2-2020

Analysis of blockchain protocol against static adversarial miners corrupted by long delay attackers

Quan YUAN

Puwen WEI

Keting JIA

Haiyang XUE
Singapore Management University, haiyangxue@smu.edu.sg

SCIENCE CHINA
Information Sciences

March 2020, Vol. 63 130104:1–130104:15
https://doi.org/10.1007/s11432-019-9916-5

• RESEARCH PAPER •
Special Focus on Security and Privacy in Blockchain-based Applications

Analysis of blockchain protocol against static adversarial miners corrupted by long delay attackers

Quan YUAN[1], Puwen WEI[1*], Keting JIA[2] & Haiyang XUE[3,4]

[1]*Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan* 250100*, China;*
[2]*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084*, China;*
[3]*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100864*, China;*
[4]*Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, Beijing* 100864*, China*

Received 1 March 2019/Revised 6 April 2019/Accepted 4 June 2019/Published online 11 February 2020

Abstract Bitcoin, which was initially introduced by Nakamoto, is the most disruptive and impactive cryptocurrency. The core Bitcoin technology is the so-called blockchain protocol. In recent years, several studies have focused on rigorous analyses of the security of Nakamoto's blockchain protocol in an asynchronous network where network delay must be considered. Wei, Yuan, and Zheng investigated the effect of a long delay attack against Nakamoto's blockchain protocol. However, their proof only holds in the honest miner setting. In this study, we improve Wei, Yuan and Zheng's result using a stronger model where the adversary can perform long delay attacks and corrupt a certain fraction of the miners. We propose a method to analyze the converge event and demonstrate that the properties of chain growth, common prefix, and chain quality still hold with reasonable parameters in our stronger model.

Keywords blockchain, bitcoin, random oracle, delay, consensus protocol

Citation Yuan Q, Wei P W, Jia K T, et al. Analysis of blockchain protocol against static adversarial miners corrupted by long delay attackers. Sci China Inf Sci, 2020, 63(3): 130104, https://doi.org/10.1007/s11432-019-9916-5

1 Introduction

Since the introduction of Bitcoin [1], a series of studies [2–11] have focused on analyzing the security of Nakamoto's blockchain protocol, which is the core of the Bitcoin system. Garay et al. [12] provided the first rigorous cryptographic analysis of the basic properties of Nakamoto's blockchain protocol. In order to improve the security of the blockchain protocol, Wu et al. [13] suggested adding a supervising auditor. Considering the power of network delays, Pass et al. [14] analyzed Nakamoto's blockchain protocol in an asynchronous network with Δ-bounded delays; however, their analysis only holds for short delays, i.e., $\Delta \ll 1/np$, where n is the number of miners and p is mining hardness. Recently, Wei et al. [15] investigated the security of Nakamoto's blockchain against long delay attacks where $\Delta = O(1/np)$ or even $\Delta > 1/np$. Notice that their security proof is only considered in the honest miner setting, which means the adversary neither has any hash power nor corrupts any miner. Thus, a natural question is: What if the adversary can corrupt some miners and control a fraction of the total computational power in long delay attacks?

Our contribution. In this paper, we extend the results of a previous study [15] and analyze Nakamoto's blockchain using a stronger model wherein the adversary can perform long delay attacks

* Corresponding author (email: pwei@sdu.edu.cn)

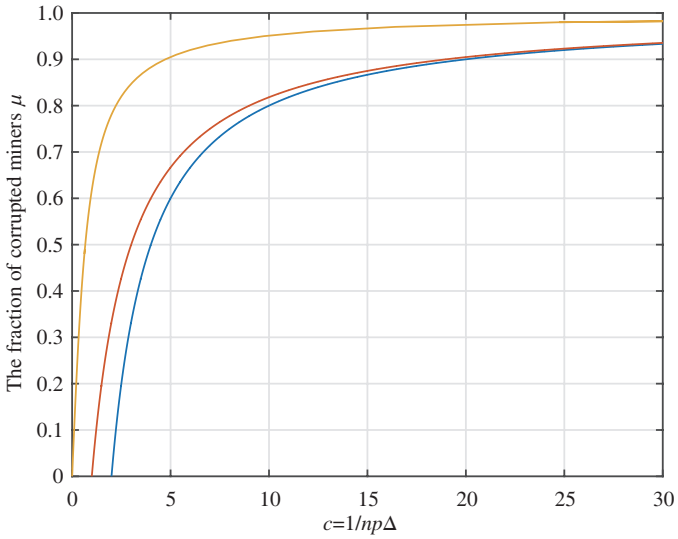

Figure 1 Comparison of the previous study [14] and this study. The blue curve indicates the maximum value of μ, i.e., the parameter under which the common prefix of the blockchain protocol can be proved [14]. The red curve indicates the maximum value of μ in this study where the probability of delay α is set to 1. The yellow curve shows when the best attack succeeds in violating the common prefix [14]. Note that if the parameter μ is above the yellow curve, the blockchain protocol will be insecure.

and corrupt a fraction of the miners. More precisely, in our model, there are $(1+\mu)n$ miners in total, μn of which are corrupted miners fully controlled by the adversary. Here, the adversary can control the mining strategy of the corrupted miners as well as their associated computational power and communication capability. For instance, the adversary can broadcast the chains maintained by μn corrupted miners at any round. Since the adversary can generate blocks using corrupted miners while performing a long delay attack, we refine the definition of chain quality in order to capture the quality of the majority of honest miners' chains. Note that the chain quality property is not discussed by Wei et al. [15] due to the honest miners setting. Our result demonstrate that the properties of chain growth, common prefix, and chain quality hold on the condition that $(1+\alpha np\Delta)\mu < (1-\eta)(1-np(1+\alpha\Delta))$, which is compared with the previous condition [14] in Figure 1.

As shown in Figure 1, we can prove the security of the blockchain protocol under a looser condition than the boundary proposed by Pass et al. [14]. We emphasize that the corruption in our model is static, which means the adversary can control a fixed group of miners. However, the adversary in previous study [14] can perform adaptive corruption but short delay attack, which means the adversary can select a certain number of the miners arbitrarily and control them.

Therefore, an interesting problem is how to evaluate the security of the blockchain protocol in a model that allows adaptive corruption and long delay.

Main analysis techniques. Ref. [15] provides a method called $\mathsf{Tree}_{\mathsf{MC}}$ to simplify the security analysis of Nakamoto's blockchain when considering long network delay. However, this types of method cannot be applied directly to our setting directly wherein the adversary can perform long delay attacks and can also control a certain fraction of the miners. The main difficulty of the analysis in our model is that the adversary may destroy the converge event defined in [15] by broadcasting "corrupted" blocks mined by corrupted miners. To address this problem, we introduce a probability experiment called the Bernoulli race to estimate the number of "corrupted" blocks during the execution of the blockchain protocol. Then, we identify a special event, which we refer to as **converge**, that may imply the elimination of all forks in our model. By comparing the number of **converge** events with the number of "corrupted" blocks, we show the probability that the adversary can maintain long forks is negligible, which implies the security of the blockchain.

2 Preliminaries

In this section, we review the blockchain protocol. Note that we adopt the notations used in previous studies [12, 14].

2.1 Notation

A blockchain C is a sequence of ordered blocks $C = \overrightarrow{B}$ where B denotes a block. $|C|$ denotes the number of blocks in C. Let C_i^r denote the chain of miner i at round r. $C^{\lfloor k}$ denotes the chain C, which removes the last k blocks of C. $C^{\lfloor k} = \varepsilon$ if $k \geqslant |C|$. $C_1 \preceq C_2$ means there exists some $k \geqslant 0$ such that $C_1 = C_2^{\lfloor k}$, i.e., C_1 is a prefix of C_2. $\mathsf{B}(n, p)$ denotes the binomial distribution with n trials and success probability p. Let D be a distribution and random variable $X \sim D$. If $\Pr[X \geqslant 0] = 1$, we consider D a non-negative distribution. Let $X_1 \sim D_1$ and $X_2 \sim D_2$ be independent random variables. If $X = X_1 + X_2$, then the distribution of X is denoted as $D_1 + D_2$. Similarly, $D_1 - D_2$ denotes the distribution of $X = X_1 - X_2$.

2.2 Blockchain protocol

Nakamoto's blockchain protocol is captured by a simplified protocol called $(\Pi, \mathcal{C})$ in [15], where the adversary is responsible for delivering all the messages. The simplified Nakamoto's blockchain protocol has been described in [12, 14, 15]. $(\Pi, \mathcal{C})$ is directed by an environment $Z(1^\kappa)$. At the beginning of the protocol, each honest miner i owns a genesis block $C_i^0 = B_0$. The protocol proceeds in rounds, which are globally referenced by miners and the adversary. Let p denote the mining hardness, which means each miner can succeed in mining with probability p per round. In each round, each honest miner runs Π to maintain a chain C as follows.

(1) If the last block of C is mined by himself, go to step (2). Otherwise, the miner receives messages from Z, creates a block B, and extends C to CB with probability p. If the miner succeeds in extending C, send the new blockchain to the adversary A. In this case, we say the miner mines a block B after chain C.

