On the Unfairness of Blockchain

Rachid Guerraoui[1] and Jingjing Wang[1]

École Polytechnique Fédérale de Lausanne, IC, Station 14, CH-1015, Lausanne, Switzerland
firstname.lastname@epfl.ch

Abstract. The success of *Bitcoin* largely relies on the perception of a *fair* underlying peer-to-peer protocol: *blockchain*. Fairness here essentially means that the reward (in bitcoins) given to any participant that helps maintain the consistency of the protocol by *mining*, is proportional to the computational power devoted by that participant to the mining task. Without such perception of fairness, honest miners might be disincentivized to maintain the protocol, leaving the space for dishonest miners to reach a majority and jeopardize the consistency of the entire system.
We prove, in this paper, that blockchain is actually *unfair*, even in a distributed system of only *two honest miners*. In a realistic setting where message delivery is not instantaneous, the ratio between the (expected) number of blocks committed by two miners is at least *exponential* in the product of the message delay and the difference between the two miners' hashrates. To obtain our result, we model the growth of blockchain, which may be of independent interest. We also apply our result to explain recent empirical observations and vulnerabilities.

1 Introduction

At the heart of the celebrated *Bitcoin* currency and payment system [1, 2, 3] lies a distributed protocol called *blockchain*, now considered of independent interest [4]. Essentially, this protocol maintains a distributed data structure, also called *the blockchain*, made of a series of transaction *blocks*, and updated by specific nodes called *miners*. To update a blockchain, a miner $\mathcal{M}$ devotes computational resources into a task called "proof-of-work" [5] in order to *mine* a block. Each block mined by $\mathcal{M}$ includes an extra *coinbase* transaction to reward $\mathcal{M}$'s computational effort with bitcoins [6]. The computational power of any miner is characterized by a *hashrate*, λ, meaning that, on average, it takes $\frac{1}{\lambda}$ units of time to mine a block. Once a block is mined, the block is propagated to the other miners. Roughly speaking, the block is said to be *committed* when it is delivered to all miners.

The success of Bitcoin relies on the perception of *fairness* [7]: in short, the reward of a given honest miner $\mathcal{M}$ is proportional to $\mathcal{M}$'s hashrate [8, 9]. Fairness in this sense is crucial, for otherwise (i.e., if the reward of an honest miner were lower than its fair proportion), honest miners could be disincentivized and stop maintaining the blockchain, leaving the space for dishonest miners (the

proportion of which could then grow to a majority) to jeopardize the correct functioning of the entire system [10].

We prove in this paper that Bitcoin blockchain is actually *unfair*. The fundamental reason is simple: blockchain is a distributed protocol, meaning that message delivery is not instantaneous. We show that with non-nil message delays, in a distributed system of *two honest miners*, $\mathcal{M}_1$ and $\mathcal{M}_2$ with hashrates λ_1 and λ_2, respectively, and message delay u between them, if $\lambda_1 > \lambda_2$, then $\mathcal{M}_1$ can commit many more blocks than $\mathcal{M}_2$ in expectation. The ratio between the expected number of blocks committed by $\mathcal{M}_1$ and $\mathcal{M}_2$ is lower bounded by $\frac{e^{\lambda_1 u}\lambda_1}{e^{\lambda_2 u}\lambda_2}$, which is *exponential* in the product of the message delay and the difference between the hashrates of $\mathcal{M}_1$ and $\mathcal{M}_2$.

To establish our lower bound, we go beyond most previous theoretical analyses of blockchain that take communication delays into account, typically assuming a delay rate (the percentage of miners receiving a certain message) [11], or dividing time into discrete steps (which may be viewed as an approximation of continuous time) [8, 12]. We rather model time in a *continuous way* to relate hashrates to communication delays, and model exactly how the blockchain grows in time (which may be of independent interest). We construct our proof of the lower bound in two steps: we first establish (1) the unfairness per se, and then (2) the exponential advantage. The key to our proof is the probability distribution of when the blockchain grows so that the blockchain includes k blocks for any $k \in \mathbb{Z}^+$. We show that the miner with a higher hashrate can grow its chain earlier for any k with a higher probability, implying unfairness, as the first step. In the second step, we extract the exponential term from the fact that no block is mined during some message delay u.

