Proceedings of the 4th International Conference on Computing and Data Science (CONF-CDS 2022)
DOI: 10.54254/2755-2721/2/20220591

Center of darkness: Attacks and defensive strategies on blockchain consensus algorithm

Liuyi Fang[1,2]

[1]Hunan Applied Technology University, CN

[2]1474378088@qq.com

Abstract. With the rapid development of blockchain technology, more and more attention has been paid to the core consensus algorithm, and the related security problems have followed. For example, due to the loopholes in the consensus algorithm, the Bitcoin Gold platform lost about 18 million USD. Therefore, how to avoid such attacks against the consensus algorithm is an unavoidable topic for everyone involved in the blockchain platform. Starting from the mainstream consensus algorithm, this paper introduces their evolution process and their respective principles. In addition, we also introduce the possible attack principles and harms from two perspectives. One is *general*, that is, attacks under this category are not limited to specific consensus algorithm and platforms. The other type is *specific*, that is, the vulnerability locates in specific consensus algorithm. At the end of this paper, we divide all the personnel involved in the blockchain into three categories, and put forward specific suggestions for each category of personnel to help them better avoid and respond to possible attacks on the consensus algorithm.

Keywords: consensus algorithm, attacks, defensive strategies, blockchain.

1. Introduction

Since the birth of Bitcoin in 2009, blockchain has been formally proposed as its underlying support technology. As of March 2022, the total market value of the global blockchain reached up to approximately 2.8 trillion USD [1], among which the Bitcoin's reached up to approximately 1.27 trillion USD. Blockchain not only brings great economic value and research value, but also attracted extensive attention and research in industry and academic circles. Based on the blockchain technology, many customized blockchain platforms are implemented. The most famous ones are Ethereum, introducing smart contracts; and EOS.IO, a decentralized application development platform, widely known as the representative of Blockchain 3.0.

Although these platforms all use the blockchain as the underlying technology, they are not exactly identical. The main reason for this difference is the *consensus algorithm* in the blockchain. Consensus in the blockchain refers to whether majority of participants have obtained the verifications and confirmations of transactions in the network [2]. In this way, participants can ensure the safe growth of the blockchain, and further safeguard the legitimate rights of users. However, if loopholes are found in the consensus algorithm in the above process, it is likely to be deliberately exploited by malicious participants, causing serious ecological damage to the blockchain platform. For example, in May 2018, Bitcoin Gold was attacked by a malicious user [3], stealing about 390,000 BTG (around 18 million

DOI: 10.54254/2755-2721/2/20220591

USD). In addition, BSV also experienced block reorganization after being attacked. It was confirmed that the direct cause of these two attacks was the "51% attack" against the PoW consensus algorithm [4].

Therefore, this paper takes the mainstream consensus algorithm as the research object and thinks about how to prevent and address the attack against consensus algorithm. We have classified all the mainstream attacks on the consensus algorithm so far, and propose corresponding best practices on how to avoid these attacks for both developers and users of blockchain platforms.

The other parts of this paper are as follows: the second section gives an overview of related technologies, and the third section describes the characteristics and innovations of our work. The following two sections briefly introduce the mainstream consensus algorithm, and classify corresponding attacks against them, respectively. In the sixth section, we propose some best practices to help developers avoid these attacks. The last two sections conduct a discussion and summarize the full text.

2. Background

In this section, we will briefly introduce the basic concepts in the blockchain.

2.1. Primer on blockchain

In blockchain network, *nodes* play an important part. Generally speaking, nodes are divided into two types according to their functions: *ordinary nodes* and *miners*. The former one can generate, receive and forward data in the network, while the latter one not only have the capabilities of ordinary nodes, but also can collect data and package the collected data into a part of the blockchain according to the consensus algorithm. Each node has a unique identifier, that is, the *address*. Nodes can interact through declaring addresses, such as sending or receiving transactions. Moreover, *transaction* is one of the important parts of a data flow in blockchain network. It contains at least the sender's address, the recipient's address and intended sent data. After the transaction is constructed, it will be broadcasted to the network, forwarded by multiple nodes, and received and processed by the miners.

However, considering the possibility of malicious transactions, when the transactions are accepted by the miners, they are not directly added to the current blockchain, but will be verified firstly. Validated transactions are stored by miners in their own *transaction pool*, and some of transactions in the transaction pool are packaged into *blocks* according to consensus algorithm. Therefore, in a nutshell, a block is a batch of legal transactions and the necessary meta-information for security purposes. Similarly, blocks are sent to the blockchain network after being packaged. Once a block has passed the verification of most miners, it is added to the blockchain, i.e., being stored "permanently" on each node's disk.