(2) On receiving the chains delivered by the adversary A, choose the longest chain, e.g., C'. If $|C'| > |C|$, set $C = C'$ and go to the next round.

Remark. In step (1), to avoid consecutive mining, the miner checks whether he has mined the last block of his chain. This means an honest miner cannot mine two consecutive blocks in a chain. This modification to the block chain protocol was proposed in a previous study [15]. This modification can prevent some possible forks. In addition, a miner is unlikely to mine two consecutive blocks in practice if the number of miners is sufficiently large. Therefore, this restriction on miner behavior is reasonable. However, we emphasize that such restriction is invalid for corrupted miners in our model. In addition, the modification may lead to a slight decline in total mining power, and, similar to a previous study [15], we ignore such a change in our proof due to the large number of miners.

2.3 Adversary with corrupted miners

The adversary in our model is similar to that of [15] with the exception that he can corrupt μn miners, where $\mu \in [0, 1]$. Note that there are $(1 + \mu)n$ miners in total. We emphasize that the corruption in our model is static. Therefore, our adversary is more powerful than that of [15] but weaker than that of [14] in terms of corruption attack, where the adversary can perform adaptive corruption and "short" delay attacks. As in [15], we also assume that if the adversary fails to delay the target chain, the chain will be diffused to other miners immediately. After the behavior of the honest miners in a round, the adversary behaves as follows.

Execution of adversary at round r:

• **Mining.** Each corrupted miner obtains a valid chain C from the adversary and attempts to mine an arbitrary block B after C with probability p. If the corrupted miner succeeds in mining a block, CB becomes a new valid chain and is sent to the adversary.

• **Receiving.** The adversary receives chains from honest and corrupted miners and selects the chain he wants to delay. However, if the selected chain is from an honest miner, it can only be delayed with probability α. The delayed chains from honest miners are tagged as "delayable". The undelayed chains are tagged as "undelayable", and chains from corrupted miners are tagged as "corrupted". Then, all the chains together with their tags and the round r are stored in a list $\mathcal{T}$.

• **Distribution.** The adversary selects chains in $\mathcal{T}$ to be distributed and these chains will be received by all miners at the next round. However, the following two types of chains must be distributed at the current round.

(1) Chains tagged as undelayable.

(2) Chains tagged as delayable for Δ rounds.

After Distribution, the current round ends.

Remark. Note that the adversary can create "forks" among honest miners by distributing more than one chains at a round. For example, the adversary distributes chains C_1 and C_2 in a round such that $|C_1| = |C_2|$. He sends (C_1, C_2) to honest miner i but (C_2, C_1) to honest miner j. Suppose C_1 and C_2 are longer than i and j's chains. Then, i will accept C_1 as his chain while j will accept C_2. As a result, a fork is created between miner i and j.

We say an honest miner is "being delayed" if his chain is being delayed by the adversary. Due to the no consecutive mining constraint, a delayed honest miner will not mine a block until he accepts a new chain mined by others.

3 Properties of the proposed blockchain model

Three basic properties characterize the security of blockchain protocol, i.e., chain growth, common prefix, and chain quality. We adopt the definitions of chain growth and common prefix given in a previous study [15] and supplement the definition of chain quality.

3.1 Chain growth

Let $\texttt{view}(\Pi, \mathcal{C}, A, Z, \kappa)$ denote the joint view of all honest miners and $|\texttt{view}(\Pi, \mathcal{C}, A, Z, \kappa)|$ denote the number of rounds during the execution of the protocol.

Definition 1 ([15]). Given $\texttt{view}(\Pi, \mathcal{C}, A, Z, \kappa)$ and two rounds r_1, r_2 such that $r_1 < r_2 \leqslant |\texttt{view}(\Pi, \mathcal{C}, A, Z, \kappa)|$. For $\lambda \in (\frac{1}{2}, 1]$ and integer $t \geqslant 0$, if

$$\Pr_{i,j}[|C_j^{r_2}| - |C_i^{r_1}| \geqslant t] \geqslant \lambda, \tag{1}$$

we say the blockchain grows by at least t blocks with majority $\lambda \in (\frac{1}{2}, 1]$ from round r_1 to r_2, where the probability is taken over all the selections of $i, j \in [n]$.

Define $\texttt{chain-increase}^{(\Pi,\mathcal{C})}_{A,Z,\kappa}(r_1, r_2, \lambda)$ as the maximum value of t satisfying (1) as follows:

$$\texttt{chain-increase}^{(\Pi,\mathcal{C})}_{A,Z,\kappa}(r_1, r_2, \lambda) = \max\left\{t | \Pr_{i,j}[|C_j^{r_2}| - |C_i^{r_1}| \geqslant t] \geqslant \lambda\right\}.$$

Definition 2 ([15]). The blockchain protocol $(\Pi, \mathcal{C})$ has chain growth rate $g \in \mathbb{R}$ with majority $\lambda \in (\frac{1}{2}, 1]$ if there exists some constant c and negligible functions ϵ_1, ϵ_2 such that for every $\kappa \in \mathbb{N}$, $T \geqslant c\log(\kappa)$ and every $r \leqslant |\texttt{view}(\Pi, C, A, Z, \kappa)| - T$, the following holds:

$$\Pr\left[\texttt{chain-increase}^{(\Pi,\mathcal{C})}_{A,Z,\kappa}(r, r+T, \lambda) \geqslant gT\right] \geqslant 1 - \epsilon_1(\kappa) - \epsilon_2(gT), \tag{2}$$

where the probability is taken over the randomness of the protocol.

3.2 Common prefix

Definition 3 ([15]). Given $\texttt{view}(\Pi, \mathcal{C}, A, Z, \kappa)$ and round $r \leqslant |\texttt{view}(\Pi, \mathcal{C}, A, Z, \kappa)|$. For $\lambda \in (\frac{1}{2}, 1]$ and integer $k \geqslant 0$, if

$$\Pr_{i,j}\left[\left(C_i^r{}^{\lfloor k} \preceq C_j^r\right) \wedge \left(C_j^r{}^{\lfloor k} \preceq C_i^r\right)\right] \geqslant \lambda, \tag{3}$$

set $\texttt{common-prefix}^{(\Pi,\mathcal{C})}_{A,Z,\kappa}(r, k, \lambda) = 1$; otherwise $\texttt{common-prefix}^{(\Pi,\mathcal{C})}_{A,Z,\kappa}(r, k, \lambda) = 0$. Here the probability is taken over all the selections of $i, j \in [n]$.

Definition 4 ([15]). A blockchain protocol $(\Pi, \mathcal{C})$ satisfies the common prefix property with parameter $\lambda \in (\frac{1}{2}, 1]$ if there exists some constant c and negligible functions ϵ_1 and ϵ_2 such that for every $\kappa \in \mathbb{N}$, $T \geqslant c\log(\kappa)$ and every $r \leqslant |\texttt{view}(\Pi, C, A, Z, \kappa)|$, the following holds

$$\Pr\left[\texttt{common-prefix}^{(\Pi,\mathcal{C})}_{A,Z,\kappa}(r, T, \lambda) = 1\right] \geqslant 1 - \epsilon_1(\kappa) - \epsilon_2(T), \tag{4}$$

where the probability is taken over the randomness of the protocol.

3.3 Chain quality

Definition 5. Let $\texttt{quality}_k(C, \rho) = 1$ if, in any consecutive sequence of more than k blocks in chain C, there are at most ρk blocks mined by corrupted miners.

Given $\texttt{view}(\Pi, \mathcal{C}, A, Z, \kappa)$, if

$$\Pr_i[\texttt{quality}_k(C_i^r, \rho) = 1] \geqslant \lambda, \tag{5}$$

define $\texttt{chain-quality}^{(\Pi,\mathcal{C})}_{A,Z,\kappa}(r, \rho, k, \lambda) = 1$. Otherwise, $\texttt{chain-quality}^{(\Pi,\mathcal{C})}_{A,Z,\kappa}(r, \rho, k, \lambda) = 0$. Here the probability is taken over all the selections of $i \in [n]$.

Definition 6. The blockchain protocol $(\Pi, \mathcal{C})$ has the chain quality at round r with parameter ρ and majority $\lambda \in (\frac{1}{2}, 1]$ if there exists some constant c and negligible functions ϵ_1, ϵ_2 such that for every $\kappa \in \mathbb{N}, k \geqslant c\log(\kappa)$, the following holds

$$\Pr\left[\texttt{chain-quality}^{(\Pi,\mathcal{C})}_{A,Z,\kappa}(r, \rho, k, \lambda) = 1\right] \geqslant 1 - \epsilon_1(\kappa) - \epsilon_2(k), \tag{6}$$

where the probability is taken over the randomness of the protocol.

