cryptography

MDPI

Article

Research on PoW Protocol Security under Optimized Long Delay Attack

Tao Feng * and Yufeng Liu

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China
* Correspondence: fengt@lut.edu.cn; Tel.: +86-136-6939-6157

Abstract: In the blockchain network, the communication delay between different nodes is a great threat to the distributed ledger consistency of each miner. Blockchain is the core technology of Bitcoin. At present, some research has proven the security of the PoW protocol when the number of delay rounds is small, but in complex asynchronous networks, the research is insufficient on the security of the PoW protocol when the number of delay rounds is large. This paper improves the proposed blockchain main chain record model under the PoW protocol and then proposes the T_{OD} model, which makes the main chain record in the model more close to the actual situation and reduces the errors caused by the establishment of the model in the analysis process. By comparing the differences between the T_{OD} model and the original model, it is verified that the improved model has a higher success rate of attack when the probability of mining the delayable block increases. Then, the long delay attack is improved on the balance attack in this paper, which makes the adversary control part of the computing power and improves the success rate of the adversary attack within a certain limit.

Keywords: blockchain; chain record model; security; long delay attack

Citation: Feng, T.; Liu, Y. Research on PoW Protocol Security under Optimized Long Delay Attack. *Cryptography* **2023**, *7*, 32. https://doi.org/10.3390/cryptography7020032

Academic Editor: Kentaroh Toyoda

Received: 22 March 2023
Revised: 6 June 2023
Accepted: 7 June 2023
Published: 16 June 2023

1. Introduction

In recent years, with the continuous research on blockchain smart contracts [1], distributed ledgers [2], and other related cryptographic technology [3], more and more people have noticed the importance of maintaining the consistency of distributed ledgers. For example, the blockchain protocol proposed by the authors in paper [4], using neural networks and machine learning algorithms, guarantees the confidentiality of the transaction phase and peer-to-peer data exchange, completes consensus in the shortest possible time, and effectively avoids 51% of attacks. In blockchain systems, maintaining the consistency of the distributed ledger in the face of complex delay environments caused by hardware devices, network communications, and human interference is a security issue that blockchains must face. For example, the eclipse attack [5] and the sybil attack [6] can restrict the communication between nodes, resulting in forking or double spending. However, in the actual P2P network environment, the success rate of the delayed attacks carried out by the adversary is not 100%, and the adversary delays the broadcast information for at most Δ rounds ($\Delta \ll 1/np$, Δ is an integer relating n and p, where n is the number of miner nodes and p is the probability of successfully mining the block, when the number of nodes in the network is high and mining is difficult, the p is relatively small). How to effectively reduce the impact caused by delay in the blockchain can be mainly solved by optimizing the original consensus mechanism, such as by verifying the number of nodes [7], changing the method of adding new blocks [8,9], etc. Some studies also pointed out that the current PoW can resist a certain delay attack [10–12]. However, most of these studies focus on the maximum delay round number $\Delta \ll 1/np$. Although some studies [13] have pointed out that the PoW protocol still provides good security when $\Delta > 1/np$, the Tree_{MC} model proposed by the authors is too idealistic, and when the number of forks is large in the model, a large amount of node information cannot be recorded on the model, resulting in errors. In addition, in

the authors' proposed model for long delay attacks against the PoW protocol, the authors set the adversary to only control the communication to indicate that the PoW protocol has some security, which does not clearly reflect the complex delay situation. This paper makes improvements to solve these problems. The main contributions of this research include:

- Optimized the main chain record model based on the original research so that the improved record model can simulate the evolution of the main chain on the blockchain accurately.
- Improved the original long delay attack model, combined it with the balanced attack where the adversaries can control a certain number of corrupted miners, and proposed the improved long delay attack model, which made the improved attack model more real and improved the success probability of the adversary attack.
- Based on the above research, this paper analyzes the security of the PoW protocol in a complex latency environment and shows that the PoW protocol still has good security in a complex delay environment.

2. Related Work

Existing work ranges from reducing the risk to chain security in latency environments by improving consensus mechanisms to proving the security of blockchain PoW protocols in theoretical latency environments. Sirer and Eyal put forward the concept of "selfish mining" [14]. The authors believe that the reward mechanism of Bitcoin has defects, which may weaken the decentralized characteristics of the blockchain. The adversary only needs to control about 33% of the computing power to attack by delaying the release of the broadcast, and at the same time, selfish mining will cause a lot of waste of resources. Later, in order to solve the delay problem of real-time payment, the authors proposed the Bitcoin-NG method [15] to shorten the time of block transaction confirmation. This method can shorten the block confirmation time and alleviate the problems about chain forks and double spending caused by delay to a certain extent.

Therefore, in the PoW protocol, the most direct contradiction is between the block interval and the block size. Sompolinsky et al. [16] proved that it is easier for the adversary to attack in the case of high throughput. At the same time, the authors proposed the Greedy Least-Observed Sub-Tree algorithm (GHOST algorithm), which effectively improves the forking problems caused by the block generation and the difficulty in determining the longest chain and eliminates the asymmetric advantage caused by different computing power. Nayak et al. [17] proposed the "stubborn" strategy, which extends the original "selfish mining". Based on this strategy, the revenue of the malicious mining pool will increase by 13.94%, and the "stubborn" strategy is further optimized. Two new strategies, "the Equal Fork Stubborn" and "Trail Stubborn", are proposed to further improve the mining revenue of corrupted mining pools.

Garay and Kiayias et al. [18,19] established an abstract model and proved that if the adversary controls a certain proportion of computing power, delay attacks can be launched by interfering with communication between miners. They also pointed out that the adversary can still cause chain attacks through delay when only controlling communication. Furthermore, it proposed three security attributes—chain growth rate, common prefix, and chain quality—to reflect the current security state of blockchain. Pass et al. [10] proved that if the adversary's delayed attack on the chain causes a fork, the number of message delay rounds is less than Δ, where $\Delta \ll 1/np$. Considering the influence of networks and hardware devices in the actual situation, Wei and Yuan [13] simulated the evolution process of the main chain under PoW through a tree structure $\mathrm{Tree}_{\mathrm{MC}}$ and modified the three security attributes proposed in [18,19] to prove that the blockchain is safe under the long delay attack of $\Delta \geq 1/np$ in an asynchronous network.

