

Analysing Attacks on Blockchain Systems in a Layer-based Approach

JOYDIP DAS* and SYED ASHRAF AL TASIN*, Shahjalal University of Science & Technology, Bangladesh

MD. FORHAD RABBI, Shahjalal University of Science & Technology, Bangladesh

MD SADEK FERDOUS, BRAC University, Bangladesh

Blockchain is a growing decentralized system built for transparency and immutability. There have been several major attacks on blockchain-based systems, leaving a gap in the trustability of this system. This article presents a comprehensive study of 23 attacks on blockchain systems and categorizes them using a layer-based approach. This approach provides an in-depth analysis of the feasibility and motivation of these attacks. In addition, a framework is proposed that enables a systematic analysis of the impact and interconnection of these attacks, thereby providing a means of identifying potential attack vectors and designing appropriate countermeasures to strengthen any blockchain system.

CCS Concepts: • **Security and privacy** → **Distributed systems security**; • **General and reference** → **Surveys and overviews**.

Additional Key Words and Phrases: Blockchain, Security of Blockchain Layers, Adversarial Factors, Attack & Mitigation Feasibility, Network & Consensus Security, Attack Analysis, Inter-connection of Attacks

1 Introduction

From the very first implementation of blockchain through Bitcoin by Satoshi Nakamoto [95], this technology has offered improved security and played a vital role in the development and enhancement of many different application domains such as banking sectors, streaming & copy-right services, wallet services, healthcare services, electronic voting, and IoT (Internet of Things). Blockchain's rise can be credited to its idea of efficient ledger management with decentralization and immutability. Unfortunately, even with increased security, blockchain has been shown to be vulnerable in different aspects. Records of security breaches in blockchain-based systems are noticeable with reports claiming a total of \$14.6B worth of funds been stolen, accumulated in 45 countries with 364 incidents, in between 2011 to 2022 [23]. In addition, the largest De-Fi (Decentralized Finance) hack to date happened in early 2022, involving more than \$650 million [23]. BonqDAO and AllianceBlock experienced a security breach on February 2, 2023, as a result of a flaw in BonqDAO's smart contract. This led to a financial loss of approximately \$120 million [22]. The biggest financial loss in a 2023 attack occurred in September 23, when an attack on the database of Mixin Network's cloud service provider resulted in a substantial loss of assets on its mainnet, amounting to approximately \$200 million [118].

Another evidence of vulnerability can be seen in this publicly revealed data confirming that money lost to blockchain hackers is around \$273 hundreds million with more than 800 such events [134]. According to data gathered by Comparitech, 6 out of the top 10 expensive crypto attacks occurred in 2021 alone [65]. Also in the early years of Bitcoin, the famous crypto-exchange network Mt. Gox lost \$474 Million due to security flaws such as transaction mutability [91]. Nomad, a crypto start-up, was hacked in early 2022, which was the eighth largest cryptocurrency hack with

*Both authors contributed equally to this research.

the damage of \$190 million token [31]. In 2016, ether worth \$9 billion was taken from the DAO (Decentralized Autonomous Organization) due to some flaws in code [99].

These attacks have key impacts on the global economy, with the prediction of the risk of losing \$30 billion per year by 2025 [92]. In order to ensure a significant adoption of blockchain, these attacks need to be studied and analyzed for vulnerabilities. In addition, these attacks can be calculated for their impacts, feasibility, and mitigation techniques. Our study aims to tackle the challenge of securing blockchain systems by conducting an in-depth analysis, breaking down the barriers to enhanced protection. Although previous studies have shed light on blockchain-based systems and attack vectors, they often lacked a thorough and systematic analysis of these attacks. Our study seeks to close these gaps by offering an in-depth attack analysis using a layer-based approach, attacker perspective and outcome analysis, and a detailed examination of the connections between attacks across several layers through a custom framework discussed in Section 4.2.

1.1 Contribution

This article provides a comprehensive analysis and systematic study of attacks related to different blockchain systems in the context of a four-layer blockchain structure as previously introduced by Ferdous *et al.* in [62]. The primary contributions of this paper are as follows.

- We conduct a thorough examination of several security attacks commonly targeted towards different blockchain systems.
- For every instance of an attack, we try to find the answers to the following questions.
 - Is there a motivation behind these attacks and which specific vulnerabilities are being exploited?
 - What are the steps in each of these attacks?
 - What are the potential challenges for attackers in initiating these attacks and what are the potential outcomes?
 - Can these attacks be considered realistic, and are there any known mitigating strategies?
 In order to answer these questions, we have formulated several criteria, which have been used to analyze each attack.
- We investigate the connections between scenarios in which one attack can provide a strategic advantage in executing additional attacks.
- Finally, we present a visual summary, in tabular format, of our analysis of each attack.

1.2 Structure

The remainder of the article is structured as follows. In Section 2, we provide the necessary background knowledge of blockchain systems. Next, in Section 3, we discuss related work relevant to this article. Section 4 gives a comprehensive overview of our framework by discussing the blockchain layers, the taxonomy of properties, and a list of analyzed attacks. Then, in Section 5, Section 6, Section 7 and Section 8, different attacks relevant to each layer are analyzed using the formulated properties. Section 9 presents a summary of the findings. Finally, we draw our conclusions in Section 10.

2 Background

In this section, we provide a brief overview of different aspects of blockchain technology to understand the attacks analyzed in this article.

A peer-to-peer computer network with a distributed ledger is the fundamental idea of blockchain. Cryptographer David Chaum's article 'Computer Systems Established, Maintained, and Trusted by Mutually Suspicious Groups' contains the first concept for a blockchain-like technology [114]. Later Haber et al. introduced the concept of a collection of blocks linked together using a cryptographic

mechanism [68]. However, the pioneer in making blockchain viable was Satoshi Nakamoto who introduced Bitcoin using a mix of well-established technologies such as a peer-to-peer (P2P) network, digital signature, secure timestamping, and cryptographic hash algorithms [95]. A blockchain is an example of distributed ledger that consists of blocks that are connected to each other with cryptographic mechanisms, thus forming the notion of chain of blocks or blockchain [44]. Each block contains a number of transactions, each transaction representing a transfer of data or value between any two entities. The distributed nature of a blockchain requires a consensus algorithm to ensure that all data stored in a blockchain are synchronized with all P2P nodes. The key innovation of Bitcoin by Satoshi Nakamoto was to achieve a network-wide consensus among the P2P nodes regarding the state of the blockchain in a decentralized way without using any trusted party. All these features enable blockchain to maintain a decentralized, immutable, transparent, efficient, and consensus-based ledger.

In addition to Bitcoin, Ethereum is another popular blockchain system that offers additional capabilities. Ethereum is a decentralized application platform developed on top of a blockchain [35]. Bitcoin is a digital currency, however, Ethereum is more concerned with application development. It accomplishes this by using the notion of smart contracts, which are self-executing code that can automate agreements and transactions. These smart contracts are implemented on a virtual machine, named Ethereum Virtual Machine (EVM), and stored on the blockchain. These contracts can be executed using a transaction that changes the state of the virtual machine. This change of states is also recorded on the Blockchain. Like bitcoin, a distributed consensus algorithm ensures a network-wide agreement over the EVM state and the blockchain data, thus facilitating the notion of immutable code and data. This opens up a wide range of possibilities, including safe marketplaces, complicated financial instruments, and even new types of digital assets. Ethereum features its own coin, Ether (ETH), which is used to pay for different transactions, smart contract execution and data stored on the blockchain.

There are mainly two different types of blockchain as discussed in the following.

- **Public Blockchain:** Public blockchains are decentralized networks that are accessible to anyone who wishes to take part in validating and recording transactions. They offer transparency and accountability as anyone can access the blockchain and verify each of its components. This transparency and decentralization make them ideal for cryptocurrencies and decentralized applications. Bitcoin [1] and Ethereum [7] are prime examples of such public blockchain systems.
- **Private Blockchain:** Private blockchains are restricted networks where only authorized participants can validate and record transactions. They are suitable for enterprise use cases like supply chain management and financial transactions, where privacy, control and efficient operations are essential. Examples include Hyperledger Fabric [63] and R3 Corda [5].

2.1 Consensus Mechanism

Any blockchain system's core component is the underlying consensus algorithm. A consensus algorithm is a fault-tolerant mechanism that is employed to reach agreement on specific decisions or states within a blockchain network. In a blockchain system, consensus is critical since it guarantees that each new block added to the ledger represents the single version of the truth agreed upon by all nodes. Any consensus mechanism consists of these three properties - consistency, availability, and fault tolerance. A consensus algorithm mechanism must guarantee the characteristics of the atomic broadcast (i.e., validity, agreement, integrity, and total order) [62]. Numerous criteria are used to establish the acceptable network condition with consensus. The following section discusses some common consensus algorithms that are employed in various blockchain systems [62].

- **Proof of Work (PoW):** PoW is based on a simple principle - ‘A solution that is difficult to find but is easy to verify’ [96]. PoW involves solving a resource-intensive computational cryptographic puzzle to add new blocks to the blockchain. The PoW mechanism has a difficulty parameter and a node repeatedly solves the cryptographic puzzle to reach that parameter value. If successful, broadcast it to other nodes. The widely implemented version of PoW is based on SHA-256 [18]. The computer nodes which engage in solving such puzzles are known as miner nodes and the process is known as mining.
- **Proof of Stake (PoS):** In PoS, miners are known as ‘Validators’. The network selects a validator through a bidding process. Each validator deposits a portion of their cryptocurrency associated with that network known as stake. The selection of a validator is typically based on this stake. The likelihood of a participant being selected as a validator increases with the amount of cryptocurrency he stakes. However, some PoS systems may also consider additional factors like how long the stakes have been held [79] and randomization [45] to prevent any particular entity from being continuously chosen as a validator. If a validator tries to cheat or use unfair ways and gets detected, he will lose all the stakes deposited previously. PoS is more energy efficient than PoW as it does not consume electricity.
- **Practical Byzantine Fault Tolerance (PBFT):** This algorithm was proposed by Castro and Liskov in 1999 [40]. In this system, a cluster of replicas process transactions and ultimately creates a new block. The primary replica orders the transaction and gathers approvals from other replicas. Upon receiving enough approvals, the primary replica creates a block and broadcasts it. The system functions properly as long as the proportion of malicious nodes is less than one-third of the total nodes and the primary replica is not compromised. This consensus mechanism is mainly used in private blockchains.

2.2 Mining Pools and Reward System

When miners collaborate to form a sizable collective network for effective mining, it is referred to as a mining pool. Each member of a mining pool contributes computing power to solve a block. If any member finds a block, the entire mining pool is rewarded with the related cryptocurrency. Usually, a pool operator maintains the pool. The pool operator is responsible for the reward distribution and other operational activities [58]. The mining pool reward system is based on ‘Shares’. A share is a partial block solution. For example, let us assume that a block solution is a number that contains 32 trailing zeros. If a solution with 28 trailing zeros is found, it may be considered as a share or partial proof of work. The share is the main indicator of an individual miner’s contribution to the mining pool to find a valid solution. When any participant finds a full proof of work (FPoW), that is, a number with 32 trailing zeros, it is submitted to the pool manager. The pool manager then publishes this FPoW to the blockchain network and the block generation reward is distributed among the participants in one of the many different methods, such as proportional, pay-per-share (PPS) and pay-per-last-N-shares (PPLNS) [105]. We briefly describe these methods next.

- **Proportional:** This reward scheme is based on rounds. A round is the time interval of finding 2 blocks. In each round, miners keep submitting shares to the pool. If the pool succeeds in finding a block, it gets rewarded and the reward is distributed to the miners by the number of shares they submitted during that round.
- **Pay-per-share (PPS):** Every miner is immediately rewarded with the expected value of the share’s contribution upon submitting a valid share. The pool operator receives all of the rewards for discovered blocks and pays out miners using the pool’s current balance.
- **Pay-per-last-N-shares (PPLNS):** This scheme is somewhat similar to the ‘Proportional’ scheme. In contrast to Proportional, the miner’s payment in this technique is determined

based on the last N shares rather than all shares from the previous round. As a result, all miners profit more if the round was short enough, and vice versa.

2.3 Stale Blocks and Forks

A successfully mined block that is ultimately discarded from the longest chain is known as a ‘Stale block’. It happens because two or more miners can simultaneously solve the PoW puzzle for a specific round and create multiple blocks with different valid solutions. When this happens, a ‘fork’ is created. A fork is a state in which there are conflicting opinions among network nodes regarding the status of the blockchain. In that case, only one block is added in the blockchain by fork resolution mechanism and other stale/orphan blocks are rejected. Transactions in these rejected blocks are sent back to the mempool (a cache used by every P2P node in a blockchain network) to await pickup in a subsequent block. When a fork occurs in Bitcoin network by the miners, the longest chain rule is used to resolve it [95]. The longest chain is the one that requires the most energy to construct. In Ethereum, fork resolution is based on the node with the heaviest sub-tree, which in short is called GHOST (Greedy Heaviest-Observed Sub-Tree) protocol [120]. However, recently Ethereum adopted a PoS-based consensus mechanism, which reduces the chance of fork [36, 37, 57].

Forks also might occur for other reasons such as to implement a new feature, correct a security vulnerability, or settle a dispute within the community over the direction the blockchain system should take. In such situations, there are two types of fork: the hard fork and the soft fork. A hard fork is effectively a persistent deviation from the most recent version of a blockchain. This results in the blockchain being split into two separate networks that operate independently of one another since some nodes can no longer reach a consensus. Bitcoin Cash is an example of such a hard fork [13]. On the other hand, a soft fork is a backward compatible modification or upgrade of a blockchain. It does not cause the network to split off or produce a new version of the blockchain. Rather, it enables the network to smoothly switch over to the new rules while preserving compatibility with the previous ones. SegWit is an example of a soft fork that took place in 2017 for Bitcoin [55].

Figure 1 shows a hard fork, longest chain rule, and a scenario of stale blocks in a blockchain system.

2.4 Double-Spending

Double-spending is a serious security risk in blockchain-based cryptocurrency systems. In contrast to a physical currency, which is tangible and cannot be replicated, digital currencies are inherently replicable. Double-spending takes advantage of this feature by attempting to spend the same digital currency several times, compromising the integrity of the system. We describe three attacks that are variants of the double spending attack in Section 6.4, Section 6.6, and Section 6.7.

3 Related Work

A layer-based attack analysis is a systematic and efficient technique for critically analyzing a blockchain system. This method, which divides the system into smaller, more manageable components, makes it easier to identify and mitigate vulnerabilities by enabling a focused examination on certain components without being overwhelmed by the complexities of the entire system. Additionally, it makes it possible to introduce security measures that are layer-specific, putting the strongest protections where they are most needed and improving the system’s overall resilience.

Other surveys have been conducted to gain insight into the attack surface of blockchain systems, which differ from the scope of our research. Guggenberger *et al.* performed a review of the current literature on blockchain system attacks, ultimately identifying 87 relevant attacks. [67]. The attacks

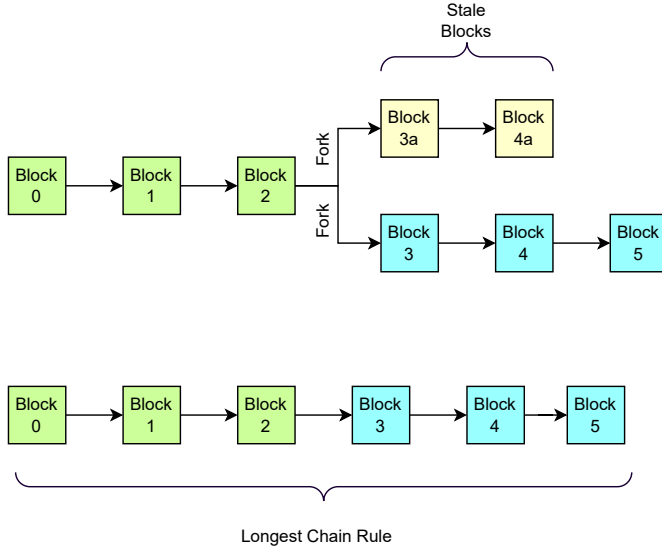


Fig. 1. Fork and Longest Chain Rule

were represented using the attack tree (AT) notation, as proposed by Mauw and Oostdijk (2006) [88]. However, a comprehensive analysis of each attack is absent, and the attacks are also not categorized in a layer-based approach.

Ferdous *et al.* while introducing four distinct layers within a blockchain system, discussed some relevant attack vectors of consensus mechanisms, however, does not cover all layers [62].

In their study, Chen *et al.* conducted a comprehensive analysis of the security aspects of the Ethereum system, encompassing vulnerabilities, assaults, and protection mechanisms [42]. The Ethereum platform and its associated layers, including the network, consensus, data, and application, were comprehensively addressed. In contrast, our work is not limited to one specific platform.

Li *et al.* did a comprehensive analysis of the security vulnerabilities associated with widely used blockchain systems [82]. The researchers conducted a blockchain security assessment by examining 20 distinct vulnerabilities, 6 attacks, and 5 corresponding defenses. Nevertheless, a layer-based categorization is missing in their work that we address. In addition, we analyze other relevant attacks.

Saad *et al.* explored the attack surface of blockchains [109] where 22 attacks and 33 defense mechanisms were covered. We take a different approach. We discuss attackers' incentives, specific vulnerabilities, stakes from attackers' perspective to launch such attack and categorize attacks using a layered approach. Furthermore, our analysis includes certain attacks that were absent in theirs.

Homoliak *et al.* introduced the security reference architecture (SRA) as a framework for blockchains. Four layers were used in the study, including network, consensus, replicating state machine, and application. The ISO/IEC 15408 threat risk assessment standard was utilized for this purpose. Their study differs from ours in several ways. Firstly, we adopt the Application layer and the Meta-Application layer as suggested by [62], which leads to different results and categorizations. Secondly, we illustrate how the components of different layers are impacted by attacks. Lastly, we utilize our own custom framework presented in Section 4.2 to gain deeper insight into each attack.

