Electronic Research Archive

http://www.aimspress.com/journal/era

ERA, 30(10): 3735–3754.
DOI: 10.3934/era.2022191
Received: 30 May 2022
Revised: 19 July 2022
Accepted: 27 July 2022
Published: 10 August 2022

Research article

The impact of network delay on Nakamoto consensus mechanism

Shaochen Lin[1], Xuyang Liu[1], Xiujuan Ma[3], Hongliang Mao[3], Zijian Zhang[1,2,*], Salabat Khan[4] and Liehuang Zhu[1,*]

[1] School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China

[2] Southeast Institute of Information Technology, Beijing Institute of Technology, Fujian 351100, China

[3] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China

[4] School of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China

* **Correspondence:** Email: zhangzijian@bit.edu.cn, liehuangz@bit.edu.cn.

Abstract: Nakamoto consensus is prevailing in the world largest blockchain-based cryptocurrency systems, such as Bitcoin and Ethereum. Since then, various attempts have been studied to attack Nakamoto consensus worldwide. In recent years, network delay has won more attention for making inconsistent ledgers in blockchain-based applications by virtue of attacking Nakamoto consensus. However, so far as we know, most of the existing works mainly focus on constructing inconsistent ledgers for blockchain systems, but not offering fine-grained theoretical analysis for how to optimize the success probability by flexibly dividing computational power and network delay from the viewpoint of adversary. The paper first utilizes network delay and the partition of controlled computation power of honest miners for making forks as long as possible. Then, formally analysis is provided to show the success probability of the proposed attack, and compute the optimal network delay and splitting for adversarial computation power in theory. Finally, simulation experiments validate the correctness of the formal analysis.

Keywords: network delay; Nakamoto consensus; blockchain

1. Introduction

Blockchain, as an append-only distributed database, has the characteristics of tamper-proof, decentralization and transparency, etc. These advantages of blockchain technology enables blockchain-based applications to be widely used in finance, healthcare and education all around the

world [1]. For an instance, Nakamoto published a white paper about Bitcoin in 2008 [1]. This was the first and largest blockchain-based cryptocurrency system. In Bitcoin, transactions in blocks can achieve consensus among miners based on Proof of Work (PoW) which is the earliest Nakamoto. From then on, Nakamoto consensus started to be applied into various of blockchain systems.

In Nakamoto consensus, all the miners try their best to create valid new blocks as soon as possible through solving cryptographic puzzles. With higher computation power comes a higher success probability of solving a puzzle and creating a valid new block by a miner.

Garay et al. [2] first built a formal adversarial model that allow a small part of computing power is controlled by an adversary. Based on there model, they analyze the security of Nakamoto consensus. Precisely, Nakamoto consensus makes the final ledger consistent to every miner under the condition that more than half of the total computational power of miners is honest and cannot be controlled by the adversary. However, there analysis premises the network without delay. This is impractical because Nakatomo consensus is designed to work on the Internet. In the real world, network delay is normal. Both message routing and network attack are likely to cause network delay. Several studies divided networks into three main types, synchronous networks, partial synchronous networks and asynchronous networks, and Nakamoto consensus is usually postulated to guarantee security in asynchronous networks [3].

Network delay poses a serious threat to violate the security of Nakamoto consensus, especially when networks have long delay. More concretely, long network delay makes forks inevitable for the Nakamoto consensus. If forks cannot be merged in time, the requirement of ledger consistency cannot be met in blockchain applications as well [4]. Pass et al. [5] broke the consistency of blockchain in a network with a-priori bounded delay Δ. The adversary is assumed to have the ability to isolate some miners'network communication in a short time period. Wei et al. analyzed the security of blockchain systems in a long network delay where no computation power is completely controlled by the adversary [6], but the adversary can divide honest miners into two groups with equal number. Yuan et al. also analyzed the security of Nakamoto consensus, when assuming that both a small part of honest miners' computation power can be corrupted and network delay can be arbitrarily made [7].

The aforementioned works inspire us to address a question. Suppose the adversary can control a part of computation power and network delay to honest miners. How does the adversary arrange the computation power and deploy the network delay to honest miners in order to optimize the success probability of making inconsistent ledger as long as possible for Nakamoto consensus?

The concrete contributions of there paper are summarized as below.

1) A new attack to Nakamoto consensus is proposed by using a part of computation power and network delay of honest miners. The proposed attack dynamically change network delay for parts of honest miners, such that the success probability can be enhanced when comparing with the previous works.
2) The success probability of attacking Nakamoto consensus is formally analyzed. Specifically, both the partition of the controlled computation power and the network delay in the proposed attack are proven to optimize the success probability.
3) Simulation experiments validate the correctness of the formal analysis and show the enhancement when comparing with the previous works.

2. Related works

The Nakamoto consensus greatly solves the problem of the decentralized peer-to-peer system [1], which has lead to new consensus protocol such as Proof of Stake (PoS) and Delegated Proof of Stake (DPoS) [8, 9]. Since then, a number of blockchain applications emerge in the scenarios of cryptocurrencies [10, 11], micropayment [12, 13], privacy protection [14–17], and fair secure computation [18–20], etc. Furthermore, a series of researches focus on security of consensus protocol occurs [21, 22].

The attack for Nakamoto consensus can mainly be classified into two categories. One is from the network aspect, the other is from the consensus aspect. The network layer encapsulates the networking mode, message transmission protocol and data verification mechanism of the blockchain system, and is the most basic technical architecture in the blockchain technology system. Like other applications running on the Internet, blockchain system may suffer DoS (denial-of-service) attack [23] in the network layer, which means attackers can use a number of network resources to attack blockchain systems or networks, causing them to stop responding or even crash, thus denying services. For p2p network, [24] propose eclipse attacks where the attacker manipulates multiple peer nodes to maintain long-term transmission connections with the target node, making the number of online connections reach the upper limit of inbound connections of the target node, thus blocking connection requests from other legitimate nodes. At there point, the target node is "isolated" from the P2P network by an attacker, causing the target node to fail to properly maintain the blockchain ledger.BGP(border gateway protocol) is a key component of the Internet. An adversary can hijack BGP to manipulate the Internet routing path [25]. Since blockchain transmits information based on the Internet, hijacking BGP can be used to mislead and intercept the traffic of blockchain nodes. Once an attacker takes over the nodes' traffic, it can disrupt consensus and transaction processes by affecting the proper functioning of the blockchain network.

In addition to attacking at the network layer, an adversary can also launch an attack from the aspect of consensus. [26] proposed an attack called bribery attacks on Bitcoin-style consensus. The adversary can temporarily gain more than 50 percent of total mining power by posting "malicious rewards" that encourage miners to mine on the adversary's designated branch chain. Once the adversary has more than 50 percent of the total network power, she can easily launch a double-spending-attack [27] or history-revision attack [26]. Eyal and Sirer propose an attack strategy that only requires about 1/3 of the total mining power,which is called "selfish mining" [28]. Furthermore based on the 'selfish mining' strategy, Gobel et al. study the effect of network delay on the evolution of the Bitcoin blockchain [29]. Sampolinsky and Zohar [30] show that at high throughput, the adversary can easily achieve double-spending attack.

