority attacks have also been widely discussed with countermeasures proposed to overcome a monopoly in Blockchain networks. Miller *et al.* [162] proposed changes to the PoW puzzle in Bitcoin in order to restrict coalitions of mining pools for majority attacks. Their proposed design incorporates *nonoutsourceable* puzzles in PoW, in which mining pools that outsource their mining work risk losing mining rewards. Saad *et al.* [163] leveraged the expected transaction confirmation height and the block publishing height to detect selfish mining behavior in PoW-based Blockchains. Using the relationship between the two features, they created a "truth state" for each published block in order to distinguish between a legitimate block and a selfishly mined block. Also addressing the 51% attack, Bastiaan [32] introduced the concept of "two phase proof-of-work" (2P-PoW). 2P-PoW is a continuous-time Markov chain (CTMC) model that incorporates two challenges for miners to solve instead of one. The states of the CTMCs prevent the pool from increasing beyond an alarming size by shrinking incentive for miners in the pool. 2P-PoW prevents large pools from creating a hegemony by either outsourcing a major chunk of their hash rate or exposing the private keys of the pool operator.

Johnson *et al.* [164] proposed a game-theoretic approach to address DDoS attacks against mining pools. Other countermeasures include putting a cap on the minimum amount in the transaction that a sender can have or increasing the block size to accommodate more transactions. Yet another approach is to reduce the difficulty in mining blocks so that more blocks can be mined with no transactions going to waste. Each of these propositions have their own caveats.

Increasing the block size might not be sufficient, since a powerful adversary can still stress the network by generating dust transactions. On the other hand, reducing difficulty will reduce the block time but it will increase the number of orphaned blocks in the system and the overall size of the Blockchain. At the time of writing this paper (October 2018), Bitcoin and Ethereum Blockchain size was recorded to be 162 GB and 450 GB, respectively [165]. Saad *et al.* [34] proposed fee-based and age-based countermeasures to prevent DDoS attack on Blockchain mempools. In their work, they shifted the transaction filtering process from the mining pools to the mempools. Their proposed countermeasures optimize the mempool size and raise the attack cost for the attacker while favoring legitimate users in the system.

To prevent DNS-based attacks, extensive research has been carried out to equip the Blockchain systems with DNS attack defenses [78], [166], [167]. Apostolaki *et al.* [30] proposed long- and short-term solutions for routing attacks. They propose routing-aware peer selections to maximize diversity of Internet paths and limit the vantage points for attacks. They also proposed peer behavior monitoring to check abrupt disconnections and unusual latency in block delivery.

Other solutions to prevent delay attacks include end-to-end encryption for message propagation. Another possible approach to prevent spatial partitioning is the decentralized hosting of mining pools and full nodes over the Internet. As shown in Table VII. 50% of Ethereum nodes are located within two countries, which makes them vulnerable to a nation-state adversary. In order to prevent that, new nodes must be hosted on cloud services that have a higher geographical spread and network diversity. The dimensions we explored in this paper

encourage additional research on Blockchain technology in the areas regarding DNS and DDoS attacks.

To counter block withholding attacks [82], [154], [168], [169], Schrijvers *et al.* [170] introduced an incentive-compatible reward scheme that discourages a malicious miner from carrying out withholding attacks against the targeted mining pool. Rosenfeld [171] introduced a Honeypot technique to lure rogue miners into a "trap", thereby catching the miner who withholds valid solutions. Bag and Sakurai [169] proposed additional incentives for finding a valid solution for a block in order to prevent mining collusion. Concurrent to their prior work, Bag *et al.* [168] introduced a new scheme that blinds the miners in the pool from the current target to obfuscate their ability to distinguish between a partial and full PoW. Their proposed solution also binds the pool operator to fairly distribute the reward to the winning miner.

The FAW attack can be countered by introducing timestampped beacons in the assignment given to the miners by the pool operators [111]. As a response to each assignment, the miners calculate the partial PoW and send the response to the pool operator embedded with the beacon value. The beacon value is updated after a few seconds to catch a malicious miner if he withhold the valid solution and later propagates it in the network. However, the authors also noticed that this solution may not be practical in some situations and conclude that FAW attacks remain an open problem for the research community to address. To address the security issues in private Blockchains, several variants of PBFT protcol have been proposed. Those protocols try to increase the fault tolerance beyond 33% [172], [173] and use hardware assistance to detect the behavior of faulty replicas [174]. The key challenge in private Blockchains is the high message complexity that restricts the scalability. As a result, in a small network, the adversary can easily compromise 33% replicas. To address this issue, Liu *et al.* [174] proposed a scalable Byzantine consensus with hardware assisted secret sharing, which reduces the message complexity of PBFT to $O(n)$. This can be leveraged to construct large private Blockchain networks that can withstand various forms of attacks.

To counter BGP attacks on Bitcoin, Aposotolaki *et al.* [175] proposed a secure and scalable Bitcoin relay network called SABRE. SABRE uses the inter-domain routing policies to discover secure and economic routing paths between relay nodes. The relay nodes provide security against routing attacks, and therefore, can be used to route the Bitcoin traffic. SABRE does not take a clean-slate approach towards redesigning the routing policies. Instead, it is easily deployable and can run alongside the existing Internet architecture.

Concurrent to the ongoing efforts of improving the network-oriented vulnerabilities in Blockchain systems, new research directions are being taken to use Blockchains for improving security and efficiency of the information-centric networking (ICN) [176]–[178]. Weiss *et al.* [176] explored the application of Blockchain in radio spectrum management. Their work outlines the utility of Blockchains properties towards ensuring some fundamental requirements for dynamic spectrum sharing applications. These requirements include decentralization, transparency, immutability, access control, and security. In a spectrum sharing environment, participants can have competing interests which may lead to conflicts in meeting these requirements. While highlighting these challenges, the paper puts Blockchains into perspective and shows that a consensus-based distributed ledger can be helpful in the context.

Ghiro *et al.* [177], explored the use of Blockchains in the distributed wireless networks. Their proposed model envisages a system in which all nodes find a rough consensus on the best path that connects them with each other. They call it the "Proof of Networking" and investigate its potential utility in ensuring trust efficiency and access control in large-scale distributed networks. Li *et al.* [178], presented Gosig; a Blockchain-based scalable Byzantine consensus on adversarial wide area networks. Gosig combines crypto-based secret leader selection and multi-round voting in the network protocol layer to optimize the gossip-based message propagation. The Blockchain data structure adds a security protection by preventing data-tampering and the Byzantine consensus enables operational consistency in the presence of adversaries.

VI. Blockchain Applications Attacks

The Blockchain and associated peer-to-peer system are separate from the application services using them. Based on the nature of the Blockchain applications, they have their own vulnerabilities and attack surface. Therefore, we expect a significant number of attacks related to various applications, which we address in this section. Our analysis is primarily on applications such as cryptocurrencies and smart contracts.

A. Blockchain Ingestion and Anonymity

Public Blockchains have a weak notion of anonymity, and they provide open data accessibility to the public. As such, the analysis of the public Blockchain can reveal useful information to an adversary. This process is known as Blockchain ingestion and it might not be desirable to the Blockchain application or its users. For example, a credit card company in the open market can use data analytics to delve into public information on the Blockchain and optimize its own schemes to compete with the digital currency. To demonstrate the potential exploitation of the public data, Fleder *et al.* [38] used graph analysis to create directed links between Blockchain data of Bitcoin and associated identities of the wallet users.

1) Mt. Gox Incident: In 2013, two attackers exploited the public nature of Bitcoin Blockchain to carry out fraudulent transactions and create a fake demand of bitcoins at multiple exchanges. The main target of attack was Mt. Gox; the biggest Bitcoin exchange in Japan in 2013. The attackers frequently carried out a sequence of fraudulent transactions at Mt. Gox. Since the Blockchain is public, the rate of transactions was noted by other exchanges too and it was assumed as if the overall demand of the coins had increased. As a result, the price of Bitcoin increased from $150 USD to $1,000 USD towards the end of 2013. The trade carried out at Mt. Gox by the attackers was not backed by the real coins, which eventually led to the bankruptcy of the exchange.

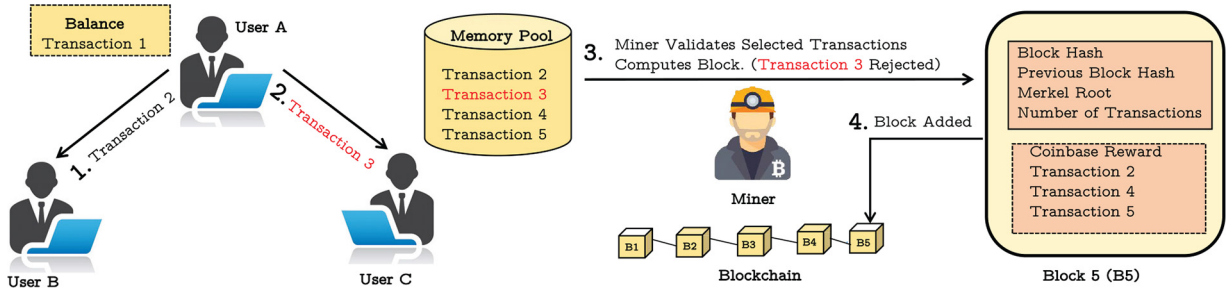


Fig. 18. Double-spending attack carried out by User A. User A has Transaction 1 in his balance. Using that as an input, he generates Transaction 2 and sends it to user B. Then he generates Transaction 3 from the an already spent Transaction 1. When miner queries the mempool, he can either select Transaction 2 or Transaction 3. If Transaction 3 gets rejected, user C suffers the loss.

2) Illegal Activities: Anonymity in Blockchain-based cryptocurrencies provides lucrative opportunities for miscreants to carry out fraudulent activities. As such, cryptocurrencies have become a popular source of funds transfer for illicit activities associated with the *Deep Web* [179], [180]. Since the use of fiat currency leaves traces on Blockchains that can be tracked by law enforcement, cryptocurrencies on the other hand, preserve the anonymity of the user. This is a key reason why various countries have banned the use of cryptocurrencies [181].

Blockchains are tamper-proof, append-only, and decentralized; once a transaction is committed, it cannot be reversed. This has led to various irreversible scam activities online, where users are tricked into sending money through Bitcoin ATMs. Furthermore, the absence of a central authority makes it harder to claim fraud and expect reimbursement. Therefore, design constructs of Blockchain applications can be exploited to facilitate cyber crimes and online frauds.

B. Double-Spending

In cryptocurrencies, double-spending refers to the use of a one-time transaction twice or multiple times. To illustrate double-spending with an example, consider the following scenario. In cryptocurrency operations, a transaction transfers the ownership of asset from a sender's address to the receiver's public address, and the value of the transaction is signed by the signer with a private key. Once the transaction is signed, it is broadcast to the network upon which the receiver validates the transaction. The validation at the recipient's end happens when the receiver looks up the unspent transaction output of the sender, verifies the sender's signature, and waits for the transaction to be mined into a valid block. The process takes a few minutes depending upon the size of the mempool, the throughput of the network, priority factor of the transaction, and the block computation time of the cryptocurrency. In Bitcoin, the average time of block mining is 10 minutes.