2.2. Mining

As described in section 2.1, miners produce blocks according to a consensus algorithm, which process is called *mining*. To prevent malicious users, mining usually requires something to prove its legitimacy. For example, in the proof-of-work (PoW) consensus algorithm, they are required to solve an identical complex mathematical problem. When a miner takes lead in solving it, it means that the miner obtains the right to package the transactions in this time period into blocks. Other miners will move on to the next one. Note, such a process requires miners to pay a certain cost (e.g., computing power and hard disk), so there are some incentive mechanisms. For example, in Bitcoin, miners who successfully generate blocks will receive a certain token as award. In Ethereum, even mining the blocks that are not located on the main chain can obtain certain tokens. The incentive mechanism can ensure that the miners will compete for rewards, and enable the blockchain to be self-maintained.

However, the existence of competition may lead to multiple miners producing and broadcasting blocks at the same time, resulting in *forking* at a certain node. Therefore, different platforms have different solutions. Bitcoin proposed the longest chain principle [5], that is, the chain with the largest inputted hashrate is regarded as the main chain, other blocks will be considered illegal and abandoned.

DOI: 10.54254/2755-2721/2/20220591

Moreover, the GHOST protocol (introduced by Ethereum) allows some off-mainchain blocks to be attached, so that their miners can get a certain percentage of extra income [6].

3. Related work

Consensus algorithm. Since Bitcoin showed its potential in 2009, more and more researchers have been involved in various fields of blockchain. PoW, as the consensus algorithm that people pay attention to, has some defects, such as low verification efficiency and high energy consumption. Therefore, many new consensus algorithms are proposed [5,7-12]. Among them, Dziembowski's Proof of Space has solved the problem of mining by hard disk capacity instead of computer computing power [10]. The early version of Proof of Stake implemented by King et al [7]. replaces computer calculation with the number of tokens hold by users, which also alleviates the problem of energy consumption to a certain extent. On this basis, Larimer proposed the Delegated Proof of Stake [9], which significantly improved the transaction verification efficiency on the premise of sacrificing a certain degree of decentralization.

Security problem of consensus algorithm. Since most blockchain platforms have no authentication mechanism, any user can access the blockchain network as an ordinary node. Therefore, malicious users can use the defects of consensus algorithm to attack the blockchain platform, resulting in illegal gains or adverse consequences such as affecting the development of the blockchain. Some work focuses on the security issue of consensus algorithm [13-19]. For example, Ittay Eyal et al started from the existing consensus algorithm [13], deeply studied the mining mechanism of Bitcoin, and proposed an attack to obtain additional illegal gains by hiding mined blocks. Natoli et al explored the relationship between power and network delay from the perspective of blockchain network [14], and proposed a theoretical attack that uses network delay to obtain illegal benefits. Vitalik Buterin proposed an attack on the Ethereum platform that uses Nash equilibrium to affect PoW [15]. Specifically, the attack took advantage of the greedy mentality of miners to indirectly control the development of the blockchain in the form of a game. In addition, this paper also gives guidance on how to avoid and resist these attacks, aiming at helping all kinds of people to better modify and perfect the existing consensus algorithm.

4. Consensus algorithm

Since 2009, the consensus algorithm, Proof of Work (PoW) of Bitcoin, has been tested by practice and has a considerable degree of security and stability. However, this does not mean that PoW has no disadvantages (e.g., energy consumption problem [20]). As mentioned in the related work, a large number of consensus algorithm designed for specific scenarios and problems have emerged in the blockchain market. We will introduce the mainstream ones that are widely used at present in detail.

4.1. Proof of Work (PoW)

PoW can be traced back to 1992 as early as possible [21]. Until 2008, Satoshi Nakamoto officially defined the PoW algorithm in the blockchain [5].

The process of PoW is roughly as follows: the miners need to constantly modify a variable named *nonce* to solve a mathematical problem. If one of them luckily enough get the correct nonce, the miner obtains the right to package a block for this time period. The packaged block, carrying the nonce, will be broadcasted. When other miners receive the block, they will verify transactions and the nonce. If the verification passes, it will be linked to the blockchain and all miners will move on to the next problem.