Another series of studies have theoretically and rigorously analyzed the security of blockchain systems. Garay et al. present a rigorous cryptographic analysis on blockchain protocol [2]. Christian and Roger analyzed the broadcast of blocks and transactions in the Bitcoin network and believed that network delay was the main reason for the folks [4]. Pass et al. introduce the concept of network delay to their blockchain model and prove the security of blockchain consensus mechanism in an asynchronous network where the adversary have adaptive mining power and can create a short network delay [5]. Wei et al. relax the concept of common prefix property and chain growth to allow fractions of miners' chains being inconsistent and shows the security of blockchain protocol with long delay in

a honest miner setting [6]. After that they extends the situation to that the adversary own some mining power and proves that the properties of chain growth, common prefix, and chain quality still hold on some conditions [7].

3. Preliminaries

In this section, we review the simplified Nakamoto consensus which has been described in [2, 5, 6] and the dominant adversarial model [7].

3.1. Notation

A sequence of blocks compose a special chain which is called blockchain. Let B denotes a block and C denotes a blockchain. Then $C = \vec{B}$. Let m denotes the message in a block (e.g., in bitcoin protocol, message m denotes the transactions in that block). $\vec{m}$ denotes the all messages in a blockchain correspondingly. Each miners maintains its own separate chain at every moment. $H : \{0,1\}^* \to \{0,1\}^{256}$ is a cryptographic hash function(bitcoin mining uses SHA256 as hash function, its output is 256 bits).

3.2. Nakamoto consensus

The Nakamoto consensus consists of a pair of algorithms (Π^V, C). Π^V is a stateful algorithm which every miner runs at each round to maintain its local blockchain C. We take the Bitcoin system as an example. A block is a tuple, i.e., $B = (h_{-1}, m, r, h)$. h_{-1} is a hash pointer to the previous block. m denotes the transactions contained in the block. r is a nonce which is a 32-bit (4-byte) field in bitcoin. h is a hash pointer to the current block and $h = H(h_{-1}, m, r)$. Every miner's blockchain is initialized to a genesis block $B_0 = (0, \perp, 0, H(0\| \perp \|0))$. The second algorithm C takes C as input and outputs the messages $\vec{m}$ contained in C, i.e, $C(C) = \vec{m}$. V is an algorithm to check the validity of $\vec{m}$. In bitcoin, m consists of transactions and V is used to check the validity of transactions in the blockchain. If m is valid, $V(C(C)) = 1$.

In the Bitcoin system, a block $B = (h_{-1}, m, r, h)$ is valid with respect to a predecessor block $B_{-1} = (h'_{-1}, m', r', h')$ only if the conditions hold: $h_{-1} = h'$, $h = H(h_{-1}, m, r)$ and $h < D_p$, where D_p is the parameter to represent the mining difficulty. If all the blocks in a blockchain C are valid, and $V(C(C)) = 1$, the blockchain C is valid.

Time sequence in the Bitcoin system is regarded as discrete rounds. Each round represents a short period of time. At each round, each honest miner receives the blockchains and the transactions from the network, and runs Π^V to maintain its blockchain as follow:

- When receiving the blockchains from the network, pick up the valid and longest chain, say C'. If C' is longer than its local chain C, discard C and select C' as the new local chain C.
- Pack up the valid transactions as a message m, so that $V(C(C)\|m) = 1$. Then pick a nonce r randomly and compute h through the hash function H, $h = H(h_{-1}, m, r)$. If $h < D_p$, add block $B = (h_{-1}, m, r, h)$ to the end of the local blockchain C, which means a new block be mined and a new chain be created. Then broadcast the new chain. In a round, a miner can try mining for most q times.

In reality, the miners' mining power are different. That is, during a fixed time, with the great mining power can the miner try more times to mining. Furthermore, the total mining power of the whole

network always changes. But here, each miner is supposed to have the same mining power and the number of miners is fixed, for simplicity [5, 6]. Let p denotes the probability that a miner successfully mine a new block at a round, where $p = 1 - (1 - \frac{D_p}{2^{256}})^q \approx \frac{qD_p}{2^{256}}$.

The algorithm Π^V is simplified in [5,6] to facilitate their analysis. Since the probability of successful mining for a miner in a round is p, it is unnecessary to focus on how many times they mine in a round. So in their model, they suppose that a miner only do once mining in a round, and the probability of success is p. For more convenient to analyze, a little restrictions is made on honest miners [6, 7]. The honest miner cannot mine in a chain where the last block is mined by himself. In other words, once an honest miner succeeds in mining a new block, it will not continue to mine in the later rounds until it receives a longer chain and choose it as its local chain. This is reasonable because it is rare for a honest miner to mine two consecutive blocks because of the large number of miners and the small probability p. In the article, we follow the model setting described in [7].

3.3. Blockchain model with long network delay

We use the model proposed in [7] to analyze long delay attack against the common prefix of blockchain. The adversary not only control a part of miners (we call them corrupted miners), but also control the communications between honest miners. In total, there are $(1 + \mu)n$ miners (n honest miners and μn corrupted miners). At every round, the adversary executes as follow:

- Mining. Each corrupt miner obtains a valid blockchain from the adversary and try extending the chain with successful probability p. If the corrupted miner succeeds in extending the chain, the extended chain will be sent to the adversary.
- Receiving. The Adversary receives the chains from honest and corrupted miners and choose which chains she wants to delay. Each selected chain from honest miners can be successfully delayed with probability α. If the chain from the honest miner is successfully delayed, it can be delayed for Δ rounds at most. While the chains from the corrupted miners can always be delayed successfully for any rounds.
- Distribution. The Adversary can choose any chains from the corrupted miners to distribute. As to the chains from honest miners, if the chain is failed to be delayed or it has already been delayed for Δ rounds, the chain must be distributed at current round.

When a honest miner succeeds in mining a new block and the adversary fails to delay the new chain, we call the new chain *undelayable* chain.

After distribution, the current round ends. Meanwhile all honest miners will receive the distributed chains at the next round. According to [4], in bitcoin network, the time for a block to be known by all miners is about 10 s. So we define 10 seconds as a round. What's more, once the adversary distributes more than one chain at a round, she can make the honest miners to receive these chains in different order at the next round, so that she can create a fork among the honest miners. For example, at some round, the adversary distributes two chains, C_1 and C_2 with the same length. She sends (C_1C_2) to the honest miner i but (C_2C_1) to another honest miner j. Suppose C_1 and C_2 are longer than i and j's chains. Then, i will accept C_1 as its chain while j will accept C_2. As a result, a fork is created between the miner i and the miner j.

3.4. Common prefix

A blockchain is essentially a distributed shared database. Ideally, all the chains would remain consistent. However, the last few blocks of the blockchain in the entire network may be different due to network delay or malicious attacks by adversaries. This will jeopardize the security of blockchain and cause a series of risks and losses. Informally, the common prefix refers that all the miners' blockchains are the same except the last several blocks. It describes the consistency of the blockchain across the whole network.

4. Long delay attack on common prefix

Based on the above model, we present a concrete attack on common prefix of blockchain common prefix and its improved one.

Supposed the adversary's goal is to use network delay and the controlled computation power to create a fork (two blockchains with same length) in the network and try to maintain the extension of the two chains as long as possible. a is the number of the honest miners work on first chain, and n-a is the number of the honest miners who work on second one. n is the number of honest miners. The number of the corrupted miners controlled by the adversary is μn. The adversary also divides corrupted miners into two part, b corrupted miners working on the first chain and μn-b corrupted miners working on the second chain. Let A = $a + b$ denotes the number of the miners working on the first chain (marked as the first group), and B = n-$a + \mu n$-b denotes the number of the miners working on the second one (marked as the second group). Let $\gamma(n, p) = 1\text{-}(1\text{-}p)^n$, which denotes the probability that among n miners, at least one miner succeeds in mining a new block in a round. When p is very small, $\gamma(n, p) \approx np$.