3. Optimized Blockchain Model

3.1. T_{OD} Record Model

Wei [13] modified the previous security attributes proposed by Garay; the attribute chain growth denotes the number of blocks in the network where the majority of the chains grow in one round, and the attribute common prefix denotes the maximum number of consecutive identical nodes on different chains starting from the root node. They also proposed a tree model to simulate the evolution of the main chain by adding and deleting operations to nodes on the model to achieve the evolution of the main chain on the PoW protocol, and used this model to represent the state of the miner chain under the ratio $\lambda(\lambda > 1/2)$. Due to the problem of the model setting method, there may be some errors with the actual main chain evolution process. In this model, the authors set two operations for adding and deleting blocks in the consensus mechanism to implement the record control of the nodes. In each round of simulation consensus, when the node addition operation is completed, the delete operation will be performed immediately according to the model setting. This operation will delete all useless nodes in the current round, also known as "illegal" nodes, and among these deleted branches are bound to exist those that have not been recorded in the model due to improvements in computing power or delay factors. In addition, the standby chain may become the main chain in the later round, resulting in the possibility of inconsistency with the record node in the actual consensus process. In this paper, an improved record model is proposed to address the relevant issues as delineated by T_{OD}. In this improved T_{OD} model, the main operations are shown below (in Figure 1):

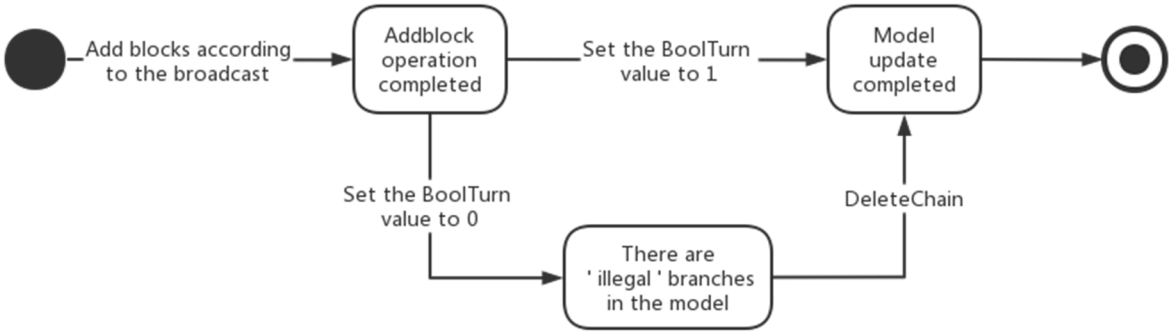


Figure 1. T_{OD} record status diagram.

Step 1. Addblock: in the network, assume that there is a branch chain C_0 on T_{OD} and that the individual nodes $C_0 = (B_0, B_1, \ldots B_k)$ are exactly the same as the first k block nodes on the broadcast chain C', i.e., there exists a value of k $(0 < k < n)$ such that the first k nodes on C' are exactly equal to the C_0 branch, and at that point k reaches its minimum value that satisfies the conditions, then adds the blocks on C' from k + 1 to n to the branch.

Step 2. BoolTurn: the return value is set according to the circumstances of each round of the consensus process. It determines whether the chain deletion operation needs to be performed in the following operation.

Step 3. DeleteChain: in the model, the "illegal" chain and block here represent branches of the model in the current round where the length is shorter than the tree depth corresponding to the chain C newly added to the model and all invalid blocks and branches added to the relevant branch after adding the last block node of the broadcasted new chain C' to the model tree in Step 1.

Let B_0 be the initial root on T_{OD}. After the Addblock operation on the current round, determine whether a new fork is created on the current model and whether the fork depth at the original node in the model increases.

3.2. Fork Problems on T_{OD}

When forking occurs during model recording, we consider the following two questions:

- After adding a new block, determine whether a new fork is generated on the current model.
- If there is a new fork after adding a new block to the original fork, judge whether the fork depth on the node increases.

If a new block is added to the model in the current rth round after the Addblock operation and no fork is created, or if the difference in depth between the longest and shortest branches on the current branch is less than 1, the BoolTurn return value is set to 1 and set to 0 in the r + 1 round. If a new fork is generated in the current round and the difference in depth between the longest and shortest branches is not less than 1, then set the BoolTurn value to 0. If the return value of BoolTurn is 0, the DeleteChain operation will be executed immediately to delete the "illegal" chain or block in the current model; otherwise, the DeleteChain operation to delete the "illegal" block will not be executed in the current round. As shown in the Figures 2–4, the specific situation of handling fork problems in the model is introduced as follows:

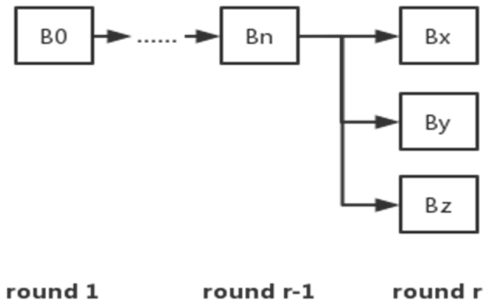


Figure 2. The fork depth of the current round does not increase.