The node that successfully mines the block will receive some tokens. Because of this incentive, miners under PoW compete to solve mathematical problems, thus ensuring the activity of the network. However, a lot of computing resources are wasted, resulting in energy consumption.[1] In addition, in order to avoid the forking problems, Bitcoin also proposes the longest chain principle to ensure reaching the consensus.

[1] Due to enormous energy consumption, many countries began to resist the mining activities [11]

DOI: 10.54254/2755-2721/2/20220591

4.2. Proof of Stake (PoS)

In order to solve the energy consumption problem of PoW, the idea of PoS was proposed in 2011 [22]. PoS attempts to convert the amount of money held by a user into the user's hashrate. In 2012, Sunny King and Scott Nadal launched Peercoin (PPC), which realized an earlier version of PoS [23]. Specifically, the PPC takes *coin age* as its mining power, which can be calculated by multiplying the amount of money in the user's wallet by how long the money is held. The higher coin age one has, the easier the mathematical problem he needs to solve. The other process is similar to Bitcoin's.

Ethereum plans to adopt PoS consensus in its 2.0 stage. Different from the PPC's PoW-PoS mixed design, Ethereum completely abandoned the competition-based mining strategy, and randomly selected a set of *validator nodes* to prove the new block [8]. It should be noted that these validators are not pre-selected or elected, ordinary nodes can become validators by committing a certain amount of Ether to an official smart contract. This not only completely avoids the problem of energy consumption, but also strengthens the decentralization of Ethereum. Moreover, for Ethereum officials, the scope of searching for attackers has also been narrowed from all nodes in the whole network to nodes that have mortgaged tokens before.

4.3. Delegate Proof of Stake (DPoS)

DPoS was proposed and applied to Bitshares blockchain in 2014 [9]. As a derivative algorithm of PoS, DPoS still uses stake as proof, but DPoS introduces the concept of *delegate*. Specifically, the DPoS algorithm allows each node to mortgage its owned tokens as votes for the trusted nodes as delegates. The top n most trusted nodes (different blockchains have different n, for example, EOS is 21 and Bitshares is 101) are responsible for generating blocks in turn and uploading them to the blockchains.

Therefore, the consensus process of DPoS is different from both PoW and PoS. The participating miners no longer need to solve the so-called mathematical problem or mortgage a certain number of tokens to become validators, but only need to vote for their own trusted nodes.

4.4. Proof of Space (PoSpace)

PoSpace is also called Proof of Capacity. Due to the energy waste problem, PoSpace was proposed as a substitute for PoW in 2013 [10]. Specifically, the PoSpace no longer requires users to spend a lot of time and computing resources to solve mathematical problems. It uses the space of hard disk of the computer instead of the CPU (or GPU) for mining. In short, the PoSpace generates a large hash table from the data on the hard disk, then randomly generates a value and calculates a value called *deadline* by looking up a table, which is the time when the user generates a new block. For example, if the minimum deadline obtained by a user through the above steps is 15 seconds, and no new block is generated within the next 15 seconds, he can generate a block and obtain a reward after 15 seconds. The consensus algorithm of the Chia chain which exploded in 2021 is based on the PoSpace [24].

4.5. Proof of Authority (PoA)

Two types of nodes exist in the network using PoA: *validator nodes* and *normal nodes*. The generation of new blocks depends entirely on the validator nodes, which are the "approved" set of nodes in the network, dramatically increasing the degree of centralization of the network. Therefore, the PoA is generally not considered as a consensus algorithm for the public chain alone. For example, the BSC public chain combines DPoS and PoA consensus algorithm. It first selects validators nodes through a mechanism similar to DPoS, so as to ensure that the number of validators is limited and can be replaced according to a certain condition. Then, the PoA algorithm is used to let validators generate new blocks in turn.

5. Attack

In this section, we discuss the attacks against consensus algorithm from two directions. One is the *general attack*, any consensus algorithm may suffer, and the other is the *specific attack*, which is specific on a certain consensus algorithm.

DOI: 10.54254/2755-2721/2/20220591

5.1. General attack

5.1.1. 51% attack. 51% attack mainly targets on blockchain platforms whose consensus algorithms are without authentication, e.g., PoW and PoS. Taking advantage of the lack of identity authentication, it is difficult for the maintainer of the blockchain to capture the attacker's information and conduct the following defense.