4 $\mathsf{Tree}_{\mathsf{MC}}$ model

We refer to the $\mathsf{Tree}_{\mathsf{MC}}$ model introduced in [15]. The $\mathsf{Tree}_{\mathsf{MC}}$ model is useful to capture the evolution of the main chains of the blockchain protocol. Informally, $\mathsf{Tree}_{\mathsf{MC}}$ is a tree that can record the state of the main chains, where the nodes of $\mathsf{Tree}_{\mathsf{MC}}$ are the blocks and the branches are chains broadcast to honest miners. $\mathsf{Tree}_{\mathsf{MC}}$ has many good properties that are suitable to prove the properties of the blockchain protocol defined above. We follow the lemmas in [15] and provide the relation between $\mathsf{Tree}_{\mathsf{MC}}$ and the view of $(\Pi, \mathcal{C})$ in term of chain quality.

4.1 $\mathsf{Tree}_{\mathsf{MC}}$

$\mathsf{Tree}_{\mathsf{MC}}$ is initialized to the root B_0. In each round, $\mathsf{Tree}_{\mathsf{MC}}$ records the chains as follows.

• **AddBlock.** When the adversary broadcasts a chain $C = (B_0, B_1, \ldots, B_l)$, search the branch (or paths from root to leaves) C' in $\mathsf{Tree}_{\mathsf{MC}}$ such that $C' = C^{\lfloor k}$ with the smallest k. Note that the C' exists and is unique. Then, extend C' with the last k ordered blocks of C. Since the adversary can broadcast more than one chain in a round, one node of $\mathsf{Tree}_{\mathsf{MC}}$ may be extended with more than one branch simultaneously. If this occurs, we say the adversary creates a "fork".

• **DeleteBlock.** After the adversary completes Distribution in a round and Addblock is done, assume the depth of $\mathsf{Tree}_{\mathsf{MC}}$ is d. Delete "useless" blocks or forks such that only the branches Cs satisfying the following conditions remain.

(1) $|C| = d$.

(2) For any branch C' with length d, the last block of C was added to $\mathsf{Tree}_{\mathsf{MC}}$ no later than the last block of C'.

As shown above, $\mathsf{Tree}_{\mathsf{MC}}$ records the chains that are broadcast by the adversary and may be accepted by honest miners. If a chain C is broadcast but is shorter than the longest broadcast chain or there was another C' of the same length broadcast earlier, C cannot be stored in the tree due to the DeleteBlock operation. In addition, all branches on $\mathsf{Tree}_{\mathsf{MC}}$ at the end of a round are of equal depth and the depth of $\mathsf{Tree}_{\mathsf{MC}}$ never decreases.

The properties of $\mathsf{Tree}_{\mathsf{MC}}$ in our static corruption setting are described as follows.

Lemma 1. Properties of $\mathsf{Tree}_{\mathsf{MC}}$.

(1) If new blocks are successfully added to $\mathsf{Tree}_{\mathsf{MC}}$ at the end of a round, then the depth of $\mathsf{Tree}_{\mathsf{MC}}$ increases.

(2) If the depth of $\mathsf{Tree}_{\mathsf{MC}}$ increases from d to $d+l$ at a round where $l \geqslant 2$, then. in this round, all blocks with depth greater than $d+1$ are mined by the adversary.

(3) (Converge) If only one block is added to $\mathsf{Tree}_{\mathsf{MC}}$ at the end of a round, then there will be no fork on $\mathsf{Tree}_{\mathsf{MC}}$ and the depth increases by 1.

Proof. Properties (1) and (3) are the same as those in [15]; however, property (2) differs due to corrupted miners. The proofs of the properties (1) and (3) given in [15] hold in our setting and thus are omitted. We only need to prove property (2).

Suppose the depth of $\mathsf{Tree}_{\mathsf{MC}}$ increases from d to $d+l$ at a round where $l \geqslant 2$. That means a chain with length of at least $l+2$ is broadcast while previously broadcast chains are no longer than l. Thus, no honest miner has accepted a chain longer than $l+1$, and no block can be mined by honest miners after position $l+2$. Therefore, the blocks after position $l+2$ are mined by the adversary.

4.2 Relation between $\mathsf{Tree}_{\mathsf{MC}}$ and the view of $(\Pi, \mathcal{C})$

$\mathsf{Tree}_{\mathsf{MC}}$ shows the possible states that are maintained by the majority of honest miners. Although there may be some chains that are not recorded in $\mathsf{Tree}_{\mathsf{MC}}$ due to the adversarial delay, we show that the difference between $\mathsf{Tree}_{\mathsf{MC}}$ and the actual view of the main chains of $(\Pi, \mathcal{C})$ is negligible. The following lemmas describe the relations between $\mathsf{Tree}_{\mathsf{MC}}$ and the view of $(\Pi, \mathcal{C})$. Note that we only provide the proof of Lemma 5. The proofs of Lemmas 2–4 are similar to those in [15]. Although the adversary controls a certain number of corrupted miners, the proofs of the lemmas still hold because the differences between $\mathsf{Tree}_{\mathsf{MC}}$ and the view of $(\Pi, \mathcal{C})$ are caused by delay rather than corrupted miners.

Lemma 2. Assume $1/2 < \lambda \leqslant 1 - 8\alpha p\Delta$. Let m^r_{delay} be the number of honest miners who are being delayed at round r. We have

$$\Pr\left[m^r_{\text{delay}} > \frac{(1-\lambda)n}{4}\right] < \mathrm{e}^{-\text{poly}(\kappa)}, \tag{7}$$

where the probability is taken over the randomness of the protocol and poly$(\cdot)$ is a polynomial function.

Over-delay$_r$ denotes an event where $m^r_{\text{delay}} > \frac{(1-\lambda)n}{4}$ occurs. Due to Lemma 2, for any r, we have $\Pr[\textbf{Over-delay}_r] < \mathrm{e}^{-\text{poly}(\kappa)}$.

Lemma 3. Given $(\Pi, \mathcal{C}, A, Z, \kappa)$, assume $1/2 < \lambda \leqslant 1 - 8\alpha p\Delta$. Let d^r_{tree} be the depth of $\mathsf{Tree}_{\mathsf{MC}}$ at round r. The chain growth property of $(\Pi, \mathcal{C})$ is described as follows:

$$\Pr\left[\texttt{chain-increase}^{(\Pi,\mathcal{C})}_{A,Z,\kappa}(r_1, r_2, \lambda) \geqslant d^{r_2}_{\text{tree}} - d^{r_1}_{\text{tree}}\right] \geqslant 1 - 2\mathrm{e}^{-\text{poly}(\kappa)}. \tag{8}$$

Lemma 4. Given $(\Pi, \mathcal{C}, A, Z, \kappa)$, assume $1/2 < \lambda \leqslant 1 - 8\alpha p\Delta$. Let d be the depth of $\mathsf{Tree}_{\mathsf{MC}}$. If all branches of $\mathsf{Tree}_{\mathsf{MC}}$ at round r have a common prefix of length $d - T$, we obtain

$$\Pr\left[\texttt{common-prefix}^{(\Pi,\mathcal{C})}_{A,Z,\kappa}(r, T, \lambda) = 1\right] \geqslant 1 - 2\mathrm{e}^{-\text{poly}(\kappa)}. \tag{9}$$

Lemma 5. Assume $1/2 < \lambda \leqslant 1 - 8\alpha p\Delta$. If in any consecutive k blocks of any branch of $\mathsf{Tree}_{\mathsf{MC}}$ in round r, there are at most ρk blocks mined by corrupted miners, we obtain

$$\Pr\left[\texttt{chain-quality}_{A,Z,\kappa}^{(\Pi,\mathcal{C})}(r,\rho,k,\lambda)=1\right] \geqslant 1-2\mathrm{e}^{-\mathrm{poly}(\kappa)}. \tag{10}$$

Proof. $\mathsf{Tree}_{\mathsf{MC}}^r$ denotes the set of all branches on $\mathsf{Tree}_{\mathsf{MC}}$ in round r, and Delay_r denotes the set of delayed chains in round r. Lemma 5 is obvious if $C_i^r \in \mathsf{Tree}_{\mathsf{MC}}^r$ for all honest miner i; however, not all C_i^rs are recorded by $\mathsf{Tree}_{\mathsf{MC}}$. Thus, we must demonstrate that most chains of honest miners are recorded.

Assume $C_i^r \notin \mathsf{Tree}_{\mathsf{MC}}^r$ and consider the following two cases.