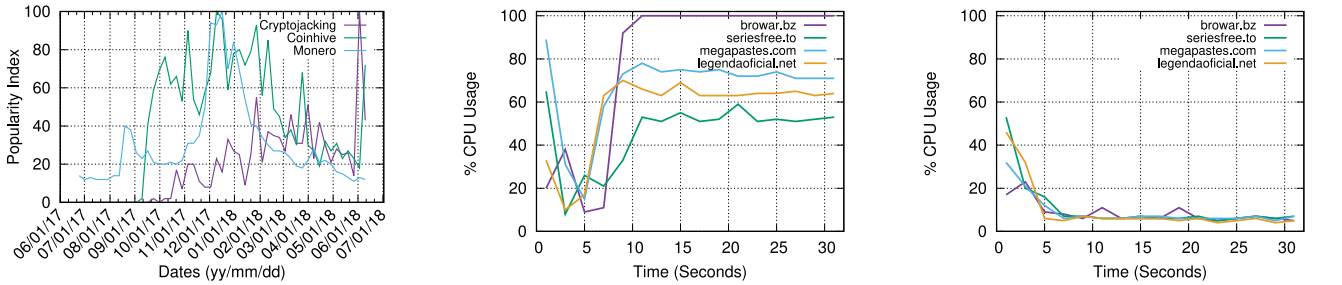
In an environment of fast transactions [54], [182] or if a receiver is optimistic, he may release the product to the sender before the transaction gets mined into the Blockchain. As such, this gives the sender an opportunity to sign the same transaction and send it to another recipient. This behavior of signing the same transaction with a private key and sending it to two different receivers is known as double-spending. In double-spending there are two transactions derived from the same unspent transaction output of the sender, and only one of them gets incorporated into the Blockchain. In Figure 18, we illustrate how a double-spending attack can be carried out in a cryptocurrency. Consensus delay in the network Section V-F, BGP attacks, flood attack on mempools, or the 51% attack Section V-B can cause additional latency in the verification and propagation process, which increase the chances of an adversary to perform double-spending. In March 2013, due to a soft fork, a successful double-spending transaction worth $10,000 USD was carried out in Bitcoin.

C. Cryptojacking

Cryptojacking is a form of an attack that is launched on Web and cloud-based services to illegally perform PoW for Blockchain-based cryptocurrencies without consent [183], [184]. The most recent as well as the most prevalent form of cryptojacking is the in-browser cryptojacking, which turns websites into mining pools [185]. PoW requires processor intensive mathematical calculations which usually involves finding a target hash value. As the aggregate hash rate of the cryptocurrency network increases, the associated difficulty to compute a block also increases. To meet the difficulty requirements, sophisticated hardware such as GPUs and ASIC chips [113] are used by miners. Mining pools expand their hashing capability by inviting more miners to join their pool and purchasing expensive hardware with better computation capabilities. As a result, the mining process in major Bitcoin systems becomes an expensive and competitive game that prevents small miners from mining blocks independently.

1) Cloud-Based Cryptojacking: To compensate for that, malicious miners have found a way of to expand their hash power by hijacking processors of remote devices for mining. This attack is known as the covert mining, or cryptojacking attack, and it involves hijacking a target device to perform PoW calculations for the attacker. Initially, these attacks were launched against cloud service providers, where malicious users performed covert mining operations on virtual machines and exhausted cloud resources. This behavior was first noticed by Tahir *et al.* [41], where they also proposed countermeasures in the form of a software tool called "MineGuard" to effectively detect and stop covert mining operations in cloud.

2) Web Cryptojacking: Cryptojacking was brought to the Web in 2017, and has been soaring in popularity as shown in Figure 19(a). Web-based cryptojacking is used by attackers who inject malicious JavaScript code into websites that

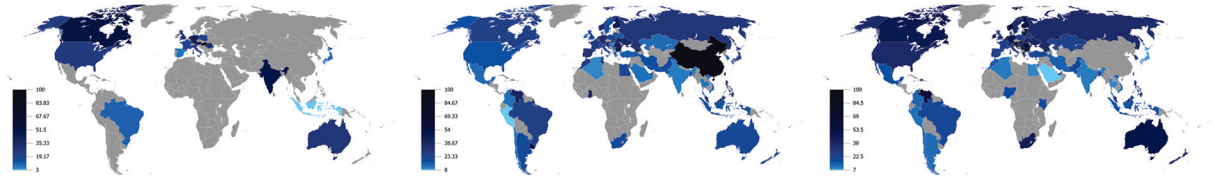


(a) Google popularity index

(b) Percentage CPU usage by four cryptojacking websites when JavaScript is enabled

(c) Percentage CPU usage by four cryptojacking websites when JavaScript is disabled

Fig. 19. (a) Shows the Google search index for the terms "Cryptojacking", "Coinhive", and "Monero." Notice that towards the end of 2017, there has been a rise in the Google search for the three terms which coincides with timing of large scale cryptojacking attack. In (b) and (c) we show the effect of four cryptojacking websites with an without JavaScript enabled. Cryptojacking consumes high CPU power upto 100% which can affect critical CPU operations.



(a) Cryptojacking (b) Coinhive (c) Monero

Fig. 20. Heatmap of the global distribution of Google searches for each term. Notice that U.S. is the most prevalent country in all three search results. Moreover there is more similarity in the search for Coinhive and Monero.

```
<script
    src="./Welcome_files/coinhive.min.js"></script>
<script>
        var miner = new
            coinhive.Anonymous("attacker-key",
        {throttle: 0.1});
        miner.start();
</script>
```

Fig. 21. Malicious JavaScript code that links a website to Coinhive.

secretly mine tokens without the consent of their visitors. In browser-based cryptojacking, the Web browser on the client device executes JavaScript code that establishes a WebSocket connection with a remote dropzone server. The server then sends the target to the client, which computes hashes for PoW and transmits them back to the server. During this process, the device owner remains unknown of this background activity and seamlessly continues to browse the website. In-browser cryptojacking not only poses a major privacy threat, it also harms the performance of the visiting device, since PoW-based hash computations are processor-intensive and may lead to excessive CPU usage and battery drainage. To further facilitate these attacks, online platforms such as coinhive and crypto-loot [186], [187] emerged in 2017, to provide simple code snippets for the attackers and website owners. Those services bind websites with their platform service and perform cryptojacking on the website's visitors machines.

Coinhive is the most popular platform for cryptojacking attacks on websites, and it is linked to the cryptocurrency called Monero [188]. In Figure 21, we provide the JavaScript cryptojacking code used by attackers to bind victim website to their account at Coinhive. The code listing shows that when a browser loads *coinhive.min.js* file, it establishes a WebSocket connection with coinhive server and passes the attacker's key to bind with the dropzone server. It then receives a *target* and submits the corresponding *hashes* to the server over the same socket connection [189]. The throttling parameter controls the hash rate of the victim device and is adjustable to the requirement of the attacker.

In Figure 19(b) and Figure 19(c), we plot the processor usage of four cryptojacking websites with JavaScript enabled and disabled. It can be noticed that each website uses different CPU power when JavaScript is enabled, indicating varying thresholds of throttling parameters. Figure 19 also shows that when the JavaScript is disabled, the browser cannot execute the malicious script and is unable to perform cryptojacking.

In-browser cryptojacking is a relatively new attack related to the PoW-based Blockchain applications, therefore no prior research is available that looks into the operations and effects of this attack. However, owing to the incidents reported in the news, it can be inferred that cryptojacking is becoming popular over time. In Figure 19(a), we show the popularity index of the terms "Cryptojacking", "Coinhive", and "Monero", as recorded by Google analytics based on the search count [190]. The results in Figure 19(a), show that since October 2017, there has been a rise in the search for each term, indicating the interest shown by the users in cryptojacking. Additionally, in Figure 20, we show the global distribution of these searches.

3) Case Studies: Cryptojacking is considered as an emerging threat to the security and privacy of Bitcoin systems, by the research community. Symantec's latest Internet Security Threat Report (ISTR) reveals that cryptojacking attacks on websites have increased by 8500% during 2017 [191]. In February 2018, a large scale cryptojacking attack was launched that compromised more than 4000 websites worldwide

including the websites of U.K. National Health Service (NHS) and U.S. Federal Judiciary [189]. U.K.'s National Cyber Security Centre (NCSC) has declared cryptojacking a "significant threat" in its yearly cyber security report [192].

D. Wallet Theft

Where credentials, such as keys, associated with peers in the system are stored in a digital wallet, the "wallet theft" attack arises with certain implications on the application. For example, in Bitcoin, the wallet with spendable balance is stored in *wallet.dat* file in the data directory of Bitcoin software. Before 2014, the wallets were stored unencrypted in the data directory, vulnerable to attacks. An adversary with access to the data directory could spend the transactions on behalf of the legitimate owner. Even when a wallet is safely guarded on the host, launching a malware attack on the host will allow the adversary to steal the wallet. A recent work by Volety *et al.* [193] analyzed the security of two Bitcoin wallet software applications, suggesting the possibility of obtaining access to Multibit HD and Electrum wallets through an offline brute-force password attempt. In a similar direction, Vasek *et al.* [194] conducted a large-scale measurement study to analyze the usage of *brain wallets* in Bitcoin. The *brain wallets* have a usability benefit for Bitcoin users, freeing them from storing their private keys on untrusted hardware. However, these *brain wallets* can be drained by adversaries by brute-force attacks. The study by Vasek *et al.* [194] showed that among 881 *brain wallets*, all but 21 were drained, indicating a high risk potential.

Case studies: In December 2017, $63 million USD worth bitcoins were stolen from the wallet of a cryptocurrency company, NiceHash [195]. During the hack, the entire contents in NiceHash's Bitcoin wallet were stolen. In November 2017, Tether Treasury wallet was hacked and $31 million USD worth bitcoins were stolen from it. Also in November 2017, $280 million USD worth of ether was locked up after a user deleted the code in the digital wallet hosted by a company named Parity Technologies. In July 2016, the social media Blockchain "Steemit" was attacked and $85,000 USD worth digital currency was stolen from 260 accounts. In January 2015, Bitstamp's Bitcoin wallet was hacked, resulting in a loss of $5.1 million USD worth bitcoins [196].

1) Key Exposure and Theft: A well-known problem in Blockchain-based cryptocurrencies is private key exposure and theft. If the attacker acquires the private key belonging to a user, he can sign and generate a new transaction on behalf of the user, and possibly spend his balance to unauthorized recipients. Brengel and Rossow [197] studied the key leakage in Bitcoin by studying the Bitcoin Blockchain for ECDSA nonce reuse. Their results show that ECDSA nonce reuse is misused in Bitcoin to generate transactions on behalf of users. Similarly, Breitner and Heninger [198] performed cryptanalytics attacks on Bitcoin, Ethereum, and Ripple to expose their private keys. They used a lattice-based algorithm to compute private ECDSA keys that were used in biased signatures.

2) Software Client Vulnerabilities: Public Blockchain applications such as Bitcoin and Ethereum have open-source software clients that enable users to connect with the network. Over time, new software versions are released, implementing new rules and upgrades. An upgrade is also released to patch vulnerability in an old version. In Bitcoin, the Bitcoin Core v0.15 and below are vulnerable to denial-of-service attacks. This vulnerability was patched in the newly released v0.16. However, not all nodes download the newly released version. They continue with the old software client and remain exposed to its vulnerabilities. In Table IX, we show the diversity in adoption of a Bitcoin software client. Notice that only 36.28% of the nodes are using the most updated software version that is immune to the denial-of-service attack.

TABLE IX
TOP 5 SOFTWARE VERSIONS USED BY BITCOIN FULL NODES ALONG WITH THEIR RELEASE DATE, LAG FROM THE DATE OF COLLECTION IN DAYS, AND PERCENTAGE OF USERS. THE RECENT VERSION 0.16.0 HAS NOT BEEN ADOPTED BY THE MAJORITY OF NETWORK AS YET

Index	Version	Release Date	Lag	Users %
1	B. Core v0.16.0	02-26-2018	59	36.28%
2	B. Core v0.15.1	11-11-2017	166	27.52%
3	B. Core v0.15.0.1	09-19-2017	219	5.01%
4	B. Core v0.14.2	06-17-2017	313	4.67%
5	B. Core v0.15.0	04-22-2017	369	2.05%

Moreover, the open-source code can be exploited by an adversary to release a new update with a malicious code and bugs. If a user installs the software, it can provide access to the attacker who can launch various attacks including DDoS, balance theft, etc.. It is therefore necessary to download the software client from a trusted platform.

Bitcoin and Ethereum nodes use the Remote Procedure Call (RPC) API to communicate data between a local node server and a client application. These API calls can be used to generate transactions, establish connections, monitor network status, retrieve data from the Blockchain ledger, etc. As such, an insecure API design can lead to vulnerabilities and attacks. Wang *et al.* [199] analyzed such vulnerabilities and exploits in the official Go-version Ethereum client (geth). They also demonstrated the attacks in an Ethereum testbed.