Take the 51% attack on the PoW as an example. In PoW, the higher the hashrate of a miner, the faster his mining speed. When a miner has more than 51% hashrate of the whole network, we can think that the blockchain has been controlled by the miner. This also means that as long as there is enough time, it can start from a certain point in the past and regenerate a chain. Once its height exceeds the current main chain, according to the longest chain principle, nodes will switch the main chain to this "fake" chain. The process can even start from the genesis block and generate a chain that completely replaces the original chain. It is worth noting that the launching requirement of 51% attack varies according to different consensus. For example, in PoS, the attacker needs to have 51% of the total mortgaged tokens to meet the launching conditions.

5.1.2. Double spending attack. Double spending attack is an attack that uses the same token twice or more in exchange for multiple gains. Specifically, the attacker will firstly publish a transaction and contact the service provider (the victim). When the victim believes that the transaction has been completed and provides services to the attacker (actually the transaction may not have been confirmed), the attacker will invalidate the transaction by some means, achieving the purpose of obtaining services without cost. For example, 51% attack can implement double spending attack. When the victim sees the transaction included in the chain and thinks that the transaction has been completed, the attacker directly rewrites the blockchain through 51% attack.

Additionally, the attacker can take advantage of rules of the blockchain to implement the double spending attack. In Ethereum, for example, each transaction costs *gas* as transaction fees. Since the miners can obtain the gas carried by transactions as reward, the higher the gas, the quicker the confirmation of transactions. Using this rule, the attacker first issues a transaction with low gas and sets the target address as the victim. After the victim provides services, the attacker issues another transaction with higher gas and sets the target address as himself. The rational miner will add the latter transaction to the candidate block first. In this way, when the transaction with low gas is finally processed, the verification will not pass due to insufficient balance, thus realizing double spending attack. Three types of attacks that can achieve the purpose of double spending attack are introduced following.

Race attack. Specifically, race attack is performed by simultaneously broadcasting two transactions with the same piece of tokens to the blockchain network at almost the same time, one to the victim and the other to attacker himself. The attacker needs to replace the former one with the latter one, thus deceiving the victim to finish double spending attack. The above example of using gas adopts the idea of race attack. In addition, race attack is also possible in Bitcoin. The RBF protocol introduced by Bitcoin enables unacknowledged transactions to be replaced by a higher fee one (similar to gas for Ethernet transactions), making race attack possible in Bitcoin.

Finney attack. Finney the attack is achieved by secretly mining blocks to achieve transaction rollback. It requires that the attacker must be a node participating in block generation. The attacker initiates two transactions, one for the victim and the other for himself. When the second transaction is issued, the attacker begins to mine candidate blocks containing the second transaction. When the candidate block is constructed, the attacker does not immediately publish it to the network, but waits for the victim to accept the unconfirmed transaction and provide services. At this point, the attacker broadcasts the self-mined block, and the first transaction is invalidated because the second transaction has been confirmed.

Vector 76 attack. This attack targets on a centralized exchange and technically combines race attack and finney attack. To be specific, the prerequisite is to obtain the support of two or more nodes participating in block generation. Take two nodes (nodes A and B) as an example. First, node A is

DOI: 10.54254/2755-2721/2/20220591

directly connected to the server of the exchange, and node B is connected to other nodes in the network. The attacker creates two transactions, whose target addresses are the exchange and the attacker himself, respectively. At this point, node A starts to secretly mine the block containing the former transaction, and once the block is generated, it is directly sent to the exchange (not broadcast on the network). After the transaction is confirmed by the exchange, the attacker immediately performs the withdrawal operation on the exchange and simultaneously releases the latter transaction to the network through the node B, in an attempt to create a "forking". If the forked chain becomes the main chain, the exchange takes the financial loss.

5.2. Specific attack

5.2.1. Selfish mining attack. Selfish mining refers to the behavior that selfish nodes mine legal blocks, but continue to mine along the blocks without broadcasting them to the network. Due to the incentive mechanism of Bitcoin, all the miners are competing to mine blocks. Thus, the nodes in the network can be divided into two types, one is honest nodes, the other is selfish mining nodes. Some selfish mining nodes may collude to form a node pool. The attack strategy is shown in Figure 1. Assuming that at the initial state of the blockchain, all nodes mine new blocks on the latest block in the main branch, two possible states appear: State A and State B.