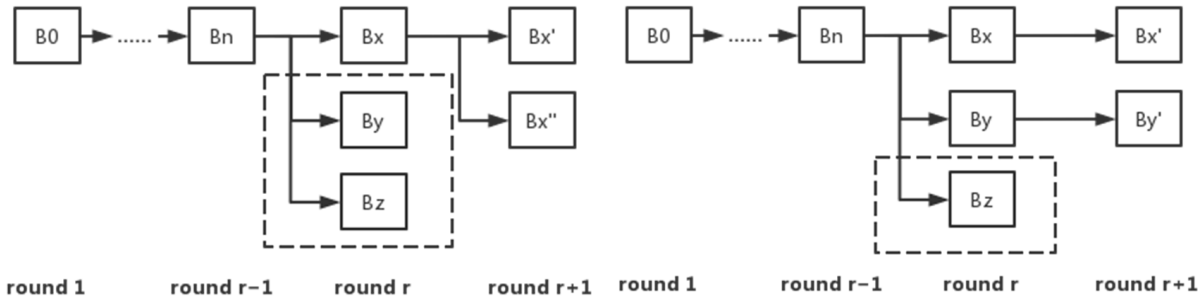


Figure 3. No new fork is generated in the current round, but the fork depth is increased, set BoolTurn = 0.

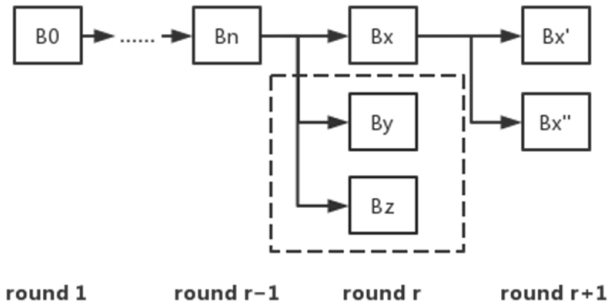


Figure 4. No fork is generated in the current round, but the fork depth is increased, set BoolTurn = 0.

If there is no fork generated after adding a new block in the current round, or if the depth difference between the longest branch and the shortest branch is not more than 1, set BoolTurn = 1, and no DeleteChain operation (3) is performed, as shown in Figure 2.

If there is no fork generated after the Addblock operation in a round but the fork depth between the longest chain and the shortest chain in the current model is more than 1,

BoolTurn is set to 0, and the DeleteChain operation is executed immediately in the current round to delete useless branches in the model.

If a new fork is generated in a round and the depth of the longest branch and the shortest branch are not less than 1, BoolTurn is set to 0 in this round, and the operation DeleteChain is executed immediately to delete the "illegal" branches in the model.

4. Long Delay Attack Based on T_{OD} Model

4.1. Effect of Long Delay Attack on Chain Growth

In this part, we use the T_{OD} model to simulate and analyze the potential threat of long delay attacks on the blockchain PoW protocol and make a comparison experiment with [20]. Similarly, we set the adversary to not control any computing power, and we divide miners into H_A and H_B to represent two branches on T_{OD}. At the same time, the number of miners in the two sets is dynamically equal. That is, assuming $|H_A| = |H_B| = n/2$, the probability of successful mining in each round is p, and the probability of mining a delayable block in each round is α. If n miners successfully mined, then the probability of successful mining is $\eta(n, p) = 1 - (1 - p)^n$, and when n is large enough, we can consider $\eta(n, p) \approx np$.

Here, we consider the impact of long delay attacks on chain growth. Considering the impact of chain growth and forks per unit round, the analysis of chain growth can be divided into three cases: Both A and B chains grow, only one chain grows, or there is no growth, and thus it can be expressed as follows:

(1) If both branches A and B grow at the end of a round, there are several possible situations:

   (a) After the consensus process, both A and B have mined non-delayable blocks, so in the next round, each branch is successfully extended, the fork depth is increased by 1, and the length of the blockchain is increased by 1; the probability is shown in Equation (1).

$$\frac{\eta(\frac{n}{2}, (1-\alpha)p)^2}{\eta(n,p)} \approx \frac{\frac{1}{4}(1-\alpha)n^2p^2}{np} = \frac{(1-\alpha)np}{4} \tag{1}$$

   (b) After the completion of the consensus processes of A and B, the miners have successfully mined the block, the two branch chains have increased, and the probability of this situation is Equation (2).

$$\frac{\eta(\frac{n}{2}, p)^2}{\eta(n,p)} \approx \frac{\frac{1}{4}n^2p^2}{np} = \frac{np}{4} \tag{2}$$

   (c) When both A and B mines the delayable block, and the adversary chooses to broadcast in the same round, the chain length will also increase. We will discuss the details in case 4.

(2) If only one branch grows after the current round r, BoolTurn is set to 1, and BoolTurn is set to 0 in round r + 1. If a new block is generated in round r + 1, no delete operation is performed, and so if a branch grows, BoolTurn is set to 1. Without loss of generality, the probability that one of the branches A or B mined a non-delayable block is A, and B failed to mine a new block, and so we can obtain Equation (3):

$$\frac{2(1-\eta(\frac{n}{2}, p))\eta(\frac{n}{2}, (1-\alpha)p)}{\eta(n,p)} \approx \frac{(1-\alpha)(2-np)}{2} \tag{3}$$

Similarly, when one of A and B mined the non-delayable block and the other mined the delayable block, it was also necessary to consider whether the number of delay rounds reached Δ, which is discussed in case 4.

(3) Branch A and branch B failed to mine a new block in the round r, and at this time, both branches did not grow; the probability of such a situation is shown in Equation (4):

$$\frac{(1-\eta(\frac{n}{2},p))^2}{\eta(n,p)} \approx \frac{\frac{1}{4}(2-np)^2}{np} = \frac{(2-np)^2}{4np} \tag{4}$$

Similarly, if A and B both mined a delayable block and delayed it in the same round, no new node will be created in this round.

(4) Here we focus on several cases after a delayable block has been mined. When one of the two branches has a delayable block, without loss of generality, set it to A and discuss the other block.