E. Attacks in Smart Contracts

As new applications are built on top of Blockchain, their own limitations along with Blockchain vulnerabilities, create a new attack surface. Smart contracts belong to the generation of Blockchain 2.0 and in this section, we will explore the attack possibilities in smart contracts. The most well known smart contract application in digital world is Ethereum, which uses Solidity programming language for coding contracts. Solidity [200] is a contract oriented language, influenced by Javascript, Python and C++. Deficiencies in programming language, execution environment, and coding style can lead to a series of attacks. In Figure 22, we demonstrate a vulnerable smart contract code that steals a sender's balance. "The DAO" had a similar vulnerability in their smart contract which resulted in a loss of $50 million USD. Some of the well known attacks on Ethereum smart contracts include reentrancy attack, over and under flow attacks, replay attacks, short address attacks and reordering attacks [43], [201].

```
// Vulnerable Smart Contract
mapping (address -> uint) private userBalance;
function withdraw() public {
    uint WithdrawAmount = userBalance[msg.sender];
if (!(msg.sender.call.value(WithdrawAmount)())) {
    throw; } // Caller's code executed and it can
    make recursive call to withdraw() again.
    userBalance[msg.sender] = 0;
}
```

Fig. 22. Reentrancy attack on smart contract code [42]. A major problem with calling external contracts is that they alter the control flow of the code that the running contract does not anticipate. In this contract, an external call is made before the user's balance is set to 0.

```
// DoS attack
contract Malicious_Auction {
    address presentLeader;
    uint maxBid;
    function bid() payable {
        require(msg.value > maxBid);
        require(presentLeader.send(maxBid)); //
            Refund the old leader, if it fails then
            revert
        presentLeader = msg.sender;
        maxBid = msg.value;}}
```

Fig. 23. DoS attack on a vulnerable smart contract in which the malicious bidder may revert funds to the old leader and prevent other bidders from calling the *bid*() function. As such the malicious bidder remains the leader of the auction for as long as he wants.

1) Reentrancy Attacks: In reentrancy attack, if the user does not update the balance before sending ether, an attacker can steal all the ether stored in the contract by recursively making calls to the *call.value*() method in a ERC20 token. As such, a careless user may lose his entire balance in the contract if he forgets to update his balance.

2) DoS Attacks: DoS attack in smart contract enables a malicious actor to keep funds and authority to himself. Consider an example of a smart contract auction in which a malicious bidder tries to become the leader of an auction illustrated in Figure 24. The vulnerable contract prevents refunds to the old leader of the contract and makes the attacker the new leader. Moreover, it cancels all the *bid*() requests sent by other bidders and keeps the attacker as the leader of the auction. Another form of DoS attack in Ethereum smart contract involves exploiting the gas limit set by the contract Figure 23. In Ethereum, if the overall gas consumed by the smart contract upon execution exceeds the gas limit, the contract transaction fails. An attacker can exploit this by adding multiple addresses with refund needs. Upon execution, the gas required to refund those addresses may exceed the total gas limit, thereby cancelling the final transaction.

3) Overflow Attacks: An overflow in a smart contract happens when the value of the *type variable* (2^{256}) is exceeded. For instance, in a smart contact of online betting, if someone sends large amount of ether, exceeding (2^{256}), the value of the bet would be set to 0. Although exchange of an ether value greater than (2^{256}) is unrealistic, but it remains a programming vulnerability in smart contracts written in Solidity.

4) Short Address Attacks: The short address attack exploits a bug in Ethereum's virtual machine to make extra tokens

```
// DoS attack on Gas Limit
struct Payee {
    address addr;
    uint256 value;}
Payee[] payees;
uint256 nextPayeeIndex;
function payOut() {
    uint256 i = nextPayeeIndex;
    while (i < payees.length && msg.gas > 200000) {
      payees[i].addr.send(payees[i].value);
      i++;}
    nextPayeeIndex = i;}
```

Fig. 24. DoS attack exploiting the gas limit in a vulnerable smart contract. The attacker initiates a list of addresses that demonstrate the need for the refund. Once all the addresses are refunded, the overall gas used by the smart contract exceeds its gas limit.

```
mapping (address => uint256) public userBalance;
// Vulnerable code
function transfer(address _to, uint256 _value) {
    // Check if the sender has sufficient balance
    require userBalance[msg.sender] >= _value);
    // Compute new balance
 userBalance[msg.sender] -= _value;
 userBalance[_to] += _value; }
```

Fig. 25. Overflow attack. When the sender's balance is being checked, the contract code does not take into the account if the balance exceeds the value 2^{256}. In such a case, the balance will be set to 0 by default and overflow attack can be launched.

```
contract Vulnerable {
    function () payable {
        revert(); }
     function somethingBad() {
        require(this.balance > 0);
        // Do something bad
        }}
```

Fig. 26. Vulnerable contract code that allows forcible balance transfer to the contract without a fallback function.

on limited purchases. The short address attack is mostly applicable on ERC20 tokens. For this attack, the attacker creates an Ethereum wallet ending with 0 digit. Then he makes a purchase on the address by removing the last 0. If the contract has a sufficient balance, then the *buy function* does not check the sender's address and Ethereum's virtual machine appends missing 0 to complete the address. As a result, for each 1000 tokens bought, the machine returns 256000 tokens.

5) Forcible Balance Transfer: In vulnerable smart contract codes, forcible balance transfer to the contract can occur without a fallback function. This can be used to exhaust the gas limit and disallow the final transaction.

6) Gas Cost Attacks: In Ethereum, the gas consumption ensures that smart contracts, executed in the Ethereum Virtual Machine (EVM), eventually terminate. If the gas cost is not configured properly, an adversary can exploit it to prevent the termination of the smart contracts, leading to a DoS attack. Although Ethereum has taken measures to prevent this attack, Chen *et al.* [202] show that the changes are insufficient and exploitable to under-priced operations. To address this problem, they also proposed an adaptive gas cost mechanism to defend against the under-priced DoS attack.

F. Replay Attacks

The replay attack involves making one transactions on two different Blockchains [203], [204]. For instance, when a cryptocurrency forks into two separate currencies, users hold equal assets on both ledgers. A user has an option of carrying out a transaction on any one of the two chains. In replay attacks, the attacker sniffs the transaction data on one ledger and replays it on the other ledger. As such, the user loses assets on both chains. A simple case can be drawn from Ethereum.

In Ethereum, a transaction signed on one Blockchain is valid on all Ethereum-based Blockchains that do not enable the *chainID*. Therefore, a transactions made on a test network can be replicated on the public network to steal funds. Although Ethereum has taken countermeasures to prevent replay attacks by incorporating *chainID* in transactions, users who do not enable this wallet feature remain vulnerable.

G. Countering Blockchain Applications Attacks

Attacks on Blockchain applications have various possible countermeasures. For example, it is advised to keep backups of the wallet and secure the keys used for signing transactions. Passwords are easy to compromise, and using a strong password is required as a defence against brute-force attack. However, changing passwords does not change the keys secured by them, making those keys vulnerable due to a previous compromised password. Wallet encryption, a standard practice in the original Bitcoin design, is highly recommended to cope with vulnerable keys.

To address the issues with un-encrypted wallets in Bitcoin, an improvement proposal (BIP38) was released in 2012 with the title "Passphrase-protected Private Key" [205]. Although the proposal was not implemented in Bitcoin Core, standalone tools of BIP38 are made available to harden the security of Bitcoin wallets. The Bitcoin documentation also provides methods of securely managing the Bitcoin wallets [206]. Among them, the most highly recommended ways of wallet management include using hardware wallets or multi-signature wallets. Hardware wallets are special-purpose devices for safely storing bitcoins on a peripheral that is trusted by the user to sign transactions. Multi-signature wallets use multiple private keys to move bitcoins instead of using a single key, thereby adding extra layers of protection.

New models of cryptocurrency, such as "Zcash", provide chain anonymity to the transactions, the users, and the amount exchanged. As such, the shielded architecture of "Zcash" Blockchain prevents block ingestion attacks. The double-spending attack is easily addressed in fast networks, but not when the network is characterized by high latency and longer block mining times. One possible approach to deal with the problem is utilizing one-time (or a few time) signatures, such as XMSS [85], which reveal the private key of the user if he tries to double-spend. However, this requires the change in the current signature algorithms that Blockchain applications have used. Other proposals include reducing the difficulty parameter of a Blockchain to enable swift block mining, which is a reasonable approach, except that it would further facilitate selfish mining and the rate of stale blocks.

All major attacks on smart contracts in Ethereum are either related to the vulnerabilities in programming platforms or careless programming practices. These attacks can be prevented by patching vulnerabilities in Ethereum virtual machine (EVM) and avoiding programming mistakes in smart contracts [43].

To counter covert mining in cloud, Tahir *et al.* [41] proposed "MineGuard" that detects anomalous use of processor in Virtual Machines. To mitiage in-browser cryptojacking, reputable Web browsers, including Chrome and Firefox have, launched Web extensions that actively detect WebSocket connections that trasnmit PoW [207]–[209]. However, as of now, the extensions use a blacklisting approach to spot the WebSocket traffic which has its limitations. For example, an attacker with access to the blacklist can easily circumvent detection by using a relay server between the host and the dropzone server. As of now, cryptojacking and its defences are open challenges and require more attention from the community.

VII. Discussion and Future Directions

Blockchains have become popular in recent years owing to the increasing use of decentralized systems and the growing need for tamper-proof data management. As such, they are being used in several domains such as IoT, health care, electronic voting, e-government solutions, and supply chain [210]–[217]. However, prior to the integration of such legacy systems with Blockchains, it is pertinent to fully understand their security properties and the attack surface. It might be possible that a conventional application, hoping to improve its security model, may further be exposed to a higher risk by using Blockchains. For example, delay-sensitive applications in supply chains cannot afford unusual latency in transaction propagation and data-sensitive applications such as electronic voting cannot afford a double-spent transaction. While these attacks might be infeasible in conventional client-server model, using Blockchains might create new attack avenues for them. An adversary can launch consensus delay attacks to stall information propagation in the supply chain or create a double-spent transaction to invalidate the vote of a legitimate user. Moreover, as mentioned in Section IV, once a fraudulent activity is part of the Blockchain, the system will require a major hard fork to reverse the transaction. Therefore, the use of Blockchains may bring new attack avenues on an otherwise secure application. In the light of these changes, we believe it is important and timely to perform a systematic treatment of Blockchain attack surface to expose its vulnerabilities and outline new threat models for emerging applications. As an outcome of our research, in the following, we discuss the key lessons learned as well as the future directions that can navigate the future research.

A. Blockchain Structure Attacks

1) Key Lessons: The goal of attacks on the Blockchain structure is to present to participants inconsistent views of the Blockchain state. The impact of these attacks depend on the period of time for the inconsistent state and the actions taken based on the inconsistent state. The attacks are typically

realized by generating intentional soft or hard forks, stale blocks and orphaned blocks. The primary reason for the generation of forks is due to the introduction of new rules and likelihood of peers switching to those rules. The likelihood of forks will reduce if peers do not switch back and forth between the rules. The occurrence of stale blocks and orphaned blocks can reduced by limiting the instances of race conditions in the mining process. Finally, the exploitation of vulnerabilities in the consensus protocols has a huge impact on the Blockchain structure. We have identified specific instances where vulnerabilities in PoW, PoS, and PBFT have been exploited resulting in either delaying or aborting the transaction validation process.

2) Future Directions: Recently, researchers have proposed approaches that provide insights into addressing the Blockchain forks, stale blocks, orphaned blocks, and resilient consensus protocols. For example, the fork problem has been addressed by a corking mechanism [218]. Insights into understanding attack surfaces for permissioned Blockchain platforms has been presented that could influence design of resilient permissioned Blockchain platforms [219].

One of the pressing needs is to systematically characterize and evaluate the performance of Blockchain protocols in asynchronous networks. Current research on this topic suggests that bounds on the majority attacks can be reduced in asynchronous networks. However, these findings are yet to be validated in the real-world settings to accurately estimate the bounds for the majority attack. Towards that, it is important to identify the mining and non-mining nodes in PoW-based Blockchains, construct the network topology, estimate delays in the block propagation, and observe the requirements for the majority attacks.