4.1. The intuitive attack

Initially, the two groups of the honest miners have the same chains. When there exists new blocks mined at a round r, the adversary begins her operation to create a fork among the honest miners and extend the length of the fork as long as possible. Consider the following cases:

1) In each of the two groups, there is a least one miner succeeds in mining a new block. The means the chains of two groups both can be extended at the next round independently.
   The adversary first broadcasts the new chains. Then make the honest miners of the first group receive the chain created by their group earlier and the same to the honest miners of the second group. As a result, the adversary succeeds in extending the length of the fork by 1. The probability that first group mines a block is $\gamma(\mathrm{A}, p)$ and The probability that second group mines a block is $\gamma(\mathrm{B}, p)$. The probability that this case happens is

$$p_{case1} = \frac{\gamma(A, p) \cdot \gamma(B, p)}{\gamma(A + B, p)} \approx \frac{ABp}{A + B}. \tag{4.1}$$

2) Only the miners of the first group succeed in mining new blocks and at least one honest miner in first group succeeds in mining new block. The probability that this case happens is:

$$\frac{(1 - \gamma(B, p)) \cdot \gamma(a, p)}{\gamma(A + B, p)}. \tag{4.2}$$

Among the new chains, if there exist *undelayable* chain (each new chain with probability 1-α being it), those *undelayable* chains will be broadcast to all honest miners including the honest miners in the second group. The adversary fails extending the fork. All honest miners' chains become consistent and the length of the fork become 0. The probability is:

$$\frac{(1-\gamma(B,p))\cdot\gamma(a,(1-\alpha)p)}{\gamma(A+B,p)}. \tag{4.3}$$

On the other case, the adversary succeeds in delaying all the new chains. The probability of this case is:

$$\begin{aligned}&\frac{(1-\gamma(B,p))\cdot(\gamma(a,p)-\gamma(a,(1-\alpha)p))}{\gamma(A+B,p)}\\ &\approx\frac{(1-Bp)a\alpha}{A+B}.\end{aligned} \tag{4.4}$$

Then, the adversary can keeps these chains until a new block to be mined in the second group in the next Δ rounds. The corrupted miners of the first group don't need to mine until the adversary succeeds in increasing the length of the fork or fails it. At each of the following Δ rounds, there happen three cases:

(a) The second group succeeds in mining a block. Then the adversary broadcast the new chain and the delayed chain which is created by the first group before. Make the honest miners of the first group receive the chain created by their group earlier and the same to the honest miners of the second group. As a result, the adversary succeeds in extending the length of the fork by 1. The probability of the case is $\gamma(B,p)\approx Bp$.

(b) The second group fails to mine new blocks while the honest miners of first group mines a block which the adversary fails to delay. Then the new chain should be broadcast and all the honest miners will accept the new chain. So the adversary fails.

(c) The second group fails to mine new blocks, while the honest miners of the first miners don't create an *undelayable* chain. Then goes to the next round. The probability of the case is

$$\begin{aligned}p_{next}&=(1-\gamma(B,p))\cdot(1-\gamma(a,(1-\alpha)p))\\ &\approx(1-Bp)(1-a(1-\alpha)p).\end{aligned} \tag{4.5}$$

In a word, when the adversary succeeds in delaying all the new blocks, the adversary can succeed in extending the length of the fork with probability Bp at each of the following Δ rounds, or go to the next round with probability p_{next}. So the probability that adversary succeeds at round $r+1$ is Bp and the probability that adversary succeeds at round $r+2$ is $p_{next}\cdot Bp$. Similarly, the probability that the adversary succeeds in extending the length of the fork at round $r+i$ is $p_{next}^{i-1}\cdot Bp$. Since the adversary can only delay the chains for most Δ rounds, the total probability that adversary succeeds in these Δ rounds is

$$\sum_{i=1}^{\Delta}p_{next}^{i-1}\cdot Bp=\frac{Bp(1-p_{next}^{\Delta})}{1-p_{next}}. \tag{4.6}$$

To sum up, the probability that case 2) happens and the adversary succeeds in extending the length of the fork is

$$p_{case2s}=\frac{(1-Bp)a\alpha}{A+B}\cdot\frac{Bp(1-p_{next}^{\Delta})}{1-p_{next}}. \tag{4.7}$$

3) Only the miners of the second group succeed in mining new blocks and at least one honest miner in second group succeeds in mining a new block. Similar to the case 2), the the probability that case 3) happens and the adversary succeeds extending the length of the fork is

$$p_{case3s} = \frac{(1-Ap)(n-a)\alpha}{A+B} \cdot \frac{Ap(1-{p'_{next}}^{\Delta})}{1-p'_{next}}, \tag{4.8}$$

where $p'_{next} \approx (1-Ap)(1-(n-a)(1-\alpha)p)$.

4) Only the miners of the first group succeed mining new blocks and none of honest miners in first group succeed mining a new block, i.e., the new blocks are all mined by the corrupted miners in first group (they can be delayed definitely and for any rounds). The probability that the case happens is

$$\begin{aligned} p_{case4} &= \frac{(1-\gamma(B,p)) \cdot (1-\gamma(a,p)) \cdot \gamma(b,p)}{\gamma(A+B,p)} \\ &\approx \frac{(1-Bp)(1-ap)b}{A+B}. \end{aligned} \tag{4.9}$$

In the case, the corrupted miners of the first group don't need to mine as well until the adversary succeeds or fails. To successfully extend the fork at later round, the adversary needs the miners of the second group successfully mining new blocks no later than the new blocks being broadcast which are mined by the honest miners in the first group (the new blocks are broadcast immediately with probability $1-\alpha$ or delayed for Δ rounds with probability α). Otherwise the adversary fails and all honest miners' chains will become consistent. In the case, the corrupted miners in first group don't need to mine until the adversary succeeds or fails.

In the later rounds, the probability p_1, that the honest miners of the first group create *undelayable* chains, is $\gamma(a,(1-\alpha)p) \approx a(1-\alpha)p$. The probability p_2, that the miners of the second group successfully mines new blocks in a round, is $\gamma(B,p) \approx Bp$. The probability p_3, that the honest miners of the first group successfully mine new blocks in a round, is $\gamma(a,p) \approx ap$.

Firstly consider the situation of the round $r+1$ to round $r+\Delta$. At round $r+1$, if the second group succeeds mining new block, then the adversary can extend the length of the fork between the blockchains of the two groups. The probability that the adversary succeeds at round $r+1$ is p_2. If the second group fails to mine new block and the honest miners of the first group don't create *undelayable* chains (i.e. at the round, the adversary succeeds in delaying all new blocks mined by the honest miners of the first group or the honest miners of the first group fail to mine new blocks), then goes to the next round. The probability that goes to the next round is $(1-p_1)(1-p_2)$. So the probability that adversary succeeds extending the fork by 1 at $r+2$ is $(1-p_1)(1-p_2)p_2$. Similarly, the the probability that adversary succeeds at $r+i$ is $((1-p_1)(1-p_2))^{i-1}p_2$. The total probability that the adversary succeeds from round $r+1$ to round $r+\Delta$ is

$$\begin{aligned} p_{s1} &= \sum_{i=1}^{\Delta} ((1-p_1)(1-p_2))^{i-1} \cdot p_2 \\ &= \frac{p_2(1-((1-p_1)(1-p_2))^{\Delta})}{1-(1-p_1)(1-p_2)}. \end{aligned} \tag{4.10}$$

Then consider the situation of the round $r+\Delta+1$ to infinite round. To go to the round $r+\Delta+1$, the following two conditions need to be met.