(a) If branch B failed to mine the block, it needs to consider whether the block mined by branch A has reached the delay limit. If it has reached the Δ round, the block must be broadcast. If it has not reached the Δ round, the adversary can choose to continue to delay; if A did not mine a non-delayable block, it will have the following probability in the following round shown in Equation (5):

$$P_n = (1-\eta(\frac{n}{2},p))\cdot(1-\eta(\frac{n}{2},(1-\alpha)p)) \approx \frac{(2-np)(2-np+\alpha np)}{4} \tag{5}$$

(b) Branch B must broadcast a non-delayable block in the current round, and in a sense, the probability of the adversary not mining a non-delayable block is almost equivalent to the probability of mining a delayable block, and so we can obtain Equation (6):

$$\begin{aligned} &\frac{2(1-\eta(\frac{n}{2},(1-\alpha)p))\cdot\eta(\frac{n}{2},(1-\alpha)p)}{\eta(n,p)} \\ &\approx (1-\frac{(1-\alpha)np}{2})(1-\alpha) = 1-\alpha-\frac{(1-\alpha)^2np}{2} \end{aligned} \tag{6}$$

As shown in Equation (7), the analysis shows that if the delayable block is mined on one chain, the possibility of the other chain is not unique, and the probability of the adversary continuing to wait for the delay in this case is:

$$1-(\frac{2-np-2\alpha+np\alpha}{2}+\frac{np}{4}) = \frac{np-2\alpha(np-2)}{4} \tag{7}$$

In this case, the probability of chain deletion and chain growth in a round is 0 because of delay, that is, the probability that the adversary succeeds in the round that can be delayed is:

$$S = \frac{(2-np)^2\alpha p}{4np} + \frac{np-2\alpha(np-2)}{4} \times \frac{np(1-P_n^{\Delta})}{2-2P_n} \tag{8}$$

4.2. Improvement of Long Delay Attack

In [20], the authors proposed the balance attack, and Wei [13] proposed that it is possible to cause a delay attack even if the adversary does not control any computing power. However, in the actual situation, when the adversary does not mine, its behavior is "free". In other words, when the adversary makes an attack, it can create a delay attack again on another chain. Therefore, combining the balance attack and the long delay attack in [13], we propose an improved long delay attack. The following table compares three attack methods.

In our attack, assuming that the adversary can control at most μn miners and the total number of honest miners is n, we divide the honest miners and the controlled corrupted miners into two parts equally, dividing the honest miners into H_A and H_B, and the chains C_A and C_B represent the two branches. Each chain has $n(\mu+1)/2$ nodes, but the corrupted miners do not always mine on the current chain. For example, when miners on one chain have finished mining, miners on the current chain can dynamically delay attacks on the

other chain, which also greatly improves the work efficiency of corrupt miners in this case. So, we have the following situation (see Table 1):

Table 1. Comparison of attack methods.

	Long Delay Attack	**Balance Attack**	**Improved Long Delay Attack**
Whether the adversary controls computing power	No	YES	YES
Purpose of attack	Extend fork	Change the target chain to the main chain	Extend and produce fork
Method of attack	The adversary delays the new blocks and broadcasts them to different miners in different order after collecting a certain number of chains	Isolating miners' communication and implementing efficient mining on the target chain, turning the target chain into the master chain	The adversary delays the new blocks, and the corrupted miners mine the delayable blocks and immediately delay them, then broadcasts them separately to different honest miners.
Corrupted miner	—	Work on one chain to increase the target chain mining efficiency	According to the mining results of each round, corrupted miners can mine dynamically in the two sets

4.3. Proof of Security

(1) When blocks are mined on both C_A and C_B, and the blocks on that chain are both obtained by at least one honest miner, and after the miners broadcast the chains on their respective nodes, a fork is formed between C_A and C_B at this point, with the fork depth increasing by 1 and the chain growth increasing by 1. So, we can obtain Equation (9), and the probability of this happening is:

$$P_1 = \frac{\eta(\frac{n(1+\mu)}{2}, p)\eta(\frac{n(1+\mu)}{2}, p)}{\eta(n, p)} = \frac{n(1+\mu)p}{4} \tag{9}$$

(2) On C_A and C_B, one of the chains has a miner successfully mining a node and the block is a non-delayable block, and the other chain has not mined. According to the T_{OD} model, it is known that when a new block is created on one chain it will not immediately delete the shorter chain and enter the BoolTurn operation, maximizing the spare chain on the model and allowing the adversary the opportunity to extend the fork. So, in the next round, if the shorter chain mines a new block, the adversary succeeds in increasing the fork, and success in the next round requires two conditions to be satisfied: one chain does not mine a non-delayable block, and the other chain does not succeed in mining, and so the probability of entering the next round is shown in Equation (10):

$$P_{next} = (1 - \eta(\frac{n}{2}, (1-\alpha)p)) \cdot (1 - \eta(\frac{n(1+2\mu)}{2}, p)) \tag{10}$$

At this time, the corrupted miners can work on the other chain, so we can obtain Equation (11) which shows that the probability of successfully increasing the fork within Δ rounds is:

$$\sum_{i=1}^{\Delta} {P_{next}}^{i-1} \cdot \eta(\frac{n(1+2\mu)}{2}, p) \tag{11}$$

On this basis, the analysis is made by considering the two cases of the first chain mining a delayable block and a non-delayable block; when a node on one chain mines

a non-delayable block and the other chain does not mine a block, there is a probability Equation (12):

$$P_2{}' = C_2^1 \frac{(1-\eta(\frac{(1+\mu)n}{2}, p))\eta(\frac{n(1+\mu)}{2}, (1-\alpha)p)}{n(1+\mu)p} \tag{12}$$

So, we obtain Equation (13), and the probability that the adversary will succeed in this case is:

$$P_2 = P_2{}' \cdot \sum_{i=1}^{\Delta} P_{next}{}^{i-1} \cdot \eta(\frac{n(1+2\mu)}{2}, p) \tag{13}$$

(3) If only one chain in C_A and C_B mined a block, which was mined by a corrupted miner, and another failed, then the probability of this case is shown in Equation (14):

$$P_3{}' = \frac{C_2^1(1-\eta(\frac{n}{2}, p)) \cdot \eta(\frac{\mu n}{2}, \alpha p)(1-\eta(\frac{(1+\mu)n}{2}, p))}{\eta(n(1+\mu), p)} \tag{14}$$

So, the probability, as shown in Equation (15), that the adversary can successfully extend the fork in this situation is:

$$P_3 = P_3{}' \cdot \sum_{i=1}^{\Delta} P_{next}{}^{i-1} \cdot \eta(\frac{n(1+2\mu)}{2}, p) \tag{15}$$