Wen *et al.* studied attacks and countermeasures using a six-layer blockchain model [127]. The corresponding layers are as follows: Data, Network, Consensus, Incentive, Contract and Application. However, while organizing attacks, they combined the consensus and incentive layers and did not cover the application layer. In addition, they provided an overview and countermeasures for each attack. In contrast, we dive deep into each attack and explicitly analyze utilizing our framework. We also analyze inter-layer connections for attacks.

Zheng *et al.* also proposed six layers: Data, Network, Consensus, Incentive, Contract, and Application [130]. Their research is primarily focused on the security and privacy features and techniques of blockchain, as well as the comparison of various consensus mechanisms. In discussing security and privacy properties, they briefly covered four major types of attacks and vulnerabilities. However, we developed a completely different strategy, diving deep into the categorization of layer-based attacks.

In addition, several studies have been conducted to categorize attacks on blockchain systems. Moubarak *et al.* examined blockchain security, with a focus on Hyperledger, Ethereum, and Bitcoin [94]. They provided a summary of several challenges and attack scenarios and briefly reviewed possible mitigation techniques. Anita *et al.* presented a taxonomy of security risks associated with blockchain technology, introducing 7 groups covering 17 attacks [25]. Chen *et al.* proposed a blockchain attack classification system using three categories encompassing 11 attacks [43]. However, none of these studies used any layer-based categorization.

4 Blockchain layers, property taxonomy and attacks

In this section, we group attacks according to the layers they target. Many attacks affect multiple layers at the same time. In those situations, we highlight the attack in the most affected layer. Toward this aim, we present the layer-based approach in blockchain as introduced in [62] (Section 4.1). Then, we introduce the taxonomy of properties (Section 4.2) and present the list of attacks (Section 4.3).

4.1 Layer-based Approach on Blockchain

In a blockchain system, various components exist, each responsible for performing one or more specific functions. Hence, it is essential to decompose the whole system into different layers to achieve modularity, scalability, security, and interoperability. David *et al.* suggested four layers of blockchain: consensus, mining, propagation, and semantic [129]. However, they mostly focused on the blockchain system, ignored the P2P component altogether. Ferdous *et al.* proposed another layer scheme with four layers: network, consensus, application, and meta-application [62] and this addressed the issue in the layer model of David *et al.*. Our attack analysis is based on the scheme proposed by Ferdous *et al.*. Here, we briefly discuss the layers and their responsibilities

- **Meta-Application Layer:** The objective of a blockchain system's meta-application layer is to create an overlay atop the application layer so that other application domains can benefit from the logical interpretation of a blockchain system.
- **Application Layer:** The logical interpretation of a blockchain system is determined by the application layer. A logical interpretation can be a cryptocurrency, a smart contract, a reward mechanism, etc.
- **Consensus Layer:** The distributed consensus is provided by the consensus layer. Various consensus algorithms such as PoW, PoS, etc. are the essential part of this layer. These algorithms are utilized to obtain the necessary consensus in the system.
- **Network Layer:** The network layer components are in charge of managing network capabilities, such as joining the peer-to-peer network, adhering to the networking protocol,

informing newly joined nodes of the current status of the blockchain, propagating and receiving transactions and blocks, and other related tasks. Figure 2 illustrates a high-level overview of blockchain layers.

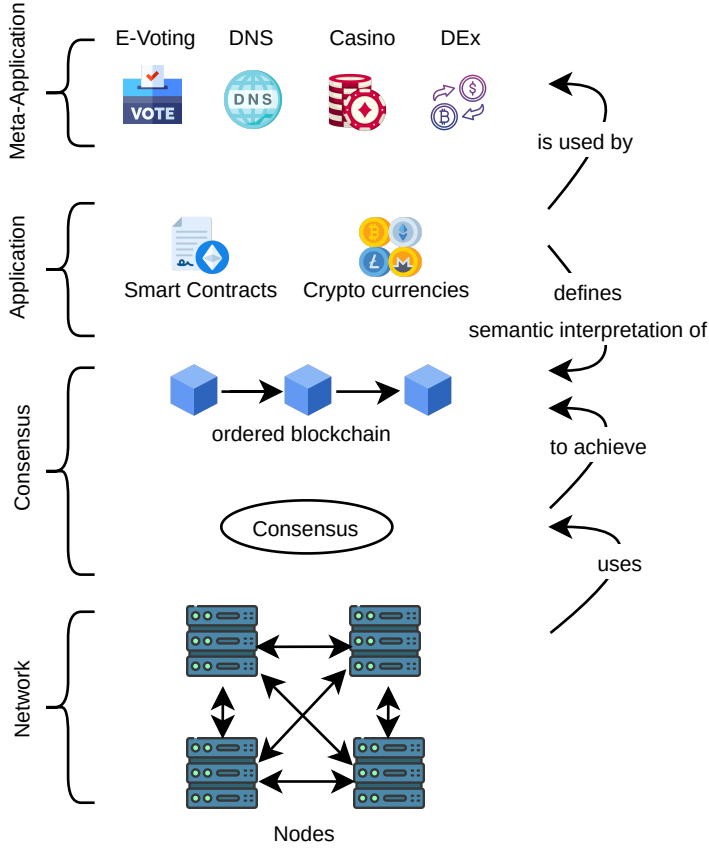


Fig. 2. Blockchain Layers

4.2 Taxonomy of properties

We study each attack using the following properties:

- I **Motivation & vulnerability:** This refers to the overview of attacker(s) intention, in regard to parts and properties of blockchain system that an attacker wants to take advantage of and specific desirable states of a blockchain system for an attacker in certain attacks.
- II **Attack strategy overview:** This property implies the generalized method of attack execution with simplified states.
- III **Conditions & outcomes:** The term reflects system condition(s) or fact(s) that affects how the attack turns out & which parts of the system strongly impacted by an attack(s). This also elaborates the risk factors regarding an attacker's motive & consequences of a successful attack(s) in a system.

IV **Enhancements:** This property explains the connection to other attacks or leveraging situations where the current attack may lead to another.

V **Plausibility & prevention measures:** This indicates the availability of a solution to a specific assault and provides a view of the practical difficulty of executing the attack.

4.3 Analyzed attacks

The list of attacks analyzed in this article is presented in Table 1. In order to compile this list of attacks, we conducted a comprehensive literature review in several sources including the arXiv preprint server, journals and conference proceedings. It was based on the frequency and consequences of such attacks as reported in earlier studies in those sources. By incorporating multiple credible sources, we ensured that the list encompassed the latest and most prevalent attacks in the field. This systematized approach provides an excellent basis for the examination carried out in the present work.

Table 1. Surveyed Attacks

Network Layer	Consensus Layer	Application Layer	Meta-Application Layer
Balance Attack	Punitive Forking Attack	Replay Attack	Front Running Attack
Sybil Attack	Block Withholding Attack	Short Address	Block Stuffing Attack
Eclipse Attack	Fork After Withholding		
BGP Hijacking attack	Vector 76		
Pool Hopping attack	Selfish Mining Attack		
	Race Attack		
	Finney Attack		
	Long Range (LR) : Simple		
	LR : Posterior Corruption		
	LR : Stake Bleeding		
	P+Epsilon Attack		
	Feather Forking Attack		
	Bribery Attack		
	Consensus Delay Attack		

5 Network Layer Attacks

Network layer attacks typically interrupt node communications and target processes that allow nodes to communicate and agree on data, exploit vulnerabilities in node detection, transaction/block propagation, and communication protocols, and disrupt the functions of the network layer, such as data transfer and synchronization. Communication channels underpin blockchain networks; therefore, network layer attacks target them. These channel disruptions have a direct influence on network performance. The attacks analyzed under the network layers are discussed next.

5.1 Balance Attack

The Balance attack was first identified by Christopher Natoli and Vincent Gramoli in 2017, which targets the blockchain fork mechanism[97]. The main target of the attack is Ethereum; however, Bitcoin is also vulnerable to the same strategy. It is to be noted that this attack is applicable to both the network and consensus layers. Next, we analyze this attack using the selected properties.

Motivation & vulnerability: In this attack, the attacker leverages Ethereum’s Ghost Protocol [120] or, in the context of Bitcoin, the longest chain rule. The objective is to hinder block propagation across the network and achieve double-spending [110], by manipulating the selection of branches.

Attack strategy overview: The attack strategy overview is presented in Figure 3.

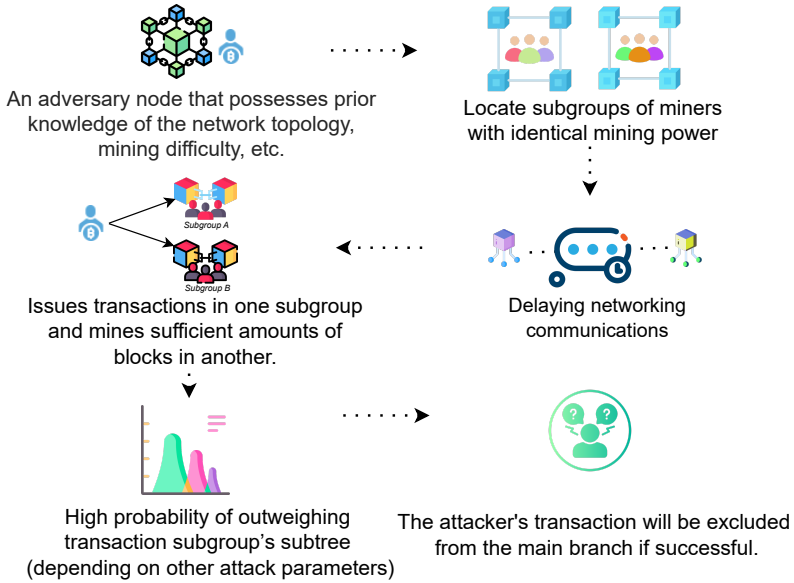


Fig. 3. Overview of Balance Attack

Conditions & outcomes: The attack requires key components such as knowledge of miners’ logical or physical communication graphs, computational capacity for mining, and the current difficulty level [97]. A Successful attack will damage honest miners, making mining efforts ineffective for one subgroup and potentially harming intended payment recipients.

Enhancements: Regardless of Balance attack success, an attacker may develop a consensus delay attack situation by trying to establish a reasonable delay between nodes, as explained later in our work in Section 6.14.

Plausibility & prevention measures: This attack is feasible with limited mining power and dynamic data from multiple sources, including blockchain communication architecture, such as propagation latency, difficulty, and linked nodes. So far, No mitigating methods for forkable blockchains have been presented for this attack [97].

5.2 Sybil Attack

First put forward by John R. Douceur in his 2002 paper "The Sybil Attack" [53], Sybil attack is an attempt to dominate a network by leveraging many aliases. Successful Attackers can host many nodes and outvote reliable users [21, 73]. Sybil attacks are a milder form of an attack known as a *51% attack*. In a 51% attack, if malicious miners happen to control more than 50% of the network hash rate, they manipulate the blockchain protocol rules.

Motivation & vulnerability: With the ultimate motive to corrupt the peer-to-peer network, the attacker aims to create and maintain large numbers of digital identities (assisted by multiple

devices, virtual machines, IP addresses, and botnets [110, 123]) and use these identities to capture new nodes by exploiting node limitations in state synchronization [26].

Attack strategy overview : The attack strategy overview is presented in Figure 4.

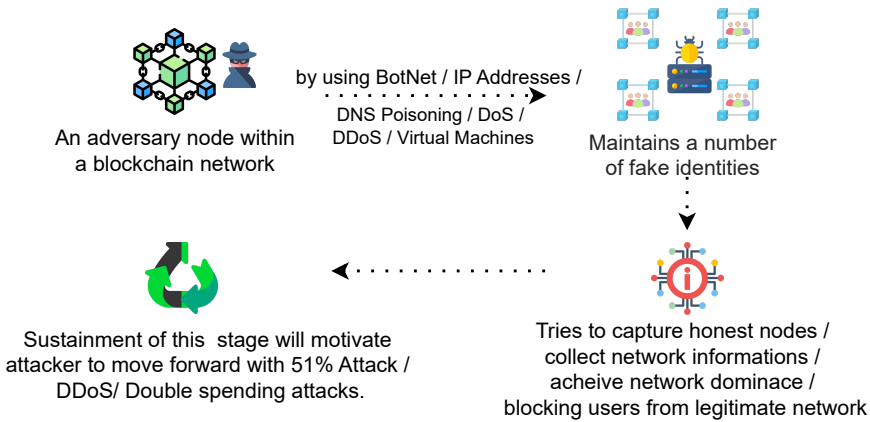


Fig. 4. Overview of Sybil Attack

Conditions & outcomes : The number of Sybil nodes controlled by the attacker and cooperation among them determine the attack severity. The attacker can influence consensus at specific lengths and lower the block propagation performance if a fake network is developed [73]. Due to the attacker's actions, the majority bar is raised, wasting more processing power [123].

Enhancements: A successful attack can lead to consensus delay, DoS and DDoS attacks [123], or 51% attacks [73]. Also, it raises the chance of double-spending attacks like Finney Attack and Race Attack, outlined later in our work.

Plausibility & prevention measures: In 'The Sybil Attack,' John R. Douceur said that Sybil attacks are always possible without any logically centralized authority, such as in the public blockchain [53]. Public blockchains have been attacked by Sybil attacks in the past [124]. In addition, Bitcoin created the Bloom Filter to protect and minimize privacy threats for lightweight nodes, or Simplified Payment Verification (SPV) nodes [26]. Different studies show that this Bloom Filter does not fully protect a node from synchronizing with a Sybil node or agent [101].

Additionally, a wallet-generated address method has been proposed to identify Sybil attacks on public blockchains [123]. Direct and indirect identity validation methods have been proposed to counter Sybil attacks [53]. Several strategies, including registration-based methods, third-party mixing protocols, neighborhood similarity, network clustering, and position verification, have been proposed to prevent Sybil attacks [123]. Some of these methods are also applied to specific networks but not in major public blockchains. In addition, application-specific defenses are useful to defend against systemic Sybil attacks [73].

5.3 Eclipse Attack

An Eclipse attack happens when an attacker segregates a particular user or node within a peer-to-peer (P2P) network. The idea of an Eclipse attack on blockchain was first discussed by Heilman *et al* [71]. Although the steps of this attack are somewhat similar to the Sybil attack, the motivations and objectives are distinct [50].

Motivation & vulnerability: Eclipse attacks exploit blockchain’s peer-to-peer protocol limitations (i.e. - outgoing and incoming connections rule, persistent network information [71], node state synchronization [78]) to isolate a node from the network. Next, the attacker entirely controls the victim’s information access to restrict their blockchain view or co-opt their computing power for additional attacks [86].

Attack strategy overview: The attack strategy overview is presented in Figure 5.

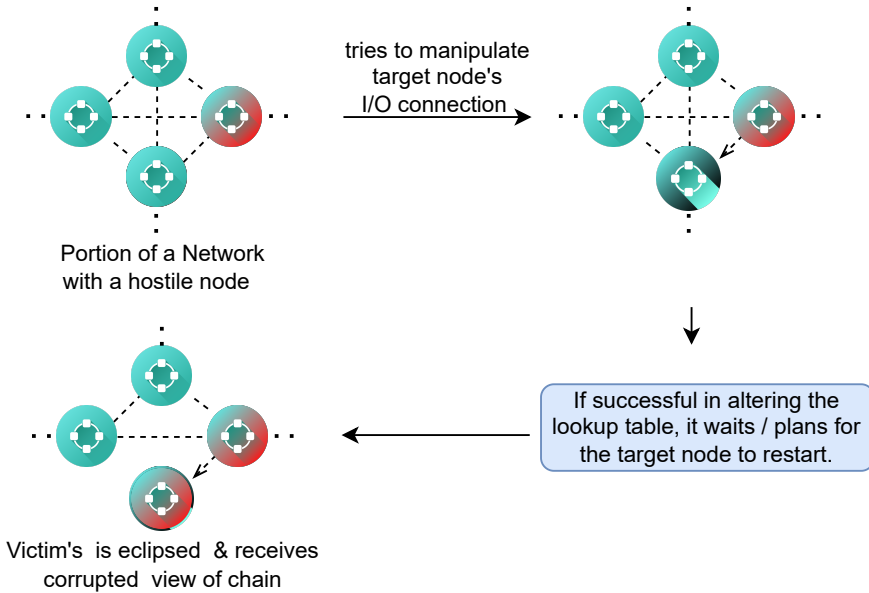


Fig. 5. Overview of Eclipse Attack

Conditions & outcomes: Depending on the blockchain, its client’s type, and the attacker’s choice, there are many determining factors in eclipsing a node. Most common scenarios require some pre-computation [78], lookup table, and outgoing connection alteration, along with ensuring victim node’s restart [71]. Successful eclipsing disconnects the victim from the genuine blockchain state, which the attacker exploits.

Enhancements: If the victim is successfully eclipsed, it gives the attacker advantages in block racing, enables transaction hiding and selfish mining scenarios, 0/N confirmation double-spending and stake bleeding attack. Also it facilitates the wastage of the victim’s mining power [50, 71, 78, 86, 131].

Plausibility & prevention measures: Without a centralized authority, the attack is feasible, although resource utilization may vary [86]. To stand against eclipse attacks in Ethereum, solutions like eliminating the reboot exploitation window, ensuring constant seeding, and limiting incoming TCP connections have been implemented [86]. Bitcoin implemented countermeasures include deterministic random eviction, random selection, increasing bucket numbers, feeler connections, and test before evict [70, 71].

5.4 BGP hijacking Attack

This attack exploits the Border Gateway Protocol (BGP) routing protocol. Established in 1989, BGP is used to determine routing decisions among autonomous systems (ASes) on the Internet. BGP Hijacking involves manipulating internet routing tables using the protocol and illegitimately obtaining clusters of IP addresses. Through the BGP hijack attack, attackers can impersonate the IP address of their targets [126]. This attack poses a threat to blockchain by allowing hostile actors to redirect mining pools, resulting in income loss. In 2014, around \$83,000 (USD) was stolen by the BGP hijacking attack [121]. On 17 August 2022, an assailant carried out a BGP hijack on Celer Bridge, a cryptocurrency service, leading to the loss of \$235,000 worth of bitcoin [83].