- At round $r+1$, both group fail to mine new blocks (with probability $(1-p_3)(1-p_2)$).
- From the round $r+2$ to the $r+\Delta$, the first group fails to create a *undelayable* chain and the second group fails to mine new blocks (with probability $(1-p_1)^{\Delta-1}(1-p_2)^{\Delta-1}$).

Then if the second group succeeds mining new block (with probability p_2), the adversary can succeed to extending the fork. The probability that the adversary succeeds at round $r+\Delta+1$ is $(1-p_3)(1-p_1)^{\Delta-1}(1-p_2)^{\Delta}p_2$.
Similarly, if the adversary succeeds at round $r+\Delta+2$, she needs that

- In round $r+1$ and $r+2$ both groups fail to mine new blocks (with probability $(1-p_3)^2(1-p_2)^2$).
- From the round $r+3$ to the $r+\Delta+1$ the first group fails to create a *undelayable* chain and the second group fails to mine new blocks (with probability $(1-p_1)^{\Delta-1}(1-p_2)^{\Delta-1}$).
- The second group succeeds mining new blocks at round $r+\Delta+2$ (with probability p_2).

The probability is $(1-p_3)^2(1-p_1)^{\Delta-1}(1-p_2)^{\Delta+1}p_2$. We can conclude that the probability that the adversary succeeds at round $r+\Delta+i$ is $(1-p_3)^i(1-p_1)^{\Delta-1}(1-p_2)^{\Delta+i-1}p_2$.
To sum up, the total probability that adversary succeeds from round $r+\Delta+1$ to infinite round is

$$\begin{aligned} p_{s2} &= \sum_{i=1}^{\infty}(1-p_3)^i(1-p_1)^{\Delta-1}(1-p_2)^{\Delta+i-1}p_2 \\ &= \frac{(1-p_3)(1-p_1)^{\Delta-1}(1-p_2)^{\Delta}p_2}{1-(1-p_3)(1-p_2)}. \end{aligned} \tag{4.11}$$

To sum up, We can conclude the probability that the case 4) happens and the adversary succeeds extending the length of the fork is

$$p_{case4s} = p_{case4} \cdot (p_{s1} + p_{s2}). \tag{4.12}$$

5) Only the miners of the second group succeed mining new blocks and none of honest miners in second group succeed mining a new block, i.e., the new blocks are all mined by the corrupted miners in second group(they can be delayed definitely and for any rounds). Similarly to case 4), the probability that the case 5) happens and the adversary succeeds extending the length of the fork is

$$p_{case5s} = p_{case5} \cdot (p'_{s1} + p'_{s2}), \tag{4.13}$$

where $p_{case5} = \frac{(1-Ap)(1-(n-a)p)(\mu-b)}{A+B}$, $p'_{s1} = \frac{p'_2(1-((1-p'_1)(1-p'_2))^{\Delta})}{1-(1-p'_1)(1-p'_2)}$, $p'_{s2} = \frac{(1-p'_3)(1-p'_1)^{\Delta-1}(1-p'_2)^{\Delta}p'_2}{1-(1-p'_3)(1-p'_2)}$, $p'_1 = (n-a)(1-\alpha)p$, $p'_2 = Ap$ and $p'_3 = (n-a)p$.

Considering the all cases above, when there is at least one miner succeeds mining a new block, the total probability that the adversary succeeds extending the length of the fork between the two chains by 1 is

$$p_{success} = p_{case1} + p_{case2s} + p_{case3s} + p_{case4s} + p_{case5s}. \tag{4.14}$$

Then the adversary waits for new block being mined and continues executing as described above. If the adversary's goal is to extend the length of the fork to T, the probability that she succeeds is $p_{success}^T$.

4.2. The improved attack

Most the blockchain protocols follow "the longest chain" principle as well as the bitcoin protocol. That means the honest miners always accept the longer chain and work on it while the corrupted miners controlled by the adversary don't need execute like that. The adversary can dynamically have any corrupted miners working on any chain.

In the initial method, when only one group succeeds mining new blocks at a round, the adversary make the corrupted miners in that group out of work to wait new blocks mined by the other group. This wastes a part of mining power that could be used for a while.

Considering that, when only one group succeeds mining new blocks at a round, the adversary can make the corrupted miners of that group work on the chain of the other group temporarily until the new blocks mined in the other group. Then the adversary can extend the length of the fork, and these corrupted miners work back on the chain of the origin group. In the attack method, the probability that the adversary succeeds gets promoted. Specific analysis is as follow:

1) In each of the two groups, there is a least one miner succeeds mining a new block. This case is the same to the case 1) of the initial attack method. When the case happens, the adversary can succeed extending the length of the fork by 1. The probability that the case happens is

$$p_{case1} = \frac{\gamma(A,p)\cdot\gamma(B,p)}{\gamma(A+B,p)} \approx \frac{ABp}{A+B}. \tag{4.15}$$

2) Only the miners of the first group succeed mining new blocks and at least one honest miner in first group succeeds mining new block.
In the case, only when the adversary succeeds delaying all the new chains mined in the round (with probability about $\frac{(1-Bp)a\alpha}{A+B}$ discussed above), she can go to the next round and has the chance to extend the length of the fork. Then the adversary can have the corrupted miners of the first group working on the chain of the second group. So the probability of case (a) becomes $\gamma(B+b,p) \approx (B+b)p$ and the probability that goes to the next round in case (c) becomes

$$\begin{aligned} q_{next} &= (1-\gamma(B+b,p))\cdot(1-\gamma(a,(1-\alpha)p)) \\ &\approx (1-(B+b)p)(1-a(1-\alpha)p). \end{aligned} \tag{4.16}$$

To sum up, the total probability that adversary succeeds in later Δ rounds is

$$\sum_{i=1}^{\Delta} q_{next}^{i-1}\cdot(B+b)p = \frac{(B+b)p(1-q_{next}^{\Delta})}{1-q_{next}}. \tag{4.17}$$

To sum up, in the improved attack methods, the probability that case 2) happens and the adversary succeeds extending the length of the fork is

$$q_{case2s} = \frac{(1-Bp)a\alpha}{A+B}\cdot\frac{(B+b)p(1-q_{next}^{\Delta})}{1-q_{next}}. \tag{4.18}$$

3) Only the miners of the second group succeed mining new blocks and at least one honest miner in second group succeeds mining a new block.

Similar to the case 2), in improved attack method, the probability q_{case3s}, that the case 3) happens and the adversary succeeds is

$$q_{case3s} = \frac{(1-Ap)(n-a)\alpha}{A+B} \cdot \frac{(A+\mu n-b)p(1-{q'_{next}}^{\Delta})}{1-q'_{next}}, \tag{4.19}$$

where $q'_{next} \approx (1-(A+\mu n-b)p)(1-(n-a)(1-\alpha)p)$.