- Case 1. $C_i^r \in \mathsf{Delay}_r$. If $\textbf{Over-delay}_r$ does not occur, $|\mathsf{Delay}_r| \leqslant \frac{(1-\lambda)n}{4}$. Here, we obtain

$$\Pr_i\left[C_i^r \in \mathsf{Delay}_r | \overline{\textbf{Over-delay}_r}\right] \leqslant \frac{|\mathsf{Delay}_r|}{n} = \frac{1-\lambda}{4}. \tag{11}$$

- Case 2. $C_i^r \notin \mathsf{Delay}_r$. Here, C_i^r was distributed by the adversary, added to $\mathsf{Tree}_{\mathsf{MC}}$ due to AddBlock, and then deleted due to DeleteBlock in round $r' \leqslant r$. Due to Lemma 1, $d_{\mathrm{tree}}^{r'} \geqslant |C_i^r|$ and $d_{\mathrm{tree}}^{r'} \leqslant d_{\mathrm{tree}}^r$; thus, we obtain $d_{\mathrm{tree}}^{r'} = |C_i^r| = d_{\mathrm{tree}}^r$. Therefore, there exists another branch C^* such that $|C_{\mathrm{tree}}^*| = d_{\mathrm{tree}}^r$ and C_{tree}^* is added to $\mathsf{Tree}_{\mathsf{MC}}$ before C_i^r. r^* denotes the round in which C_{tree}^* is added. C_{tree}^* is distributed by the adversary in round r^*; however, miner i did not update his state with C_{tree}^*. Thus, $C_i^{r^*}$ must be no shorter than C_{tree}^*. Therefore, $|C_i^{r^*}| = |C_i^r| = d_{\mathrm{tree}}^r$ and $C_i^{r^*} = C_i^r$. Thus, we conclude that C_i^r was created no later than r^* but was distributed in round $r' > r^*$, which means miner i was delayed in round r^* and $C_i^r \in \mathsf{Delay}_{r^*}$. Similarly, if $\textbf{Over-delay}_{r^*}$ does not occur, then $|\mathsf{Delay}_{r^*}| \leqslant \frac{(1-\lambda)n}{4}$. Here, we obtain

$$\Pr_i\left[C_i^r \notin \mathsf{Tree}_{\mathsf{MC}}^r \wedge \text{Case 2} | \overline{\textbf{Over-delay}_{r^*}}\right] \leqslant \Pr_i\left[C_i^r \in \mathsf{Delay}_{r^*} | \overline{\textbf{Over-delay}_{r^*}}\right] \leqslant \frac{|\mathsf{Delay}_{r^*}|}{n} = \frac{1-\lambda}{4}.$$

Therefore the probability of $C_i^r \notin \mathsf{Tree}_{\mathsf{MC}}^r$ conditioned by $\overline{\textbf{Over-delay}_r} \wedge \overline{\textbf{Over-delay}_{r^*}}$ is given as follows:

$$\begin{aligned}
&\Pr_i\left[C_i^r \notin \mathsf{Tree}_{\mathsf{MC}}^r | \overline{\textbf{Over-delay}_r} \wedge \overline{\textbf{Over-delay}_{r^*}}\right] \\
&\leqslant \Pr_i\left[C_i^r \notin \mathsf{Tree}_{\mathsf{MC}}^r \wedge \text{Case 1} | \overline{\textbf{Over-delay}_r}\right] + \Pr_i\left[C_i^r \notin \mathsf{Tree}_{\mathsf{MC}}^r \wedge \text{Case 2} | \overline{\textbf{Over-delay}_{r^*}}\right] \\
&\leqslant \frac{1-\lambda}{4} + \frac{1-\lambda}{4} = \frac{1-\lambda}{2}.
\end{aligned}$$

In any consecutive at least k blocks of any branch of $\mathsf{Tree}_{\mathsf{MC}}$ in round r, there are at most ρk blocks mined by corrupted miners; thus, we obtain $\texttt{quality}_k(r, C, \rho) = 1$ for all $C \in \mathsf{Tree}_{\mathsf{MC}}^r$. Suppose **Over-delay** does not occur in round r and r^*. Then, we obtain $\Pr_i[C_i^r \notin \mathsf{Tree}_{\mathsf{MC}}^r] \leqslant \frac{1-\lambda}{2}$. Therefore, we obtain

$$\Pr_i\left[\texttt{quality}_k(C_i^r, \rho) = | \overline{\textbf{Over-delay}_r} \wedge \overline{\textbf{Over-delay}_{r^*}}\right] \geqslant 1 - \frac{1-\lambda}{2} > \lambda, \tag{12}$$

which means if $\overline{\textbf{Over-delay}_r} \wedge \overline{\textbf{Over-delay}_{r^*}}$ occurs, we obtain $\texttt{chain-quality}_{A,Z,\kappa}^{(\Pi,\mathcal{C})}(r,\rho,k,\lambda) = 1$.

Consider the event $\overline{\textbf{Over-delay}_r} \wedge \overline{\textbf{Over-delay}_{r^*}}$. If $r^* \leqslant r - \Delta$, chains delayed at r^* are broadcast prior to round r. Since m_{delay}^r is determined by the number of miners delayed from rounds $r - \Delta + 1$ to r, here, $\textbf{Over-delay}_r$ and $\textbf{Over-delay}_{r^*}$ are independent events. If $r - \Delta < r^* < r$, the probability of $\textbf{Over-delay}_{r^*}$ will be less than when $\textbf{Over-delay}_r$ does not occur. Therefore, we obtain

$$\begin{aligned}
&\Pr\left[\texttt{chain-quality}_{A,Z,\kappa}^{(\Pi,\mathcal{C})}(r,\rho,k,\lambda)=1\right] \\
&\geqslant \Pr\left[\overline{\textbf{Over-delay}_r} \wedge \overline{\textbf{Over-delay}_{r^*}}\right] \\
&= 1 - \Pr[\textbf{Over-delay}_r] - \Pr\left[\textbf{Over-delay}_{r^*} | \overline{\textbf{Over-delay}_r}\right] \\
&\geqslant 1 - \Pr[\textbf{Over-delay}_r] - \Pr[\textbf{Over-delay}_{r^*}] \\
&\geqslant 1 - 2\mathrm{e}^{-\mathrm{poly}(\kappa)}.
\end{aligned}$$

This completes the proof.

5 Main theorems and proofs

In this section, we analyse the three basic properties of $(\Pi, \mathcal{C})$ using $\mathsf{Tree}_{\mathsf{MC}}$ in our model, where the adversary can perform the long delay attack and corrupt miners.

5.1 Chain growth

Theorem 1 (Chain growth)**.** Assume $1/2 < \lambda \leqslant 1 - 8\alpha p\Delta$. The blockchain protocol $(\Pi, \mathcal{C})$ has the chain growth rate $g = \frac{(1-\delta)f}{1+\alpha f\Delta}$ with majority λ, where $f = 1 - (1-p)^n \approx np$.

Proof. Due to Lemma 3, we only need to focus on the depth growth of $\mathsf{Tree}_{\mathsf{MC}}$. Given $\mathsf{Tree}_{\mathsf{MC}}$ during the execution of $(\Pi, \mathcal{C})$, in round r_0, the depth of $\mathsf{Tree}_{\mathsf{MC}}$ is $d^{r_0}_{\text{tree}}$. Assume the depth becomes $d^{r_0}_{\text{tree}} + t$ at round r_t; thus, we should prove that $\frac{t}{r_t - r_0} \geqslant g$ with overwhelming probability.

Without loss of generality, assume there is no miner delayed in round r_0; otherwise, select r'_0 such that there is no miner delayed in round r'_0 and $d^{r'_0}_{\text{tree}} = d^{r_0}_{\text{tree}}$. Obviously, $r'_0 < r_0$. In the following, we prove $\frac{t}{r_t - r'_0} \geqslant g$ and then we obtain $\frac{t}{r_t - r_0} > \frac{t}{r_t - r'_0} \geqslant g$.

For any integer i, there may be more than one block at position $d^{r_0}_{\text{tree}} + i$. Here, let B_i be the first block at position $d^{r_0}_{\text{tree}} + i$ mined by honest miners. Note that B_i may not exist because it is possible that all blocks at position i are mined by corrupted miners. Let r_i be the round in which the depth of $\mathsf{Tree}_{\mathsf{MC}}$ increases to $d^{r_0}_{\text{tree}} + i$. Rounds r_0 to r_t are divided into t periods, where period i denotes rounds from r_{i-1} to r_i Obviously, B_i can only be mined in period i, and each period i comprises a mining phase and a delay phase. The mining phase of period i begins when B_{i-1} is added to $\mathsf{Tree}_{\mathsf{MC}}$ and ends when B_i is mined. The rest of period i is the delay phase. Note that the mining phase of period 1 begins at round r_0. If B_i does not exist, the mining phase of period i is set to the entire period.

Let R^i_{mine} and R^i_{delay} denote the number of rounds of mining and delay phases of period i, respectively. Here, $R_{\text{mine}} = \Sigma^t_{i=1} R^i_{\text{mine}}$ and $R_{\text{delay}} = \Sigma^t_{i=1} R^i_{\text{delay}}$; thus, $R_{\text{mine}} + R_{\text{delay}} = r_t - r_0$.

To estimate R_{mine} and R_{delay}, we consider the extended periods. Period i is extended by not changing the mining process in period i; however, the adversary attempts to delay the distribution of chains of length $d^{r_0}_{\text{tree}} + i$ for as long as possible. Here, let R^{i*}_{mine} and R^{i*}_{delay} be the number of rounds in the mining and delay phases, respectively, in extended period i. It is obvious that $R^i_{\text{mine}} \leqslant R^{i*}_{\text{mine}}$ and $R^i_{\text{delay}} \leqslant R^{i*}_{\text{delay}}$.