(4) If only one chain in C_A and C_B mined a block, which was mined by an honest miner, and the other chain failed to mine the block, then the probability of this case is:

$$P_4{}' = \frac{C_2^1 \cdot \eta(\frac{n}{2}, p) \cdot (1-\eta(\frac{(1+\mu)n}{2}, p)))}{\eta(n(1+\mu), p)} \tag{16}$$

So, the probability that the adversary can successfully extend the fork in this situation is:

$$P_4 = P_4{}' \cdot \sum_{i=1}^{\Delta} P_{next}{}^{i-1} \cdot \eta(\frac{n(1+2\mu)}{2}, p) \tag{17}$$

Therefore, as shown in Equation (18), the probability that the adversary can successfully increase the fork depth by 1 in this case is:

$$P_{suc} = P_1 + P_2 + P_3 + P_4 \tag{18}$$

It is easy to see that the probability that the adversary will extend the length of the fork to T is P_{suc}^T.

5. Experimental Analysis

In this section, we experimentally analyze the evolution of the long delay attack and the improved attack model on the T_{OD} model, experimentally verifying the feasibility of the T_{OD} model and the change in attack efficiency after the improved attack.

5.1. Experimental Analysis of Long Delay Attack on Chain Growth

We simulate and analyze the potential threat of a long delay attack on the growth rate of the blockchain chain through the model, assuming $\mu = 0$, that is, the adversary does not control any miner's computing power. Here, we analyze the change in chain length for one unit round. If the adversary successfully prevents the new block from being added to T_{OD}, the adversary is called successful. The fork depth and delay success probability are shown in the following table.

According to the experimental data in Table 2, when the number of miners and the mining success rate are fixed, the growth of the number of delayed rounds and the probability of mining the delayable block will greatly improve the adversary's success

probability of attack, and with the increase of the fork depth, the success probability of attack will gradually decrease; that is, in order to improve the attack efficiency, it must be controlled within a certain fork depth.

Table 2. Comparison of success probabilities with different parameters.

α	Δ	T	f	n	P_{suc}
0.70	60	5	1/60	100,000	0.0002942
0.70	60	3	1/60	100,000	0.0011407
0.75	80	5	1/60	100,000	0.0015351
0.75	80	3	1/60	100,000	0.0045197
0.80	100	5	1/60	100,000	0.0056138
0.80	100	3	1/60	100,000	0.0133162
0.85	120	5	1/60	100,000	0.0165696
0.85	120	3	1/60	100,000	0.0328166
0.85	140	5	1/60	100,000	0.0403558
0.85	140	3	1/60	100,000	0.0766876

As can be seen in Figure 5, the adversary's attack success rate in each round is closely related to the probability of the delayable block. As the block delayable probability increases, the adversary's attack success rate increases, and when the number of delay rounds increases, the adversary's attack success rate is higher under the same block delayable probability.