Another takeaway from our work is the need to develop energy efficient and secure consensus protocols that may substitute PoW and PoS. Through Bitcoin, we have learned that PoW is highly energy inefficiency. Moreover, PoW also leads to the race conditions in Blockchains in which miners compete for block rewards [220]. The race condition eventually facilitates attacks such as selfish mining, the 51% attack, double-spending, forks, and stale blocks. To address the energy inefficiency and avoid race conditions, PoS has been proposed that uses an auction process for block mining. However, we have shown that PoS can create network centralization and unfairness in system. Also, recently, stake bleeding attacks have been presented as a new attack surface that inflicts PoS protocols [123]. While PBFT offers an alternative to both PoW and PoS, it has some major shortcomings. PBFT-based Blockchains have a low fault tolerance, thus are more vulnerability to Sybil attacks. If an adversary plants Sybil nodes that account for one-third of the system, they can eventually prevent the consensus over the Blockchain. Moreover, PBFT-based private Blockchains have a high message complexity and low-scalability. As a result, the network size does not expand beyond a few hundred nodes. This makes it easier for the adversary to introduce Sybil nodes to prevent consensus. As such, low scalability, high message complexity, and low fault tolerance are some of the key challenges in PBFT-based private Blockchains. Currently, efforts are made to improve the fault tolerance and scalability of PBFT-based Blockchains [221]. With the knowledge of the attack surface in consensus protocols, one could design a meta-consensus process that would lead to a hybrid consensus. In the hybrid consensus process, the network designer can switch the consensus protocols by balancing risks and other notions, such as the quality of service. In addition, resilience can be incorporated into the hybrid consensus process to ensure that the primary Blockchain processes meet acceptable quality of service. Efforts such as attack-tolerant agreement algorithms provide insights to achieve a resilient consensus process.

B. Blockchain Peer-To-Peer System Attacks

1) Key Lessons: From our analysis, we noticed that the peer-to-peer architecture of Blockchains is the most dominant class of the Blockchain attack surface. In public Blockchains particularly, the topological asymmetry of the network can be easily exploited to compromise the system. Moreover, and since the public Blockchains are permissionless, the network remains impartial towards legitimate users as well as the adversary. This property further weakens the security model since the adversary has an open access to all the resources.

Moreover, the network layer allows external entities to influence the internal operations of the Blockchain application. For example, an ISP, external to the Blockchain network, can hijack BGP prefixes to isolate peers. If such an attack is launched against a mining pool, the hash rate of the network will be affected leading to transaction stall. Other external adversaries include competing Blockchain applications, nation states, cloud service providers, and DNS servers, that can disrupt the flow of traffic to affect the activities of the target Blockchain application. While the effect of external entities can be reduced by using private Blockchains, however, this may only partially solve the problem. Private Blockchains can strengthen the network conditions by limiting the exposure of system information, however, they also limit the scope of the application by allowing selective peers to participate.

2) Future Directions: Some of the open challenges in the Blockchains attack surface are shown in Table III. There have been new insights into exploiting and mitigating the attack surfaces in mining process. For instance, further analysis of Block withholding attacks for rational uncooperative miners that present a new threat to the Blockchain mining process [222]. The problem of random mining group selection has the potential to prevent 51% attacks. An effort to conduct silent timestamping has proven to be effective to mitigate the attack surfaces in the mining process [223].

It can be observed that routing attacks do not have effective countermeasures and current Blockchain applications have not taken initiatives to address them. For example, as shown in Table VIII, if a malicious ISP hijacks ASes owned by *Alibaba*, it can hijack more than 50% of the Bitcoin hash rate. As a result, block generation in Bitcoin will stall, leading to delays in transaction confirmation. If we analyze the spatial behavior of Bitcoin [147], we observe an increase in the centralization of nodes over time, indicating that the network is not responding to the threats of a hijack.

C. Blockchain Application Attacks

1) Key Lessons: Blockchain applications have been subject to several attacks with double spending being the dominant threat. Double spending has manifested in several forms over the years. Recently, double spending was realized using a sybil attack in bitcoin network [224]. We have also shown that the increasing programming flexibility of smart contracts have made conventional Blockchain applications more vulnerable. In Ethereum, for example, the reentrancy attack and the overflow attack can be launched to steal the user's balance. Such attacks cannot be launched on Bitcoin, Ripple, and Zcash which do not offer programming flexibility to users. Additionally, we have reported that the use of a Blockchains at the application layer also creates new attack avenues. For example, by exploiting the open-source client software, an attacker can get access to his private keys and balance. Therefore, the application-oriented use of Blockchains needs to be carefully addressed to avoid attacks.

2) Future Directions: Also shown in Table III, certain policies of Blockchain applications have created attack avenues that remain an open problem. There have been attempts at mitigating double spending attacks. For instance, an approach to use recipient oriented transactions has proved successful in preventing double spending attacks in private Blockchain [225]. A scalable and proactive solution to leverage Software Defined Networking's orchestration features for protecting Blockchain applications has been proposed [226]. In Bitcoin and Ethereum, the block size limit and the block generation time have led to flooding attacks and delays [34]. These applications should revise their policies to prevent such attacks. Furthermore, a developing problem that most Blockchain applications are likely to encounter in future is their high storage footprint. Due to the append-only model, Blockchains linearly grow in size leading to a high storage cost. While this problem appears trivial in cryptocurrencies, it will become significant when Blockchains will be introduced in data intensive applications such as supply chains. A naïve solution is the use of payment channel networks to offload the transaction activity from the main Blockchain [227], [228]. However, the use of payment channels obscures the data transparency on the main Blockchain and may also suffer from privacy issues. Therefore, more research is required to come up with effective solutions.

In summary, the key takeaways of our work point towards: 1) more secure deployment of Blockchains in distributed environment, 2) development of fair and efficient consensus algorithms, and 3) careful interaction of Blockchain layer with the application layer to avoid vulnerabilities and attacks.

VIII. Conclusion

In this paper, we explore the attack surface of Blockchain technology. We attribute attacks to the cryptographic constructs of the blockchain, the underlying communication architecture, and the context in which they are applied. In doing so, we highlight major threats and ongoing defense research activities. We believe that various attacks against Blockchain can be still launched, not withstanding the current and existing defenses, and that some of those attacks can be used to facilitate several others. By outlining these attacks and surveying their countermeasures, we highlight new research directions that need to be pursued towards more secure and effective use of Blockchains.

References

[1] L. Mauri, S. Cimato, and E. Damiani, "A comparative analysis of current cryptocurrencies," in *Proc. 4th Int. Conf. Inf. Syst. Security Privacy (ICISSP)*, Jan. 2018, pp. 127–138. [Online]. Available: https://doi.org/10.5220/0006648801270138

[2] G. Danezis and S. Meiklejohn, "Centrally banked cryptocurrencies," in *Proc. Annu. Netw. Distrib. Syst. Security Symp. (NDSS)*, San Diego, CA, USA, Feb. 2016, p. 502. [Online]. Available: http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2017/09/centrally-banked-cryptocurrencies.pdf

[3] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "SoK: Research perspectives and challenges for bitcoin and cryptocurrencies," in *Proc. IEEE Symp. Security Privacy*, San Jose, CA, 2015, pp. 104–121. [Online]. Available: https://ieeexplore.ieee.org/document/7163021

[4] A. E. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "HAWK: The blockchain model of cryptography and privacy-preserving smart contracts," in *Proc. 37th IEEE Symp. Security Privacy (Oakland)*, San Jose, CA, USA, May 2016, pp. 839–858. [Online]. Available: https://doi.org/10.1109/SP.2016.55

[5] K. Bhargavan *et al.*, "Formal verification of smart contracts: Short paper," in *Proc. 23rd ACM Conf. Comput. Commun. Security (CCS)*, Vienna, Austria, Oct. 2016, pp. 91–96. [Online]. Available: http://doi.acm.org/10.1145/2993600.2993611

[6] P. K. Sharma, S. Rathore, and J. H. Park, "DistArch-SCNET: Blockchain-based distributed architecture with Li-Fi communication for a scalable smart city network," *IEEE Consum. Electron. Mag.*, vol. 7, no. 4, pp. 55–64, Jul. 2018. [Online]. Available: https://doi.org/10.1109/MCE.2018.2816745

[7] K. Fan, Y. Ren, Y. Wang, H. Li, and Y. Yang, "Blockchain-based efficient privacy preserving and data sharing scheme of content-centric network in 5G," *IET Commun.*, vol. 12, no. 5, pp. 527–532, Mar. 2018. [Online]. Available: https://doi.org/10.1049/iet-com.2017.0619

[8] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11676–11686, 2018. [Online]. Available: https://doi.org/10.1109/ACCESS.2018.2801266

[9] D. Rakic, "Blockchain technology in healthcare," in *Proc. 4th Int. Conf. Inf. Commun. Technol. Ageing Well e-Health*, Mar. 2018, pp. 13–20. [Online]. Available: https://doi.org/10.5220/0006531600130020

[10] E. F. Jesus, V. R. L. Chicarino, C. V. N. de Albuquerque, and A. A. de A. Rocha, "A survey of how to use blockchain to secure Internet of Things and the stalker attack," *Security Commun. Netw.*, vol. 2018, pp. 1–9, Apr. 2018. [Online]. Available: https://doi.org/10.1155/2018/9675050

[11] P. K. Sharma, S. Singh, Y.-S. Jeong, and J. H. Park, "DistBlockNet: A distributed blockchains-based secure SDN architecture for IoT networks," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 78–85, Sep. 2017. [Online]. Available: https://goo.gl/UBv1Sf

[12] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1676–1717, 2nd Quart., 2019. [Online]. Available: https://doi.org/10.1109/COMST.2018.2886932

[13] H. Hyvärinen, M. Risius, and G. Friis, "A blockchain-based approach towards overcoming financial fraud in public sector services," *Bus. Inf. Syst. Eng.*, vol. 59, no. 6, pp. 441–456, 2017. [Online]. Available: https://doi.org/10.1007/s12599-017-0502-4

[14] F. Holotiuk, F. Pisani, and J. Moormann, "The impact of blockchain technology on business models in the payments industry," in *Proc. Leadership Digit. Transf. Internationale Tagung Wirtschaftsinformatik*, Feb. 2017, p. 6. [Online]. Available: http://aisel.aisnet.org/wi2017/track09/paper/6

[15] E. Heilman, F. Baldimtsi, and S. Goldberg, "Blindly signed contracts: Anonymous on-blockchain and off-blockchain bitcoin transactions," in *Proc. Financ. Cryptography Data Security Int. Workshops BITCOIN VOTING WAHC*, Feb. 2016, pp. 43–60. [Online]. Available: https://doi.org/10.1007/978-3-662-53357-4_4

[16] G. G. Dagher, P. B. Marella, M. Milojkovic, and J. Mohler, "BroncoVote: Secure voting system using Ethereum's blockchain," in *Proc. 4th Int. Conf. Inf. Syst. Security Privacy (ICISSP)*, Jan. 2018, pp. 96–107. [Online]. Available: https://doi.org/10.5220/0006609700960107

[17] F. S. Hardwick, A. Gioulis, R. N. Akram, and K. Markantonakis, "E-voting with blockchain: An E-voting protocol with decentralisation and voter privacy," in *Proc. IEEE Int. Conf. Internet Things (iThings)*, Halifax, NS, Canada, 2018, pp. 1561–1567.

[18] M. M. Eljazzar, M. A. Amr, S. S. Kassem, and M. Ezzat, "Merging supply chain and blockchain technologies," 2018. [Online]. Available: http://arxiv.org/abs/1804.04149.