Next, we analyse R^i_{mine}s and R^i_{delay}s separately.

At the beginning of the mining phase of extended period i, all honest miners have chains of length $d^{r_0}_{\text{tree}} + i - 1$. Here, let f be the probability that some honest miners mine successfully in a round. Thus, we obtain $f = 1 - (1-p)^n$. The mining phase of extended period i ends when B_i is mined. Therefore, R^{i*}_{mine} is distributed geometrically with probability f. Since all R^{i*}_{mine} are independent variables, the sum $R^*_{\text{mine}} = \Sigma^t_{i=1} R^*_{\text{mine}}$ follows negative binomial distribution $\mathsf{NB}(t, f)$. Due to Chernoff bound for negative Binomial distribution [15], we obtain

$$\Pr\left[R^*_{\text{mine}} \geqslant \frac{(1+\delta_1)t}{f}\right] < \mathrm{e}^{-\text{poly}_1(\delta_1^2 t)}, \tag{13}$$

where $0 < \delta_1 < 1/2$ and $\text{poly}_1(\delta_1^2 t) = \frac{\delta_1^2 t}{3f}$.

Here, $R^i_{\text{mine}} \leqslant R^{i*}_{\text{mine}}$; thus, we obtain $R_{\text{mine}} \leqslant R^*_{\text{mine}}$. Therefore, we obtain

$$\Pr\left[R_{\text{mine}} \geqslant \frac{(1+\delta_1)t}{f}\right] \leqslant \Pr\left[R^*_{\text{mine}} \geqslant \frac{(1+\delta_1)t}{f}\right] < \mathrm{e}^{-\text{poly}(\delta_1^2 t)}. \tag{14}$$

At the beginning of the delay phase of extended period i, block B_i is mined by an honest miner. With probability $1-\alpha$, B_i is undelayable, which implies that B_i will be broadcast immediately. Then, $R^{i*}_{\text{delay}} = 0$; otherwise, the adversary can delay B_i for at most Δ rounds. As a result, we obtain $E[R^{i*}_{\text{delay}}] \leqslant \alpha\Delta$.

Here, R^{i*}_{delay}s are independent variables with the same distribution. Let $R^*_{\text{delay}} = \Sigma^t_{i=1} R^i_{\text{delay}}$. Due to the Chernoff bound, we obtain

$$\Pr[R^*_{\text{delay}} \geqslant (1+\delta_2) t E[R^{i*}_{\text{delay}}]] < \mathrm{e}^{-\text{poly}_2(\delta_2^2 t)}, \tag{15}$$

where $0 \leqslant \delta_2 \leqslant 1/2$ and $\mathrm{poly}_2(\delta_2^2 t) = \frac{\delta_2^2 t E[R^{i*}_{\mathrm{delay}}]}{3}$.

Since $R^i_{\mathrm{delay}} \leqslant R^{i*}_{\mathrm{delay}}$, we obtain $R_{\mathrm{delay}} \leqslant R^*_{\mathrm{delay}}$. Thus, we obtain

$$\Pr[R_{\mathrm{delay}} \geqslant (1+\delta_2)tE[R^{i*}_{\mathrm{delay}}]] \leqslant \Pr[R^*_{\mathrm{delay}} \geqslant (1+\delta_2)tE[R^{i*}_{\mathrm{delay}}]] < \mathrm{e}^{-\mathrm{poly}_2(\delta_2^2 t)}. \tag{16}$$

According to inequalities (14) and (16), we obtain

$$\Pr\left[r_t - r_0 \leqslant \frac{(1+\delta_1)t}{f} + (1+\delta_2)tE[R^{i*}_{\mathrm{delay}}]\right] > 1 - \mathrm{negl}(t), \tag{17}$$

where $\mathrm{negl}(t) = \mathrm{e}^{-\mathrm{poly}_1(\delta_1^2 t)} + \mathrm{e}^{-\mathrm{poly}_2(\delta_2^2 t)}$ is a negligible function with t.

With sufficiently small δ_1, δ_2, pick $\delta > 0$ such that

$$\frac{1}{1-\delta}\left(\frac{1}{f} + E[R^{i*}_{\mathrm{delay}}]\right) = (1+\delta_1)\frac{1}{f} + (1+\delta_2)E[R^{i*}_{\mathrm{delay}}]. \tag{18}$$

Then, we obtain

$$\Pr\left[r_t - r_0 \leqslant \frac{t}{(1-\delta)f}(1 + fE[R^{i*}_{\mathrm{delay}}])\right] > 1 - \mathrm{negl}(t). \tag{19}$$

Therefore, we have

$$\Pr\left[t \geqslant \frac{(1-\delta)f}{1 + fE[R^{i*}_{\mathrm{delay}}]}(r_t - r_0)\right] > 1 - \mathrm{negl}(t). \tag{20}$$

Since $E[R^{i*}_{\mathrm{delay}}] \leqslant \alpha\Delta$, we obtain

$$\Pr\left[t \geqslant \frac{(1-\delta)f}{1 + \alpha f\Delta}(r_t - r_0)\right] > 1 - \mathrm{negl}(t). \tag{21}$$

Due to Lemma 3, $\texttt{chain-increase}^{(\Pi,\mathcal{C})}_{A,Z,\kappa}(r_1, r_2, \lambda) \geqslant d^{r_t}_{\mathrm{tree}} - d^{r_0}_{\mathrm{tree}}$ with probability at least $1 - 2\mathrm{e}^{-\mathrm{poly}(\kappa)}$. Due to inequality (21), $d^{r_t}_{\mathrm{tree}} - d^{r_0}_{\mathrm{tree}} \geqslant g(r_t - r_0)$ with probability at least $1 - \mathrm{negl}(t)$. Thus, we obtain

$$\Pr\left[\texttt{chain-increase}^{(\Pi,\mathcal{C})}_{A,Z,\kappa}(r_0, r_t, \lambda) \geqslant g(r_t - r_0)\right] \geqslant 1 - 2\mathrm{e}^{-\mathrm{poly}(\kappa)} - \mathrm{negl}(t), \tag{22}$$

which completes the proof of Theorem 1.

5.2 Common prefix

For the common prefix, the proof in [15] does not hold in our corrupted miner setting because it is difficult to analyze the convergence event. To address this issue, we introduce a new probability experiment called the Bernoulli race.

Definition 7 (Bernoulli race experiment)**.** Let X_i, Y_i be independent random variables. For integer i, $X_i = 1$ with probability p (otherwise, $X_i = 0$), while $Y_i = 1$ with probability q (otherwise, $Y_i = 0$). Here, let $R = \Sigma_{i=1}^k X_i$, where k denotes the first i such that $Y_i = 1$. Variable R is said to have a Bernoulli race distribution $\mathsf{BR}(p, q)$.

We then consider the sum of variables that have Bernoulli race distribution with the same parameters.

Definition 8 (Repeated Bernoulli race)**.** Here, $R_1, R_2, \ldots, R_k$ are independent random variables. For $i \in [k]$, R_i has a Bernoulli race distribution $\mathsf{BR}(p, q)$. Let $R = \sum_{i=1}^k R_i$. Variable R is considered to have a repeated Bernoulli race distribution $\mathsf{RBR}(k, p, q)$.

Lemma 6. $\mathsf{RBR}(k, p, q) = \mathsf{NB}(k, \frac{q}{1-(1-p)(1-q)}) + \mathsf{B}(k, p) - k$.

Proof. Here, let X_i, Y_i be the independent random variables in a Bernoulli race experiment, and $\Pr[X_i] = p$ and $\Pr[Y_i] = q$. Then, the joint distribution of (X_i, Y_i) is given as follows:

$$(X_i, Y_i) = \begin{cases} (1,1), & \text{with probability } pq, \\ (1,0), & \text{with probability } p(1-q), \\ (0,1), & \text{with probability } q(1-p), \\ (0,0), & \text{with probability } (1-p)(1-q). \end{cases} \tag{23}$$

The counter variable counts the number of events $X_i = 1$. If $(X_i, Y_i) = (1, 0)$, counter = counter + 1. Here, if $(X_i, Y_i) = (0, 1)$, counter is output. If $(X_i, Y_i) = (1, 1)$, counter + 1 is output. Thus, the output of counter follows the distribution p $\mathsf{BR}(p, q)$.