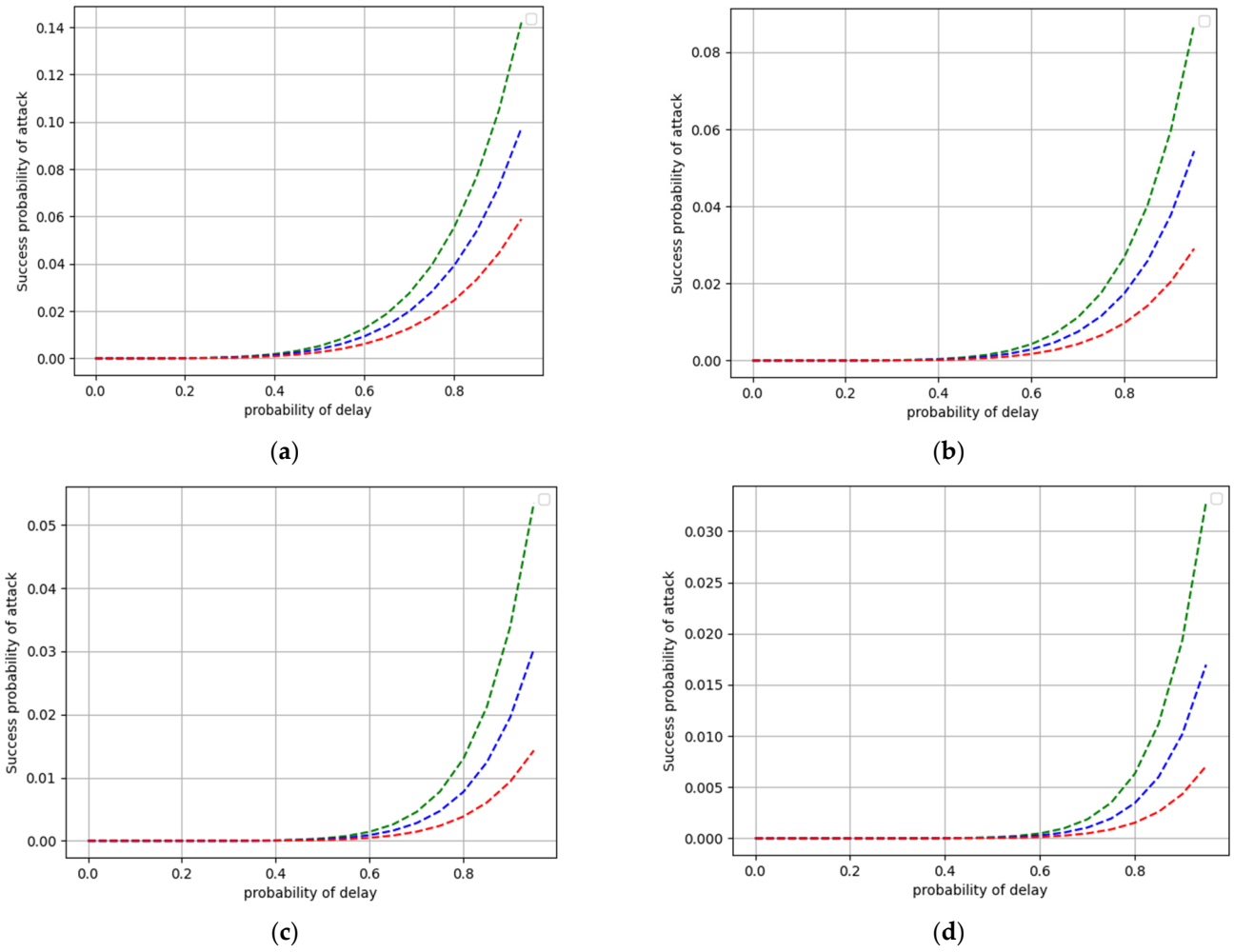


Figure 5. Relationship between probability of delay and success probability of attack (red: $\Delta = 90$, blue: $\Delta = 110$, green: $\Delta = 130$). (**a**) T = 4, (**b**) T = 5, (**c**) T = 4, (**d**) T = 5.

We compared and analyzed the adversary's attack on the security property of chain growth rate after mining a delayable block on T_{OD} and $Tree_{MC}$ in a round with $np = 1/60$. From Figure 6, we can see that under the same value of α, the optimized T_{OD} model shows that the success rate of the adversary's attack is higher than that of the $Tree_{MC}$ model, and the success rate of the adversary's attack on the chain growth rate increases with the increase in a round but does not exceed 0.8. Combining the previous experimental data, it can be concluded that although the success rate of the attack may increase after the adversary has mined the delayable block, considering the actual situation, with the increase in the number of consecutive rounds T and the decrease in the number of delayable blocks and delayable rounds, the success rate is still at a low level, and so the PoW protocol still has good security in the face of long delay attacks.

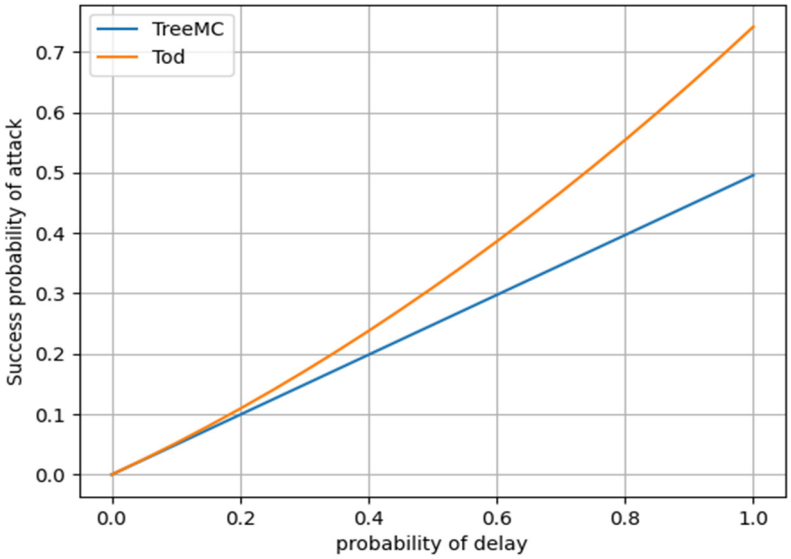


Figure 6. The relationship between success probability of attack and probability of delay under $Tree_{MC}$ and T_{OD} models.

5.2. Improved Experimental Analysis of Long Delay Attack

The values of p change in blockchain networks at different levels of computing power and mining difficulty; in this experiment, we assume that the adversary can control a certain amount of computing power, and the upper bound is μn, where $0 < \mu < 1$, $n = 100{,}000$, and $np = 1/60$. We analyzed the relationship between the probability of mining a delayable block α, the number of delay rounds Δ, the proportion of control adversaries μ, and the probability of a successful attack.

Figure 7 shows that in the improved long delay attack, when the proportion of corrupted miners is given, the success rate of the adversary attack can increase significantly with the increase in the probability of delayable blocks, and the success rate of the adversary attack will gradually decrease as the fork depth gradually increases. The attack success rate also increases significantly when the number of corrupted miners controlled by the adversary is large.

As shown in Figure 8, when the maximum delayable round is set to 90, we can see the success probability of the adversary attack under different delayable block probabilities when the fork depth is 3 and 5, respectively. We can see that the increase in fork depth will lead to a decrease in the adversary attack success rate.

The increase in fork depth and the decrease in Δ will reduce the adversary's success probability. Experiments show that when the fork depth increases to 7, the success rate of the attack will decrease sharply (see Figure 9).