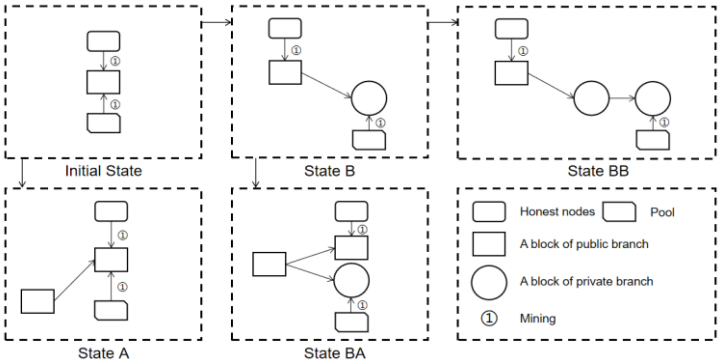

Figure 1. Selfish mining attack strategy.

State A: Honest nodes firstly mine new blocks on the main chain. It will be unprofitable for the node pool to selfishly mine the old blocks (as the distinction of hashrate between node pool and honest nodes), so it will move on to the latest blocks.

State B: The node pool firstly mines a new block on the main chain. Instead of broadcasting it to the network, the node pool continues to mine along this new block, forming a "hidden" private branch in the network, and then producing two possible subsequent states: state BA and state BB.

State BA: Honest nodes mine another new block soon after the state B, indicating the two chains are at the same height. Thus, the node pool immediately broadcasts the block to make two chains competitive. Honest nodes will choose one of them to mine (because honest nodes do not aware the malicious nodes). After that, there will be three results. First, the node pool takes the lead and directly announces the new block, thus the forking chain is the new main chain, resulting in two blocks' award. Second, if honest nodes based on the forking chain are the first, the node pool obtains the revenue of one block. Third, if the honest nodes based on the main chain are the first, then the node pool cannot obtain any revenue.

State BB: Another new block has been mined from the node pool, and the height of the forking chain is two blocks ahead of the main chain. Node pool still does not broadcast the fork chain, but continues to mine new blocks along the forking chain. The height of the main chain, however, will finally catch up. Therefore, once a new block is mined on the main chain, the node pool announces a block on the forking chain (similar to State BA). When the forking chain is only one block ahead of the main chain, the node pool immediately announces the entire forking chain. Due to the longest chain principle, it will be the new main chain, and the node pool receives all blocks' award.

DOI: 10.54254/2755-2721/2/20220591

[13] points out that when the node pool has 1/4 of the hashrate of the whole network and half of the honest nodes mine based on the forking chain, the profit of the node pool for selfish mining will be greater than that of the nodes for honest mining. It is worth noting that not only Bitcoin may be attacked by selfish mining, [17] indicating that Ethereum is more vulnerable to selfish mining attack.

5.2.2. Balance attack. Balance attack is an attack against the PoW consensus algorithm, and the target is usually a set of nodes under a certain hashrate threshold. This attack affects the network communication between node groups in the network by adding communication delay between them, and then implements a double spending attack.

The process of balance attack is as follows. First, the attacker finds two node groups with similar hashrate (for example, node groups A and B), and introduces communication delay between them through eclipse attack [16]. Then the attacker initiates a transaction targeting the victim in group A, simultaneously initiates another transaction targeting himself in group B. Moreover, the attacker tries to mine blocks in group B to ensure that the height in group B exceeds that of the group A. Once the victim confirms the transaction and provides the service, the attacker cancels the communication delay between the two node groups and allows the blockchain in group B to replace the one in group A, which leads to a double spending attack.

The paper shows that if the attacker has 5% computing power of the network and can introduce communication delay into multiple node groups, then the probability of double spending attack on the GHOST protocol is very high [14].

5.2.3. Nothing-at-stake attack. Nothing-at-stake attack is specifically against earlier versions of PoS (like what PPC adopts), in which the attacker usually initiates two transactions and devotes himself to mining the block containing one of them to implement double spending. In the PoW, miners have to gamble to choose one of the chains for mining because it takes a lot of calculation to generate new blocks. In the earlier versions of PoS, the miners had enough calculation power to mine all chains for their own benefit (because there was always one branch that would become the main chain).

Assuming that the PoS blockchain has and only has two chains at this time, and all miners except the attacker mine on both chains simultaneously. At this time, the attacker initiates two transactions on two chains respectively, with the target address of an exchange (the victim) and himself. When exchange confirms the transaction, the attacker immediately initiates a withdrawal request. After the exchange responds to the demand for the withdrawal, the attacker can put all his effort into another branch, eventually making it as the main chain, thus causing a double spending attack.