Since event $(X_i, Y_i) = (0, 0)$ has no effect on counter, we focus on the conditional probability under $(X_i, Y_i) \neq (0, 0)$. The probability distribution of (X_i, Y_i) conditioned on $(X_i, Y_i) \neq (0, 0)$ is given as follows:

$$(X_{j_i}, Y_{j_i}) = \begin{cases} (1,1), & \text{with probability } \dfrac{pq}{1-(1-p)(1-q)}, \\ (1,0), & \text{with probability } \dfrac{p(1-q)}{1-(1-p)(1-q)}, \\ (0,1), & \text{with probability } \dfrac{q(1-p)}{1-(1-p)(1-q)}. \end{cases} \tag{24}$$

Here, let $Z_i = 1$ if $Y_{j_i} = 1$; otherwise, $Z_i = 0$. Then, we obtain

$$Z_i = \begin{cases} 1, & \text{with probability } \dfrac{q}{1-(1-p)(1-q)}, \\ 0, & \text{with probability } \dfrac{p(1-q)}{1-(1-p)(1-q)}. \end{cases} \tag{25}$$

Consider the series of Z_i, which follows Bernoulli distribution with probability $\frac{q}{1-(1-p)(1-q)}$. When $Z_i = 0$, counter = counter + 1. When $Z_i = 1$, counter is output if $X_{j_i} = 0$ or counter + 1 is output if $X_{j_i} = 1$. Here, $\mathsf{G}(p)$ denotes a geometric distribution with probability p. Before counter is output, the number of events that $Z_i = 0$ follows the distribution $\mathsf{G}(\frac{q}{1-(1-p)(1-q)}) - 1$. Therefore, the probability distribution of the output of counter is obtained as follows:

$$\text{counter} \sim \begin{cases} \mathsf{G}\left(\dfrac{q}{1-(1-p)(1-q)}\right), & \text{with probability } p, \\ \mathsf{G}\left(\dfrac{q}{1-(1-p)(1-q)}\right) - 1, & \text{with probability } 1-p. \end{cases} \tag{26}$$

Here, $\mathsf{TP}(p)$ denotes a Bernoulli distribution with probability p. Then, we obtain

$$\text{counter} \sim \mathsf{G}\left(\frac{q}{1-(1-p)(1-q)}\right) + \mathsf{TP}(p) - 1. \tag{27}$$

This means that $\mathsf{BR}(p, q) = \mathsf{G}(\frac{q}{1-(1-p)(1-q)}) + \mathsf{TP}(p) - 1$. Moreover, $\mathsf{RBR}(k, p, q)$ is the sum of k independent random variables with distribution $\mathsf{BR}(p, q)$. Here, we obtain $\mathsf{RBR}(k, p, q) = \mathsf{NB}(k, \frac{q}{1-(1-p)(1-q)}) + \mathsf{B}(k, p) - k$ because the negative binomial distribution is the sum of the independent identical geometric distribution, and the binomial distribution is the sum of the independent identical Bernoulli distribution.

Lemma 7. Here, let $R_1, R_2, \ldots, R_n$ be independent random variables with identical distribution such that $R_i \sim \mathsf{RBR}(k, p, q)$ for all $i \in [n]$. In addition, let $R = \sum_{i=1}^{n} R_i$. Then, we obtain $R \sim \mathsf{RBR}(kn, p, q)$.

The proof of Lemma 7 is obvious due to Lemma 6; thus, this proof is omitted.

Next, we investigate the structure of $\mathsf{Tree}_{\mathsf{MC}}$. If there is a fork in $\mathsf{Tree}_{\mathsf{MC}}$, we define the fork depth as follows.

Definition 9 (Fork depth)**.** Given $\mathsf{Tree}_{\mathsf{MC}}$ at round r, B is the latest block in $\mathsf{Tree}_{\mathsf{MC}}$ such that

(1) B is the only block with depth d;

(2) B is mined by an honest miner.

Suppose the depth of $\mathsf{Tree}_{\mathsf{MC}}$ is $d + T$. Here, $\mathsf{Tree}_{\mathsf{MC}}$ has fork depth T, and B denotes the fork block of $\mathsf{Tree}_{\mathsf{MC}}$.

Thus, the depth can be considered the estimation of the length of the longest fork in $\mathsf{Tree}_{\mathsf{MC}}$. The key to the proof of the common prefix is given in Lemma 8.

Lemma 8. Assume $(1+\alpha np\Delta)\mu < (1-\eta)(1-np(1+\alpha\Delta))$ for constant η. Here, $\mathsf{Tree}_{\mathsf{MC}}$ has fork depth T with probability $\mathrm{negl}(T)$, where negl is a negligible function.

Proof. Here, the main idea is to prove that the rate of mining by corrupted miners is less than the rate of "convergence" with overwhelming probability.

Suppose $\mathsf{Tree}_{\mathsf{MC}}$ has fork depth T. The adversary must create a fork from depth $d+1$ to $d+T$ on $\mathsf{Tree}_{\mathsf{MC}}$. We divide set $W=\{d+1,d+2,\ldots,d+T\}$ into two subsets W_1, W_2. Here, $W_1=\{d+t|t\in[T]$, there is a block mined by honest miners at depth $d+t\}$, and $W_2=W\setminus W_1$. Note that T_1 and T_2 denote the number of elements in W_1 and W_2, respectively.

We also divide rounds during which the depth of $\mathsf{Tree}_{\mathsf{MC}}$ increases from d to $d+T$ into T periods, as in the proof of chain growth. We then analyze the upper bound of the number of blocks mined by corrupted miners over T rounds.

(1) Consider t such that $d+t\in W_1$. Let B_t be the first block mined by honest miners in period t. The mining phase ends at the round wherein B_t is mined, and then the delay phase starts. Here, R^t_{mine} and R^t_{delay} denote the number of rounds in the mining and delay phases, respectively.

In the mining phase, a corrupt miner i can mine blocks until the first block is mined by honest miners. The probability of miner i mining a block is p in each round, while the probability of honest miners mining a block is $f=1-(1-p)^n$. Here, let $S^{t,i}_{\text{mine}}$ be the number of blocks mined by miner i in the mining phase of period $d+t$. Therefore, we obtain $S^{t,i}_{\text{mine}}\sim\mathsf{BR}(p,f)$.

Since $S^{t,i}_{\text{mine}}$ are independently identically distributed, we obtain $S^t_{\text{mine}}=\sum_{i=1}^{\mu n}S^{t,i}_{\text{mine}}$ and $S^t_{\text{mine}}\sim\mathsf{RBR}(\mu n,p,f)$,where S^t_{mine} is the number of blocks mined by all corrupted miners.

Here, $S_{\text{mine}}=\sum^{t\in W_1}S^t_{\text{mine}}$ is the number of the blocks mined by corrupted miners in all mining phases in period $t\in W_1$. According to Lemma 7, we obtain $S_{\text{mine}}\sim\mathsf{RBR}(\mu nT_1,p,f)$.

(2) In the delay phase, $R^t_{\text{delay}}=0$ if B_t is undelayable. If B_t is delayable, then $R_{\text{delay}}\leqslant\Delta$. Here, let $R_{\text{delay}}=\sum^{t\in W_1}R^t_{\text{delay}}$. As in the proof of (16), we obtain

$$\Pr[R_{\text{delay}}>(1+\delta_1)T_1E[R^i_{\text{delay}}]]<\mathrm{e}^{-\frac{\delta_1^2T_1E[R^i_{\text{delay}}]}{3}},\quad\text{for } 0\leqslant\delta_1\leqslant 1. \tag{28}$$

$E[R^i_{\text{delay}}]\leqslant\alpha\Delta$ and $T_1\leqslant T$; thus, we obtain

$$\Pr[R_{\text{delay}}>(1+\delta_1)T\alpha\Delta]<\mathrm{e}^{-\frac{\delta_1^2T\alpha\Delta}{3}},\quad\text{for } 0\leqslant\delta_1\leqslant 1. \tag{29}$$

Here, we select a sufficiently small δ_1 such that $(1+\delta_1)T\alpha\Delta$ is an integer. Let **bad1** be an event where $R_{\text{delay}}>(1+\delta_1)T\alpha\Delta$. If **bad1** does not occur, R_{delay} is less than $(1+\delta_1)T\alpha\Delta$ rounds. In these rounds, the adversary can also use corrupted miners to mine blocks. Since each miner can mine successfully with probability p, μn corrupted miners have at most $(1+\delta_1)T\alpha\Delta\mu n$ opportunities to mine a block. Let S_{delay} be random variable such that $S_{\text{delay}}\sim\mathsf{B}((1+\delta_1)T\alpha\Delta\mu n,p)$, which can be considered the upper bound of the number of blocks mined by corrupted miners in the delay phases.

(3) Consider t such that $d+t\in W_2$. Here, we cannot divide rounds into two phases because no honest miner succeeds. In other words, in period t, corrupted miner i can only mine blocks before an honest miner succeeds. Let $S^{t,i}_{\text{mine}*}$ be the number of blocks mined by miner i in period t. We extend period t as long as possible to obtain the upper bound of $S^{t,i}_{\text{mine}*}$. In fact, when period t is extended until an honest miner succeeds and the upper bound of $S^{t,i}_{\text{mine}*}$ has a Bernoulli race distribution $\mathsf{BR}(p,f)$. Similarly, $S_{\text{mine}*}$ denotes the upper bound of the number of blocks mined by all corrupted miners. Thus, we obtain $S_{\text{mine}*}\sim\mathsf{RBR}(\mu nT_2,p,f)$.