Motivation & vulnerability: The attacker utilizes BGP route redirection to hijack cryptocurrency mining operations without payment, capturing honest miners in a malicious pool and maintaining the false pool for each honest miner in brief intervals. To keep the activity undetected [121], the attacker can partition the network or intercept a portion of connections to induce delays in blockchain traffic through BGP hijacks [27].

Attack strategy overview: The attack strategy overview is presented in Figure 6.

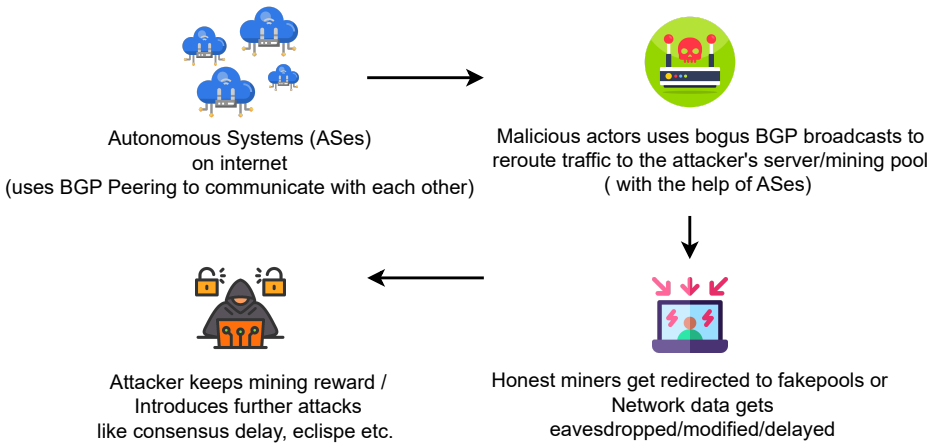


Fig. 6. Overview of BGP hijacking Attack

Conditions & outcomes: Access to Internet Routing Table is crucial, and efficiently corrupting it indicates susceptibility to the attack. It will leave miners in a rewardless state [27]. Also, the attacker needs to ensure the bogus BGP announcements are unfiltered by the upstream network. Moreover, a BGP attack may lead to unexpected forks in the chain [83].

Enhancements: A successful attack can isolate a node, creating a barrier in communication that increases vulnerability in multiple domains such as node eclipse, 0-confirmation double-spends, selfish mining, and consensus delay [27].

Plausibility & prevention measures: One proposed solution is RouteChain, which utilizes a blockchain-based routing system to address BGP Hijacking [107]. Similar solutions like as BlockJack, which is developed using Hyperledger Fabric and Quagga, are also recommended [113]. Lukas *et al.* proposed an alternative method to improve the security in the BGP Protocol by overseeing AS border routers [87]. For long-term defenses, employing separate control and data channels, alongside UDP heartbeats and encrypted communications, serve as effective solutions [27].

5.5 Pool Hopping Attack

A mining Pool is formed by miners to speed up the new block mining process and receive rewards with accumulated effort and less difficulty. Depending on where the pool is right now, predicted profits, volatility, and maturity time will change. The practice of mining just when the payoff is high and the difficulty is low and leaving when the opposite is true is known as pool hopping [117]. Thus, hoppable pool mining is unfavorable, as more miners hop, the pool becomes increasingly unstable. Everyone mining alone or at a hopping-proof or very hopping-resistant pool is the only sustainable approach [105].

Motivation & vulnerability: Various mining pools employ different reward methods as explained in Section 2.2. A logical miner has a choice to join a larger reward pool to maximize the monetary outcome of their mining efforts. The exploitation of block reward distribution can be achieved by the analysis of mining pool behavior and the selection of an effective mining method [41]. Therefore, the assailant has the ability to develop a steady level of revenue with pool hopping.

Attack strategy overview: The attack strategy overview is presented in Figure 7.

Conditions & outcomes: For attackers to execute this attack successfully, they need to time the pool switch effectively, and keep track of longer mining rounds [133], network delays in pool joining, and the current hashrate of pool [105, 133]. If executed properly, it will impact the pool’s honest miners, the pool’s share and fluctuations in total hashrate of the pool. Overall, an unsustainable mining pool with increased block generation time and an increased chance for an attacker to secure roughly 28% more rewards, depending on the hashrate ratio between between hoppers and honest miners [105].

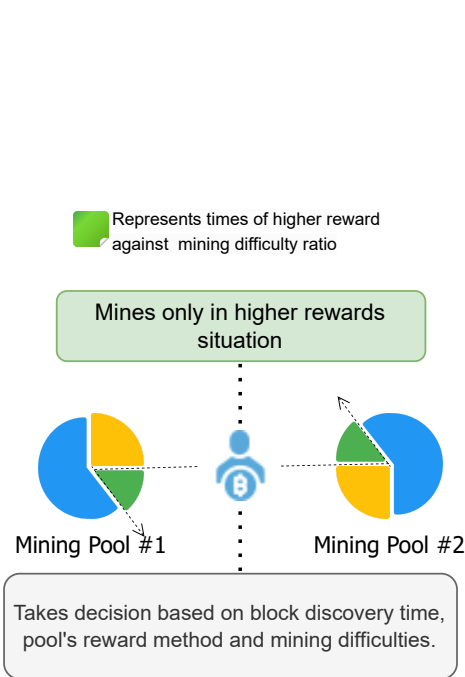


Fig. 7. Overview of Pool Hopping Attack

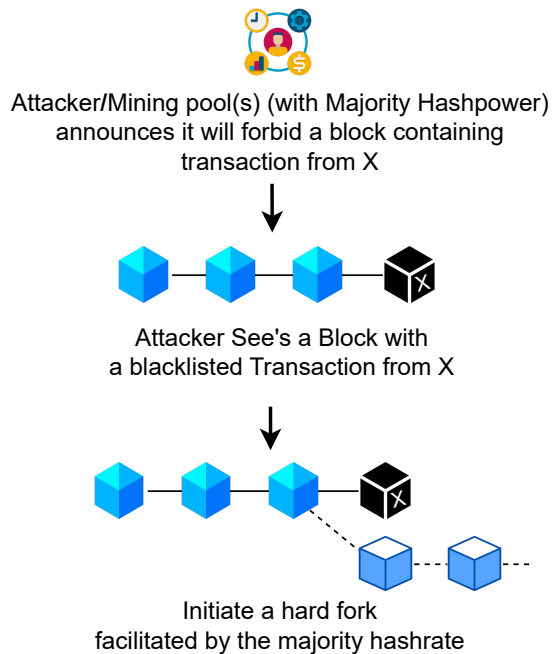


Fig. 8. Overview of Punitive Forking

Enhancements: This attack does not lead to another attack.

Plausibility & prevention measures: There have been previously created technologies, such as Bithopper, for pool hopping that have been successful [39]. Proposed solution to mitigation of Pool hopping includes mining resource allocation mechanism (developed with python and cobyta) [41]. A pool hopping prevention strategy using a smart contract is also proposed which aims to detect and prevent the attacks using miner certificates and by introducing risk factors in pool switching during incomplete block mining [117]. In addition to these, a novel mining pool design is also proposed, based on zero-determinant theory and iterated prisoner's dilemma (IPD) game which is fee-free and introduces fairness as long as mutual cooperation exist between miners and the pool [115].

6 Consensus Layer Attacks

Blockchain systems use a consensus method to ensure that all participants agree on the current state of the ledger. This process serves as the foundation for trust and security in any blockchain system. Malicious actors can disrupt the consensus method used by participating nodes to validate malicious transactions and manipulate the blockchain state. These attacks aim to disrupt, corrupt, or misuse the consensus process by taking advantage of weaknesses in the consensus algorithm of the blockchain system, potentially causing catastrophic effects for the blockchain. The attacks in the consensus layer are discussed next.

6.1 Punitive Forking Attack

Punitive forking refers to the act of targeting a specific entity in order to enforce laws or ban its transactions. The attack targets the consensus layer and compromises the integrity of the chain, potentially leading to the centralization of hashpower [46].

Motivation & vulnerability: This attack can be carried out in a variety of ways by exploiting hashpower advantage. It is also viable to announce an address blacklisted and enforce it for pools owned by the attacker if they own more than 51% of the network hashrate [75, 89].

Attack strategy overview: The attack strategy overview is presented in Figure 8.

Conditions & outcomes: It is required by the attackers to possess more than 51% of the network's hashpower in order to convince miners to suspend all transactions from a particular address. Hard forks and restricted addresses are the outcomes of successful assaults [24, 106]. It also has an effect on the reliability of a lesser part ($\leq 49\%$) of the consensus as well.

Enhancements: This attack does not provide any leverage for subsequent attacks.

Plausibility & prevention measures: Despite blockchain's decentralized nature, mining pools can, in the worst-case scenario, consolidate hashpowers that could be used for this assault, making any solution proposal impractical at this time. Currently, achieving the goal of hashpower domination on the network requires the accumulation of hashpower from a just 2 or 3 mining pools [29]. Thus, the prevention of punitive forking remains an open challenge [46].

6.2 Block Withholding (BWH) Attack

Rosenfeld et al. introduced the Block Withholding (BWH) attack [105]. In this attack, the attacker or a group of attackers prevents the mining pool from receiving a legitimate block to affect the pool manager or the other honest pool miners.

Motivation & vulnerability: The motivation of this attack is to increase profits from mining or to harm the pool manager. The attack impairs the mining pool's reward-sharing scheme. This attack

can be carried out by a single dishonest miner or a group of dishonest miners with a significant quantity of mining power. Also, this attack can be launched by a pool to another pool [58]. Most of the mining pools are open. Miners can join in these pools through a public Internet interface. However, these open pools are susceptible to traditional block withholding attacks.

Attack strategy overview: Rosenfeld talks about two different variants of this attack: Sabotage and Lie in Wait [105]. In ‘Sabotage’, if a dishonest miner finds a ‘share’, he never submits it to the pool manager. In ‘Lie In Wait’, if the attacker solves a block, he withholds it and keeps adding his share to the pool. After a certain amount of time, he submits the block. Thus he receives more profit. The attack strategy overview is presented in Figure 9.

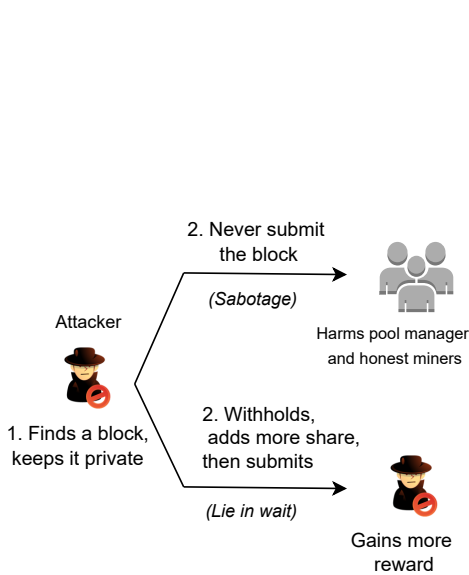


Fig. 9. Overview of Block Withholding Attack

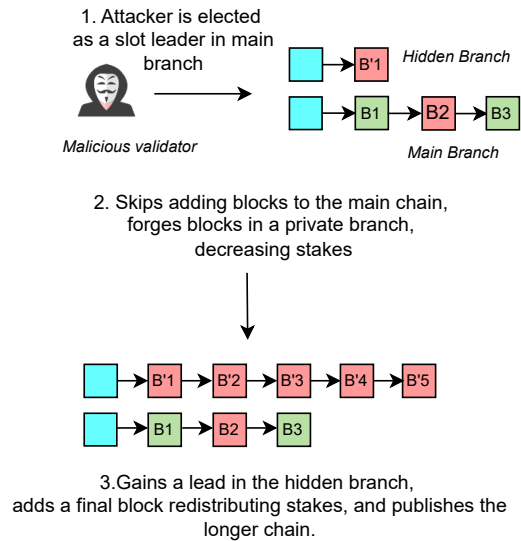


Fig. 10. Overview of Long Range Attack (Stake Bleeding)

Conditions & outcomes: This attack has a significant relation with the attacker’s hash power as he needs to add more shares or solve a block. This attack, in any form, negatively impacts both the pool operator and the honest miners who operate within the pool. If the PPS reward sharing system is applied in the pool as described in Section 2.2, the manager has to bear the loss. The ‘Lie in Wait’ variant can be useful for the attacker to gain monetary value. However, Curtois *et al.* argued that the ‘Sabotage’ variant can also bring monetary value to the attacker [47].

Enhancements: This attack can lead to Race attack described in Section 6.6 [58].

Plausibility & prevention measures: In 2014, the ‘Eligius’ mining pool experienced a 300 BTC loss [128], demonstrating the attack’s feasibility. A two-step PoW mechanism called ‘Oblivious Share’ is proposed to prevent this attack, but it wastes miners’ computing resources [105]. To avoid this attack, open mining pools should only include known and trusted individuals [47].

6.3 Fork After Withholding (FAW)

This is another variant of the Block Withholding attack, but with a different goal and result. Kwon *et al.* first proposed this strategy and came to the conclusion that this attack is always advantageous to the attacker [80].

Motivation & vulnerability: The motivation of this attack is to gain profit by creating an intentional fork in the blockchain. This attack can be carried out by a single miner or a pool against another pool. Similar to the BWH attack, the open pools are vulnerable to the FAW attack.

Attack strategy overview: In this attack, the attacker splits his mining power into two different fractions. One part is invested into the target mining pool, and the other part is utilized for legitimate solo mining. The rest of the attack strategy is briefly presented in Figure 17.

Similarly, an attacker can simultaneously target numerous pools and distribute his mining power among them. This approach creates a fork with $n+1$ branches by focusing on n pools. Furthermore, an attack can be launched by two pools against one another.

Conditions & outcomes: The attacker needs find an FPoW. Mining power plays a vital role here. He needs to split his mining power optimally. Therefore, he must be aware of the computational power of the target pool and the likelihood that his FPoW will be chosen for the main chain [80]. If the attacker possesses Sybil nodes, it will be beneficial for him to notice external block propagation faster. A successful attack may lead to a fork in the blockchain or increase the attacker's rewards. The FAW attack can significantly increase an attacker's rewards by up to four times in comparison to a BWH attacker [80].

Enhancements: This attack can lead to Race attack as it is similar to BWH attack.

Plausibility & prevention measures: This attack is feasible and can be launched against Ethereum, Dogecoin, Permacoin and Litecoin as well as Bitcoin blockchain systems [80]. Several methods have been proposed to prevent this attack, like 'Oblivious Share' [105] and 'Two Phase Proof of Work' [60]. In addition, Kwon *et al.* [80] propose a reward-sharing mechanism which decreases the risk of this attack but causes high reward variance.

6.4 Vector76 Attack

This attack was initially suggested by a user named vector76 in a bitcoin forum [125]. It combines the elements of the Race attack and the Finney attack (discussed later). Here, the attacker pre-mines a block and then, attempts to broadcast and add the pre-mined block to the main chain. This is also known as a '1-confirmation' attack.

Motivation & vulnerability: The cryptocurrency exchanges that allow for deposits and withdrawals of funds are the main targets of this attack. A cryptocurrency exchange is a digital platform that allows traders to exchange cryptocurrencies for various assets, including fiat currencies or other cryptocurrencies. If the exchange service allows an incoming connection, the attacker gets a chance to launch this attack.

Attack strategy overview: The attack strategy overview is presented in Figure 11.

Conditions & outcomes: The attacker must pre-mine a block. The targeted exchange service must allow other nodes to discover and directly connect to their nodes. A successful attacker keeps the withdrawn money along with the later transaction as shown in Figure 11. In the event of a failed attack when the pre-mined block is not added to the main chain, the attacker will still have a deposit in the exchange. If the pre-mined block remains in the main chain, the block generation reward also goes to the attacker.

Enhancements: This attack does not provide any leverage for subsequent attacks.

Plausibility & prevention measures: There have been no occurrences of this attack thus far. An effective mitigation strategy involves employing a waiting period for transactions until numerous

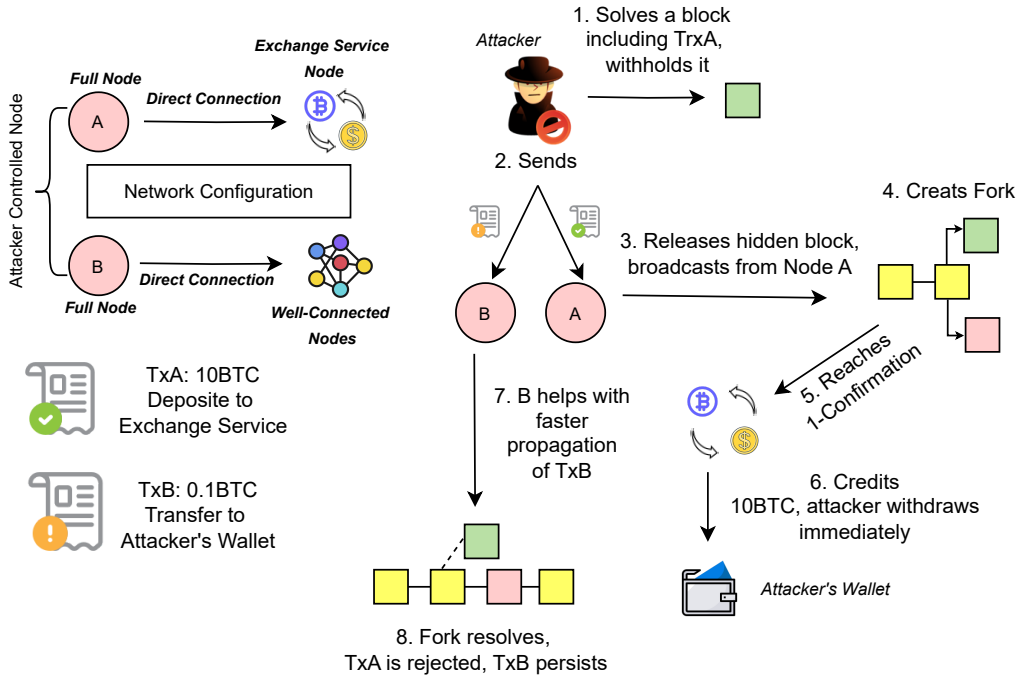


Fig. 11. Overview of Vector 76 Attack

confirmations are received. Additionally, exchange services should refrain from accepting incoming connections.