[19] G. Baruffaldi and H. Sternberg, "Chains in chains—Logic and challenges of blockchains in supply chains," in *Proc. 51st Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2018, pp. 1–8. [Online]. Available: http://aisel.aisnet.org/hicss-51/in/digital_supply_chain/3

[20] N. Fotiou and G. C. Polyzos, "Decentralized name-based security for content distribution using blockchains," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM)*, San Francisco, CA, USA, Apr. 2016, pp. 415–420. [Online]. Available: https://doi.org/10.1109/INFCOMW.2016.7562112

[21] M. Zhang and Y. Ji, "Blockchain for healthcare records: A data perspective," *PeerJ PrePrints*, vol. 6, Mar. 2018, Art. no. e26942. [Online]. Available: https://doi.org/10.7287/peerj.preprints.26942v1

[22] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *Proc. 18th IEEE Int. Conf. e-Health Netw. Appl. Services*, Munich, Germany, Sep. 2016, pp. 1–3. [Online]. Available: https://doi.org/10.1109/HealthCom.2016.7749510

[23] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Symp. Security Privacy Workshops (SPW)*, San Jose, CA, USA, May 2015, pp. 180–184. [Online]. Available: https://goo.gl/kTNim3

[24] A. Back *et al.* (2014). *Enabling Blockchain Innovations With Pegged Sidechains*. [Online]. Available: http://kevinriggen.com/files/sidechains.pdf

[25] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[26] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "P2P mixing and unlinkable bitcoin transactions," in *Proc. Annu. Netw. Distrib. Syst. Security Symp. (NDSS)*, San Diego, CA, USA, Feb./Mar. 2017, pp. 1–15. [Online]. Available: https://www.ndss-symposium.org/ndss2017/ndss-2017-programme/p2p-mixing-and-unlinkable-bitcoin-transactions/

[27] I. Eyal, "The miner's dilemma," in *Proc. 36th IEEE Symp. Security Privacy (Oakland)*, San Jose, CA, USA, May 2015, pp. 89–103. [Online]. Available: https://doi.org/10.1109/SP.2015.13

[28] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Proc. 13th IEEE Int. Conf. Peer-to-Peer Comput. (P2P)*, Sep. 2013, pp. 1–10. [Online]. Available: https://doi.org/10.1109/P2P.2013.6688704

[29] *Bitcoin Developer Guide*. Accessed: Sep. 13, 2019. [Online]. Available: https://bitcoin.org/en/developer-guidepeer-discovery

[30] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *Proc. 38th IEEE Symp. Security Privacy (Oakland)*, San Jose, CA, USA, May 2017, pp. 375–392. [Online]. Available: https://doi.org/10.1109/SP.2017.29

[31] Y. Marcus, E. Heilman, and S. Goldberg, "Low-resource eclipse attacks on Ethereum's peer-to-peer network," in *Proc. IACR Cryptol. ePrint Archive*, vol. 2018, 2018, p. 236. [Online]. Available: http://eprint.iacr.org/2018/236

[32] M. Bastiaan. (2015). *Preventing the 51%-Attack: A Stochastic Analysis of Two Phase Proof of Work in Bitcoin*. [Online]. Available: https://goo.gl/nJsMzV

[33] T. Leelavimolsilp, L. Tran-Thanh, and S. Stein, "On the preliminary investigation of selfish mining strategy with multiple selfish miners," 2018. [Online]. Available: http://arxiv.org/abs/1802.02218.

[34] M. Saad, M. T. Thai, and A. Mohaisen, "POSTER: Deterring DDoS attacks on blockchain-based cryptocurrencies through Mempool optimization," in *Proc. Asia Conf. Comput. Commun. Security (ASIACCS)*, Jun. 2018, pp. 809–811. [Online]. Available: https://goo.gl/4kgiCM

[35] I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *Proc. 13th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Santa Clara, CA, USA, Mar. 2016, pp. 45–59. [Online]. Available: https://goo.gl/VGN4yw

[36] C. A. Vyas and M. Lunagaria, "Security concerns and issues for bitcoin," in *Proc. Nat. Conf. Workshop Bioinformat. Comput. Biol. (NCWBCB)*, 2014, pp. 1–18.

[37] A. Kaushik, A. Choudhary, C. Ektare, D. Thomas, and S. Akram, "Blockchain—Literature survey," in *Proc. 2nd IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. (RTEICT)*, Bengaluru, 2017, pp. 2145–2148. [Online]. Available: https://ieeexplore.ieee.org/document/8256979

[38] M. Fleder, M. S. Kester, and S. Pillai, "Bitcoin transaction graph analysis," 2015. [Online]. Available: http://arxiv.org/abs/1502.01657.

[39] T. Bamert, C. Decker, R. Wattenhofer, and S. Welten, "BlueWallet: The secure bitcoin wallet," in *Proc. Int. Workshop Security Trust Manag.*, 2014, pp. 65–80.

[40] I. D. Rubasinghe and D. T. N. K. Zoysa, "Transaction verification model over double spending for peer-to-peer digital currency transactions based on blockchain architecture," *Int. J. Comput. Appl.*, vol. 163, no. 5, pp. 24–31, 2012.

[41] R. Tahir *et al.*, "Mining on someone else's dime: Mitigating covert mining operations in clouds and enterprises," in *Proc. 20th Int. Symp. Res. Attacks Intrusions Defenses (RAID)*, Atlanta, GA, USA, Sep. 2017, pp. 287–310. [Online]. Available: https://doi.org/10.1007/978-3-319-66332-6_13

[42] Ethereum. *Ethereum Contract Security Techniques and Tips*. Accessed: Sep. 13, 2019. [Online]. Available: https://github.com/ethereum/wiki/wiki/Safety

[43] M. Grincalaitis. (Sep. 2017). *The Ultimate Guide to Audit a Smart Contract*. [Online]. Available: https://goo.gl/TD7suo

[44] S. Underwood, "Blockchain beyond bitcoin," *Commun. ACM*, vol. 59, no. 11, pp. 15–17, 2016. [Online]. Available: http://doi.acm.org/10.1145/2994581

[45] I.-C. Lin and T.-C. Liao, "A survey of blockchain security issues and challenges," *Int. J. Netw. Security*, vol. 19, no. 5, pp. 653–659, 2017.

[46] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts SoK," in *Proc. 6th Int. Conf. Principles Security Trust*, vol. 10204, 2017, pp. 164–186. [Online]. Available: https://doi.org/10.1007/978-3-662-54455-6_8

[47] M. C. K. Khalilov and A. Levi, "A survey on anonymity and privacy in bitcoin-like digital cash systems," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2543–2585, 3rd Quart., 2018. [Online]. Available: https://doi.org/10.1109/COMST.2018.2818623

[48] D. Siegel. *Understanding the DAO Attack*. Accessed: Sep. 13, 2019. [Online]. Available: https://www.coindesk.com/understanding-dao-hack-journalists/

[49] C. Baldwin. (Aug. 2016). *Bitcoin Worth 72 Million Stolen From Bitfinex Exchange in Hong Kong*. [Online]. Available: http://reut.rs/2gc7iQ9

[50] F. Memoria. (Nov. 2017). *700 Million Stuck in 115,000 Unconfirmed Bitcoin Transactions*. [Online]. Available: https://www.cryptocoinsnews.com/700-million-stuck-115000-unconfirmed-bitcoin-transactions/

[51] R. McMillan. (2014). *The Inside Story of MT. GOX, Bitcoin's 460 Million USD Disaster*. [Online]. Available: https://www.wired.com/2014/03/bitcoin-exchange/

[52] B. Community. (Oct. 2017). *The 51% Attack*. [Online]. Available: https://learncryptography.com/cryptocurrency/51-attack

[53] C. Pérez-Solà, S. Delgado-Segura, G. Navarro-Arribas, and J. Herrera-Joancomartí, "Double-spending prevention for Bitcoin zero-confirmation transactions," *Int. J. Inf. Security*, vol. 18, pp. 451–463, 2019. [Online]. Available: https://doi.org/10.1007/s10207-018-0422-4

[54] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in *Proc. 19th ACM Conf. Comput. Commun. Security (CCS)*, Raleigh, NC, USA, Oct. 2012, pp. 906–917. [Online]. Available: http://doi.acm.org/10.1145/2382196.2382292

[55] M. Pilkington, "Blockchain technology: Principles and applications," in *Proc. Res. Handbook Digit. Transf.*, 2016, p. 225.

[56] R. Shrestha, R. Bajracharya, A. P. Shrestha, and S. Y. Nam, "A new-type of blockchain for secure message exchange in VANET," *Digit. Commun. Netw.*, to be published. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352864818303092

[57] R. Casado-Vara, F. de la Prieta, J. Prieto, and J. M. Corchado, "Blockchain framework for IoT data quality via edge computing," in *Proc. 1st Workshop Blockchain Enabled Netw. Sensor Syst.*, Shenzhen, China, 2018, pp. 19–24. [Online]. Available: https://doi.org/10.1145/3282278.3282282

[58] A. Dmitrienko, D. Noack, and M. Yung, "Secure wallet-assisted offline bitcoin payments with double-spender revocation," in *Proc. Asia Conf. Comput. Commun. Security (ASIACCS)*, Abu Dhabi, UAE, Apr. 2017, pp. 520–531. [Online]. Available: http://doi.acm.org/10.1145/3052973.3052980

[59] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3416–3452, 4th Quart., 2018. [Online]. Available: https://doi.org/10.1109/COMST.2018.2842460

[60] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K. Tan, "BLOCKBENCH: A framework for analyzing private blockchains," in *Proc. Int. Conf. Manag. Data SIGMOD Conf.*, Chicago, IL, USA, May 2017, pp. 1085–1100. [Online]. Available: https://doi.org/10.1145/3035918.3064033

[61] G. Baralla, S. Ibba, M. Marchesi, R. Tonelli, and S. Missineo, "A blockchain based system to ensure transparency and reliability in food supply chain," in *Proc. Int. Workshops Parallel Process.*, Turin, Italy, Aug. 2018, pp. 379–391. [Online]. Available: https://doi.org/10.1007/978-3-030-10549-5_30

[62] A. Ahmad, M. Saad, M. Bassiouni, and A. Mohaisen, "Towards blockchain-driven, secure and transparent audit logs," in *Proc. Int. Conf. Mobile Ubiquitous Syst. Comput. Netw. Services (MobiQuitous)*, Nov. 2018, pp. 443–448. [Online]. Available: https://doi.org/10.1145/3286978.3286985

[63] A. Ouaddah, A. A. E. Kalam, and A. A. Ouahman, "FairAccess: A new blockchain-based access control framework for the Internet of Things," *Security Commun. Netw.*, vol. 9, no. 18, pp. 5943–5964, 2016. [Online]. Available: https://doi.org/10.1002/sec.1748

[64] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016. [Online]. Available: https://doi.org/10.1109/ACCESS.2016.2566339

[65] A. Miller, I. Bentov, R. Kumaresan, and P. McCorry, "Sprites: Payment channels that go faster than lightning," 2017. [Online]. Available: http://arxiv.org/abs/1702.05812.

[66] J. Lind, O. Naor, I. Eyal, F. Kelbert, P. R. Pietzuch, and E. G. Sirer, "Teechain: Reducing storage costs on the blockchain with offline payment channels," in *Proc. 11th ACM Int. Syst. Storage Conf. (SYSTOR)*, Jun. 2018, p. 125. [Online]. Available: http://doi.acm.org/10.1145/3211890.3211904

[67] L. Lundbaek, A. C. D'Iddio, and M. Huth, "Optimizing governed blockchains for financial process authentications," 2016. [Online]. Available: arXiv:1612.00407.

[68] M. Minaei, P. Moreno-Sanchez, and A. Kate, "R3C3: Cryptographically secure censorship resistant rendezvous using cryptocurrencies," in *Proc. IACR Cryptol. ePrint Archive*, vol. 2018, 2018, p. 454. [Online]. Available: https://eprint.iacr.org/2018/454

[69] G. Bissias, B. N. Levine, A. P. Ozisik, G. Andresen, and A. Houmansadr, "An analysis of attacks on blockchain consensus," 2016. [Online]. Available: http://arxiv.org/abs/1610.07985.