(4) In addition, corrupted miners can mine blocks prior to the T periods. Here, assume B is the fork block of $\mathsf{Tree}_{\mathsf{MC}}$, and the position of B is d. Since B is mined by an honest miner, it may be delayed by the adversary. If so, corrupted miners can mine blocks after B until period 1 starts. Suppose that the number of blocks mined in this manner is S_{pre}. Thus, we obtain $S_{\text{pre}}\sim\mathsf{B}(\mu n\Delta,p)$.

Let S^T be the upper bound of the number of blocks mined by corrupted miners in T periods. Thus, we obtain

$$\begin{aligned}S^T&=S_{\text{mine}}+S_{\text{Delay}}+S_{\text{mine}*}+S_{\text{pre}}\\&\sim\mathsf{RBR}(\mu nT_1,p,f)+\mathsf{B}((1+\delta_1)T\alpha\Delta\mu n,p)+\mathsf{RBR}(\mu nT_2,p,f)+\mathsf{B}(\mu n\Delta,p)\end{aligned}$$

$$\begin{aligned}
&\sim \mathsf{RBR}(\mu nT, p, f) + \mathsf{B}((1+\delta_1)T\alpha\Delta\mu n, p) + \mathsf{B}(\mu n\Delta, p)\\
&\sim \mathsf{NB}\left(\mu nT, \frac{f}{1-(1-p)(1-f)}\right) + \mathsf{B}((((1+\delta_1)\alpha T+1)\Delta+T)\mu n, p) - \mu nT\\
&\sim \mathsf{NB}\left(\mu nT, \frac{1-(1-p)^n}{1-(1-p)^{n+1}}\right) + \mathsf{B}((((1+\delta_1)\alpha T+1)\Delta+T)\mu n, p) - \mu nT.
\end{aligned}$$

Let $S_1^T \sim \mathsf{NB}(\mu nT, \frac{1-(1-p)^n}{1-(1-p)^{n+1}})$. Therefore, according to the Chernoff bound, we obtain

$$\Pr[S_1^T > (1+\delta_2)\mu T(n+1)] < \mathrm{e}^{-\frac{\delta_2^2\mu T(n+1)}{3}}, \tag{30}$$

where we use $\frac{1-(1-p)^n}{1-(1-p)^{n+1}} \approx \frac{np}{(n+1)p} = \frac{n}{n+1}$ to obtain the inequality above. Here, an event where $S_1^T > (1+\delta_2)\mu T(n+1)$ is denoted **bad2**.

Let $S_2^T \sim \mathsf{B}((((1+\delta_1)\alpha T+1)\Delta+T)\mu n, p)$. According to the Chernoff bound, we obtain

$$\Pr[S_2^T > (1+\delta_3)\alpha\Delta\mu npT] < \mathrm{e}^{-\frac{\delta_3^2\alpha\Delta\mu npT}{3}}. \tag{31}$$

An event where $S_2^T > (1+\delta_3)\alpha\Delta\mu npT$ is denoted **bad3**.

Therefore, if **bad1**, **bad2**, and **bad3** do not occur, we have

$$\begin{aligned}
S^T &= S_1^T + S_2^T - \mu nT\\
&\leqslant (1+\delta_2)\mu T(n+1) + (1+\delta_3)\alpha\Delta\mu npT - \mu nT\\
&\leqslant (1+\delta_4)\mu T(1+\alpha np\Delta),
\end{aligned}$$

where δ_2, δ_3 are selected as sufficiently small.

In addition, we focus on an event called **converge**. Here, random variable $\mathbf{converge}_i = 1$ denotes an event where **converge** occurs in period $i \in W_1$. Let B_i be the first block mined by honest miners in period i and suppose B_i is mined in round r_i. For $i \in W_1$, $\mathbf{converge}_i = 1$ in period i implies the following:

(1) Only one honest miner mines successfully in round r_i.

(2) The chain in which B_i lies is undelayable or is delayable while there is no new block mined by honest miners in following Δ rounds.

If $\mathbf{converge}_i = 1$, B_i is the only block mined by honest miners in period i. In addition, B_i can be the only block at position $d+i$ mined by honest miners because an honest miner can only mine a block at position $d+i$ in period i. If $\mathbf{converge}_i = 1$ and corrupted miners do not mine a block at position $d+i$, B_i will become the only block at position $d+i$. As a result, there will be no fork when the depth of $\mathsf{Tree}_{\mathsf{MC}}$ increases to $d+t$.

Note that the above two conditions of **converge** are independent. Since $i \in W_i$, B_i exists; thus, the probability of condition (1) is obtained as follows:

$$\frac{np(1-p)^{n-1}}{1-(1-p)^n} > \frac{np(1-p)^{n-1}}{np} = (1-p)^{n-1} > 1-np. \tag{32}$$

The probability of condition (2) is expressed as follows:

$$1-\alpha+\alpha(1-p)^{n\Delta} > 1-\alpha+\alpha(1-np\Delta) = 1-\alpha np\Delta. \tag{33}$$

Since these two conditions are independent, we obtain

$$\Pr[\mathbf{converge}_i = 1] > (1-np)(1-\alpha np\Delta) > 1-np(1+\alpha\Delta). \tag{34}$$

To create a fork from depth $d+1$ to $d+T$, the adversary must use corrupted miners to mine at least one block at position $d+t$ such that $\mathbf{converge}_t = 1$. In addition, since there is no block mined by honest miners at position $d+t$ such that $t \in W_2$, the adversary must also mine at least one block at such a

position. Here, D^T denotes the lower bound of the number of blocks mined by an adversary in T periods. Thus, we obtain

$$D^T \sim \mathsf{B}(T_1, 1 - np(1+\alpha\Delta)) + T_2. \tag{35}$$

To obtain the lower bound, let $D_1^T \sim \mathsf{B}(T_1, 1-np(1+\alpha\Delta))$ and $D_2^T \sim \mathsf{B}(T_2, 1-np(1+\alpha\Delta))$. Here, it is obvious that $D_2^T < T_2$. According to the Chernoff bound, we obtain the following:

$$\Pr[D_1^T + D_2^T < (1-\delta_5)(T_1+T_2)(1-np(1+\alpha\Delta))] < \mathrm{e}^{-\frac{(1+\delta_5^2)(T_1+T_2)(1-np(1+\alpha\Delta))}{3}}. \tag{36}$$

An event where $D_1^T + D_2^T < (1-\delta_5)(T_1+T_2)(1-np(1+\alpha\Delta))$ is denoted **bad4**.

Therefore, if **bad4** does not occur, we obtain

$$D^T \geqslant D_1^T + D_2^T \geqslant (1-\delta_5)(1-np(1+\alpha\Delta))T. \tag{37}$$

Here, the adversary has successfully created a fork from depth $d+1$ to $d+T$; thus, we obtain $S_T \geqslant D_T$. If all "bad" events do not happen, we obtain the following:

$$(1+\delta_4)\mu T(1+\alpha np\Delta) \geqslant (1-\delta_5)(1-np(1+\alpha\Delta))T. \tag{38}$$

Thus, there exist some $\eta' > 0$ such that

$$(1+\alpha np\Delta)\mu \geqslant (1-\eta')(1-np(1+\alpha\Delta)), \tag{39}$$

where η' is determined by δ_1, δ_2, δ_3. We pick these parameters properly such that $\eta' = \eta$. As a result, the inequality (39) contradicts the assumption $(1+\alpha np\Delta)\mu < (1-\eta)(1-np(1+\alpha\Delta))$.

Therefore, at least one of the four "bad" events occurs. Thus, the probability that $\mathsf{Tree}_{\mathsf{MC}}$ will have fork depth T is at most $\Pr[\mathbf{bad1}] + \Pr[\mathbf{bad2}] + \Pr[\mathbf{bad3}] + \Pr[\mathbf{bad4}]$, which is expressed as follows:

$$\exp\left(-\frac{\delta_1^2\alpha\Delta T}{3}\right) + \exp\left(-\frac{\delta_2^2\mu T(n+1)}{3}\right) + \exp\left(-\frac{\delta_3^2\alpha\Delta\mu np T}{3}\right) + \exp\left(-\frac{(1+\delta_5^2)(1-np(1+\alpha\Delta))}{3}T\right), \tag{40}$$

where the four terms in (40) are negligible in $\alpha\Delta T$, μnT, $\alpha\mu np\Delta T$, and $\alpha\mu np\Delta T$, respectively, and the last inequality follows from the assumption. In our model, we have $\Delta = O(1/np)$; thus, Eq. (40) can be expressed as $\mathrm{negl}(\alpha\mu T) = \mathrm{negl}'(T)$, where negl and negl$'$ are negligible functions. This completes the proof.

Theorem 2. Assume $1/2 < \lambda \leqslant 1 - 8\alpha p\Delta$ and $(1+\alpha np\Delta)\mu < (1-\eta)(1-np(1+\alpha\Delta))$ for constant η. Here, the blockchain protocol $(\Pi, \mathcal{C})$ satisfies the common prefix property with parameter λ.