6.5 Selfish Mining Attack

Eyal *et al.* first modeled this attack on the Bitcoin system [61]. In this attack, the attackers purposefully keep their blocks secret to maximize the rewards. Instead of adding their discovered blocks to the public blockchain, the attackers build a separate, private version of the blockchain and continue to add new blocks to it. Meanwhile, honest miners, unaware of the existence of the private chain, continue to mine on the public version of the chain, which is actually lagging behind. The race between public and private versions of the same chain ends when the attacker publishes the longest private chain. If the network is built upon the longest chain rule, the network adopts the attackers' chain, causing the honest miners' work to be invalidated.

Motivation & vulnerability: Attackers want to enhance their rewards and control the network by withholding mined blocks and surreptitiously mining on top of them. This approach uses the longest chain rule in the proof-of-work consensus mechanism to fork the blockchain.

Attack strategy overview: The attack strategy overview is presented in Figure 12. This attack can also take place in a mining pool. The malicious pool can act as a single agent and repeat the event [61].

Conditions & outcomes: The attacker must initiate the process by discovering a valid block. The mining power of the attacker is crucial in this context. According to [80], 9% hash power

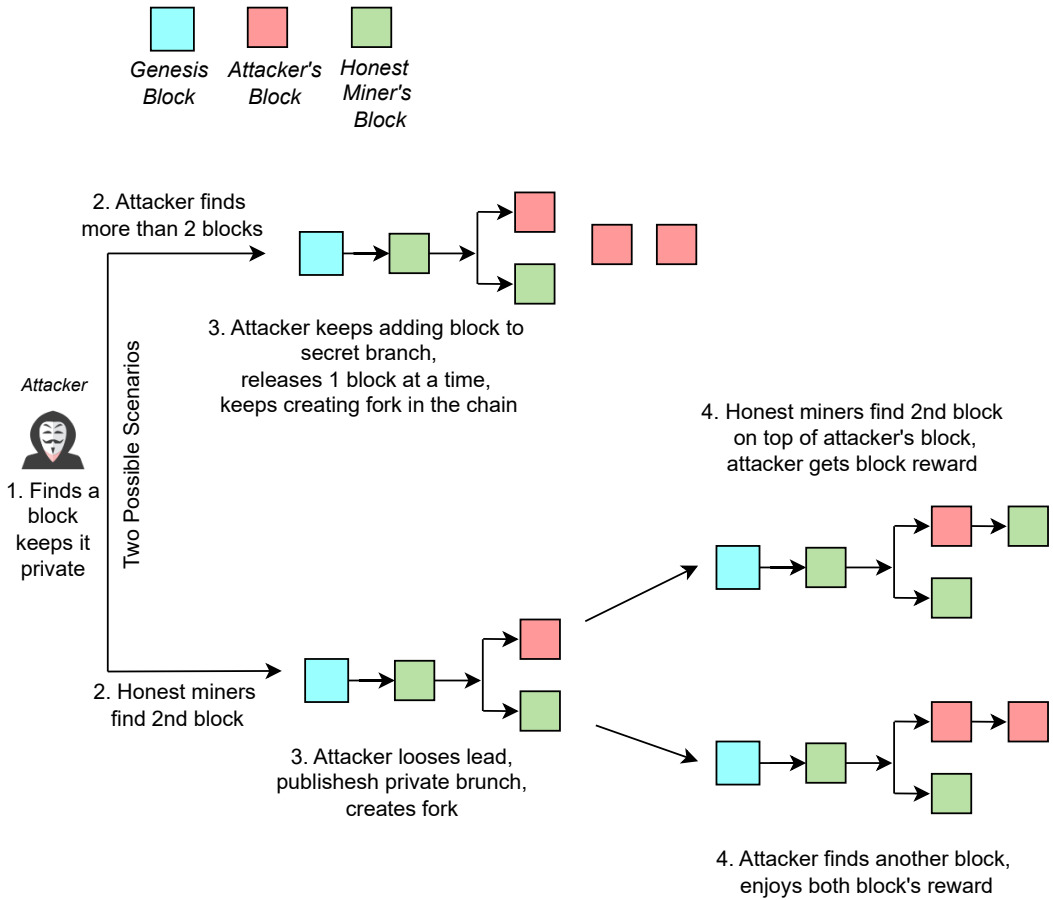


Fig. 12. Overview of Selfish Mining Attack

is required. In addition, to achieve a successful fork, the attacker must transmit the block more rapidly over the network. The consequences of such an attack are significant. It interferes with the consensus protocol and compromises the integrity of the system. Miners expend their resources on unproductive blocks. Furthermore, the inclusion of all transactions in the block of the honest miner is likewise met with rejection, thus creating opportunities for various forms of other attacks.

Enhancements: A successful attack can lead to a double-spending and fork after withholding attack.

Plausibility & prevention measures: This attack is considered to be impractical [80]. To counter this attack, a timestamp-based solution is proposed by Solat *et al.* [119]. Saad *et al.* proposed the concept of 'truth state' for blocks and included the expected confirmation height parameter in transaction data structures [108].

6.6 Race Attack

Karame *et al.* proposed and modeled this attack [76]. This is an example of a 'double-spending' attack, where an attacker tries to use the same currency for two separate transactions.

Motivation & vulnerability: The ‘Fast Payment System’ of Bitcoin is exploited by this attack. On average, a new block generation in Bitcoin network takes approximately ten minutes [76]. It is clear that while taking Bitcoin payments, vendors and merchants such as supermarkets, take-away stores, vending machines, etc. cannot rely on transaction completion (i.e. new blocks being added and having enough confirmation). Therefore, as soon as the network transmits a transaction containing the required amount of BTCs from the client to one of its addresses, the merchant can accept bitcoin payments with no confirmations for low-cost transactions [2]. The attacker can use this loophole to their advantage and perform this attack.

Attack strategy overview: In this attack, an attacker generates two distinct transactions that utilize the same fund. One transaction goes to the merchant and the other one goes to a wallet controlled by the attacker. Eventually, the merchant releases the product without confirmation. The attack strategy overview is presented in Figure 13.

Conditions & outcomes: The faster transmission of the fraudulent transaction is necessary for this attack to succeed. In a peer-to-peer system, the attacker must establish a direct connection with the merchant. By doing this, the actual transaction will be sent to the victim merchant more quickly. Moreover, the attacker must also have control over a large number of Sybil nodes to disseminate the fraudulent transaction more quickly than the original one. Thus, a race condition is created. The transaction added to the blockchain is the one that gets to the miners of the network first. The attacker will be able to keep the money and the product if he wins the race. After a successful attack, the merchant suffers from financial loss, and a blockchain fork may occur.

Enhancements: This attack does not provide any leverage for subsequent attacks.

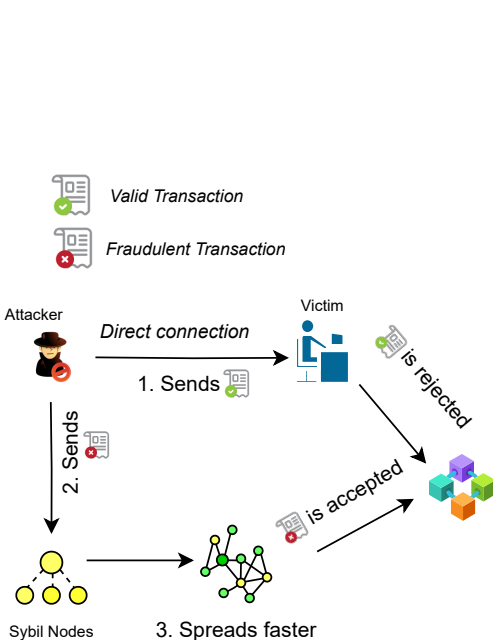


Fig. 13. Overview of Race Attack

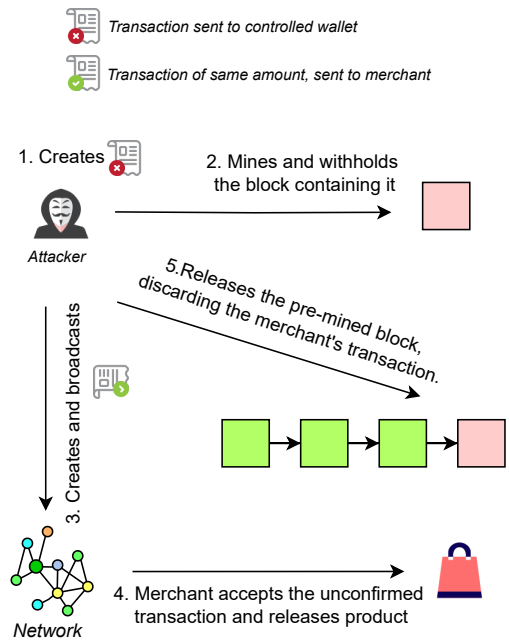


Fig. 14. Overview of Finney Attack

Plausibility & prevention measures: If a 0-confirmation payment is accepted anywhere, this attack is possible. Several ways to prevent this attack have been suggested and implemented. For example, adding listening period and observer nodes in network [76], reward-based observers [102], forwarding double-spending attempts [77], connecting with a large random sample of nodes and not accepting incoming request [28] etc. However, complete protection against double-spending attacks is not available yet [28].

6.7 Finney Attack

This attack was first suggested by Hal Finney in a bitcoin forum back in 2011 [69]. This is an instance of a double-spending attack where the attacker purposefully spends the same currency multiple times.

Motivation & vulnerability: The attacker wants to double-spend the cryptocurrency. Similar to the Race attack described in Section 6.6, the attacker targets those merchants who accept ‘Fast Payment’.

Attack strategy overview: The attack strategy overview is presented in Figure 14.

Conditions & outcomes: The attacker has to find a valid block first. Additionally, he must verify that the block includes his counterfeit transaction. Merchant also needs to accept 0-confirmation payments. In a successful attack, the attacker receives both the block generation reward and the merchant’s product without incurring any additional expenses. The merchant suffers from financial loss. An additional possibility of this attack is the occurrence of a blockchain fork.

Enhancements: This attack does not provide any leverage for subsequent attacks.

Plausibility & prevention measures: The complex and time-sensitive nature of this attack makes its occurrence quite improbable. So far, this attack has not been reported for any blockchain system [109] and hence, it remains only theoretical. No mitigation technique is exclusively available for this attack. However, some general mitigation techniques have been proposed to prevent double-spending attacks described in Section 6.6.

6.8 Long Range Attack: Simple

In a Proof-of-Stake (PoS) system, where blocks are generated without the requirement of solving computationally difficult mathematical problems, the possibility of a long-range attack exists [32]. A malicious block validator initiates a process of forking the chain by reverting to the genesis block. He creates a separate branch that has a partially or an entirely different history from the main branch. As the forged branch stretches further than the main branch, previous transactions are eliminated from the chain. Ultimately, the alternate branch is released and the attacker achieves his goal.

Motivation & vulnerability: The main motivation of this attack is to modify the timestamp and manipulate the block history. The Proof of Work (PoW) based systems necessitate a substantial amount of computational power to modify the history of a block. On the contrary, the PoS-based systems utilize a technique known as ‘Nothing At Stake’ [32]. The validator encounters no risk when making consensus decisions. Theoretically, a dishonest validator can construct an alternative branch from the main chain of a Proof of Stake (PoS) blockchain at any desired point, without suffering any tangible costs.

Attack strategy overview: The attack strategy overview is presented in Figure 15.

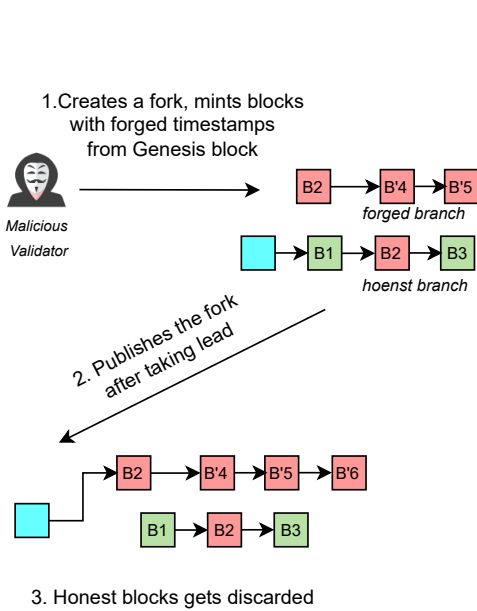


Fig. 15. Overview of Long Range (Simple) Attack

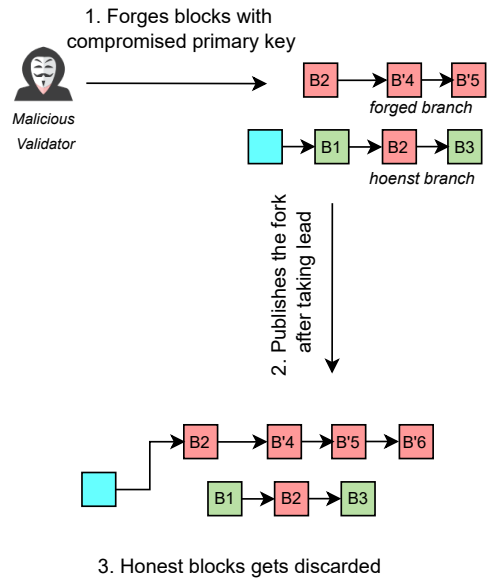


Fig. 16. Overview of Long Range Attack (Posterior Corruption).

Conditions & outcomes: For this attack to be successful, it is imperative that the block timestamp is disregarded and the longest chain is always selected as the main branch [51]. The attacker must generate blocks in advance to gain a competitive advantage. If this attack is successful, the history of blocks becomes corrupted and double-spending is possible. Furthermore, the attack results in the manifestation of ‘Weak Subjectivity’ [51]. A Weak Subjectivity refers to the challenge that is faced by newly added nodes and those that are reconnected to the network after a prolonged period of disconnection. Due to their lack of synchronization, these nodes do not possess accurate information regarding the development of the blockchain, which hinders its ability to determine the primary chain.

Enhancements: This attack does not provide any leverage for subsequent attacks.

Plausibility & prevention measures: This attack is purely theoretical. No occurrence has been observed till date. This attack can be prevented by considering timestamps while selecting the chain [64].

6.9 Long Range Attack: Posterior Corruption

If timestamp forging is not feasible in a PoS system and an attacker possesses the private key of different validators, then this attack can be carried out. Within the alternate branch, the attacker alternates block histories using both his private key and the compromised keys. Following a significant advantage, the attacker deploys the alternative branch of the blockchain, which is subsequently acknowledged and adopted.

Motivation & vulnerability: The attacker wants to rewrite the block history. He does this with compromised private keys. To maintain an equitable PoS consensus based system, validators must engage in rotation. Additionally, there should be a means to voluntarily or forcibly remove a

validator from the system. In a real-world system, a validator may be unreliable due to potential changes in incentives or compromises to the system. Therefore, a validator has the option to retire once they have generated a specific number of blocks (denoted as n). He can withdraw his investment from the system by liquidating it. And will no longer be included in it. However, the blocks he created remain within the system. By utilizing his private key, it is possible to fabricate the preceding n blocks [103]. The attacker can perhaps engage in bribery with the retired validator or somehow hack his private key. Then, whenever the attacker is elected as a block validator, he possesses the ability to generate further counterfeit blocks by utilizing both their own and the hacked private keys at a faster pace.

Attack strategy overview: The attack strategy overview is presented in Figure 16. In this case, the attacker has access to the private key of the B1 block validator.

Conditions & outcomes: To make this strategy successful, the attacker needs to control one or more private keys. The outcome of this attack is similar to the simple variant of the same attack described in Section 6.8.

Enhancements: This attack does not provide any leverage for subsequent attacks.

Plausibility & prevention measures: This attack is not recorded in any blockchain system so far. A few mitigation techniques have been suggested like Frequent checkpoints [64], Key Evolving Cryptography [49], and Trusted Execution Environment [81].

6.10 Long Range Attack: Stake Bleeding

This strategy was proposed by Gaži *et al.* [64]. This attack can be launched against PoS consensus-based blockchain systems.

Motivation & vulnerability: The attacker wants to rewrite the history of transactions. In a blockchain system with no frequent checkpoints, absence of context-based transactions, and longest chain rule with transaction fees to be used as a reward mechanism, this attack can be launched. Block validators with any proportion of shares can launch this attack.

Attack strategy overview: The attack strategy overview is presented in Figure 10.

Conditions & outcomes: This attack has a significant relation with relative stakes. The more stakes the attacker possesses, the less time it requires to complete the attack. So, attackers may form a coalition to launch such an attack [64]. Additionally, a Stake Bleeding attack could be launched after the prior blockchain has been operational for several years. A successful attack corrupts the block history which can further lead to double-spending.

Enhancements: This attack does not provide any leverage for subsequent attacks.

Plausibility & prevention measures: A stake-bleeding attack is difficult to carry out as it would require several years of blockchain history. No such attack has been recorded so far. To effectively launch this attack, an attacker with a 30% stake would need approximately six years of blockchain history [64]. However, an Eclipse attack-based stake bleeding attack is proposed by Zhang *et al.* which significantly reduces the attack completion time [131]. To prevent this attack, Gaži *et al.* proposed two methods: Context Sensitive Transaction and Density Detect Mechanism [64].