4) Only the miners of the first group succeed mining new blocks and none of honest miners in the first group succeed mining a new block, i.e., the new blocks are all mined by the corrupted miners in first group (they can be delayed definitely and for any rounds).
Then the adversary have the corrupted miners of the first group working on the chain of the second group. So the probability, that the miners of the second group successfully mines new blocks in a round becomes $q_2 = \gamma(B+b,p) \approx (B+b)p$, while the probability p_1, that the honest miners of the first group create *undelayable* chain and the probability p_3, that the honest miners of the first group successfully mine new blocks in a round don't change. Similarly, we can deduce the probability that the case 4) happens and adversary succeeds is

$$q_{case4s} = p_{case4} \cdot (q_{s1} + q_{s2}), \tag{4.20}$$

where $q_{s1} = \frac{q_2(1-((1-p_1)(1-q_2))^{\Delta})}{1-(1-p_1)(1-q_2)}$ and $q_{s2} = \frac{(1-p_3)(1-p_1)^{\Delta-1}(1-q_2)^{\Delta}q_2}{1-(1-p_3)(1-q_2)}$.

5) Only the miners of the second group succeed mining new blocks and none of honest miners in the second group succeed mining a new block, i.e., the new blocks are all mined by the corrupted miners in second group (they can be delayed definitely and for any rounds).
Similarly to case 4), in the improved attack method, the probability that the case 5) happens and the adversary succeeds extending the length of the fork is

$$q_{case5s} = p_{case5} \cdot (q'_{s1} + q'_{s2}), \tag{4.21}$$

where $q'_{s1} = \frac{q'_2(1-((1-p'_1)(1-q'_2))^{\Delta})}{1-(1-p'_1)(1-q'_2)}$, $q'_{s2} = \frac{(1-p'_3)(1-p'_1)^{\Delta-1}(1-q'_2)^{\Delta}q'_2}{1-(1-p'_3)(1-q'_2)}$ and $q'_2 = (A+\mu n-b)p$.

Considering the cases above in improved attack method, when there is at least one miner succeeds mining a new block, the total probability that the adversary succeeds extending the length of the fork between the two chains by 1 is

$$q_{success} = p_{case1} + q_{case2s} + q_{case3s} + q_{case4s} + q_{case5s}. \tag{4.22}$$

If the adversary's goal is to extend the length of the fork to T, the probability that she succeeds is $q^T_{success}$.

5. Experiments

Through experiments, we would like to investigate: 1) How divide the miners can the adversary to succeed more easily. 2) The difference between our methods and the methods proposed by [6]. So given the parameters, we calculated the probabilities obtain from all combinations of a and b, and plot figures to show it. Furthermore we use Kubernets (K8S) and run Bitcoin Core on the regtest mode to simulate the long delay attack.

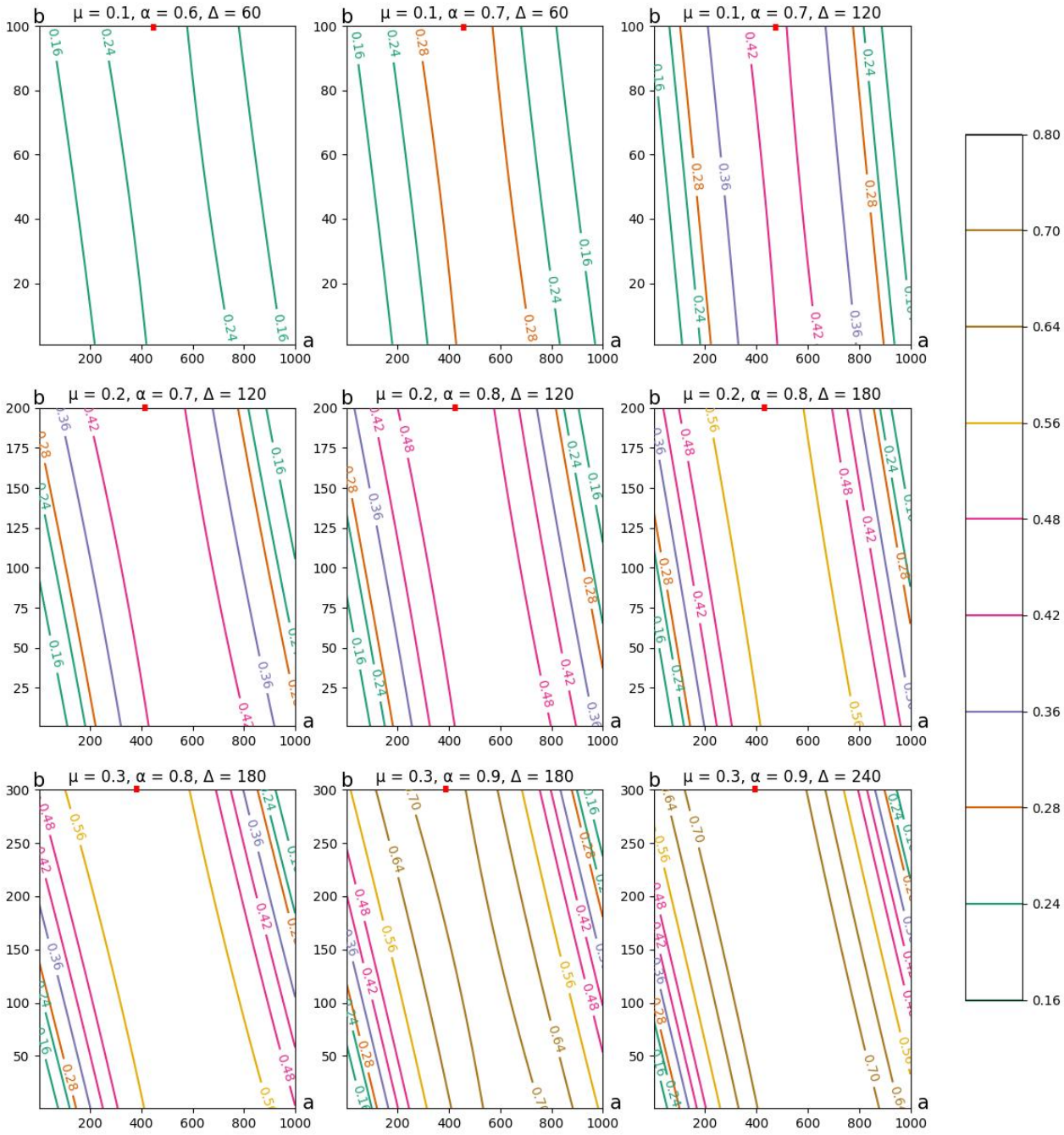

Figure 1. The probability of success for different values of a and b in initial method.

5.1. The impact of the division on probability

For an experimental interpretation of the success probability of the attack, the parameters are set as follows: The time span of a round is set to 10 seconds which is approximate to the time that a new block is known by all the miners in reality [4]. Since in bitcoin protocol, the expected time for a block

to be mined is 10 minutes, the probability that, at least one miner mines a new block in a round is $f = 1 - (1 - p)^{n+\mu n} \approx (n + \mu n)p = 1/60$. So we have $p = \frac{1}{60(n+\mu n)}$.

Consider there are $n = 1000$ honest miners in the network. Figures 1 and 2 show relationship between the probabilities of success and a and b under different parameter groups.