Note that the nothing-at-stake attack is different from the vector 76 attack. First, nothing-at-stake attack only targets early PoS consensus and the vector 76 can target all consensus algorithm. Secondly, all the miners in nothing-at-stake attack will carry out rational mining, while it is not necessary for the miners in vector 76.

5.2.4. Grinding attack. Grinding attack means an attacker indirectly increases the probability of generating a block by controlling seeds of random number generator (RNG). In PoS, although the probability of nodes generating new blocks is positively correlated with stakes, this probability still depends on the RNG. Most of the blockchains use deterministic values such as block headers and creation time as the seeds for the RNG. Therefore, the attacker of grinding attack can indirectly control his probability of becoming a producer of new blocks by controlling the random number seeds. In [25], a user pointed out that such a weakness exists in the security model of PPC. In [26], the author points out that NXT platform may also be subject to this attack in theory [27].

5.2.5. Long range attack. Long range attack is an attack against the PoS consensus algorithm. It tries to acquire past stakes and rewrite the history of blockchain in some way. In a nutshell, the attacker will start from the genesis block to re-mine blocks. The attack strategy is generally as follows. The attacker obtains the private key of users who had a high stake in the past period of time by social engineering.

DOI: 10.54254/2755-2721/2/20220591

Ideally, an attacker can acquire more than 51% of stake over a certain period of time in the past without corresponding cost. So, the attacker can start from the past to mine a new chain and make it the main chain. For example, suppose a PoS network with a total of three miners, all of whom have a 1/3 stake. At this time, miner A announces to quit when the height reaches a certain threshold. When miner A left, miner B used some means to obtain A's private key. At this point, miner B obtained the 2/3 stake, so it could start mining a new chain from genesis block and replace the main chain.

5.2.6. P+epsilon attack. The P+epsilon attack is a theoretical attack to the PoW-based blockchains [15]. The attack's strategy is based on a voting game. Assuming that there are two options, A and B, if a participant votes for A and A wins, the participant will be rewarded with P. The participants' expected earnings are shown in Table 1.

Table 1. Participants' revenue statement.

	Vote for A	Vote for B
A wins	P	0
B wins	0	P

Now, we introduce this model into the PoW network, assuming there are two chains A and B. Miners need to choose one of them for mining. At present, a miner is mining at chain A, and successfully generates a new block and broadcasts it to the network. Thus, A becomes the main chain, and then the miner can receive a reward P. At this point, the miners' income statement is shown in Table 2.

Table 2. Miners' revenue statement.

	Mining on A	Mining on B
A becomes main chain	P	0
B becomes main chain	0	P

Assuming that all the miners in the PoW network are selfish, we can construct the following attack: the attacker sends a message to all the miners: *If the branch you mine does not become the main chain eventually, you can not only get the award P, but also the additional award epsilon that I offer*. If a miner chooses to perform the requirements in the information, his revenue statement is shown in Table 3. At this time, because miners in the network are selfish, they will choose to mine on B for extra benefit. Consequently, branch B would become the main chain because of the input hashrate. So, the attacker does not need to pay extra epsilon and his reputation is not harmed, the blockchain, however, is grown in the direction the attacker wants.

Table 3. Revenue statement of miners' performance requirements.

	Mining on A	Mining on B
A becomes main chain	P	P + epslion
B becomes main chain	0	P

6. Best practice

Based on the introduction and understanding of consensus algorithm in the previous sections, we put forward some best practices on security issues, aiming at reducing or even avoiding the economic losses caused by vulnerabilities in consensus algorithm. We will split the audience into three categories: blockchain platform developers, application developers and users on blockchain platform.

6.1. For blockchain platform developers

How to resist the attack that the consensus algorithm may suffer is the core of the whole blockchain platform. From the point of view of the aforementioned attacks, an attacker must meet certain special conditions or pay a certain attack cost before it can successfully carry out the attack. In addition, the attack transaction can reveal some information, which can be used to make corresponding punishment. Therefore, it is necessary for developers to increase the cost of attack implementation and set up efficient evidence collection and punishment measures.

DOI: 10.54254/2755-2721/2/20220591

6.1.1. Increase the cost of attack. For 51% attack, increasing attack cost is the most effective defense measure. For example, in the PoS-based blockchain, the cost is 51% of the whole network's stake. And for PoA, the cost becomes how to control 51% of the validator nodes. Therefore, developers need to consider what to use as proof to maximize the cost of 51% attack.