6.11 P+ ϵ Attack

This attack was first introduced by Vitalik Buterin in [34]. Many cryptocurrency systems like SchellingCoin [33] operate on the assumption that every participant will act honestly because they

Table 2. Reward Matrix for P+ε Attack

	User Votes 0	User Votes 1
Other Nodes Vote 0	P	0
Other Nodes Vote 1	0	P

believe everyone else will do so. The attacker takes advantage of this assumption and manipulates the user to achieve any desired state.

Motivation & vulnerability: The attacker’s motivation is to take over the mechanism at zero cost. He does this by making a promise to reward people who voted in a certain way after the game is over. This reward is only paid if the majority voted differently from what they expected. Because the attacker is only paying out the reward if they win, it is always in the best interest of the voters to vote in the way that the attacker wants, regardless of what they believe the majority will do. The vulnerability of the system is that it relies on the assumption that people will act honestly in a simultaneous consensus game. The attack shows that this assumption is not always valid.

Attack strategy overview: Let us assume a system where each participant gets to vote whether to adopt a new consensus mechanism. A user gets a reward of P if he votes with the majority; 0 otherwise. The reward matrix is given in Table 2. In this case, each user has the incentive to vote honestly with the majority. Now, let us again assume that an attacker wants to change the consensus by manipulating other users. He announces to pay a little extra reward of ε to those who

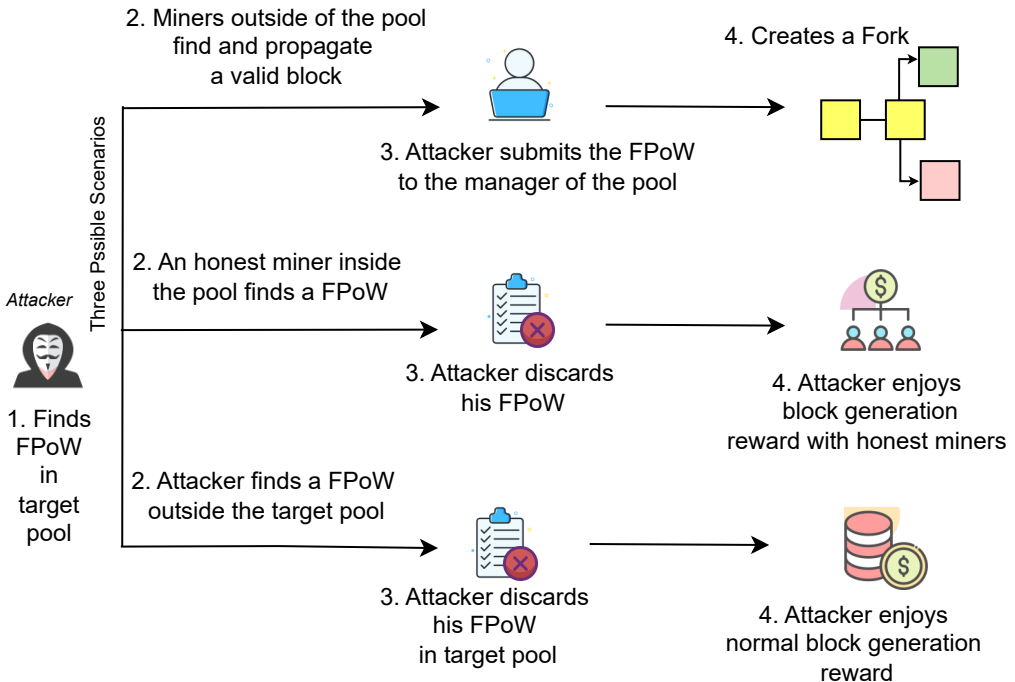


Fig. 17. Overview of Fork After Withholding Attack

Table 3. Modified Reward Matrix for $P+\epsilon$ Attack

	User Votes 0	User Votes 1
Other Nodes Vote 0	P	$P+\epsilon$
Other Nodes Vote 1	0	P

vote 1 in addition to P after the voting period, if the majority votes against adopting the consensus, that is, the majority votes 0. And if the majority goes with adopting a new consensus, that is, votes 1, nobody gets any extra reward. The reward matrix now will be similar to Table 3.

In this modified case, the users get manipulated and vote 1. But according to the attacker's contract, if the majority votes 1, nobody gets an extra reward. Ultimately, the attacker loses nothing and achieves the desired state of the system.

Conditions & outcomes: The attacker needs to broadcast the bribe. It may be done by a smart contract or by giving bribes. A demo Ethereum smart contract is available [3]. By launching this attack, the attacker achieves the desired state of a blockchain system without even paying anything.

Enhancements: This attack does not lead to any subsequent attacks.

Plausibility & prevention measures: This is a theoretical attack. Vitalik proposed two strategies [34] to mitigate such attacks. The first one is to require users to put down a deposit. Another one is to use counter-coordination strategies.

6.12 Feather Forking Attack

The feather forking attack is a more affordable version of the Punitive Forking Attack. It does not require a majority hashpower, similar to Punitive Forking. The first mention of Feather Fork and the strategy of the attack was made on the BitcoinTalk forum [93].

Motivation & vulnerability: Feather forking differs from punitive forking in that the majority hashpower is optional, however, the final goal remains the same. It can be used to force victims to pay huge amounts to validate blocks or have their transactions banned but the attacker must have mining pool authority [89, 93].

Attack strategy overview: The attack strategy overview is presented in Figure 18 [84, 89].

Conditions & outcomes: With owning some significant hashpower ($\leq 49\%$), and wagering on new block discovery time, the attacker can enforce significant transaction fee or if attacker discovers some blocks chronologically (value of K from Figure 18) in forked chain, then the victim's transaction is blacklisted [106].

Enhancements: This attack does not provide any leverage for subsequent attacks.

Plausibility & prevention measures: All miners in pools utilize reference client programs, therefore this attack would require the majority of the network to use rationally motivated clients, which is unlikely. With the "RationalMiner" client program, mitigation is still unproposed [24, 46, 93].

6.13 Bribery Attack

A bribery attack is a way to double-spend without owning the majority of hash power and renting the amount of hash power needed from rational miners. The Bribery Attack was first introduced by Joseph Bonneau [30] and first implementation was in "Smart Contracts for Bribing Miners" by Mccorry *et al.* [90].

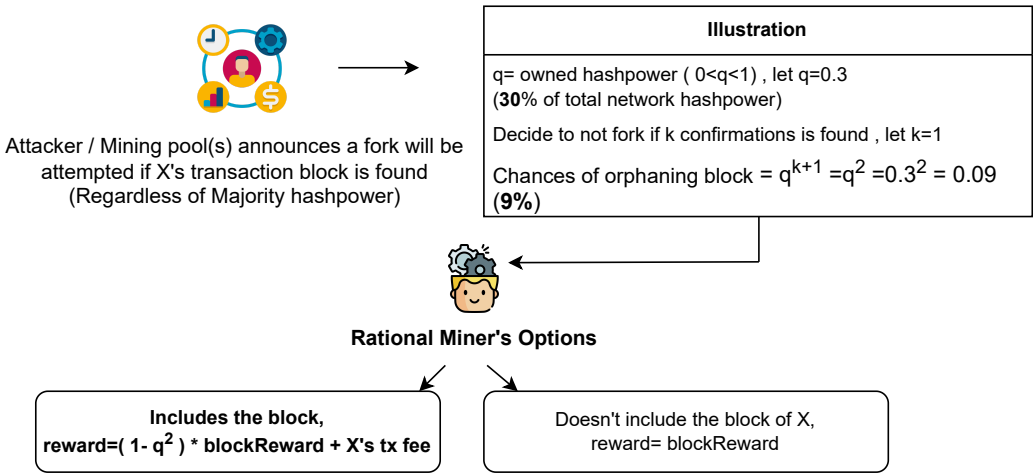


Fig. 18. Overview of Feather Forking Attack

Motivation & vulnerability: The motive is to acquire short lived mining power by leveraging rational miners (through bribing) with an aim to gain profit for both attacking individuals and bribed miners. The main chain may face a little less hashpower for a short time, however, it is dependent on the choices of rational miners. And in short bursts, it is hard to detect or mitigate. To incentivize, the attacker must keep rational miners engaged with bribed payments [30, 46].

Attack strategy overview: The attack strategy overview is presented in Figure 19.

Conditions & outcomes: The stakes of attacker are to convince miners for cooperation and join his double-spending cause and proposed methodologies to bribe miners are [30] :

- (1) Out-of-band payment/ Enforced third party mining arrangements
- (2) Running a negative-fee mining pool with cash burns to attract miners (if miner's don't stick, there's no established financial trust)
- (3) In-band payment via forking, create a soft fork to test if miners adopt this with funds available/ announced on the forked chain . A fork failure does not waste money, hence this is safer.

For a strong attack, it strongly encouraged to own around 15% hashrate. This allows participation in the block race after publishing the double-spending block [54], resulting in nearly 90% relative gains, as quantified by Sun *et al* [122].

Enhancements: This attack does not provide any leverage for subsequent attacks.

Plausibility & prevention measures: The attack is plausible with good strategies, but there is no precedent. After detection, victims / miners can counter-bribe, although it is not certain to work [30]. Attempting to bribe individual miners requires a large budget and tolerance. Other solutions include fund transfer limits and block limited transfers [54]. Another mitigation approach is to increase the amount of confirmation blocks [122].

6.14 Consensus delay attack

This attack, which causes temporary block and transaction delays, has been mentioned in several studies [66]. The latency of system consensus is the time it takes to reach consensus. This attack aims to postpone consensus and reveal the PoW consensus constraints on node synchronization and time limits [59]. This attack may occur before a full-scale attack.

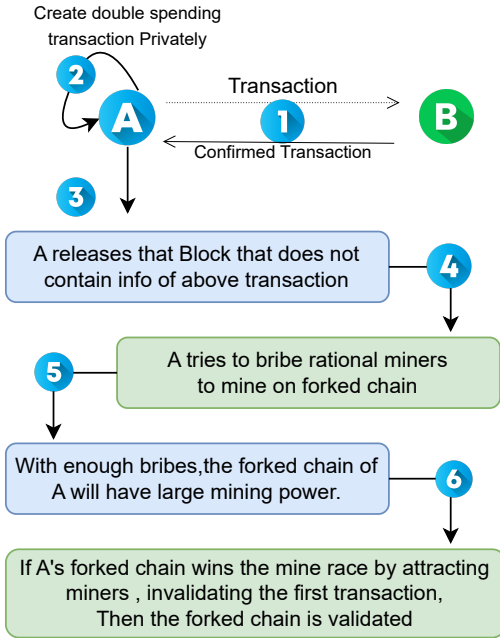


Fig. 19. Overview of Bribery Attack

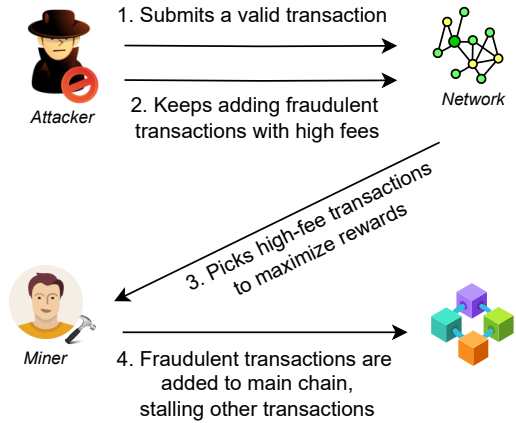


Fig. 20. Overview of Block Stuffing Attack

Motivation & vulnerability: The consensus attack exploits the mechanisms of block verifications and authenticity checks in order to exploit the time required for blockchains to achieve consensus. This, in turn, causes delays in targeted transactions by impeding the propagation of blocks. In Bitcoin, the PoW consensus system implemented by Nakamoto includes a timeout of 20 minutes for receiving updated information on newly found blocks (inv messages [19]). This timeout is being exploited in this context. In different consensus systems, there are many methods available to postpone the dissemination of information in order to achieve the same objective [40, 66].

Attack strategy overview: The block management overview (which is exploited in Bitcoin) is presented in Figure 21 [66].

While Bitcoin exploits the previous block management overview, some distributed systems and platforms, such as Zilliqa [11], Hyperledger Fabric [63], and Tendermint [8], utilize the Practical Byzantine Fault Tolerance (PBFT) consensus algorithm [40]. PBFT is employed in private and permissioned blockchains to achieve consensus, even in the presence of a small number of faulty nodes.

In PBFT, the primary node with other backup nodes executes a 4 steps algorithm to confirm a transaction.

During Prepare and Commit Phase (Figure 22) :

- Control a few replicas and delay introduction is possible [132].
- Send bogus signatures to other replicas.

Conditions & outcomes: First, the attacker must be a full-node, not SPV or other clients. A full node maintains the entire blockchain, independently verifying all transactions, while an SPV (Simplified Payment Verification) node relies on partial data and external sources for transaction

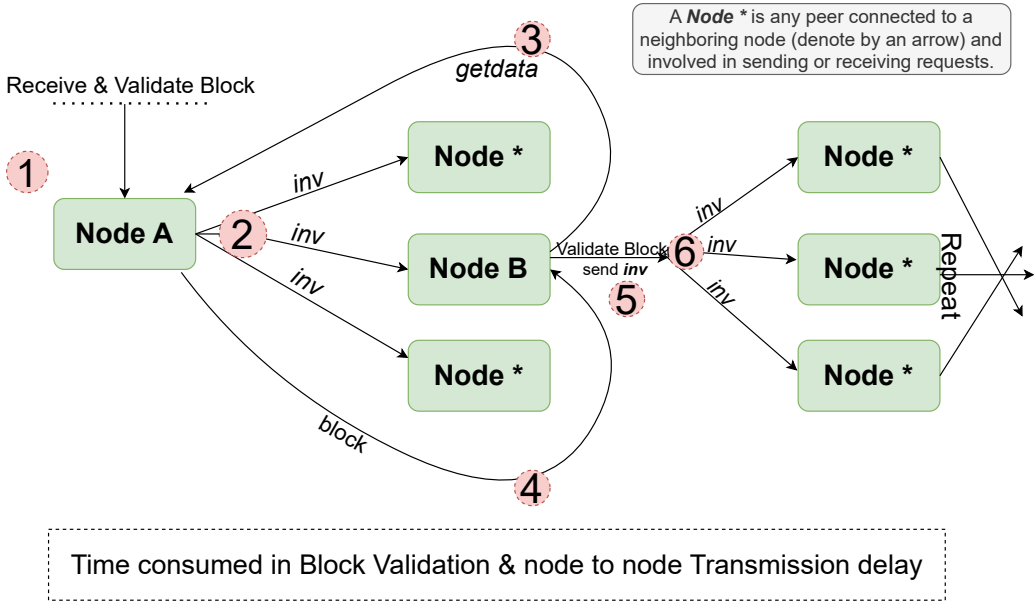


Fig. 21. Overview of Consensus Delay Attack

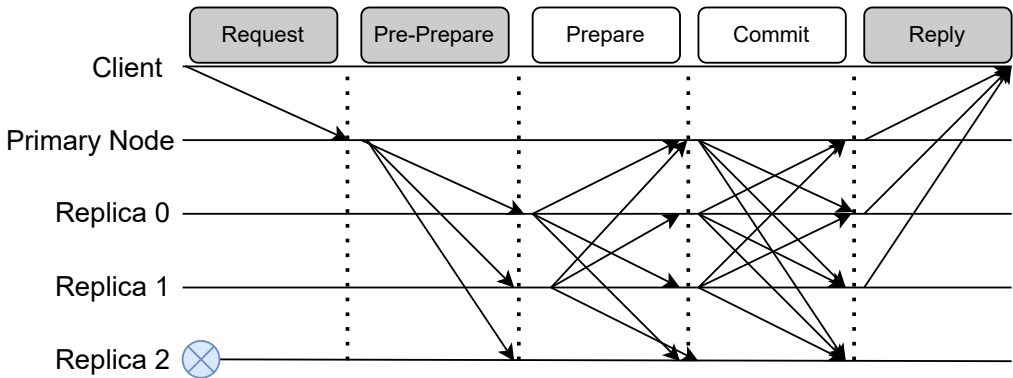


Fig. 22. PBFT Consensus Algorithm Process

validation [26]. Attackers often control numerous sybil nodes since delaying information to other nodes is difficult. Timing is critical for targeting node transaction delays. If the strategy works, genuine blocks will be wasted and honest miners would lose crucial time. With higher hashpower (33% advocated by Eyal *et al.* [61]), selfish mining attacks could occur simultaneously.

Enhancements: Successful attacks creates leverage for Sybil Attack, 51% Attack, DoS attack and Double-Spending [66].

Plausibility & prevention measures: Strategies make it viable, and suggested mitigation methods are additional relay network data, Change inv [19] messages and transaction advertisers and Non-responders penalty [66].

7 Application Layer Attacks

The application layer describes how we engage with the blockchain by reading its data and enables operations such as cryptocurrency creation, cryptocurrency distribution, smart contract deployment, and even incentive systems. Application layer attacks are especially directed at these functionalities. Unlike attacks that directly target the main blockchain, these attacks leverage flaws in the blockchain applications, potentially causing catastrophic repercussions.

7.1 Replay Attack

Replay attacks are common on blockchains. This exploit tries to spend transaction data from one chain on another legal split chain. Hard forks, which change or improve blockchain systems, facilitate the ability to replay attacks [48].

Motivation & vulnerability: After a hard fork on blockchain, the attacker targets to replay transactions from captured/sniffed/collected data. This attack in financial organizations can duplicate transactions and steal funds from unsuspecting accounts [74].

Attack strategy overview: The attack strategy overview is presented in Figure 23 [48].