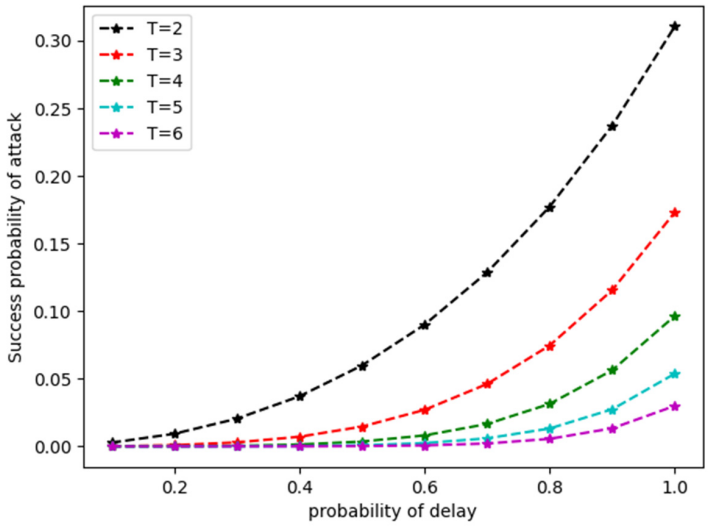
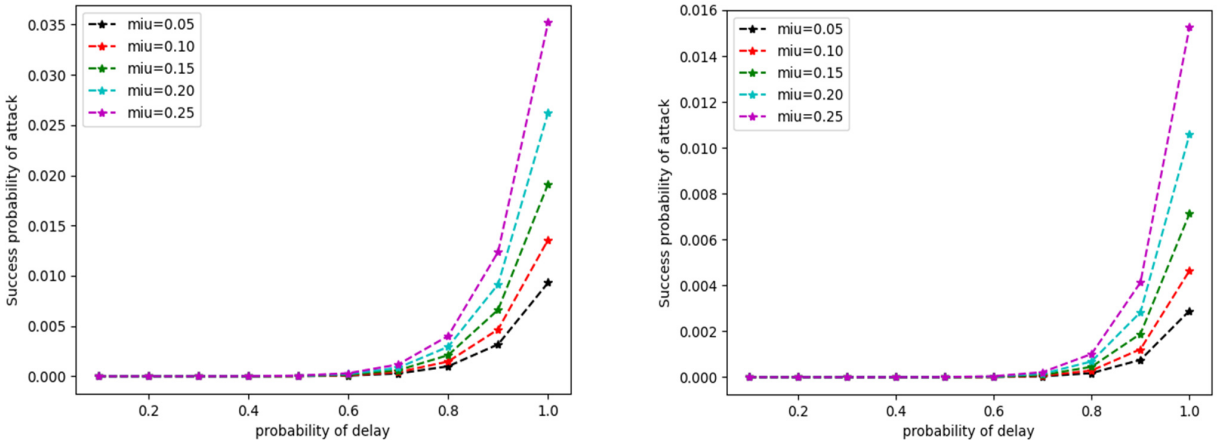


Figure 7. Relationship between block delayability probability and adversary success rate at different fork depth.



(**a**) (**b**)

Figure 8. The relationship between proportion of corrupted miners and the success rate of attacks. (**a**) Δ = 90, T = 3, (**b**) Δ = 90, T = 5.

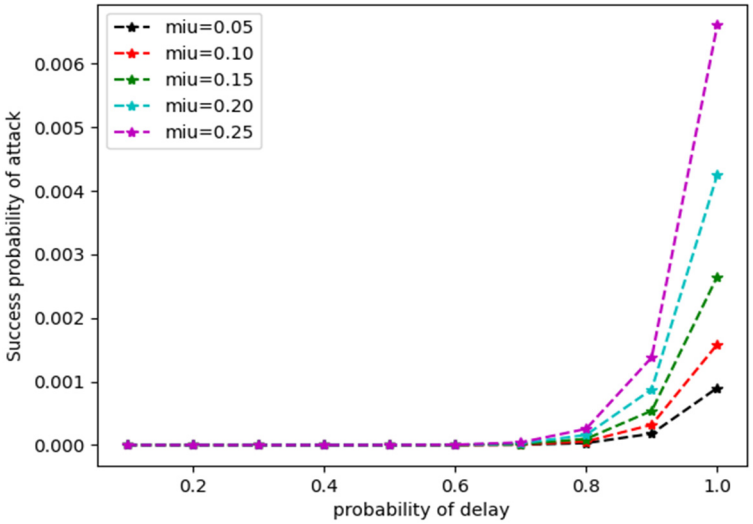


Figure 9. The relationship between proportion of corrupted miners and the success rate of attacks. Δ = 80, T = 6.

In the experiment, it is obvious that the successful attack probability of the adversary increases with the increase of the probability α of the delayable block, the number of delayed rounds Δ, and the proportion of the control adversary μ. When the proportion of the adversary is close to the extreme value, the successful attack probability increases significantly. Additionally, when the adversary controls the proportion of computing power and the probability of mining the delayable block is low, the adversary's attack success rate is still low, and with the increase in the fork depth, the adversary's attack success rate will gradually decrease. It can be seen that the PoW protocol still has good security in the face of a more complex long delay environment (see Figure 10).

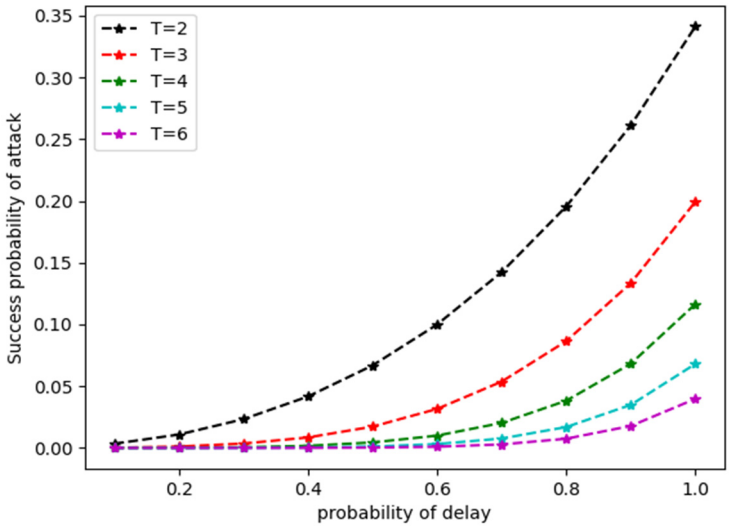


Figure 10. Relationship between probability of delayable block and attack success rate at different depths, $\mu = 0.35$, $\Delta = 15$.

6. Conclusions

This paper investigated the security of the PoW protocol in a long delay attack, and improves the existing recording model, effectively reducing the errors caused by the model in the recording process. At the same time, based on the improved model proposed, this paper verified the security of the blockchain PoW protocol in the long delay environment and proves that blockchain can guarantee certain security in the long delay environment. In the improved attack model in this paper, the adversary can control μn miners to mine, and similar to honest miners being evenly distributed among two sets, corrupted miners are also evenly distributed among them, and miners are set to be free to carry out attacks. Through the analysis of the long delay environment by the T_{OD} model, it can be seen that the probability of mining a delayable block in the mining process will greatly affect the success rate of the adversary attack, and the success rate of the adversary attack has a great relationship with the current fork depth, the maximum number of delayable rounds, and the corresponding activities of corrupted miners. This paper provides a theoretical analysis of the success rate of adversary attacks and the security of PoW protocols in the face of complex latency environments by providing a flexible division of adversary computing power and network latency from the adversary perspective. According to the experiments in this paper, we can see that when the probability of the delayable block, the maximum number of delayable rounds, and the proportion of corrupted miners in the protocol increase, the success rate of the delay attack will increase, and therefore its security will be affected to some extent. However, in combination with the actual situation, it is difficult for the adversary to control the computing power of a large number of miners in the mining process, and the probability of mining the delayable block and the number of multiple delay rounds will be greatly limited. When the fork depth reaches a certain level, the success rate of the attack will decrease significantly. Therefore, it can be seen that the PoW protocol still provides good security in the face of complex long delay attacks.