[70] S. Goldberg and E. Heilman, "Technical perspective: The rewards of selfish mining," *Commun. ACM*, vol. 61, no. 7, p. 94, 2018. [Online]. Available: https://doi.org/10.1145/3213006

[71] F. Ritz and A. Zugenmaier, "The impact of uncle rewards on selfish mining in Ethereum," in *Proc. IEEE Eur. Symp. Security Privacy Workshops (EuroS&P W)*, London, U.K., Apr. 2018, pp. 50–57. [Online]. Available: https://doi.org/10.1109/EuroSPW.2018.00013

[72] C. Decker. (Jan. 2018). *cDecker/btcresearch*. [Online]. Available: https://github.com/cdecker/btcresearch

[73] B. Scott. *Bitcoin Academic Research*. Accessed: Sep. 13, 2019. [Online]. Available: https://docs.google.com/spreadsheets/d/1VaWhbAj7hWNdiE73P-W-wrl5a0WNgzjofmZXe0Rh5sg/edit#gid=0

[74] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Gener. Comput. Syst.*, vol. 107, pp. 841–853, 2020. [Online]. Available: https://doi.org/10.1016/j.future.2017.08.020

[75] T. Salman, M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, "Security services using blockchains: A state of the art survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 858–880, 1st Quart., 2019. [Online]. Available: https://doi.org/10.1109/COMST.2018.2863956

[76] L. Anderson, R. Holz, A. Ponomarev, P. Rimba, and I. Weber, "New kids on the block: An analysis of modern blockchains," 2016. [Online]. Available: http://arxiv.org/abs/1606.06530.

[77] Y. Velner, J. Teutsch, and L. Luu, "Smart contracts make bitcoin mining pools vulnerable," in *Proc. Financ. Cryptography Data Security*, Apr. 2017, pp. 298–316. [Online]. Available: https://doi.org/10.1007/978-3-319-70278-0_19

[78] P. Silva, *DNSSEC: The Antidote to DNS Cache Poisoning and Other DNS Attacks*, F5 Netw. Inc., Seattle, WA, USA, 2009.

[79] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *Proc. IEEE Eur. Symp. Security Privacy (EuroS&P)*, Mar. 2016, pp. 305–320. [Online]. Available: https://doi.org/10.1109/EuroSP.2016.32

[80] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Proc. Financ. Cryptography Data Security*, 2016, pp. 515–532.

[81] E. Heilman, "One weird trick to stop selfish miners: Fresh bitcoins, a solution for the honest miner," in *Proc. Financial Cryptography Data Security*, 2014, pp. 161–162.

[82] N. T. Courtois and L. Bahack, "On subversive miner strategies and block withholding attack in bitcoin digital currency," 2014. [Online]. Available: arXiv:1402.1718.

[83] S. Solat and M. Potop-Butucaru, "ZeroBlock: Preventing selfish mining in bitcoin," 2016. [Online]. Available: arXiv:1605.02435.

[84] G. Kappos, H. Yousaf, M. Maller, and S. Meiklejohn, "An empirical analysis of anonymity in ZCASH," in *Proc. USENIX Security Symp.*, 2018, pp. 463–477. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/kappos

[85] A. Hülsing, D. Butin, S. Gazdag, and A. Mohaisen. (2015). *XMSS: Extended Hash-Based Signatures*. [Online]. Available: https://www.ietf.org/id/draft-irtf-cfrg-xmss-hash-based-signatures-10.txt

[86] M. Kiran and M. Stanett, *Bitcoin Risk Analysis*, NEMODE, Columbus, GA, USA, 2015. [Online]. Available: http://hdl.handle.net/10454/10717

[87] J. Becker, D. Breuker, T. Heide, J. Holler, H. P. Rauer, and R. Böhme, "Can we afford integrity by proof-of-work? Scenarios inspired by the bitcoin currency," in *Proc. Econ. Inf. Security Privacy*, 2013, pp. 135–156. [Online]. Available: https://doi.org/10.1007/978-3-642-39498-0_7

[88] J. Moubarak, E. Filiol, and M. Chamoun, "On blockchain security and relevant attacks," in *Proc. IEEE Middle East North Africa Commun. Conf. (MENACOMM)*, 2018, pp. 1–6. [Online]. Available: https://doi.org/10.1109/MENACOMM.2018.8371010

[89] M. Carlsten, H. A. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *Proc. ACM Conf. Comput. Commun. Security (SIGSAC)*, Vienna, Austria, Oct. 2016, pp. 154–167. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978408

[90] S. Bano *et al.*, "SoK: Consensus in the age of blockchains," in *Proc. ACM Conf. Adv. Financial Technol. (AFT)*, New York, NY, USA, 2019, pp. 183–198. [Online]. Available: https://doi.org/10.1145/3318041.3355458

[91] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proc. 1st ACM Conf. Comput. Commun. Security*, Nov. 1993, pp. 62–73. [Online]. Available: http://doi.acm.org/10.1145/168588.168596

[92] A. Juels and J. G. Brainard, "Client puzzles: A cryptographic countermeasure against connection depletion attacks," in *Proc. Netw. Distrib. Syst. Security Symp. (NDSS)*, San Diego, CA, USA, 1999, pp. 151–165. [Online]. Available: http://www.isoc.org/isoc/conferences/ndss/99/proceedings/papers/juels.pdf

[93] A. Castor. (May 2017). *A Short Guide to Blockchain Consensus Protocols*. [Online]. Available: https://goo.gl/kdR2r4

[94] M. Saad and A. Mohaisen, "Towards characterizing blockchain-based cryptocurrencies for highly-accurate predictions," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM Workshops)*, Honolulu, HI, USA, Apr. 2018, pp. 704–709. [Online]. Available: https://doi.org/10.1109/INFCOMW.2018.8406859

[95] K. Zheng, Y. Liu, C. Dai, Y. Duan, and X. Huang, "Model checking PBFT consensus mechanism in healthcare blockchain network," in *Proc. IEEE Int. Conf. Inf. Technol. Med. Edu. (ITME)*, Hangzhou, China, 2018, pp. 877–881. [Online]. Available: https://ieeexplore.ieee.org/document/8589428

[96] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *Proc. Int. Convent. Inf. Commun. Technol. Electron. Microelectron. (MIPRO)* Opatija, Croatia, May 2018, pp. 1545–1550. [Online]. Available: https://doi.org/10.23919/MIPRO.2018.8400278

[97] D. Fullmer and A. S. Morse, "Analysis of difficulty control in bitcoin and proof-of-work blockchains," in *Proc. IEEE Conf. Decis. Control (CDC)*, Miami Beach, FL, USA, 2018, pp. 5988–5992. [Online]. Available: https://ieeexplore.ieee.org/document/8619082

[98] M. Bartoletti, S. Lande, and A. S. Podda, "A proof-of-stake protocol for consensus on bitcoin subchains," in *Proc. Financ. Cryptography Data Security (FC Workshops)*, Sliema, Malta, Apr. 2017, pp. 568–584. [Online]. Available: https://doi.org/10.1007/978-3-319-70278-0_36

[99] W. Y. M. M. Thin, N. Dong, G. Bai, and J. S. Dong, "Formal analysis of a proof-of-stake blockchain," in *Proc. 23rd Int. Conf. Eng. Complex Comput. Syst. (ICECCS)*, Dec. 2018, pp. 197–200. [Online]. Available: https://doi.org/10.1109/ICECCS2018.2018.00031

[100] G. Gui, A. Hortaçsu, and J. Tudon, "A memo on the proof-of-stake mechanism," 2018. [Online]. Available: http://arxiv.org/abs/1807.09626.

[101] T. Duong, A. Chepurnoy, L. Fan, and H. Zhou, "TwinsCoin: A cryptocurrency via proof-of-work and proof-of-stake," in *Proc. 2nd ACM Workshop Blockchains Cryptocurrencies Contracts (BCC@AsiaCCS)*, 2018, pp. 1–13. [Online]. Available: https://doi.org/10.1145/3205230.3205233

[102] S. D. Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain," in *Proc. 2nd Italian Conf. Cyber Security*, vol. 2058. Milan, Italy, Feb. 2018, pp. 1–11. [Online]. Available: http://ceur-ws.org/Vol-2058/paper-06.pdf

[103] H. Sukhwani, J. M. Martínez, X. Chang, K. S. Trivedi, and A. Rindos, "Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric)," in *Proc. 36th IEEE Symp. Rel. Distrib. Syst. (SRDS)*, Hong Kong, Sep. 2017, pp. 253–255. [Online]. Available: https://doi.org/10.1109/SRDS.2017.36

[104] S. Kim, Y. Kwon, and S. Cho, "A survey of scalability solutions on blockchain," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Jeju Island, South Korea, Oct. 2018, pp. 1204–1207. [Online]. Available: https://doi.org/10.1109/ICTC.2018.8539529

[105] A. Chauhan, O. P. Malviya, M. Verma, and T. S. Mor, "Blockchain and scalability," in *Proc. Int. Conf. Softw. Qual. Rel. Security Companion (QRS Companion)*, Lisbon, Portugal, Jul. 2018, pp. 122–128. [Online]. Available: https://doi.org/10.1109/QRS-C.2018.00034

[106] J. A. Garay and A. Kiayias, "SOK: A consensus taxonomy in the blockchain era," in *Proc. IACR Cryptol. ePrint Archive*, vol. 2018, 2018, p. 754. [Online]. Available: https://eprint.iacr.org/2018/754

[107] C. Cachin and M. Vukolic, "Blockchain consensus protocols in the wild (keynote talk)," in *Proc. Int. Symp. Distrib. Comput. (DISC)*, Vienna, Austria, Oct. 2017, pp. 1–16. [Online]. Available: https://doi.org/10.4230/LIPIcs.DISC.2017.1

[108] O. Konashevych and M. Poblet, "Is blockchain hashing an effective method for electronic governance?" in *Proc. Annu. Conf. Legal Knowl. Inf. Syst. Annu. Conf.*, vol. 313. Groningen, The Netherlands, Dec. 2018, pp. 195–199. [Online]. Available: https://doi.org/10.3233/978-1-61499-935-5-195

[109] F. Chen, Z. Liu, Y. Long, Z. Liu, and N. Ding, "Secure scheme against compromised hash in proof-of-work blockchain," in *Proc. Int. Conf. Netw. Syst. Security*, vol. 11058. Hong Kong, Aug. 2018, pp. 1–15. [Online]. Available: https://doi.org/10.1007/978-3-030-02744-5_1

[110] S. Delgado-Segura, C. Pérez-Solà, G. Navarro-Arribas, and J. Herrera-Joancomartí, "Analysis of the bitcoin UTXO set," in *Proc. Int. Workshop Financ. Cryptography Data Security*, 2018, pp. 78–91. [Online]. Available: https://doi.org/10.1007/978-3-662-58820-8_6

[111] Y. Kwon, D. Kim, Y. Son, E. Vasserman, and Y. Kim, "Be selfish and avoid dilemmas: Fork after withholding (FAW) attacks on bitcoin," in *Proc. ACM CCS*, Dallas, TX, USA, Oct./Nov. 2017, pp. 195–209. [Online]. Available: http://doi.acm.org/10.1145/3133956.3134019

[112] M. A. Javarone and C. S. Wright, "From bitcoin to bitcoin cash: A network analysis," in *Proc. 1st Workshop Cryptocurrencies Blockchains Distrib. Syst. (CRYBLOCK@MobiSys)*, Munich, Germany, Jun. 2018, pp. 77–81. [Online]. Available: https://goo.gl/AYJ68C

[113] T. Hanke, "AsicBoost—A speedup for bitcoin mining," 2016. [Online]. Available: arXiv:1604.00575.

[114] L. A. de la Porte. (2012). *The Bitcoin Transaction System*. [Online]. Available: https://pdfs.semanticscholar.org/fd87/fd511c7416eea4bbaa02b9c2f819738b50f8.pdf

[115] *Bitcoin Block Explorer—Blockchain*. Accessed: Sep. 13, 2019. [Online]. Available: http://bit.ly/1srPhPs

[116] B. Community. *Difficulty in Bitcoin*. Accessed: Sep. 13, 2019. [Online]. Available: https://en.bitcoin.it/wiki/Difficulty

[117] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 695–708, Jan. 2019. [Online]. Available: https://doi.org/10.1109/TWC.2018.2885266

[118] Greene. (Feb. 2018). *A Brief History of Bitcoin Mining Hardware*. [Online]. Available: https://thenextweb.com/hardfork/2018/02/02/a-brief-history-of-bitcoin-mining-hardware/

[119] A. de Vries, "Bitcoin's growing energy problem," *Joule*, vol. 2, no. 5, pp. 801–805, 2018.

[120] A. Kang, "Bitcoin's growing pains: Intermediation and the need for an effective loss allocation mechanism," *Michigan Bus. Entrepreneurial Law Rev.*, vol. 6, no. 2, p. 263, 2016.