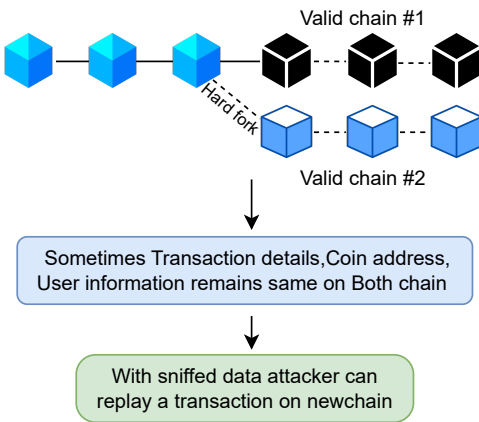


Fig. 23. Overview of Replay Attack

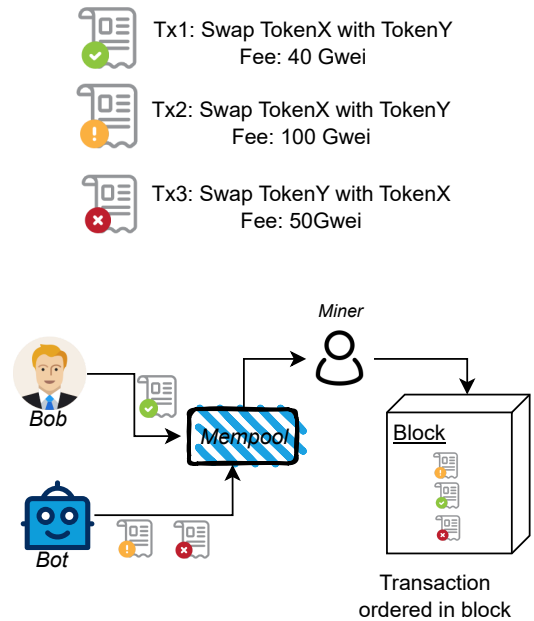


Fig. 24. Overview of Front Running Attack

Conditions & outcomes: The attack depends on the replay timing. The attacker must attempt the transaction on the new/updated chain with correct data after the hard fork and before replay protection on the upgraded chain. A successful execution could earn funds in this way [9].

Enhancements: This attack does not provide any leverage for subsequent attacks.

Plausibility & prevention measures: System improvements have caused hard forks, making the attack possible with timing. However, replay protection, time stamping, and unique transaction attributes have been shown to minimize this [9, 48].

7.2 Short-Address Attack

This is not a full-scale attack, but rather a representation of a minor vulnerability found in a third-party application that communicates with Solidity contracts [38]. [38].

Motivation & vulnerability: The bug's that existed in the Contract ABI Specification [4] allowed to send shorter parameters than expected, which allowed stealing tokens from exchanges. The main factor is address generations, which leads to cryptocurrency inflation [85].

Enhancements: This attack does not provide any leverage for subsequent attacks.

Plausibility & prevention measures: Once detected, the flaw was fixed, making the attack impossible in the present time [38].

8 Meta Application Layer Attacks

The Meta-application layer attacks in a blockchain system target the overlay that leverages the semantic interpretation of blockchain for a range of application domains. These attacks target the semantics of the blockchain system and the functionality beyond specific applications. The meta-application layer attacks exploit flaws inherent in decentralized applications. Decentralized applications or DApps are programs that run on a blockchain or p2p network. These apps allow direct user interactions without any intermediaries. They use smart contracts to automate transactions and processes, eliminating the need for intermediaries. OpenSea [16], Ethereum Name Service [15], Cryptokitties [14] etc. are some of the examples of such apps. Decentralized Exchanges or DEXs are examples of meta-applications. These are cryptocurrency trading platforms based on blockchain network. DEXs, unlike traditional exchanges, are not centralized, strengthening their resistance to manipulation and censorship. Users can trade directly with one another through smart contracts, which ensures transparency and security. Uniswap [10], Sushiswap [17], Bancor [12] etc. are some famous DEXs. The attacks on this layer can breach the integrity, security, and interoperability of blockchain-based services across various application domains.

8.0.1 Front Running Attack Front-running is a term associated with the share market. It refers to a scenario where a broker has confidential information about a large upcoming transaction that will significantly influence the price of the associated shares. Consequently, the attacker buys the share before it becomes publicly available. This is considered ill-practice [20]. Eskandari *et al.* organized a front-running attack associated with the blockchain-based system in four different categories: decentralized exchanges, crypto-collectible games, naming services and gambling apps [56].

Motivation & vulnerability: Front-running attacks in blockchain are motivated by financial gain. With knowledge of impending transactions, attackers influence the system to secure profit from price manipulation, arbitrage and fee extraction. DApps are vulnerable to such attacks. Most blockchains have public mempools with visible pending transactions before confirmation. This transparency lets attackers learn about upcoming events. Also, faulty smart contracts can cause such an attack [98]. Maximum Extractable Value (MEV) bots play a vital role in front-running attacks. These bots are operated by validators or independent actors. They strategically reshuffle, include, or remove mempool transactions to maximize value [52].

Attack strategy overview: Many cryptocurrency exchanges have bots. They constantly search the mempool for transactions and purchases as necessary. Let us assume a case where an honest user, Bob, wants to swap a significant amount of TokenX with TokenY. The demand-supply theory says this transaction will raise TokenY's price. A malicious bot observes this and broadcasts a similar transaction, but a slightly smaller amount of token. In this transaction, the bot attaches a slightly more gas fee. Another transaction by the bot swaps TokenY for TokenX, reducing its price.

This time, the bot attaches a slightly lower gas fee. Miners will arrange these transactions in the ascending order of gas fee. So, Bob will ultimately gain less than expected.

An overview of the attack strategy is presented in Figure 24.

Conditions & outcomes: The attacker needs a total knowledge of market and transaction history. This type of attack disrupts the market or provides an unfair competition advantage. Also, such attacks may lead to a double-spending scenario [56].

Enhancements: This attack does not provide any leverage for subsequent attacks.

Plausibility & prevention measures: This attack is a very common issue in Ethereum DApps or in any DEX. DODO DEX suffered \$3.8 million loss [104] and bZx DEX suffered a massive loss of \$350,000 in such an attack [116]. Several mitigation techniques such as transaction sequencing, confidentiality, and improved design practices [56] have been proposed to prevent such attacks.

8.1 Block Stuffing Attack

This is a Denial-of-Service (DoS) attack where the attacker repeatedly sends fake transactions to the network, thus slowing future transactions from being added to the chain. We study this attack in light of a real-life event. FOMO3D [6] is a famous gambling game that can be played via an Ethereum smart contract. In this game, players buy keys from a smart contract and deposit money in a pot. Each round begins with a 24-hour time counter. A key purchasing event adds 30 seconds to the counter. When the counter strikes 0, the last person to buy a key wins most of the pot and the rest is shared among the players.

Motivation & vulnerability: The main motivation of such an attack is to stall the network. In case of FOMO3D, the attacker took advantage of the block gas limit and launched this attack. Each block has a certain amount of gas limit. Miners include high-fee transactions to increase profit. Thus, they selected a set of transactions that maximizes profit per block. An attacker's main purpose is to manipulate the selection process by creating a set of transactions with precise calculations that have the best likelihood of being mined to deplete blocks' gas limits and block other transactions from being added to the chain.

Attack strategy overview: The attack strategy overview is presented in Figure 20.

Conditions & outcomes: Each transaction incurs a gas price that the attacker must pay. If the attack fails, he will suffer financial losses. The outcome is context-dependent. In case of FOMO3D, the attacker won 10,469 ETH in the first round and 3264.668 ETH in the second round which is equivalent to 24,978,405.86\$ and 7,768,440.74\$ respectively as of February 2024.

Enhancements: This attack creates no leverage for another attack.

Plausibility & prevention measures: The Ethereum network experienced multiple instances of this attack in 2018 [100, 111]. For developers of smart contracts, several preventive steps have been suggested to avoid smart contract issues and transaction-blocking events at the network level for games and applications of a similar nature like FOMO3D [111].

9 Discussion

In this section, we provide a summary of our findings from the analysis of a number of attacks (Section 9.1) and a comparative analysis of our analysis with similar existing research works (Section 9.2).

9.1 Summary of Layer-based Attack Analysis

In this article, we have explored several attacks related to blockchain-based systems. We also organize them using a layer-based modeling technique. The inter-relation of these attacks is presented in Figure 25

In Figure 25, we have placed our studied attacks on the vertical axis (denoted circles) with the corresponding layers on the horizontal axis. If an attack impacts multiple layers, we have pointed them out in the figure as explained by the legend in Figure 25. For an attack spanning multiple layers, the primary/targeted layer is denoted with a colorful circle. Different colorful circles have different semantics. These semantics have been added as legends in Figure Figure 25. Each arrow connecting the circles suggests a potential scenario of attack enhancement. The starting of the arrow denotes the attack from which the leverage is obtained, while the point of the arrow shows where this leverage can be applied to initiate a new attack.

From our analysis, it is evident that:

- The network layer is the initial vulnerable layer on the blockchain attack surface.
- Most attacks aim to exploit the consensus layer to cause significant damage.
- Some of the attacks (i.e.- BGP Hijacking, Sybil, Eclipse) are very lucrative for attackers as they provide unparalleled advantage for further attacks by enhancement. And all of them targets the network layer primarily.
- Moreover, few attacks like Consensus Delay, Race & Finney are most vulnerable by enhancement from other attacks.

We also present a summary of the affected layers in Table 4. Here, we describe the properties of each layer that is exploited in order to make an attack successful for that particular layer. We have the following findings.

- The consensus layer has the most exploitable properties as almost all of our studied attacks take advantage of protocols or rules of this layer.
- The network layer is also very crucial because many attacks have to subvert or circumvent this layer to affect the next layer.
- The application and meta-application layers are crucial for blockchain security, as high-level development involving various API gateways and architectures is prone to vulnerabilities. Bugs in these layers can cause significant financial losses and erode trust in the blockchain. Ensuring the security of these layers is imperative, as they are the primary interface for users in decentralized applications (DApps) and similar platforms, where critical issues can arise.

Table 5 provides a summary of our findings on individual attacks based on the taxonomy outlined in Section 4.2. In this table, symbols such as ‘ \emptyset ’ and ‘?’ are used to indicate a lack of information or insufficient data for that particular property. On the other hand, symbols such as ‘●’, ‘○’ and ‘◐’ are used to indicate the presence, absence and partially presence with respect to a particular property. For other properties, descriptive texts have been provided for clarification.

Our findings from Table 5 are the following.

- Of all the attacks, the balance attack is the only attack without any mitigation proposal & can be executed with the lowest mining power.
- Except for the Balance attack, all other studied attacks have at least one mitigation proposal. For Replay & Short-address the solution has already implemented.
- For Eclipse & Sybil attacks, the attack surface is complex, and prevention methods are continually being developed. Some of the proposed methods are implemented.
- Most of the attacks require the motivation of a single adversary miner. In some cases, agents like Sybil nodes help to facilitate the attack faster.

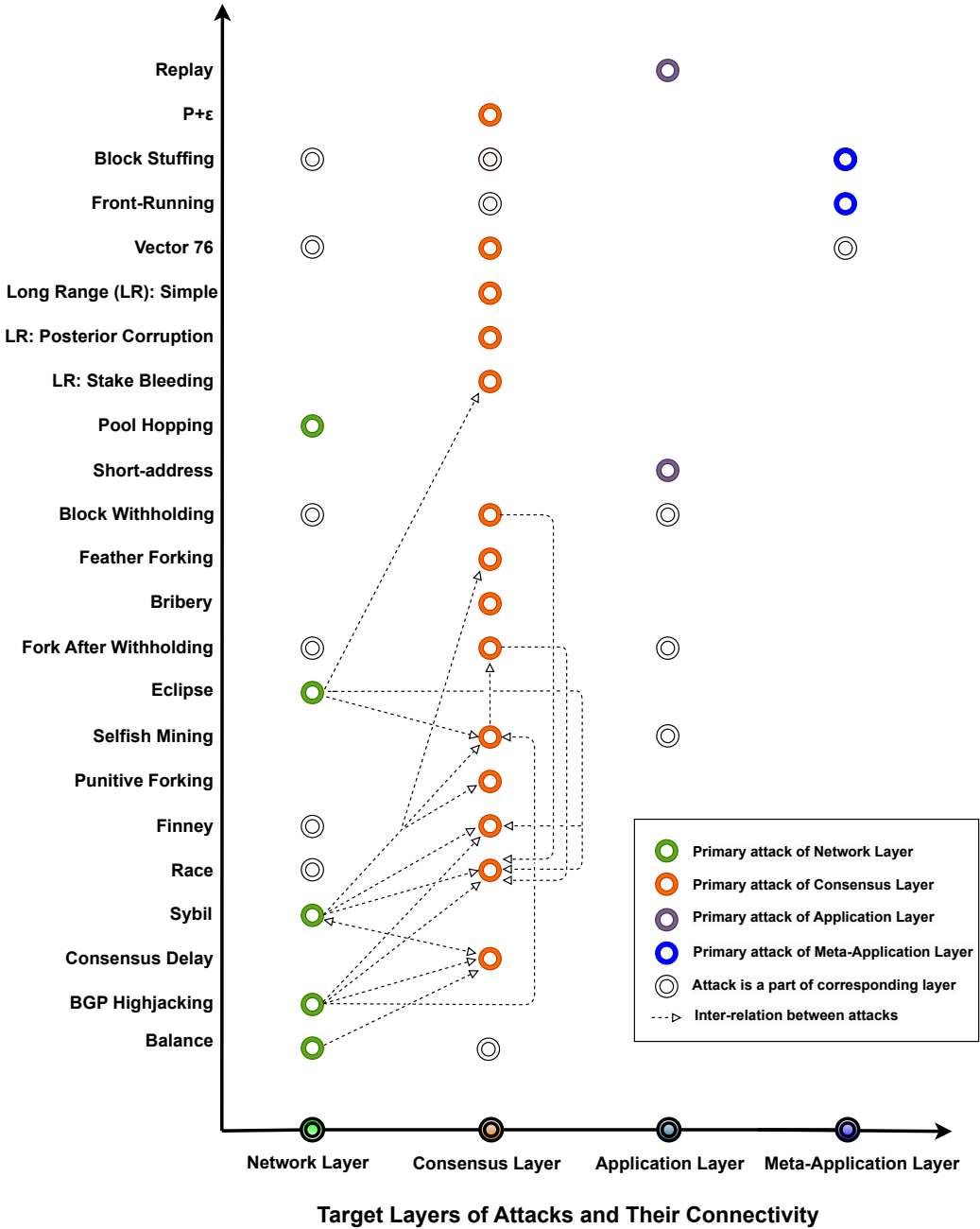


Fig. 25. Distribution of studied attacks in different layers

Table 4. Layer-based attack categorization and exploiting points for each layer

Attack Name	Network Layer	Consensus Layer	Application Layer	Meta-Application Layer
Balance Attack	Leveraging knowledge of miner's network structures, computational power, and mining difficulty to introduce communication delays.	Leverages Ethereum's GHOST Protocol / Bitcoin's longest chain rule.		
BGP Hijacking	Manipulating internet routing table			
Bribery Attack		Manipulating miners through bribing		
Consensus Delay		Exploits the mechanisms of block verifications and authenticity checks		
Eclipse Attack	P2P Protocol limitations			
Sybil Attack	Capture new nodes (assisted by multiple devices, VM, IP)			
Feather Forking		Exploit by forks (using Majority hashpower)		
Punitive Forking		Exploits by forks (without Majority hashpower)		
Replay Attack			Absence of replay protection on the upgraded chain	
Short-Address			Third Party API Exploit	
Selfish Mining		PoW mechanism exploit by hiding blocks	Reward mechanism exploit	
Block Withholding	Mining Pool Infiltration	PoW mechanism exploit by hiding blocks or avoiding block submission	Reward mechanism exploit	

Attack Name	Network Layer	Consensus Layer	Application Layer	Meta-Application Layer
Fork After Withholding	Mining Pool Infiltration	PoW mechanism exploit by hiding blocks	Reward mechanism exploit	
Block Stuffing	Denial of Service	PoW mechanism exploit by using higher trx fee		FOMO3D Application exploit
Finney Attack	Block propagation manipulation	PoW mechanism exploit by replacing 0-confirmation transaction		
Race Attack	Block propagation manipulation	PoW mechanism exploit by replacing 0-confirmation transaction		
Vector 76	P2P network joining manipulation	PoW mechanism exploit by replacing 1-confirmation transaction	Crypto Exchange application exploit	
Long Range: Simple		PoS mechanism exploit by rewriting block history		
Long Range: Posterior Corruption		PoS mechanism exploit by rewriting block history		
Long Range: Stake Bleeding		PoS mechanism exploit by rewriting block history		
P+Epsilon		Consensus mechanism exploit through bribing user		
Front Running		Consensus mechanism through transaction reordering		DNS, Gambling, DeX etc. application exploit
Pool Hopping	Timely join & exit mining pools based on profitability and anticipated behavior			