It is undeniable that as the adversary controls a large amount of computing power in the network and implements delay attacks in a diversified and deepening manner, it is clear that the high energy consumption and inefficient PoW protocols cannot meet the actual needs of the network nodes. Exploring the security of PoW protocols in more complex network environments and under attack conditions will also be our next research direction.

Author Contributions: Conceptualization, T.F. and Y.L.; methodology, T.F. and Y.L.; software, Y.L.; validation, T.F. and Y.L.; investigation, T.F. and Y.L.; resources, T.F.; data curation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, T.F. and Y.L.; visualization, T.F. and Y.L.; supervision, T.F.; project administration, T.F. and Y.L.; funding acquisition, T.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (Grant No. 62162039, 61762060), Foundation for the Key Research and Development Program of Gansu Province, China (Grant No. 20YF3GA016).

Data Availability Statement: No data were used to support this study.

Acknowledgments: The authors would like to thank all editors and reviewers for their valuable comments and suggestions and all funding for the great help and support of this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Longo, R.; Mascia, C.; Meneghetti, A.; Santilli, G.; Tognolini, G. Adaptable Cryptographic Primitives in Blockchains via Smart Contracts. *Cryptography* **2022**, *6*, 32. [CrossRef]
2. Romano, D.; Schmid, G. Beyond Bitcoin: Recent Trends and Perspectives in Distributed Ledger Technology. *Cryptography* **2021**, *5*, 36. [CrossRef]
3. Martínez, V.G.; Hernández-Álvarez, L.; Encinas, L.H. Analysis of the Cryptographic Tools for Blockchain and Bitcoin. *Mathematics* **2020**, *8*, 131. [CrossRef]
4. Caldarola, F.; d'Atri, G.; Zanardo, E. Neural Fairness Blockchain Protocol Using an Elliptic Curves Lottery. *Mathematics* **2022**, *10*, 3040. [CrossRef]
5. Heilman, E.; Kendler, A.; Zohar, A.; Goldberg, S. Eclipse Attacks on Bitcoin's Peer-to-Peer Network. In Proceedings of the 24th USENIX Security Symposium, Washington, DC, USA, 12–14 August 2015; pp. 129–144.
6. Douceur, J.R. The Sybil Attack. In Proceedings of the Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Cambridge, MA, USA, 7–8 March 2002.
7. Yang, X.; Chen, Y.; Chen, X. Effective Scheme against 51% attack on proof-of-Work Blockchain with History Weighted Information. In Proceedings of the IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019; pp. 261–265.
8. Wang, H.; Zhang, X.W. SRRS: A blockchain fast propagation protocol based on non-Markovian process. *Comput. Netw.* **2022**, *219*, 109435. [CrossRef]
9. Trom, J. Cuckoo cycle: A memory bound graph-theoretic proof-of-work. In Proceedings of the International Conference on Financial Cryptography and Data Security, San Juan, Puerto Rico, 26–30 January 2015; Springer: Berlin, Germany, 2015; pp. 49–62.
10. Pass, R.; Seeman, L.; Shelat, A. Analysis of the blockchain protocol in asynchronous networks. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, 30 April–4 May 2017; pp. 643–673.
11. Gazi, P.; Kiayias, A.; Russell, A. Tight consistency bounds for bitcoin. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 9–13 November 2020; pp. 819–838.
12. Dembo, A.; Kannan, S.; Tas, E.N.; Tse, D.; Viswanath, P.; Wang, X.; Zeitouni, O. Everything is a race and Nakamoto always wins. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 9–13 November 2020; pp. 859–878.
13. Wei, P.W.; Yuan, Q.; Zheng, Y.L. Security of the blockchain protocol against long delay attack. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, Australia, 2–6 December 2018; pp. 250–275.
14. Eyal, I.; Sirer, E.G. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography and Data Security*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8437, pp. 436–454.
15. Eyal, I.; Gencer, A.E.; Sirer, E.G.; Renesse, R.V. Bitcoin-NG: Ascalable blockchain protocol. In Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation, Santa Clara, CA, USA, 16–18 March 2016; pp. 45–59.
16. Sompolinsky, Y.; Zohar, A. Secure high-rate transaction processing in bitcoin. In Proceedings of the International Conference on Financial Cryptography and Data Security, San Juan, Puerto Rico, 26–30 January 2015; pp. 507–527.

17. Nayak, K.; Kumar, S.; Miller, A.; Shi, E. Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack. In Proceedings of the IEEE European Symposium on Security & Privacy, Saarbruecken, Germany, 21–24 March 2016; pp. 305–320.
18. Kiayas, A.; Panagiotakos, G. Speed-Security Tradeoffs in Blockchain Protocols. IACR ePrint Archive Report. 2016. Available online: https://eprint.iacr.org/2015/1019 (accessed on 5 June 2023).
19. Garay, J.; Kiayias, A.; Leonardos, N. The Bitcoin Backbone Protocol with Chains of Variable Difficulty. In Proceedings of the International Cryptology Conference, Santa Barbara, CA, USA, 20–24 August 2017; Springer: Cham, Switzerland, 2017; pp. 291–323.
20. Natoli, C.; Granmoli, V. The balance attack against proof-of-work blockchains: The R3 testbed as an example. In Computing Research Repository. *arXiv* **2016**, arXiv:1612.09426.