As for double spending attack, although it can be implemented in many ways, it essentially invalidates some transactions. The invalidation can be roughly divided into two categories, i.e., replacing transactions through competition of the block or transaction. In the former case, the cost can be increased by reducing the possibility of competition between blocks. For example, the cost of producing blocks can be increased by calculating mathematical problems like Bitcoin, but this will harm the performance. It is also possible to produce blocks in turn like PoS, but this increases the centralization. Developers may have to balance the performance and the degree of centralization. For the latter case, it is not recommended to directly remove fee mechanism to reduce competition. Moreover, as attackers often use insufficient balances to invalidate the original transactions to realize double spending, the developers can verify the balances of users so that transactions in low-balance accounts can proceed normally only after a certain margin is paid.

For other attacks, the possibility of attack implementation can also be avoided by increasing the cost of the attack. For example, the Bitcoin protocol can be modified to increase the cost of launching selfish mining attacks [13]. Reduce reliance on network communications to increase the cost of implementing balance attacks.

6.1.2. Set up efficient forensics and punishment measures. After the attack, it is also important on how to efficiently collect evidence and punish the attacker. For example, for a selfish mining attack, the attack can be mitigated to some extent by reducing the reward of nodes for successfully mining blocks continuously. As for the nothing-at-stake attack, Ethereum gives two kinds of punishment measures: the first is called Slasher [28], when users carry out rational mining, the deposits of users will be deducted; and the second is to punish users who have carried out rational mining (e.g., reducing incentives). As for long range attack, developers cannot restrict attackers to gain access to other people's accounts. Therefore, we propose to verify every n blocks in the blockchain. After verification, all chains before this block are considered permanent.

Unlike other attacks which exploit loopholes in the mechanism, the P+epsilon attack uses the entire community to carry out the attack. No matter from the perspective of increasing the cost of the attack or setting up punishment measures, it cannot achieve good results. Although this attack is difficult to achieve, we still recommend that developers conduct anti-fraud publicity in the community to increase the alertness.

6.2. For decentralized application developers

As for most of the current decentralized applications are written in smart contracts, which are independent to the consensus algorithm, some problems, however, still need to be paid attention. Take the grinding attack as an example, we already know that the attacker can tamper with the random seeds of the RNG to control the output, which can also be performed in smart contracts. Assuming that there is a gambling DApp, the attacker can manipulate the winners by modifying or selecting the random seeds. Therefore, developers of DApps should also try not to use static data as random seeds, and should use oracle to obtain external random sources as random seeds.

6.3. For users

Users of blockchains can be roughly divided into two categories: service provider and individual users. The goal of most of these attacks is to conduct double spending. Therefore, service providers have a high potential to be attacked. For them, how to avoid double-spending transactions is the most important. The most typical example is finney attack, in which the victim provides services after seeing a buyer initiate a transaction. This is clearly not desirable, as he should wait for a period of time before offering services until the transaction is fully confirmed (cannot be rolled back). In Bitcoin, the transaction in the

DOI: 10.54254/2755-2721/2/20220591

latest 6 blocks in the blockchain may be invalidated. For individual users, although they do not need to provide services in the application, they also need to be alert to the possibility of being attacked by double spending.

7. Discussion

Limitations of attack types. In a large number of attack examples and theoretical speculation, we finally selected these six typical attacks against consensus algorithm. Of course, many attacks have not been shown by us, but we argue that other attacks can be basically regarded as similar or derivative of these six attacks. For example, the selfish mining attack originates from PoW, but may be conducted on other consensus algorithms. In addition, the existence of the P+epsilon attack indicates that developers should be vigilant against similar attacks from the perspective of game theory.

Not detailed best practice. In the previous section, we put forward the best practice for each attack, but these defensive suggestions are general without specific practical advice. This is because we start from the principle of vulnerability, and specific practices may require developers to deploy in combination with the practical platform. For example, to defense long range attack, our proposal is to set a checkpoint every n blocks, but we do not give the optimal value of n. This is because the optimal value of n needs developers to find out according to the actual situation. The n can not only effectively resist long range attack, but also achieve an optimal balance among the maintenance costs, gains and financial losses after attacks.

8. Conclusion

In this paper, we introduce the mainstream consensus algorithm and corresponding attacks. Specifically, we introduce in detail how various mainstream consensus algorithm maintain the blockchain, and classify attacks into *general* and *specific* categories according to if they target on specific blockchains. For each attack, we introduce its principle, the triggering conditions, the benefits that the attacker can obtain and the detailed attack process. At the end of this paper, we divide the users involved in the blockchain into three categories, and give best practices on how to avoid and resist attacks against consensus algorithm. We argue that these best practices can significantly benefit the healthy and steady growth of blockchain platforms.