Table 5. Taxonomical properties of studied attacks

Attack Name	Initiator	Mining Power Requirement (Minimum / Advised)	Initial Breach Layer	Mitigation Techniques	Enhancements	Reference
Balance Attack	Single Miner	5%	Network	?	●	[97, 110]
BGP Hijacking	Single Miner	∅	Network	●	●	[27, 87, 107, 113]
Bribery Attack	Single Miner	15%	Consensus	●	○	[30, 54, 122]
Consensus Delay	Single (Agents)	∅	Consensus	●	●	[19, 61, 66]
Eclipse Attack	Single (Agents), Groups	∅	Network	●	●	[50, 70, 71, 78, 86, 131]
Sybil Attack	Single (Agents)	∅	Network	●	●	[21, 26, 53, 73, 123]
Feather Forking	Pools	<51%	Consensus	●	○	[24, 46, 89, 93, 106]
Punitive Forking	Pools	51%	Consensus	●	○	[29, 46, 75]
Replay Attack	Single Miner	∅	Application	●	○	[9, 48]
Short-Address	Single Miner	∅	Application	●	○	[38, 85]
Selfish Mining	Single Miner	9%	Consensus	●	●	[80, 108, 119]
Block Withholding	Single Miner, Groups, Pools	∅	Network	●	●	[47, 58, 105, 128]
Fork After Withholding	Single Miner	∅	Network	●	●	[60, 80, 105]
Block Stuffing	Single (User)	∅	Consensus	●	○	[100, 111]
Finney Attack	Single Miner	∅	Consensus	●	○	[109]
Race Attack	Single (User)	∅	Network	●	○	[28, 64, 76]
Vector 76	Single Miner	∅	Application	●	○	[125]
Long Range :Simple	Single (Validator)	∅	Consensus	●	○	[32, 51]
Long Range :Posterior Corruption	Single (Validator)	∅	Application	●	○	[49, 64, 81]
Long Range :Stake Bleeding	Single (Validator)	∅	Consensus	●	○	[64, 131]
P+Epsilon	Single (User)	∅	Consensus	●	○	[3, 34]

Attack Name	Initiator	Mining Power Requirement Minimum / Advised)	Initial Breach Layer	Mitigation Techniques	Enhancements	Reference
Front Running	Single (User)	∅	Meta - Application	●	○	[52, 56]
Pool Hopping	Single Miner	∅	Network	●	○	[39, 105, 115, 117]

9.2 Comparative Analysis

In Table 6 we compare this work with previous survey of attacks on blockchain systems based on distinct properties such as number of attacks covered, number of layers, inter-relation of attacks between layers, mitigation techniques, attacker’s perspective analysis and layer vulnerability analysis. In this table, the symbols ‘●’, ‘○’ and ‘◐’ have been used to indicate the presence, absence and partially presence of the corresponding property respectively. This table distinguishes our work from previous studies in the following manner:

- Our attack categorization consists of four layers, closely aligning with [72]. However, in [72], Homoliak *et al.* employs a Replication State Machine layer, we instead utilize an Application layer and a Meta-Application layer. As a result, the categorization in this work is more refined compared to the four-layer model of [72] and more concise than the six-layer model presented in [127]. Furthermore, our layer categorization is platform-agnostic, unlike [42], which is specifically tailored for Ethereum. Therefore, in terms of layer vulnerability analysis, we also differ from [72] and [127]. This analysis is missing in [109], [67], [82], [130] and [127].
- We present a detailed examination of the inter-relationship of attacks between layers in terms of our custom framework presented in Section 4.2. For example, how an attack might lead to subsequent attacks, how attacks can affect the components of several levels at the same time, and so on. This detailed examination is not present in previous works.
- We discuss mitigation techniques for each attack we studied which are not present in [67] and partially discussed in [81], [43], [42], [94]. While [25] points out some mitigation techniques, our work covers a broader range of attacks. Our discussion closely aligns with the approaches found in [72], [109], and [127].
- We provide an in-depth analysis from the attacker’s perspective, detailing which vulnerabilities in specific layers are exploited, the motivations behind these attacks, and the associated stakes of launching an attack. This analysis is missing in most of the aforementioned previous works and only briefly discussed in [127].

10 Conclusion

In recent years, blockchain has seen a dramatic increase in popularity, driven by the rise of decentralized systems and its widespread applications across various fields like cryptocurrencies, banking sectors, crypto-assets, IOT and health services. Popularity comes with a considerable cost of security threat & attacks on established systems using blockchain. Over the years, the losses have been mounting [112]. These exposed vulnerabilities weaken the legitimacy of the blockchain as a decentralized system and demands to be addressed & studied thoroughly. In our study, to evaluate the scenarios of security threats & impacts, we used the four layers: network, consensus, application, and meta-application. For every studied attack, we discussed: i) attacker’s motivation & vulnerabilities that might lead to an attack, ii) the attack strategy for each attack is represented

visually, iii) the conditions for the attack and the possible attack outcome, iv) the possibility of leading to another attack and v) the practicality of the attack & possible countermeasures. Through detailed analysis, it is evident that attacks can transcend layers, with vulnerabilities in one layer potentially leading to exploits in another. Furthermore, our findings are presented in tabular formats and demonstrate that a single attack can span multiple layers, depending on the targeted attributes of each layer, thereby amplifying the complexity of securing blockchain systems.

In conclusion, we believe that no system is completely secure. Therefore, additional research is required to develop a more secure and functional architecture for blockchain-based systems. This survey aims to provide valuable insight into different aspects of the examined attacks, as well as to highlight the intricate connectivity across layers under various attack scenarios. This survey can serve as an effective guide to limit adversarial activity across blockchain layers, leading to intriguing outcomes based on future research in this field.

Table 6. Comparison with previous works

Reference	Number of Attacks Covered	Number of Layers	Inter-relation of Attacks Between layers	Mitigation Techniques	Attacker's Perspective Analysis	Layer Vulnerability Analysis	Remarks
Homoliak <i>et al.</i> [72]	≈ 21	4	○	●	○	●	<ol style="list-style-type: none"> 1. Focused on SRA of blockchain systems. 2. Used 4 layers to categorize security threats and vulnerabilities.
Saad <i>et al.</i> [109]	22	0	○	●	○	○	<ol style="list-style-type: none"> 1. Emphasize on blockchain attack surface. 2. Outlined effective defense measures. 3. Explored cryptographic constructions, distributed system architecture, and application of blockchain.
Guggenberger <i>et al.</i> [67]	87	0	○	○	○	○	<ol style="list-style-type: none"> 1. Structured attacks using AT notation.
Li <i>et al.</i> [82]	6	0	○	●	○	○	<ol style="list-style-type: none"> 1. Discussed security vulnerabilities related to blockchain. 2. Reviewed security enhancement solutions for blockchain.
Chen <i>et al.</i> [42]	26	4	○	●	○	●	<ol style="list-style-type: none"> 1. Focused on Ethereum platform security. 2. Considered three perspectives- vulnerabilities, attacks and defense.
Wen <i>et al.</i> [127]	17	6	○	●	●	○	<ol style="list-style-type: none"> 1. Focused on attacks on blockchain system and their countermeasures.
Moubarak <i>et al.</i> [94]	7	0	○	●	○	○	<ol style="list-style-type: none"> 1. Evaluated blockchain security. 2. Identified flaws that may lead to attacks.

Reference	Number of Attacks Covered	Number of Layers	Inter-relation of Attacks Between layers	Mitigation Techniques	Attacker's Perspective Analysis	Layer Vulnerability Analysis	Remarks
Anita <i>et al.</i> [25]	17	0	○	●	○	○	1. 1. Presented taxonomy of security threats related to blockchain systems.
Chen <i>et al.</i> [43]	11	0	○	●	○	○	1. Reviewed the attack and defense methods of the blockchain. 2. Categorized attacks into 3 different groups.
Zhang <i>et al.</i> [130]	4	6	○	○	○	○	1. Concentrated on the security and privacy aspects and mechanisms of blockchain. 2. Compared different types of consensus mechanism.
This work	23	4	●	●	●	●	1. Focused on Layer-based attack analysis. 2. Covered attacker's perspective analysis in detail. 3. Covered layer vulnerability analysis. 4. Covered mitigation techniques.

References

- [1] [n. d.]. Bitcoin. <https://bitcoin.org/en/> Accessed: 10-09-2024.
- [2] [n. d.]. Bitcoin FAQ. <https://en.bitcoin.it/wiki/Help:FAQ#> [Online; accessed 14-September-2023].
- [3] [n. d.]. Contract 0x3607608A1907Acc2042eb83195ffe733b04F0ED4. <https://etherscan.io/address/0x3607608a1907acc2042eb83195ffe733b04f0ed4#code>. [Online; accessed 14-September-2022].
- [4] [n. d.]. Contract ABI Specification. <https://docs.soliditylang.org/en/latest/abi-spec.html>. [Online; accessed 16-September-2022].
- [5] [n. d.]. Corda - R3 — r3digitalmarkets.com. <https://r3digitalmarkets.com/corda/> Accessed: 10-09-2024.
- [6] [n. d.]. FOMO3D. <https://cryptoslate.com/products/fomo3d>. Accessed: 2024-01-30.
- [7] [n. d.]. Home | ethereum.org — ethereum.org. <https://ethereum.org/en/>. [Accessed 10-09-2024].
- [8] [n. d.]. Tendermint — tendermint.com. <https://tendermint.com/> Accessed: 10-09-2024.
- [9] [n. d.]. Transaction Replay and Replay Protection With Hard Forks Explained. ([n. d.]). <https://cryptocurrencyfacts.com/transaction-replay-and-replay-protection-with-hard-forks-explained/> Accessed on 12.09.2022.
- [10] [n. d.]. UNISWAP PROTOCOL Swap, earn, and build on the leading decentralized crypto trading protocol. <https://uniswap.org> Accessed: 2024-09-09.
- [11] [n. d.]. Zilliqa. <https://www.zilliqa.com/> Accessed: 10-09-2024.
- [12] 2017. Bancor. <https://bancor.network> Accessed: 2024-09-09.
- [13] 2017. Bitcoin Cash: Peer-to-Peer Electronic Cash. <https://www.bitcoincash.org/>
- [14] 2017. CryptoKitties: Collect and Breed Furrever Friends. <https://www.cryptokitties.co> Accessed: 2024-09-09.
- [15] 2017. Ethereum Name Service. <https://ens.domains> Accessed: 2024-09-09.
- [16] 2017. OpenSea, the largest NFT Marketplace. <https://opensea.io> Accessed: 2024-09-09.
- [17] 2020. SushiSwap | Sushi. <https://sushi.com> Accessed: 2024-09-09.
- [18] 2022. Blockchain Basics : Proof-of-work. (March 2022). <https://opentozos.com/blockchain-basics/proof-of-work/> Accessed on 22.08.2022.
- [19] 2022. P2P Network-Bitcoin Developer. (2022). https://developer.bitcoin.org/reference/p2p_networking.html Accessed on 09.09.2022.
- [20] Cory Mitchell. [n. d.]. Front-Running. <https://www.investopedia.com/terms/f/frontrunning.asp>. [Online; accessed 17-September-2022].
- [21] Binance Academy. 2020. Sybil attacks explained. <https://www.binance.com/en-ZA/feed/post/43183> Accessed on 20.12.2023.
- [22] AllianceBlock. [n. d.]. AllianceBlock Issues Statement in Response to BonqDAO Hack. <https://medium.com/allianceblock/allianceblock-issues-statement-in-response-to-bonqdao-hack-6510a61fcf5c>. [Online; accessed 30-March-2024].
- [23] Crystal Blockchain's Analytics. 2022. Crypto & DeFi Security Breaches, Fraud & Scams Report. (2022). <https://crystalblockchain.com/security-breaches-and-fraud-involving-crypto/> Accessed on 11.08.2022.
- [24] Rajanikanta sahu Anil Kumar Mishra1. 2017. Bitcoin Approach: Its Challenges and Attacks. IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) (October 2017). <https://www.iosrjournals.org/iosr-jeee/Papers/Vol12%20Issue%205/Version-2/K1205028793.pdf> Accessed on 12.09.2022.
- [25] N Anita and M Vijayalakshmi. 2019. Blockchain security attack: A brief survey. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 1–6.
- [26] Andreas M. Antonopoulos. 2014. Mastering Bitcoin: Unlocking Digital Crypto-Currencies (1st ed.). O'Reilly Media, Inc., Chapter 6.
- [27] Maria Apostolaki, Aviv Zohar, and Laurent Vanbever. 2017. Hijacking bitcoin: Routing attacks on cryptocurrencies. In 2017 IEEE symposium on security and privacy (SP). IEEE, 375–392.
- [28] Tobias Bamert, Christian Decker, Lennart Elsen, Roger Wattenhofer, and Samuel Welten. 2013. Have a snack, pay with Bitcoins. In IEEE P2P 2013 Proceedings. IEEE, 1–5.
- [29] Raynor de Best. 2024. Biggest bitcoin mining pools 2024. <https://www.statista.com/statistics/731416/market-share-of-mining-pools/>
- [30] Joseph Bonneau. 2016. Why buy when you can rent?. In International Conference on Financial Cryptography and Data Security. Springer, 19–26.
- [31] Ryan Browne. 2022. Hacked crypto startup Nomad offers a 10% bounty for return of funds after \$190 million attack. (August 2022). <https://www.cnn.com/2022/08/05/crypto-startup-nomad-offers-10percent-bounty-after-190-million-hack.html> Accessed on 12.08.2022.
- [32] Vitalik Buterin. 2014. Long-Range Attacks: The Serious Problem With Adaptive Proof of Work. (2014). <https://blog.ethereum.org/2014/05/15/long-range-attacks-the-serious-problem-with-adaptive-proof-of-work> Accessed on 15.01.2024.

- [33] Vitalik Buterin. 2014. SchellingCoin: A Minimal-Trust Universal Data Feed. <https://blog.ethereum.org/2014/03/28/schellingcoin-a-minimal-trust-universal-data-feed>. Accessed: 2024-02-23.
- [34] Vitalik Buterin. 2015. The P + epsilon Attack. <https://blog.ethereum.org/2015/01/28/p-epsilon-attack>. Accessed: 2024-02-23.
- [35] Vitalik Buterin et al. [n. d.]. Ethereum white paper. ([n. d.]). <https://ethereum.org/en/whitepaper/> Accessed on 14.08.2022.
- [36] Vitalik Buterin and Virgil Griffith. 2017. Casper the friendly finality gadget. *arXiv preprint arXiv:1710.09437* (2017).
- [37] Vitalik Buterin, Diego Hernandez, Thor Kampefner, Khiem Pham, Zhi Qiao, Danny Ryan, Juhyeok Sin, Ying Wang, and Yan X Zhang. 2020. Combining GHOST and casper. *arXiv preprint arXiv:2003.03052* (2020).
- [38] Pawel Bylica. 2017. How to Find \$10M Just by Reading the Blockchain. (2017). <https://medium.com/golem-project/how-to-find-10m-by-just-reading-blockchain-6ae9d39fcd95> Accessed on 12.09.2022.
- [39] c00w. 2011. bitHopper: Python Pool Hopper Proxy. (2011). <https://bitcointalk.org/?topic=26866> Accessed on 12.09.2022.
- [40] Miguel Castro, Barbara Liskov, et al. 1999. Practical byzantine fault tolerance. In *OsDI*, Vol. 99. 173–186.
- [41] Panagiotis Chatzigiannis, Foteini Baldimtsi, Igor Griva, and Jiasun Li. 2022. Diversification across mining pools: optimal mining strategies under PoW. *Journal of Cybersecurity* 8, 1 (03 2022). <https://doi.org/10.1093/cybsec/tyab027>.
- [42] Huashan Chen, Marcus Pendleton, Laurent Njilla, and Shouhuai Xu. 2020. A survey on ethereum systems security: Vulnerabilities, attacks, and defenses. *ACM Computing Surveys (CSUR)* 53, 3 (2020), 1–43.
- [43] Yourong Chen, Hao Chen, Yang Zhang, Meng Han, Madhuri Siddula, and Zhipeng Cai. 2022. A survey on blockchain systems: Attacks, defenses, and privacy preservation. *High-Confidence Computing* 2, 2 (2022), 100048.
- [44] Mohammad Javed Morshed Chowdhury, Md Sadek Ferdous, Kamanashis Biswas, Niaz Chowdhury, ASM Kayes, M Alazab, and Paul Watters. 2019. A Comparative Analysis of Distributed Ledger Technology Platforms. *IEEE Access* 7, 1 (2019), 167930–167943.
- [45] Nxt Community. 2013. Nxt Whitepaper. <https://nxtwiki.org/wiki/Whitepaper:Nxt>. Accessed: 2024-09-02.
- [46] Mauro Conti, E Sandeep Kumar, Chhagan Lal, and Sushmita Ruj. 2018. A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys & Tutorials* 20, 4 (2018), 3416–3452.
- [47] Nicolas T Courtois and Lear Bahack. 2014. On subversive miner strategies and block withholding attack in bitcoin digital currency. *arXiv preprint arXiv:1402.1718* (2014).
- [48] Mr. Creatonics. 2019. What are replay attacks in Blockchain & how To Prepare For Replay Attacks? (September 2019). <https://coinsutra.com/what-are-replay-attacks/> Accessed on 12.09.2022.
- [49] Bernardo David, Peter Gaži, Aggelos Kiayias, and Alexander Russell. 2018. Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In *Advances in Cryptology—EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29–May 3, 2018 Proceedings, Part II 37*. Springer, 66–98.
- [50] Marcel Deer. 2021. What is an eclipse attack? <https://cointelegraph.com/explained/what-is-an-eclipse-attack> (12 2021). Accessed on 14.09.2022.
- [51] Evangelos Deirmentzoglou, Georgios Papakyriakopoulos, and Constantinos Patsakis. 2019. A survey on long-range attacks for proof of stake protocols. *IEEE Access* 7 (2019), 28712–28725.
- [52] Diana Ambolis. [n. d.]. A Comprehensive Analysis Of Front-Running Attacks In Blockchain. <https://blockchainmagazine.net/a-comprehensive-analysis-of-front-running-attacks-in-blockchain/>. [Online; accessed 14-February-2024].
- [53] John R. Douceur. 2002. The Sybil Attack. In *Peer-to-Peer Systems*, Peter Druschel, Frans Kaashoek, and Antony Rowstron (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 251–260.
- [54] Ghader Ebrahimpour and Mohammad Sayad Haghghi. 2021. Analysis of bitcoin vulnerability to bribery attacks launched through large transactions. *arXiv preprint arXiv:2105.07501* (2021).
- [55] Johnson Lau Eric Lombrozo and Pieter Wuille. 2017. Segregated Witness (Consensus layer). <https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki>
- [56] Shayan Eskandari, Seyedehmahsa Moosavi, and Jeremy Clark. 2019. Sok: Transparent dishonesty: front-running attacks on blockchain. In *International Conference on Financial Cryptography and Data Security*. Springer, 170–189.
- [57] Ethereum. [n. d.]. Proof-of-stake (POS) | Ethereum.org. <https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/>
- [58] Ittay Eyal. 2015. The miner’s dilemma. In *2015 IEEE symposium on security and privacy*. IEEE, 89–103.
- [59] Ittay Eyal, Adem Efe Gencer, Emin Gun Sirer, and Robbert Van Renesse. 2016. Bitcoin-NG: A Scalable Blockchain Protocol. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. USENIX Association, Santa Clara, CA, 45–59. <https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/eyal>