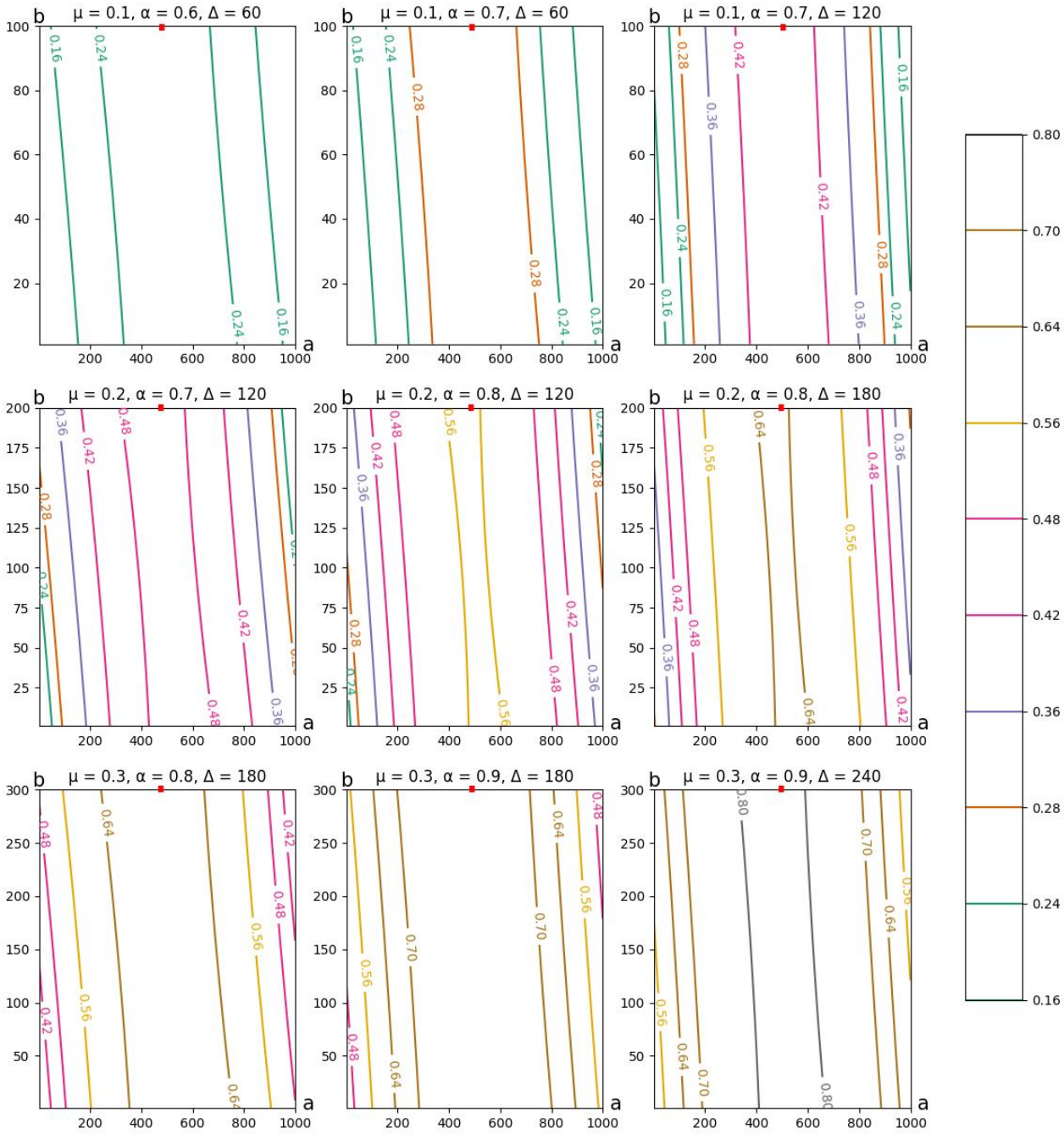

Figure 2. The probability of success for different values of a and b in improved method.

As the ability of the adversary gets promoted (i.e., μ,α and Δ get larger), she can more easily succeed extending the length of the fork when there is a new block mined. Moreover, the improved attack method indeed performs better than the initial one. When the number of the first group $a + b$ is closed to half of the number of the total miners $\frac{n+\mu n}{2}$, the probability gets higher. There is consistent with our intuitive cognition that it is easier for the adversary to maintain a fork between two groups of equal number. We mark the point with the highest probability in red. Under all parameter groups, the highest probabilities are obtained from $b = \mu n$ and the value of a closing to $\frac{n+\mu n}{2} - b$. While $a = n/2$ and $b = \mu n/2$ can still gets a very high probability as shown in Tables 1 and 2. We call the division that gets the highest probability of success best division and call the division, $a = n/2$ and $b = \mu n/2$, equal division.

To further investigate the differences between the two divisions, Tables 1 and 2 also show every probabilities of success of five cases analysed above of two divisions. We round the probability to six decimal places.

Table 1. Probabilities of success of two divisions under different parameter groups of initial attack method.

μ	α	Δ	a	b	p_{case1}	p_{case2s}	p_{case3s}	p_{case4s}	p_{case5s}	$p_{success}$
0.1	0.6	60	415	100	0.004150	0.086864	0.108012	0.061059	0	0.260085
			$n/2$	$\mu n/2$	0.004167	0.098493	0.098493	0.028106	0.028106	0.257365
0.1	0.7	60	424	100	0.004157	0.103920	0.128878	0.061915	0	0.298870
			$n/2$	$\mu n/2$	0.004167	0.117220	0.117220	0.028975	0.028975	0.296557
0.1	0.7	120	440	100	0.004165	0.161377	0.195878	0.066216	0	0.427636
			$n/2$	$\mu n/2$	0.004167	0.179115	0.179115	0.031918	0.031918	0.426233
0.2	0.7	120	380	200	0.004162	0.131434	0.196573	0.128135	0	0.460304
			$n/2$	$\mu n/2$	0.004167	0.165665	0.165665	0.059758	0.059758	0.455013
0.2	0.8	120	389	200	0.004165	0.156192	0.233078	0.131527	0	0.524962
			$n/2$	$\mu n/2$	0.004167	0.195715	0.195715	0.062443	0.062443	0.520483
0.2	0.8	180	398	200	0.004167	0.190561	0.277056	0.136905	0	0.608689
			$n/2$	$\mu n/2$	0.004167	0.234694	0.234694	0.065981	0.065981	0.605517
0.3	0.8	180	348	300	0.004167	0.155944	0.277026	0.195191	0	0.632328
			$n/2$	$\mu n/2$	0.004167	0.218141	0.218141	0.092572	0.092572	0.625593
0.3	0.9	180	355	300	0.004166	0.183354	0.327219	0.201714	0	0.716453
			$n/2$	$\mu n/2$	0.004167	0.255930	0.255930	0.097431	0.097431	0.710889
0.3	0.9	240	360	300	0.004166	0.205113	0.357971	0.207088	0	0.774338
			$n/2$	$\mu n/2$	0.004167	0.282242	0.282242	0.100855	0.100855	0.770361

In both attack method, equal division gets a little higher p_{case1} due to the number of two groups are the same and it also gets higher $p_{case2s} + p_{case3s}$ and $q_{case2s} + q_{case3s}$. But the part of the impact on total probability is minimal as long as the number of two groups are close.

The best division get a much higher $p_{case4s} + p_{case5s}$ and $q_{case4s} + q_{case5s}$. By dividing all the corrupted miners in the first group, case 5) will not happen while the probability that case 4) happens get promoted a lot which compensates for the loss of the probability of case 5). This finally helps it

get a much higher p_{case4s} and q_{case4s} and make its total probability of success exceeds equal division's.

Table 2. Probabilities of success of two divisions under different parameter groups of improved attack method.