[121] Digiconomist. (2018). *Bitcoin Energy Consumption Index*. [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption

[122] S. King and S. Nadal. *PPCoin: Peer-to-Peer Crypto-Currency With Proof-of-Stake*. Accessed: Aug. 19, 2012. [Online]. Available: https://www.chainwhy.com/upload/default/20180619/126a057fef926dc286accb372da46955.pdf

[123] P. Gazi, A. Kiayias, and A. Russell, "Stake-bleeding attacks on proof-of-stake blockchains," in *Proc. IACR Cryptol. ePrint Archive*, vol. 5, 2018, p. 248. [Online]. Available: http://eprint.iacr.org/2018/248

[124] A. Kiayias, I. Konstantinou, A. Russell, B. David, and R. Oliynykov, "A provably secure proof-of-stake blockchain protocol," in *Proc. IACR Cryptol. ePrint Archive*, vol. 2016, 2016, p. 889. [Online]. Available: http://eprint.iacr.org/2016/889

[125] P.-Y. Chang, M.-S. Hwang, and C.-C. Yang, "A blockchain-based traceable certification system," in *Proc. Int. Conf. Security Intell. Comput. Big data*, 2017, p. 363.

[126] A. Ahmad, M. Saad, L. Njilla, C. A. Kamhoua, M. Bassiouni, and A. Mohaisen, "BlockTrail: A scalable multichain solution for blockchain-based audit trails," in *Proc. Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ICC.2019.8761448

[127] R. Pass and E. Shi, "Thunderella: Blockchains with optimistic instant confirmation," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, vol. 10821, Apr. 2018, pp. 3–33. [Online]. Available: https://doi.org/10.1007/978-3-319-78375-8_1

[128] *Team Rocket*. (2018). *Snowflake to Avalanche: A Novel Metastable Consensus Protocol Family for Cryptocurrencies*. [Online]. Available: https://pdfs.semanticscholar.org/85ec/19594046bbcfe12137c7c2e3744677129820.pdf

[129] C. Berger and H. P. Reiser, "Scaling Byzantine consensus: A broad analysis," in *Proc. ACM Workshop Scalable Resilient Infrastructures Distrib. Ledgers*, Rennes, France, Dec. 2018, pp. 13–18. [Online]. Available: https://doi.org/10.1145/3284764.3284767

[130] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Proc. Financ. Cryptography Data Security*, 2014, pp. 436–454.

[131] C. Grunspan and R. Pérez-Marco, "On profitability of selfish mining," 2018. [Online]. Available: http://arxiv.org/abs/1805.08281.

[132] *Thetangle.org—Iota Tangle Explorer and Statistics*. Accessed: Sep. 13, 2019. [Online]. Available: https://thetangle.org/

[133] L. Bahack, "Theoretical bitcoin attacks with less than half of the computational power (draft)," 2013. [Online]. Available: arXiv:1312.7013.

[134] Nicehash. *Largest Crypto-Mining Marketplace*. Accessed: Sep. 13, 2019. [Online]. Available: https://www.nicehash.com/

[135] J. Bonneau, "Why buy when you can rent? Bribery attacks on bitcoin-style consensus," in *Proc. Financ. Cryptography Data Security (FC Workshops)*, 2016, pp. 19–26. [Online]. Available: https://doi.org/10.1007/978-3-662-53357-4_2

[136] B. Community. *Pow 51% Attack Cost*. Accessed: Sep. 13, 2019. [Online]. Available: https://www.crypto51.app/

[137] J. Roberts. *Bitcoin Spinoff Hacked in Rare '51% Attack'*. Accessed: Sep. 13, 2019. [Online]. Available: http://fortune.com/2018/05/29/bitcoin-gold-hack/

[138] R. Pass, L. Seeman, and A. Shelat, "Analysis of the blockchain protocol in asynchronous networks," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, Paris, France, 2017, pp. 643–673. [Online]. Available: https://doi.org/10.1007/978-3-319-56614-6_22

[139] J. Zhao, "An analysis of blockchain consistency in asynchronous networks: Deriving a neat bound," 2019. [Online]. Available: http://arxiv.org/abs/1909.06587.

[140] L. Ren, "Analysis of Nakamoto consensus," Cryptol. ePrint Archive, Rep. 2019/943, 2019. [Online]. Available: https://eprint.iacr.org/2019/943

[141] M. Castro and B. Liskov, "Practical Byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, 2002. [Online]. Available: https://doi.org/10.1145/571637.571640

[142] A. R. Kang, J. Spaulding, and A. Mohaisen, "Domain name system security and privacy: Old problems and new challenges," 2016. [Online]. Available: http://arxiv.org/abs/1606.07080.

[143] L. Gao, "On inferring autonomous system relationships in the Internet," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 733–745, Dec. 2001. [Online]. Available: https://doi.org/10.1109/90.974527

[144] M. Kumar and S. Kumar, "Improving routing in large networks inside autonomous system," *Int. J. Syst. Assurance Eng. Manag.*, vol. 5, no. 3, pp. 383–390, 2014. [Online]. Available: https://doi.org/10.1007/s13198-013-0179-0

[145] M. Saad, A. Anwar, A. Ahmad, H. Alasmary, M. Yuksel, and A. Mohaisen, "RouteChain: Towards blockchain-based secure and efficient BGP routing," in *Proc. Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2019, pp. 210–218. [Online]. Available: https://doi.org/10.1109/BLOC.2019.8751229

[146] M. Saad, V. Cook, L. Nguyen, M. T. Thai, and A. Mohaisen, "Partitioning attacks on bitcoin: Colliding space, time, and logic," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Dellas, TX, USA, Jul. 2019, pp. 1175–1187.

[147] Bitnodes. *Global Bitcoin Nodes Distribution*. Accessed: Sep. 13, 2019. [Online]. Available: https://bitnodes.earn.com/

[148] A. Greenberg. (Jun. 2017). *Hacker Redirects Traffic From 19 Internet Providers to Steal Bitcoins*. [Online]. Available: https://www.wired.com/2014/08/isp-bitcoin-theft/

[149] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *Proc. USENIX Security Symp.*, Aug. 2015, pp. 129–144. [Online]. Available: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/heilman

[150] A. Wang, A. Mohaisen, and S. Chen, "An adversary-centric behavior modeling of DDoS attacks," in *Proc. 37th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Atlanta, GA, USA, Jun. 2017, pp. 1126–1136. [Online]. Available: https://doi.org/10.1109/ICDCS.2017.213

[151] M. Vasek, M. Thornton, and T. Moore, "Empirical analysis of denial-of-service attacks in the bitcoin ecosystem," in *Proc. Financ. Cryptography Data Security*, 2014, pp. 57–71.

[152] M. Saad, L. Njilla, C. A. Kamhoua, J. Kim, D. Nyang, and A. Mohaisen, "Mempool optimization for defending against DDoS attacks in pow-based blockchain systems," in *Proc. Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2019, pp. 285–292. [Online]. Available: https://doi.org/10.1109/BLOC.2019.8751476

[153] C. Mempool. (Jun. 2018). *Report: Bitcoin (BTC) Mempool Shows Backlogged Transactions, Increased Fees If So?* [Online]. Available: https://goo.gl/LsU6Hq

[154] D. K. Tosh, S. Shetty, X. Liang, C. A. Kamhoua, K. A. Kwiat, and L. Njilla, "Security implications of blockchain cloud with analysis of block withholding attack," in *Proc. 17th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput.*. 2017, pp. 458–467.

[155] Mark. (Oct. 2017). *The Finney attack*. [Online]. Available: https://bitcoincoreacademy.com/the-finney-attack/

[156] S. Exchange. *What Is a Finney Attack?* Accessed: Sep. 13, 2019. [Online]. Available: https://bitcoin.stackexchange.com/questions/4942/what-is-a-finney-attack

[157] L. Luu, R. Saha, I. Parameshwaran, P. Saxena, and A. Hobor, "On power splitting games in distributed computation: The case of bitcoin pooled mining," in *Proc. IEEE 28th Comput. Security Found. Symp. (CSF)*, Verona, Italy, Jul. 2015, pp. 397–411. [Online]. Available: https://doi.org/10.1109/CSF.2015.34

[158] S. Exchange. *What Is a Block Withholding Attack?* Accessed: Sep. 13, 2019. [Online]. Available: https://goo.gl/ccAsAi

[159] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, "Tampering with the delivery of blocks and transactions in bitcoin," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, Denver, CO, USA, Oct. 2015, pp. 692–705. [Online]. Available: http://doi.acm.org/10.1145/2810103.2813655

[160] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, Feb. 1999, pp. 173–186. [Online]. Available: https://dl.acm.org/citation.cfm?id=296824

[161] H. Xu, Y. Long, Z. Liu, Z. Liu, and D. Gu, "Dynamic practical Byzantine fault tolerance," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, Beijing, China, May 2018, pp. 1–8. [Online]. Available: https://doi.org/10.1109/CNS.2018.8433150

[162] A. Miller, A. E. Kosba, J. Katz, and E. Shi, "Nonoutsourceable scratch-off puzzles to discourage bitcoin mining coalitions," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, Denver, CO, USA, Oct. 2015, pp. 680–691. [Online]. Available: http://doi.acm.org/10.1145/2810103.2813621

[163] M. Saad, L. Njilla, C. Kamhoua, and A. Mohaisen, "Countering selfish mining in blockchains," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, Honolulu, HI, USA, 2019, pp. 360–364. [Online]. Available: https://ieeexplore.ieee.org/document/8685577

[164] B. Johnson, A. Laszka, J. Grossklags, M. Vasek, and T. Moore, "Game-theoretic analysis of DDoS attacks against bitcoin mining pools," in *Proc. Financ. Cryptography Data Security*, 2014, p. 72.

[165] J. Göbel and A. E. Krzesinski, "Increased block size and bitcoin blockchain dynamics," in *Proc. 27th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2017, pp. 1–6. [Online]. Available: https://goo.gl/rz4zoB

[166] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Comput. Surveys*, vol. 39, no. 1, p. 3, 2007.

[167] J. Etheridge and R. Anton, "System and method for detecting and countering a network attack," U.S. Patent App. 10243 631, Sep. 13, 2002.

[168] S. Bag, S. Ruj, and K. Sakurai, "Bitcoin block withholding attack: Analysis and mitigation," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1967–1978, Aug. 2017. [Online]. Available: https://doi.org/10.1109/TIFS.2016.2623588

[169] S. Bag and K. Sakurai, "Yet another note on block withholding attack on bitcoin mining pools," in *Proc. 19th Int. Conf. Inf. Security (ISC)*, Honolulu, HI, USA, Sep. 2016, pp. 167–180. [Online]. Available: https://doi.org/10.1007/978-3-319-45871-7_11

[170] O. Schrijvers, J. Bonneau, D. Boneh, and T. Roughgarden, "Incentive compatibility of bitcoin mining pool reward functions," in *Proc. 20th Int. Conf. Financ. Cryptography Data Security (FC)*, Feb. 2016, pp. 477–498. [Online]. Available: https://doi.org/10.1007/978-3-662-54970-4_28

[171] M. Rosenfeld, "Analysis of bitcoin pooled mining reward systems," 2011. [Online]. Available: http://arxiv.org/abs/1112.4980.