- [60] Ittay Eyal and Emin Gün Sirer. 2014. How to Disincentivize Large Bitcoin Mining Pools. (2014). <https://hackingdistributed.com/2014/06/18/how-to-disincentivize-large-bitcoin-mining-pools/> Accessed on 06.03.2023.
- [61] Ittay Eyal and Emin Gün Sirer. 2014. Majority is not enough: Bitcoin mining is vulnerable. In International conference on financial cryptography and data security. Springer, 436–454.
- [62] Md Sadek Ferdous, Mohammad Javed Morshed Chowdhury, Mohammad A Hoque, and Alan Colman. 2020. Blockchain consensus algorithms: A survey. arXiv preprint arXiv:2001.07091 (2020).
- [63] Hyperledger Foundation. [n. d.]. Hyperledger Fabric – hyperledger.org. <https://www.hyperledger.org/projects/fabric> Accessed: 10-09-2024.
- [64] Peter Gaži, Aggelos Kiayias, and Alexander Russell. 2018. Stake-bleeding attacks on proof-of-stake blockchains. In 2018 Crypto Valley conference on Blockchain technology (CVCBT). IEEE, 85–92.
- [65] Rebecca Moody George Moody. 2022. Worldwide cryptocurrency heists tracker (updated daily). (2022). <https://www.comparitech.com/crypto/biggest-cryptocurrency-heists/> Accessed on 12.08.2022.
- [66] Arthur Gervais, Hubert Ritzdorf, Ghassan O. Karame, and Srdjan Capkun. 2015. Tampering with the Delivery of Blocks and Transactions in Bitcoin. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (Denver, Colorado, USA) (CCS '15). Association for Computing Machinery, New York, NY, USA, 692–705. <https://doi.org/10.1145/2810103.2813655>
- [67] Tobias Guggenberger, Vincent Schlatt, Jonathan Schmid, and Nils Urbach. 2021. A Structured Overview of Attacks on Blockchain Systems. PACIS (2021), 100.
- [68] Stuart Haber and W Scott Stornetta. 1991. How to time-stamp a digital document. Springer.
- [69] Hal. 2011. Best practice for fast transaction acceptance - how high is the risk? <https://bitcointalk.org/index.php?topic=3441.msg48384#msg48384>. [Online; accessed 10-September-2022].
- [70] Ethan Heilman. 2015. Added test-before-evict discipline in addrman, feeler connections. by Ethanheilman · pull request #6355 · Bitcoin/Bitcoin. <https://github.com/bitcoin/bitcoin/pull/6355>
- [71] Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. 2015. Eclipse Attacks on Bitcoin’s Peer-to-Peer Network. In 24th USENIX Security Symposium (USENIX Security 15). 129–144.
- [72] Ivan Homoliak, Sarad Venugopalan, Daniël Reijsbergen, Qingze Hum, Richard Schumi, and Pawel Szalachowski. 2020. The security reference architecture for blockchains: Toward a standardized model for studying vulnerabilities, threats, and defenses. IEEE Communications Surveys & Tutorials 23, 1 (2020), 341–390.
- [73] Imperva. [n. d.]. What is a sybil attack: Examples & prevention: Imperva. <https://www.imperva.com/learn/application-security/sybil-attack/> Accessed on 20.12.2023.
- [74] Rhonda Jacalynn. 2022. What is a blockchain replay attack? <https://www.certik.com/resources/blog/blockchain-replay-attack>
- [75] Abhishek Jain. 2018. Lecture 8 : Bitcoin mining. (2018). <https://www.cs.jhu.edu/~abhishek/classes/CS601-641-441-Spring2018/Lecture8.pdf> Accessed on 12.09.2022.
- [76] Ghassan O Karame, Elli Androulaki, and Srdjan Capkun. 2012. Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin. Cryptology EPrint Archive (2012).
- [77] Ghassan O Karame, Elli Androulaki, Marc Roeschlin, Arthur Gervais, and Srdjan Capkun. 2015. Misbehavior in bitcoin: A study of double-spending and accountability. ACM Transactions on Information and System Security (TISSEC) 18, 1 (2015), 1–32.
- [78] Wüst Karl and A Gervais. 2016. Ethereum eclipse attacks. Zurich, Switzerland: ETH Zurich (2016), 1–7.
- [79] Sunny King and Scott Nadal. 2012. Peercoin Whitepaper. <https://peercoin.net/assets/paper/peercoin-paper.pdf>. Accessed: 2024-09-02.
- [80] Yujin Kwon, Dohyun Kim, Yunmok Son, Eugene Vasserman, and Yongdae Kim. 2017. Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 195–209.
- [81] Wenting Li, Sébastien Andreina, Jens-Matthias Bohli, and Ghassan Karame. 2017. Securing proof-of-stake blockchain protocols. In Data Privacy Management, Cryptocurrencies and Blockchain Technology: ESORICS 2017 International Workshops, DPM 2017 and CBT 2017, Oslo, Norway, September 14-15, 2017, Proceedings. Springer, 297–315.
- [82] Xiaoli Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. 2020. A survey on the security of blockchain systems. Future generation computer systems 107 (2020), 841–853.
- [83] Doug Madory. 2022. What can be learned from recent BGP hijacks targeting cryptocurrency services? <https://www.kentik.com/blog/bgp-hijacks-targeting-cryptocurrency-services/>
- [84] Antonio Magnani, Luca Calderoni, and Paolo Palmieri. 2018. Feather forking as a positive force: incentivising green energy production in a blockchain-based smart grid. In Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems. 99–104.
- [85] Dr Adrian Manning. 2018. Short Address/Parameter Attack. (2018). <https://blog.sigmaprime.io/solidity-security.html#short-address> Accessed on 12.09.2022.

- [86] Yuval Marcus, Ethan Heilman, and Sharon Goldberg. 2018. Low-resource eclipse attacks on ethereum's peer-to-peer network. Cryptology ePrint Archive (2018).
- [87] Lukas Mastilak, Marek Galinski, Pavol Helebrandt, Ivan Kotuliak, and Michal Ries. 2020. Enhancing border gateway protocol security using public blockchain. Sensors 20, 16 (2020), 4482.
- [88] Sjouke Mauw and Martijn Oostdijk. 2006. Foundations of attack trees. In Information Security and Cryptology-ICISC 2005: 8th International Conference, Seoul, Korea, December 1-2, 2005, Revised Selected Papers 8. Springer, 186–198.
- [89] Philip Hayes Max Fang. 2017. HOW TO DESTROY BITCOIN. (May 2017). <https://www.bitcoin.org.hk/2017-05-destroy-bitcoin/> Accessed on 12.09.2022.
- [90] Patrick McCorry, Alexander Hicks, and Sarah Meiklejohn. 2018. Smart contracts for bribing miners. In International Conference on Financial Cryptography and Data Security. Springer, 3–18.
- [91] Robert McMillan. 2014. The Inside Story of Mt. Gox, Bitcoin's \$460 Million Disaster. (March 2014). <https://www.wired.com/2014/03/bitcoin-exchange/> Accessed on 12.08.2022.
- [92] Regina Mihindukulasuriya. 2022. Blinded by crypto buzz? Watch out. Crypto crime will rob world of \$30 bn/yr, says US cyber firm. (August 2022). <https://theprint.in/tech/blinded-by-crypto-buzz-watch-out-crypto-crime-will-rob-world-of-30-bn-yr-says-us-cyber-firm/1078951/> Accessed on 12.08.2022.
- [93] Andrew Miller. 2013. Feather-forks: enforcing a blacklist with sub-50% hash power. (October 2013). <https://bitcointalk.org/index.php?topic=312668.0> Accessed on 12.08.2022.
- [94] Joanna Moubarak, Eric Filiol, and Maroun Chamoun. 2018. On blockchain security and relevant attacks. In 2018 IEEE Middle East and North Africa Communications Conference (MENACOMM). IEEE, 1–6.
- [95] Satoshi Nakamoto. 2009. Bitcoin: A Peer-to-Peer Electronic Cash System. <https://bitcoin.org/bitcoin.pdf> (03 2009).
- [96] Satoshi Nakamoto and A Bitcoin. 2008. A peer-to-peer electronic cash system. Bitcoin.–URL: <https://bitcoin.org/bitcoin.pdf> 4 (2008), 2.
- [97] Christopher Natoli and Vincent Gramoli. 2017. The Balance Attack or Why Forkable Blockchains are Ill-Suited for Consortium. In 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). 579–590. <https://doi.org/10.1109/DSN.2017.44>
- [98] Neptune Mutual. [n. d.]. Solidity Front Running Attack. <https://neptunemutual.com/blog/solidity-front-running-attack/>. [Online; accessed 14-February-2024].
- [99] Emily Nicolle. 2022. Attacker Behind Record 2016 Crypto Hack Might Have Been Found. (February 2022). <https://www.bloomberg.com/news/articles/2022-02-22/attacker-behind-record-2016-crypto-hack-might-have-been-found> Accessed on 12.08.2022.
- [100] Onur Solmaz. [n. d.]. The Anatomy of a Block Stuffing Attack. <https://solmaz.io/2018/10/18/anatomy-block-stuffing/>. [Online; accessed 12-September-2022].
- [101] Ripon Patgiri, Sabuzima Nayak, and Naresh Babu Muppalaneni. 2021. Is Bloom Filter a Bad Choice for Security and Privacy?. In 2021 International Conference on Information Networking (ICOIN). 648–653. <https://doi.org/10.1109/ICOIN50884.2021.9333950>
- [102] Cristina Pérez-Solà, Sergi Delgado-Segura, Guillermo Navarro-Arribas, and Jordi Herrera-Joancomartí. 2019. Double-spending prevention for bitcoin zero-confirmation transactions. International Journal of Information Security 18, 4 (2019), 451–463.
- [103] Andrew Poelstra. 2015. On Stake and Consensus. (2015). <https://download.wpsoftware.net/bitcoin/pos.pdf> Accessed on 15.01.2024.
- [104] Rob Behnke. [n. d.]. EXPLAINED: THE DODO DEX HACK (MARCH 2021). <https://www.halborn.com/blog/post/explained-the-dodo-dex-hack-march-2021>. [Online; accessed 16-September-2022].
- [105] Meni Rosenfeld. 2011. Analysis of bitcoin pooled mining reward systems. arXiv preprint arXiv:1112.4980 (2011).
- [106] Ameer Rosic. 2019. Hypothetical Attacks on Cryptocurrencies. (August 2019). https://blockgeeks.com/guides/hypothetical-attacks-on-cryptocurrencies/#What_is_a_blacklisting Accessed on 12.09.2022.
- [107] Muhammad Saad, Afsah Anwar, Ashar Ahmad, Hisham Alasmay, Murat Yuksel, and David Mohaisen. 2022. RouteChain: Towards blockchain-based secure and efficient BGP routing. Computer Networks 217 (2022), 109362.
- [108] Muhammad Saad, Laurent Njilla, Charles Kamhoua, and Aziz Mohaisen. 2019. Countering selfish mining in blockchains. In 2019 International Conference on Computing, Networking and Communications (ICNC). IEEE, 360–364.
- [109] Muhammad Saad, Jeffrey Spaulding, Laurent Njilla, Charles Kamhoua, Sachin Shetty, DaeHun Nyang, and David Mohaisen. 2020. Exploring the attack surface of blockchain: A comprehensive survey. IEEE Communications Surveys & Tutorials 22, 3 (2020), 1977–2008.
- [110] Sarwar Sayeed and Hector Marco-Gisbert. 2019. Assessing Blockchain Consensus and Security Mechanisms against the 51% Attack. Applied Sciences 9, 9 (2019). <https://doi.org/10.3390/app9091788>
- [111] SECBIT. [n. d.]. A Comprehensive Solution to Bugs in Fomo3D-like Games. <https://hackernoon.com/a-comprehensive-solution-to-bugs-in-fomo3d-like-games-ab3b054f3cc5>. [Online; accessed 14-September-2022].

- [112] NEFTURE SECURITY I Blockchain Security. 2024. A year of crypto crimes in Review - the 2023 report. <https://medium.com/coinmonks/a-year-of-crypto-crimes-in-review-the-2023-report-7bb3ae6d9782>
- [113] I Sentana, Muhammad Ikram, and Mohamed Ali Kaafar. 2021. BlockJack: Towards Improved Prevention of IP Prefix Hijacking Attacks in Inter-Domain Routing Via Blockchain. *arXiv preprint arXiv:2107.07063* (2021).
- [114] Alan T. Sherman, Farid Javani, Haibin Zhang, and Enis Golaszewski. 2019. On the Origins and Variations of Blockchain Technologies. *IEEE Security & Privacy* 17, 1 (2019), 72–77. <https://doi.org/10.1109/MSEC.2019.2893730>
- [115] Hongwei Shi, Shengling Wang, Qin Hu, Xiuzhen Cheng, Junshan Zhang, and Jiguo Yu. 2021. Fee-Free Pooled Mining for Countering Pool-Hopping Attack in Blockchain. *IEEE Transactions on Dependable and Secure Computing* 18, 4 (2021), 1580–1590. <https://doi.org/10.1109/TDSC.2020.3021686>
- [116] Simon Taylor. [n. d.]. What the bZx flash loan exploit says about the future of DeFi. <https://content.11fs.com/article/what-the-bzx-flash-loan-exploit-says-about-the-future-of-defi>. [Online; accessed 15-September-2023].
- [117] Sushil Kumar Singh, Mikail Mohammed Salim, Minjeong Cho, Jeonghun Cha, Yi Pan, and Jong Hyuk Park. 2019. Smart contract-based pool hopping attack prevention for blockchain networks. *Symmetry* 11, 7 (2019), 941.
- [118] SlowMist. [n. d.]. Top Ten Blockchain Attacks of 2023. <https://slowmist.medium.com/top-ten-blockchain-attacks-of-2023-6dd681214fcf>. [Online; accessed 30-March-2024].
- [119] Siamak Solat and Maria Potop-Butucaru. 2016. Zeroblock: Preventing selfish mining in bitcoin. *arXiv preprint arXiv:1605.02435* (2016).
- [120] Yonatan Sompolinsky and Aviv Zohar. 2015. Secure high-rate transaction processing in bitcoin. In *Financial Cryptography and Data Security: 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers* 19, 507–527.
- [121] Joe Stewart. 2014. BGP hijacking for cryptocurrency profit. <https://www.secureworks.com/research/bgp-hijacking-for-cryptocurrency-profit>
- [122] Hanyi Sun, Na Ruan, and Chunhua Su. 2020. How to model the bribery attack: A practical quantification method in blockchain. In *European Symposium on Research in Computer Security*. Springer, 569–589.
- [123] P Swathi, Chirag Modi, and Dhiren Patel. 2019. Preventing sybil attack in blockchain using distributed behavior monitoring of miners. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 1–6.
- [124] Nicolas Tang. 2021. What is Sybil attack: How Blockchains Prevent Sybil Attacks. <https://phemex.com/academy/what-is-a-sybil-attack> (03 2021). Accessed on 13.09.2022.
- [125] vector76. 2011. Fake Bitcoins? <https://bitcointalk.org/index.php?topic=36788.msg463391#msg463391>. [Online; accessed 11-September-2022].
- [126] Pierre-Antoine Vervier. 2018. Why BGP Hijacking Remains a Security Scourge. <https://symantec-enterprise-blogs.security.com/blogs/feature-stories/why-bgp-hijacking-remains-security-scourge>
- [127] Yujian Wen, Fengyuan Lu, Yufei Liu, and Xinli Huang. 2021. Attacks and countermeasures on blockchains: A survey from layering perspective. *Computer Networks* 191 (2021), 107978.
- [128] wizkid057. 2014. Eligius: 0% Fee BTC, 105% PPS NMC, No registration, CPPSRB (New Thread). <https://bitcointalk.org/?topic=441465.msg7282674>. [Online; accessed 21-August-2022].
- [129] David Xiao. 2016. The Four Layers of the Blockchain. (2016). <https://medium.com/@coriacetic/the-four-layers-of-the-blockchain-dc1376efa10f> Accessed on 14.09.2022.
- [130] Rui Zhang, Rui Xue, and Ling Liu. 2019. Security and privacy on blockchain. *ACM Computing Surveys (CSUR)* 52, 3 (2019), 1–34.
- [131] Shijie Zhang and Jong-Hyook Lee. 2019. Eclipse-based stake-bleeding attacks in pos blockchain systems. In *Proceedings of the 2019 ACM International Symposium on Blockchain and Secure Critical Infrastructure*. 67–72.
- [132] Weiyu Zhong, Ce Yang, Wei Liang, Jiahong Cai, Lin Chen, Jing Liao, and Naixue Xiong. 2023. Byzantine fault-tolerant consensus algorithms: A survey. *Electronics* 12, 18 (2023), 3801.
- [133] Saide Zhu, Wei Li, Hong Li, Ling Tian, Guangchun Luo, and Zhipeng Cai. 2019. Coin Hopping Attack in Blockchain-Based IoT. *IEEE Internet of Things Journal* 6, 3 (2019), 4614–4626. <https://doi.org/10.1109/JIOT.2018.2872458>
- [134] SlowMist Zone. 2022. SlowMist Hacked Statistical Parameters. (2022). <https://hacked.slowmist.io/>, <https://www.slowmist.io/#intelligence> Accessed on 12.08.2022.