μ	α	Δ	a	b	p_{case1}	q_{case2s}	q_{case3s}	q_{case4s}	q_{case5s}	$q_{success}$
0.1	0.6	60	446	100	0.004166	0.101100	0.107591	0.062422	0	0.275279
			$n/2$	$\mu n/2$	0.004167	0.105334	0.105334	0.029176	0.029176	0.273187
0.1	0.7	60	455	100	0.004166	0.121090	0.127971	0.063459	0	0.316686
			$n/2$	$\mu n/2$	0.004167	0.125340	0.125340	0.030062	0.030062	0.314971
0.1	0.7	120	471	100	0.004161	0.183612	0.192549	0.067828	0	0.448150
			$n/2$	$\mu n/2$	0.004167	0.188556	0.188556	0.032932	0.032932	0.447143
0.2	0.7	120	444	200	0.004144	0.169468	0.189898	0.131733	0	0.495243
			$n/2$	$\mu n/2$	0.004167	0.181082	0.181082	0.062910	0.062910	0.492151
0.2	0.8	120	454	200	0.004133	0.202055	0.223190	0.135699	0	0.565077
			$n/2$	$\mu n/2$	0.004167	0.213697	0.213697	0.065580	0.065580	0.562721
0.2	0.8	180	462	200	0.004122	0.238205	0.261733	0.140613	0	0.644673
			$n/2$	$\mu n/2$	0.004167	0.250737	0.250737	0.068662	0.068662	0.642965
0.3	0.8	180	444	300	0.004080	0.218541	0.254514	0.200280	0	0.677415
			$n/2$	$\mu n/2$	0.004167	0.237835	0.237835	0.097289	0.097289	0.674415
0.3	0.9	180	457	300	0.004054	0.260241	0.294589	0.207480	0	0.766364
			$n/2$	$\mu n/2$	0.004167	0.278239	0.278239	0.101880	0.101880	0.764405
0.3	0.9	240	461	300	0.004045	0.281148	0.317483	0.211591	0	0.814267
			$n/2$	$\mu n/2$	0.004167	0.299928	0.299928	0.104378	0.104378	0.812779

As shown in Tables 1 and 2, though equal division is not the one that get the highest probability of success, the difference in probability between two divisions is very small.

5.2. Experiment on bitcoin regtest mode through Kubernets

5.2.1. Bitcoin regtest mode

Bitcoin usually refers to Bitcoin Core that runs on mainnet mode. In addition, Bitcoin provides several modes for testing and development such as testnet, segnet and regteset. Regtest mode are designed to operate as a closed system for local testing. In the mode, we can easily create a brand-new private block chain with the same basic rules as mainnet.

5.2.2. Kubernets

Kubernetes (K8s) is an open source Linux container automation operation and maintenance platform, used to manage containerized applications on multiple hosts. The goal of Kubernetes is to make deployment containerized applications simple and powerful. Kubernetes provides a mechanism for application deployment, planning, updating, and maintenance.

The traditional way to deploy applications is to install applications through plug-ins or scripts. The disadvantages of the method are that the operation, configuration, management, and life cycle of

applications are bound to the current operating system, which is not conducive to the upgrade, update, or rollback of applications. Of course, we can also create VMS to implement certain functions, but VMS are very heavy, which is not conducive to portability.

Table 3. Experimental results of three long delay attacks.

Method	(α, Δ)	The number of attempts	The number of success	Frequency
Initial attack	(0.6,60)	1074	301	0.280261
	(0.7,60)	998	344	0.344689
	(0.7,120)	893	410	0.459127
	(0.8,120)	836	433	0.517942
	(0.8,180)	731	443	0.606019
	(0.9,180)	700	483	0.690000
	(0.9,240)	670	517	0.771642
Improved attack	(0.6,60)	1107	361	0.326107
	(0.7,60)	1060	383	0.361321
	(0.7,120)	869	411	0.472957
	(0.8,120)	863	497	0.575898
	(0.8,180)	805	524	0.650932
	(0.9,180)	733	540	0.736698
	(0.9,240)	714	573	0.802521
Attack proposed by Wei et al.	(0.6,60)	1169	244	0.208725
	(0.7,60)	1064	257	0.241541
	(0.7,120)	940	366	0.389361
	(0.8,120)	856	387	0.452103
	(0.8,180)	796	454	0.570352
	(0.9,180)	702	444	0.632479
	(0.9,240)	692	469	0.677746

The new approach is implemented through the deployment of containers, each container isolated from each other, each container has its own file system, processes between containers do not affect each other, can distinguish computing resources. Compared to virtual machines, containers can be deployed quickly. Because containers are decoupled from the underlying infrastructure and machine file systems, they can be migrated between different clouds and operating systems of different versions.

Containers occupy fewer resources and are quickly deployed. Each application can be packaged into a container image. The one-to-one relationship between each application and the container also gives the container a greater advantage. Using containers, you can create container images for the application at the build or release stage, because each application does not need to be combined with the rest of the application stack and is not dependent on the production infrastructure. Similarly, containers are lighter and more ”transparent” than virtual machines, which makes them easier to monitor and manage.

5.2.3. Experimental environment

30 physical machines are used to conduct the experiment, each of which with CPU of 4 cores, 500 GB storage, 8 GB memory. In terms of software, centos7.9 and K8S are installed on each machine.

All the machines form a cluster via K8S and LAN. After that, one machine is picked up as the master node that is responsible for executing commands to manipulate containers and scheduling containers to work on other machines.

5.2.4. Experimental results

Wei et al. introduce a long delay attack on the prefix of blockchain int the network. In their model, the adversary can control the communication of the net work and has no mining power. Their attack creates and maintains a fork in two groups of equally honest miners. In experiment, we take their attack as a comparison and choose equal division for our attack methods.

We deploy 120 containers that runs Bitcoin Core on regtest mode. In simulating our attack, we take 100 containers as honest miners and 20 containers as corrupted miners that are controlled by the adversary. The goal of the adversary is to create a fork in two groups of 50 honest miners each. While in simulating [6]'s attack method, all 120 containers is honest miners and the goal of the adversary is to create a fork in two groups of 60 honest miners each. We choose seven parameter groups (α, Δ), and run 100,000 rounds for each experiment. Note that only when there is at least a new block mined in a round, the adversary can start her attack. Moreover, the entire attack takes many rounds. So the number of attempts by the adversary is far less than 100,000.

As shown in Table 3, under the same experimental setup, our attack behavior is better than the attack proposed in [6]. This is consistent with intuition. When the adversary owns extra mining power in the network, she has higher success probability in creating a fork and extending the length of the fork between two groups of the honest miners. Moreover, the number of attempts by the adversary decreases with the increasing of α and Δ, because it gives the adversary more opportunity to achieve each of her attack, after given the total rounds.

6. Conclusions

In this paper we introduce a long delay attack and improved it under the model that the adversary can control network communication and has her own mining power. Through the attack, an adversary can easily achieves malicious acts such as double spending. We analyze the success probability of the proposed attack, and compute the optimal division to create and maintain a fork as long as possible among the honest miners. Finally, we simulate the proposed attack and compare it with the attack proposed in [6].

Acknowledgments

This work is supported by National Key Research and Development Program of China under the grant No. 2021YFB2701200, National Natural Science Foundation of China (NSFC) under the grant No. 62172040, National Natural Science Foundation of China No. U1836212 and No. 61872041, and National Key Research and Development Program of China No. 2021YFB2701104.