Proof. Due to Lemma 8, the probability that $\mathsf{Tree}_{\mathsf{MC}}$ has fork depth T is $\mathrm{negl}'(T)$. In other words, if the depth of $\mathsf{Tree}_{\mathsf{MC}}$ is d, all branches in $\mathsf{Tree}_{\mathsf{MC}}$ have a common prefix of length $d-T$ with probability $1 - \mathrm{negl}'(T)$. Due to Lemma 4, the view of $(\Pi, \mathcal{C})$ satisfies $\texttt{common-prefix}^{(\Pi,\mathcal{C})}_{A,Z,\kappa}(r, T, \lambda) = 1$ with a probability of at least $1 - 2\mathrm{e}^{-\mathrm{poly}(\kappa)}$. Therefore, given the view of $(\Pi, \mathcal{C})$, we obtain

$$\Pr\left[\texttt{common-prefix}^{(\Pi,\mathcal{C})}_{A,Z,\kappa}(r, T, \lambda) = 1\right] \geqslant 1 - 2\mathrm{e}^{-\mathrm{poly}(\kappa)} - \mathrm{negl}'(T),$$

which completes the proof of Theorem 2.

5.3 Chain quality

Theorem 3. Assume $1/2 < \lambda \leqslant 1 - 8\alpha p\Delta$. Here, the blockchain protocol $(\Pi, \mathcal{C})$ satisfies the chain quality property at round r with parameters $\rho = (1+\epsilon)(\frac{1}{g} + \frac{\Delta}{k})\mu np$ and majority λ.

Proof. Let $B_{j+1}, \ldots, B_{j+k}$ be consecutive k blocks of a branch C of $\mathsf{Tree}_{\mathsf{MC}}$. We say that B is honest if it is mined by an honest miner or B is corrupted if it is mined by a corrupted miner. Here, the goal is to prove that the fraction of corrupted blocks in $B_{j+1}, \ldots, B_{j+k}$ is at most $\rho = (1+\delta)(\frac{1}{g} + \frac{\Delta}{k})\mu np$ under the condition that B_j and B_{j+k+1} are honest. If B_j or B_{j+k+1} is honest, we can extend $B_{j+1}, \ldots, B_{j+k}$ to

$B_{j'}, \ldots, B_{j''}$ such that $j' \leqslant j+1 < j+k \leqslant j''$ and $B_{j'}$ and $B_{j''}$ are honest, and the fraction of corrupted blocks among $B_{j'}, \ldots, B_{j''}$ is greater than that of $B_{j+1}, \ldots, B_{j+k}$.

Here, r_l denotes the round in which B_l is mined. Since B_j is honest, B_j must be broadcast no later than round $r_j + \Delta$. As a result, $d_{\text{tree}}^{r_j+\Delta} \geqslant j$. In contrast, B_{j+k+1} is honest; thus, the longest chain broadcast prior to r_{j+k+1} is of length $j+k$, which means that $d_{\text{tree}}^{r_{j+k+1}-1} = j+k$. Therefore, the depth of $\mathsf{Tree}_{\mathsf{MC}}$ increases by at most k from round $r_j + \Delta$ to round $r_{j+k+1} - 1$. Due to Theorem 1, the depth of $\mathsf{Tree}_{\mathsf{MC}}$ increases by at least $g(r_{j+k+1} - 1 - r_j - \Delta)$ from round $r_j + \Delta$ to round $r_{j+k+1} - 1$ with probability $1 - \text{negl}(k)$. Thus, we obtain

$$g(r_{j+k+1} - 1 - r_j - \Delta) < k, \tag{41}$$

and

$$r_{j+k+1} - 1 - r_j < \frac{k}{g} + \Delta \tag{42}$$

with probability $1 - \text{negl}(k)$.

In addition, the sequence of blocks $B_{j+1}, \ldots, B_{j+k}$ can only be mined from round r_j to round $r_{j+k+1}-1$. Here, let k_{corrupt} be the number of corrupted blocks mined from round r_j to round $r_{j+k+1} - 1$. Due to the Chernoff bound, we obtain

$$\Pr[k_{\text{corrupt}} > (1+\delta)(r_{j+k+1} - 1 - r_j)\mu np] < \mathrm{e}^{-\frac{\delta^2(r_{j+k+1}-1-r_j)\mu np}{3}}. \tag{43}$$

Note that all k blocks are mined in the $r_{j+k+1} - 1 - r_j$ rounds. According to the Chernoff bound, we obtain

$$\Pr[k > (1+\delta')(r_{j+k+1} - 1 - r_j)(1+\mu)np] < \mathrm{e}^{-\frac{\delta'^2(r_{j+k+1}-1-r_j)(1+\mu)np}{3}} < \mathrm{e}^{-\frac{\delta'^2}{3}k}. \tag{44}$$

This means that $(r_{j+k+1} - 1 - r_j)np \geqslant \frac{k}{(1+\delta')(1+\mu)}$ with a probability of at least $1 - \mathrm{e}^{-\frac{\delta'^2 k}{3}}$. Thus, in (43), $\mathrm{e}^{-\frac{\delta^2(r_{j+k+1}-1-r_j)\mu np}{3}} \leqslant \mathrm{e}^{-\frac{\delta^2\mu}{3(1+\delta')(1+\mu)}k} + \mathrm{e}^{-\frac{\delta'^2}{3}k}$, which is considered the negligible function $\text{negl}'(k)$.

From Eqs. (42) and (43), there are at most $(1+\delta)(\frac{k}{g}+\Delta)\mu np = \rho k$ corrupted blocks in $B_{j+1}, \ldots, B_{j+k}$ with probability at least $1 - \text{negl}(k) - \text{negl}'(k)$. Therefore, due to Lemma 5, we obtain

$$\Pr\left[\texttt{chain-quality}_{A,Z,\kappa}^{(\Pi,\mathcal{C})}(r,\rho,k,\lambda) = 1\right] \geqslant 1 - \mathrm{e}^{-\text{poly}(\kappa)} - \text{negl}(k) - \text{negl}'(k), \tag{45}$$

which completes the proof.

Acknowledgements Quan YUAN and Puwen WEI were supported by National Natural Science Foundation of China (Grant No. 61502276). Keting JIA was supported by National Key Research and Development Program of China (Grant No. 2017YFA0303903), National Cryptography Development Fund (Grant No. MMJJ20170121), and Zhejiang Province Key R&D Project (Grant No. 2017C01062). Haiyang XUE was supported by National Natural Science Foundation of China (Grant No. 61602473) and National Cryptography Development Fund (Grant No. MMJJ20170116). We would like to thank the anonymous reviewers for their insightful comments.

References

1 Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. 2008. https://bitcoin.org/en/bitcoin-paper
2 Decker C, Wattenhofer R. Information propagation in the bitcoin network. In: Proceedings of the 13th IEEE International Conference on Peer-to-Peer Computing, 2013. 1–10
3 Eyal I, Sirer E G. Majority is not enough: bitcoin mining is vulnerable. In: Proceedings of International Conference on Financial Cryptography and Data Security, 2014. 436–454
4 Bonneau J, Miller A, Clark J, et al. SoK: research perspectives and challenges for Bitcoin and cryptocurrencies. In: Proceedings of IEEE Symposium on Security and Privacy, 2015. 104–121
5 Göbel J, Keeler H P, Krzesinski A E, et al. Bitcoin blockchain dynamics: the selfish-mine strategy in the presence of propagation delay. Perform Eval, 2016, 104: 23–41
6 Sompolinsky Y, Zohar A. Secure high-rate transaction processing in bitcoin. In: Proceedings of Financial Cryptography and Data Security, 2015. 507–527
7 Sapirshtein A, Sompolinsky Y, Zohar A. Optimal selfish mining strategies in bitcoin. In: Proceedings of Financial Cryptography and Data Security, 2016. 515–532

8 Gervais A, Karame G O, Wust K, et al. On the security and performance of proof of work blockchains. In: Proceedings of ACM SIGSAC Conference on Computer and Communications Security, 2016. 3–16
9 Kiayias A, Koutsoupias E, Kyropoulou M, et al. Blockchain mining games. In: Proceedings of ACM Conference on Economics and Computation, 2016. 365–382
10 Natoli C, Gramoli V. The balance attack against proof-of-work blockchains: the R3 testbed as an example. In: Proceedings of Computing Research Repository, 2016
11 Nayak K, Kumar S, Miller A, et al. Stubborn mining: generalizing selfish mining and combining with an eclipse attack. In: Proceedings of IEEE European Symposium on Security and Privacy, 2016. 305–320
12 Garay J, Kiayias A, Leonardos N. The bitcoin backbone protocol: analysis and applications. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2015. 281–310
13 Wu Y B, Fan H N, Wang X Y, et al. A regulated digital currency. Sci China Inf Sci, 2019, 62: 032109
14 Pass R, Seeman L, shelat A. Analysis of the blockchain protocol in asynchronous networks. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2017. 643–673
15 Wei P W, Yuan Q, Zheng Y L. Security of the blockchain protocol against long delay attack. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, 2018. 250–275