[172] G. S. Veronese, M. Correia, A. N. Bessani, L. C. Lung, and P. Veríssimo, "Efficient Byzantine fault-tolerance," *IEEE Trans. Comput.*, vol. 62, no. 1, pp. 16–30, Jan. 2013. [Online]. Available: https://doi.org/10.1109/TC.2011.221

[173] T. Distler, C. Cachin, and R. Kapitza, "Resource-efficient Byzantine fault tolerance," *IEEE Trans. Comput.*, vol. 65, no. 9, pp. 2807–2819, Sep. 2016. [Online]. Available: https://doi.org/10.1109/TC.2015.2495213

[174] J. Liu, W. Li, G. O. Karame, and N. Asokan, "Scalable Byzantine consensus via hardware-assisted secret sharing," *IEEE Trans. Comput.*, vol. 68, no. 1, pp. 139–151, Jan. 2019. [Online]. Available: https://doi.org/10.1109/TC.2018.2860009

[175] M. Apostolaki, G. Marti, J. Müller, and L. Vanbever, "SABRE: Protecting bitcoin against routing attacks," in *Proc. Annu. Netw. Distrib. Syst. Security Symp. (NDSS)*, San Diego, CA, USA, Feb. 2019, pp. 1–15. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/sabre-protecting-bitcoin-against-routing-attacks/

[176] M. B. H. Weiss, K. Werbach, D. C. Sicker, and C. E. C. Bastidas, "On the application of blockchains to spectrum management," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 2, pp. 193–205, Jun. 2019. [Online]. Available: https://doi.org/10.1109/TCCN.2019.2914052

[177] L. Ghiro, L. Maccari, and R. L. Cigno, "Proof of networking: Can blockchains boost the next generation of distributed networks?" in *Proc. 14th Annu. Conf. Wireless On-Demand Netw. Syst. Services (WONS)*, Feb. 2018, pp. 29–32. [Online]. Available: https://doi.org/10.23919/WONS.2018.8311658

[178] P. Li, G. Wang, X. Chen, and W. Xu, "GOSIG: Scalable Byzantine consensus on adversarial wide area network for blockchains," 2018. [Online]. Available: http://arxiv.org/abs/1802.01315.

[179] C. Janze, "Are cryptocurrencies criminals best friends? Examining the co-evolution of bitcoin and darknet markets," in *Proc. Amer. Conf. Inf. Syst. (AMCIS)*, Boston, MA, USA, Aug. 2017. [Online]. Available: http://aisel.aisnet.org/amcis2017/InformationSystems/Presentations/2

[180] R. Stokes, "Virtual money laundering: The case of bitcoin and the linden dollar," *Inf. Commun. Technol. Law*, vol. 21, no. 3, pp. 221–236, 2012. [Online]. Available: https://doi.org/10.1080/13600834.2012.744225

[181] S. Williams. (2017). *Bitcoin Banned Countries*. [Online]. Available: https://tinyurl.com/y8r5gdhl

[182] G. O. Karame, E. Androulaki, M. Roeschlin, A. Gervais, and S. Capkun, "Misbehavior in bitcoin: A study of double-spending and accountability," *ACM Trans. Inf. Syst. Secur.*, vol. 18, no. 1, pp. 1–32, 2015. [Online]. Available: http://doi.acm.org/10.1145/2732196

[183] M. Nadeau. (May 2018). *What Is Cryptojacking? How to Prevent, Detect, and Recover From It*. [Online]. Available: https://goo.gl/DdGq1i

[184] R. Li and C. Kyle. (Jan. 2018). *What Is Cryptojacking?* [Online]. Available: https://hackerbits.com/programming/what-is-cryptojacking/

[185] M. Saad, A. Khormali, and A. Mohaisen, "Dine and dash: Static, dynamic, and economic analysis of in-browser cryptojacking," in *Proc. APWG Symp. Electron. Crime Res. (eCrime)*, Pittsburgh, PA, USA, 2019, pp. 1–12. [Online]. Available: https://ieeexplore.ieee.org/document/9037576

[186] (2018). *Coinhive*. [Online]. Available: https://coinhive.com/

[187] CryptoLoot. (2018). *Earn More From Your Visitors*. [Online]. Available: https://crypto-loot.com/

[188] M. Community. (2018). *Monero: Home*. [Online]. Available: https://getmonero.org/

[189] J. Condliffe. (2018). *A Cryptojacking Attack Hit Thousands of Websites, Including Government Ones*. [Online]. Available: https://goo.gl/FPgTo9

[190] Google. (2018). *Google Analytics and Trends*. [Online]. Available: https://goo.gl/9sSpGL

[191] D. Singh. (2018). *Cryptojacking Attacks Rose by 8,500% Globally in 2017: Report*. [Online]. Available: https://goo.gl/qpGcZy

[192] NCSC. (Apr. 2018). *The Cyber Threat to U.K. Business 2017–2018 Report*. [Online]. Available: https://www.ncsc.gov.uk/cyberthreat

[193] T. Volety, S. Saini, T. McGhin, C. Z. Liu, and K. R. Choo, "Cracking bitcoin wallets: I want what you have in the wallets," *Future Gener. Comput. Syst.*, vol. 91, pp. 136–143, Feb. 2019. [Online]. Available: https://doi.org/10.1016/j.future.2018.08.029

[194] M. Vasek, J. Bonneau, R. Castellucci, C. Keith, and T. Moore, "The bitcoin brain drain: Examining the use and abuse of bitcoin brain wallets," in *Proc. Financ. Cryptography Data Security*, Feb. 2016, pp. 609–618. [Online]. Available: https://doi.org/10.1007/978-3-662-54970-4_36

[195] B. Peterson. (Dec. 2017). *Thieves Stole Potentially Millions of Dollars in Bitcoin in a Hacking Attack on a Cryptocurrency Company*. [Online]. Available: https://goo.gl/znceAF

[196] W. Duggan. *The 12 Biggest Cryptocurrency Hacks in History*. Accessed: Sep. 13, 2019. [Online]. Available: https://www.benzinga.com/fintech/17/11/10824764/12-biggest-cryptocurrency-hacks-in-history

[197] M. Brengel and C. Rossow, "Identifying key leakage of bitcoin users," in *Proc. Int. Symp. Res. Attacks Intrusions Defenses (RAID)*, vol. 11050, Sep. 2018, pp. 623–643. [Online]. Available: https://doi.org/10.1007/978-3-030-00470-5_29

[198] J. Breitner and N. Heninger, "Biased nonce sense: Lattice attacks against weak ECDSA signatures in cryptocurrencies," Cryptol. ePrint Archive, Rep. 2019/023, 2019. [Online]. Available: https://eprint.iacr.org/2019/023

[199] X. Wang *et al.*, "Attack and defence of Ethereum remote APIS," in *Proc. IEEE Globecom Workshops*, Dec. 2018, pp. 1–6. [Online]. Available: https://doi.org/10.1109/GLOCOMW.2018.8644498

[200] M. Wohrer and U. Zdun, "Smart contracts: Security patterns in the Ethereum ecosystem and solidity," in *Proc. Int. Workshop Blockchain Oriented Softw. Eng. (IWBOSE@SANER)*, Campobasso, Italy, Mar. 2018, pp. 2–8. [Online]. Available: https://doi.org/10.1109/IWBOSE.2018.8327565

[201] ConsenSys. *Consensys/Smart-Contract-Best-Practices*. Accessed: Sep. 13, 2019. [Online]. Available: https://github.com/ConsenSys/smart-contract-best-practices/blob/master/docs/known_attacks.md

[202] T. Chen *et al.*, "An adaptive gas cost mechanism for Ethereum to defend against under-priced DoS attacks," in *Proc. Int. Conf. Inf. Security Pract. Exp. (ISPEC)*, Melbourne, VIC, Australia, Dec. 2017, pp. 3–24. [Online]. Available: https://doi.org/10.1007/978-3-319-72359-4_1

[203] S. Banerjee, B. Karp, and M. Walfish, Eds., "HotNets-XVI," in *Proc. ACM Workshop Hot Topics Netw.*, Palo Alto, CA, USA, Dec. 2017, pp. 1–7. [Online]. Available: https://doi.org/10.1145/3152434

[204] R. Norvill, B. B. F. Pontiveros, R. State, I. Awan, and A. J. Cullen, "Automated labeling of unknown contracts in Ethereum," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, Vancouver, BC, Canada, Jul. 2017, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ICCCN.2017.8038513

[205] BitcoinCommunityWallets. *bitcoin/bips*. Accessed: Sep. 13, 2019. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0038.mediawiki

[206] BitcoinCommunityStoring. *Storing Bitcoins*. Accessed: Sep. 13, 2019. [Online]. Available: https://en.bitcoin.it/wiki/Storing_bitcoins

[207] AntiMiner. (2018). *Anti Miner—No 1 Coin Minerblock*. [Online]. Available: https://goo.gl/BiwzUU

[208] CoinMiner. (2018). *Coin Miner Block*. [Online]. Available: https://goo.gl/MWPNv4

[209] AdGuard. (2018). *Adguard Adblocker*. [Online]. Available: https://goo.gl/AXg186

[210] T. M. Fernández-Caramés and P. Fraga-Lamas, "A review on the use of blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018. [Online]. Available: https://doi.org/10.1109/ACCESS.2018.2842685

[211] G. Perboli, S. Musso, and M. Rosano, "Blockchain in logistics and supply chain: A lean approach for designing real-world use cases," *IEEE Access*, vol. 6, pp. 62018–62028, 2018. [Online]. Available: https://doi.org/10.1109/ACCESS.2018.2875782

[212] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Zug, Switzerland, Ethereum Project, Yellow Paper, 2014.

[213] K. Lee, J. I. James, T. G. Ejeta, and H. Kim, "Electronic voting service using blockchain," *J. Digit. Forensics Security Law*, vol. 11, no. 2, p. 123, 2016.

[214] P. Noizat, "Blockchain electronic vote," in *Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments, and Big Data*. Amsterdam, The Netherlands: Elsevier, 2015, p. 453.

[215] T. I. Ron and S. Attias, "The effect of blockchain technology in the gaming regulatory environment," *Gaming Law Rev.*, vol. 21, no. 6, pp. 459–460, 2017.

[216] G. Karame, "On the security and scalability of bitcoin's blockchain," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, Vienna, Austria, Oct. 2016, pp. 1861–1862. [Online]. Available: https://doi.org/10.1145/2976749.2976756

[217] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2084–2123, 3rd Quart., 2016. [Online]. Available: https://doi.org/10.1109/COMST.2016.2535718

[218] S. Wang, C. Wang, and Q. Hu, "Corking by forking: Vulnerability analysis of blockchain," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 829–837.

[219] A. Davenport, S. Shetty, and X. Liang, "Attack surface analysis of permissioned blockchain platforms for smart cities," in *Proc. IEEE Int. Smart Cities Conf. (ISC2)*, Kansas City, MO, USA, Sep. 2018, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ISC2.2018.8656983

[220] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, Vienna, Austria, Oct. 2016, pp. 3–16. [Online]. Available: https://doi.org/10.1145/2976749.2978341

[221] B. Choi, J. Sohn, D. Han, and J. Moon, "Scalable network-coded PBFT consensus algorithm," in *Proc. Int. Symp. Inf. Theory (ISIT)*, Paris, France, 2019, pp. 857–861. [Online]. Available: https://doi.org/10.1109/ISIT.2019.8849573

[222] S. Wuthier and C.-W. Chen, "Uncle-block attack: Blockchain mining threat beyond block withholding for rational and uncooperative miners," in *Proc. Int. Conf. Appl. Cryptograph. Netw. Security*, Bogota, Colombia, Jun. 2019, pp. 241–258.

[223] S. Chang and Y. Park, "Silent timestamping for blockchain mining pool security," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, Feb. 2019, pp. 1–5.

[224] S. Zhang and J.-H. Lee, "Double-spending with a sybil attack in the bitcoin decentralized network," *IEEE Trans. Ind. Informat.*, vol. 15, no. 10, pp. 5715–5722, Oct. 2019.

[225] H. Lee, M. Shin, K. S. Kim, Y. Kang, and J. Kim, "Recipient-oriented transaction for preventing double spending attacks in private blockchain," in *Proc. 15th Annu. IEEE Int. Conf. Sens. Commun. Netw. (SECON)*, Jun. 2018, pp. 1–2.

[226] Z. A. El Houda, L. Khoukhi, and A. Hafid, "ChainSecure—A scalable and proactive solution for protecting blockchain applications using SDN," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[227] S. Werman and A. Zohar, "Avoiding deadlocks in payment channel networks," in *Proc. Int. Workshop Data Privacy Manag. Cryptocurrencies Blockchain Technol. DPM CBT*, Barcelona, Spain, Sep. 2018, pp. 175–187. [Online]. Available: https://doi.org/10.1007/978-3-030-00305-0_13

[228] R. Yu, G. Xue, V. T. Kilari, D. Yang, and J. Tang, "CoinExpress: A fast payment routing mechanism in blockchain-based payment channel networks," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, 2018, pp. 1–9. [Online]. Available: https://doi.org/10.1109/ICCCN.2018.8487351