References

[1] Total Cryptocurrency Market Cap. https://coinmarketcap.com/charts/ (2020).
[2] Consensus (computer_science). https://en.wikipedia.org/wiki/Consensus_(computer_science). (2022).
[3] Bitcoin Gold suffers 51% attack. https://bitcoingold.org/responding-to-attacks/ (2018).
[4] BSV suffers 51% attack. https://twitter.com/LucasNuzzi/status/1422637361138130944 (2021).
[5] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 21260.
[6] Sompolinsky, Y., & Zohar, A. Secure high-rate transaction processing in bitcoin. In International Conference on Financial Cryptography and Data Security (pp. 507-527). Springer, Berlin, Heidelberg (2015).
[7] King, S., & Nadal, S. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. self-published paper, August, 19(1) (2012).
[8] Ethereum's proof of stake. https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/?msclkid=a9fb4e6ccebb11ec874ac87e4da2202f (2022).
[9] Larimer, D. Delegated proof-of-stake (dpos). *Bitshare whitepaper*, *81*, 85 (2014).
[10] Dziembowski, S., Faust, S., Kolmogorov, V., & Pietrzak, K. Proofs of Space. Cryptology ePrint Archive (2013).
[11] Binance Smart Chain's consensus algorithm. https://github.com/bnb-chain/whitepaper/blob/master/WHITEPAPER.md#proof-of-staked-authority (2020).

DOI: 10.54254/2755-2721/2/20220591

[12] Schwartz, D., Youngs, N., & Britto, A. The ripple protocol consensus algorithm. Ripple Labs Inc White Paper, 5(8), 151 (2014).
[13] Eyal, I., & Sirer, E. G. Majority is not enough: Bitcoin mining is vulnerable. In International conference on financial cryptography and data security (pp. 436-454). Springer, Berlin, Heidelberg (2014).
[14] Natoli, C., & Gramoli, V. The balance attack against proof-of-work blockchains: The R3 testbed as an example. arXiv preprint arXiv:1612.09426 (2016).
[15] The P + epsilon Attack. https://blog.ethereum.org/2015/01/28/p-epsilon-attack/ (2015).
[16] Heilman, E., Kendler, A., Zohar, A., & Goldberg, S. Eclipse Attacks on {Bitcoin's}{Peer-to-Peer} Network. In 24th USENIX Security Symposium (USENIX Security 15) (pp. 129-144) (2015).
[17] Niu, J., & Feng, C. Selfish mining in ethereum. arXiv preprint arXiv:1901.04620 (2019).
[18] Schwarz-Schilling, C., Neu, J., Monnot, B., Asgaonkar, A., Tas, E. N., & Tse, D. Three Attacks on Proof-of-Stake Ethereum. arXiv preprint arXiv:2110.10086 (2021).
[19] Gaži, P., Kiayias, A., & Russell, A. Stake-bleeding attacks on proof-of-stake blockchains. In 2018 Crypto Valley conference on Blockchain technology (CVCBT) (pp. 85-92). IEEE (2018).
[20] The Environmental Impact of Bitcoin Mining. https://coincentral.com/what-is-the-environmental-impact-of-bitcoin-mining/ (2018).
[21] Castro, M., & Liskov, B. Practical Byzantine fault tolerance and proactive recovery. ACM Transactions on Computer Systems (TOCS), 20(4), 398-461 (2002).
[22] Proof of stake instead of proof of work. https://bitcointalk.org/index.php?topic=27787 (2011)..
[23] Peercoin & Proof of Stake Consensus. https://github.com/peercoin/PeercoinUniversity/blob/master/app/assets/docs/09-peercoin-and-proof-of-stake-consensus.md (2019).
[24] Chia Consensus. https://docs.chia.net/docs/03consensus/consensus_intro (2022).
[25] ppcoin - stake burn-through vulnerability. https://bitcointalk.org/index.php?topic=131901.0 (2012).
[26] NXT POS Block Skipping Attack Myth. https://hackernoon.com/nxt-pos-block-skipping-attack-myth-de88cf4b3363 (2018).
[27] NXT blockchain platform. https://www.jelurida.com/nxt (2022).
[28] Ethereum Wiki. https://eth.wiki/concepts/proof-of-stake-faqs (2022).