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, *Decentralized Bus. Rev.*, (2008), 21260. Available from: https://people.eecs.berkeley.edu/ raluca/cs261-f15/readings/bitcoin.pdf.
2. J. Garay, A. Kiayias, N. Leonardos, The bitcoin backbone protocol: Analysis and applications, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, **9057** (2015), 281–310. https://doi.org/10.1007/978-3-662-46803-6_10
3. C. Dwork, N. Lynch, L. Stockmeyer, Consensus in the presence of partial synchrony, *J. ACM*, **35** (1988), 288–323. https://doi.org/10.1145/42282.42283
4. C. Decker, R. Wattenhofer, Information propagation in the bitcoin network, in *IEEE P2P 2013 Proceedings*, IEEE, (2013), 1–10. https://doi.org/10.1109/P2P.2013.6688704
5. R. Pass, L. Seeman, A. Shelat, Analysis of the blockchain protocol in asynchronous networks, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, **10211** (2017), 643–673. https://doi.org/10.1007/978-3-319-56614-6_22
6. P. Wei, Q. Yuan, Y. Zheng, Security of the blockchain against long delay attack, in *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, **11274** (2018), 250–275. https://doi.org/10.1007/978-3-030-03332-3_10
7. Q. Yuan, P. Wei, K. Jia, H. Xue, Analysis of blockchain protocol against static adversarial miners corrupted by long delay attackers, *Sci. China Inf. Sci.*, **63** (2020), 1–15. https://doi.org/10.1007/s11432-019-9916-5
8. L. Shi, T. Wang, J. Li, S. Zhang, Pooling is not favorable: Decentralize mining power of pow blockchain using age-of-work, preprint, arXiv:2104.01918.
9. C. Li, B. Palanisamy, Comparison of decentralization in dpos and pow blockchains, *Lect. Notes Comput. Sci.*, **12404** (2020). https://doi.org/10.1007/978-3-030-59638-5_2
10. J. Bhosale, S. Mavale, Volatility of select crypto-currencies: A comparison of bitcoin, ethereum and litecoin, *Annu. Res. J. SCMS, Pune*, **6** (2018). Available from: https://www.scmspune.ac.in/journal/pdf/current/Paper%2010%20-%20Jaysing%20Bhosale.pdf.
11. H. Chen, M. Pendleton, L. Njilla, S. Xu, A survey on ethereum systems security: Vulnerabilities, attacks, and defenses, *ACM Comput. Surv.*, **53** (2020), 1–43. https://doi.org/10.1145/3391195
12. R. Pass, A. Shelat, Micropayments for decentralized currencies, in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, (2015), 207–218. https://doi.org/10.1145/2810103.2813713
13. J. Poon, T. Dryja, The bitcoin lightning network: Scalable off-chain instant payments, 2016. Available from: https://lightning.network/lightning-network-paper.pdf.

14. K. Gai, Y. Wu, L. Zhu, M. Qiu, M. Shen, Privacy-preserving energy trading using consortium blockchain in smart grid, *IEEE Trans. Ind. Inf.*, **15** (2019), 3548–3558. https://doi.org/10.1109/TII.2019.2893433

15. L. Zhu, Y. Wu, K. Gai, K. Choo, Controllable and trustworthy blockchain-based cloud data management, *Future Gener. Comput. Syst.*, **91** (2018), 527–535. https://doi.org/10.1016/j.future.2018.09.019

16. K. Gai, Y. Wu, L. Zhu, Z. Zhang, M. Qiu, Differential privacy-based blockchain for industrial internet-of-things, *IEEE Trans. Ind. Inf.*, **16** (2020), 4156–4165. https://doi.org/10.1109/TII.2019.2948094

17. G. Xu, J. Dong, C. Ma, J. Liu, U. G. O. Cliff, A certificateless signcryption mechanism based on blockchain for edge computing, *IEEE Internet Things J.*, **2022** (2022). https://doi.org/10.1109/JIOT.2022.3151359

18. I. Bentov, R. Kumaresan, How to use bitcoin to design fair protocols, *Lect. Notes Comput. Sci.*, Springer, Berlin, Heidelberg, **8617** (2014), 421–439. https://doi.org/10.1007/978-3-662-44381-1_24

19. K. Gai, J. Guo, L. Zhu, S. Yu, Blockchain meets cloud computing: A survey, *IEEE Commun. Surv. Tutorials*, **22** (2020), 2009–2030. https://doi.org/10.1109/COMST.2020.2989392

20. F. Yang, J. Tian, T. Feng, F. Xu, C. Qiu, C. Zhao, Blockchain-enabled parallel learning in industrial edge-cloud network: A fuzzy dpost-pbft approach, in *2021 IEEE Globecom Workshops (GC Wkshps)*, (2021), 1–6. https://doi.org/10.1109/GCWkshps52748.2021.9681977

21. G. Xu, Y. Liu, P. W. Khan, Improvement of the dpos consensus mechanism in blockchain based on vague sets, *IEEE Trans. Ind. Inf.*, **16** (2020), 4252–4259. https://doi.org/10.1109/TII.2019.2955719

22. Y. Liu, G. Xu, Fixed degree of decentralization dpos consensus mechanism in blockchain based on adjacency vote and the average fuzziness of vague value, *Comput. Networks*, **199** (2021), 108432. https://doi.org/10.1016/j.comnet.2021.108432

23. K. M. Elleithy, D. Blagovic, W. Cheng, P. Sideleau, Denial of service attack techniques: Analysis, implementation and comparison, *J. Syst. Cybern. Inf.*, **3** (2005), 66–71. Available from: https://www.researchgate.net/publication/242497142.

24. E. Heilman, A. Kendler, A. Zohar, S. Goldberg, Eclipse attacks on bitcoin's peer-to-peer network, in *Proceedings of the 24th USENIX Conference on Security Symposium*, (2015), 129–144. Available from: https://eprint.iacr.org/2015/263.pdf.

25. M. Apostolaki, A. Zohar, L. Vanbever, Hijacking bitcoin: Routing attacks on cryptocurrencies, in *2017 IEEE Symposium on Security and Privacy (SP)*, IEEE, **2017** (2017), 375–392. https://doi.org/10.1109/SP.2017.29

26. J. Bonneau, E. W. Felten, S. Goldfeder, J. A. Kroll, A. Narayanan, Why buy when you can rent? Bribery attacks on bitcoin consensus. Available from: https://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=C77871712694860A06EFCEC0665234B6?doi=10.1.1.698.738&rep=rep1&type=pdf.

27. G. O. Karame, E. Androulaki, S. Capkun, Double-spending fast payments in bitcoin, in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, (2012), 906–917. https://doi.org/10.1145/2382196.2382292

28. I. Eyal, E. G. Sirer, Majority is not enough: Bitcoin mining is vulnerable, *Lect. Notes Comput. Sci.*, Springer, Berlin, Heidelberg, **8437** (2013). https://doi.org/10.1007/978-3-662-45472-5_28

29. J. Göbel, H. P. Keeler, A. E. Krzesinski, P. G. Taylor, Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay, *Perform. Eval.*, **104** (2016), 23–41. https://doi.org/10.1016/j.peva.2016.07.001

30. Y. Sompolinsky, A. Zohar, Secure high-rate transaction processing in bitcoin, in *International Conference on Financial Cryptography and Data Security*, Springer, Berlin, Heidelberg, **8975** (2015), 507–527. https://doi.org/10.1007/978-3-662-47854-7_32

AIMS Press