Our result on unfairness of blockchain among honest miners has several applications. For instance, our result explains disproportionate rewards reported via empirical experiments on blockchain [13, 14]. It also implies a trade-off between the speed to mine a block and the fairness of committed blocks in blockchain as well as its variants (such as Bitcoin-NG [13], and GHOST [15][1]): namely, the legitimate temptation to increase the throughput of blockchain by reducing its mining time can however cause even more unfairness. Our result can also help extend previous results on the benefits of selfish miners (which are dishonest, deviating from the mining algorithm to maximize their rewards). Indeed, Eyal and Sirer [11] (and follow-up work [16, 9]) showed that a *selfish* miner $\mathcal{M}$'s reward can be more than its fair share if the proportion α of $\mathcal{M}$'s hashrate, among all hashrates, passes the threshold $th = \frac{1}{3}$ (which is a sufficient condition). Their result assumed, however, no message delay. In a setting with message delays, we show a lower bound $\mathcal{L}$ such that, for selfish mining to be feasible, it is necessary that $th > \mathcal{L}$, where $\mathcal{L}$ is a function of u, λ_1 and λ_2. (For reasonable message delays measured for the Bitcoin blockchain implementation [17], $\mathcal{L} > \frac{1}{3}$.) Another application of our result is the unfairness of blockchain in the context of two clusters of miners with some negligible message delay within each cluster and a larger message delay between the two clusters. In this case, we show that even

[1] In the case of two honest miners, GHOST is equivalent to Bitcoin blockchain [15].

between two miners with the same hashrate, blockchain can favor a miner $\mathcal{M}$ with an advantage that is exponential in the product of $\mathcal{M}$'s hashrate and the message delay, if $\mathcal{M}$ is closer to the cluster of miners with higher hashrates.

The rest of the paper is organized as follows. Section 2 recalls the basic blockchain scheme and mining algorithm. Section 3 presents our main result. Section 4 and Section 5 present applications of our result. Section 6 discusses related work. To improve the readability of the paper, we defer some proofs to the appendix.

2 Model

2.1 Miners

We establish our main result in a system of $m = 2$ processes, called *miners*, denoted $\mathcal{M}_1$ and $\mathcal{M}_2$. (A miner is sometimes also denoted $\mathcal{M}$ or $\mathcal{M}^*$.) Both follow the algorithm assigned to them, and none crashes. The two miners interact by exchanging messages. Communication channels do not modify, inject, duplicate or lose messages. Between the two miners, the delay on message transmission is denoted u.

We consider the classical mining scheme of [18, 6] in which a *(block) chain* is a series of blocks starting from a *genesis* block G (the initial block in any chain). Let $\mathcal{C} = CB_0(= G), CB_1, CB_2, \ldots, CB_l, l \geq 1$ be any chain of length l. As in Garay et al.'s analysis of the Bitcoin backbone algorithm [8], as well as Eyal and Sirer's analysis of selfish mining [11], we define the length of a chain as the total number of blocks in a chain. For each $j \in \{1, 2, \ldots, l\}$, CB_j has reference $h_{\mathcal{C},j-1}$ to CB_{j-1}. Reference $h_{\mathcal{C},j-1} = \mathcal{H}(CB_{j-1})$ is the hash of the previous block where $\mathcal{H}$ is a hash function agreed on by all miners. The hash $h_{\mathcal{C},l}$ of the last block is sometimes called the hash of the chain $\mathcal{C}$, and is simply denoted $h_{\mathcal{C}}$.

Every miner $\mathcal{M}$ stores a chain $\mathcal{C}$ as a local variable. ($\mathcal{M}$'s chain denotes the value of $\mathcal{M}$'s local variable $\mathcal{C}$.) Two miners might have two different chains. For each miner $\mathcal{M}$, a chain that is different from $\mathcal{M}$'s $\mathcal{C}$ is called an *alternative* chain (in $\mathcal{M}$'s perspective). The mining scheme maintains and updates $\mathcal{M}$'s chain $\mathcal{C}$.

Algorithm 1 depicts the basic mining scheme, which, for pedagogical reasons, is a simplified variant of the algorithm of [18, 6]: after a chain is updated, the original mining algorithm exchanges the data and inventories of newly created blocks [19] (for performance reasons), while Algorithm 1 sends the whole new chain. The two algorithms are equivalent for the purpose of establishing our results. Before Algorithm 1 starts, every miner assigns $\mathcal{C}$ to a chain of length 0 (which only contains G), and all miners agree on a *difficulty level* d for finding a *preimage* of $\mathcal{H}$ (which we explain later). In addition, we assume that $\mathcal{M}$ has access to an infinite number of blocks (denoted by a pool Π of blocks in Algorithm 1); in other words, we assume that, at any point in time, $\mathcal{M}$ has a block to append.

If task **SolvePuzzle** (defined in Algorithm 1) returns a chain $\mathcal{C}$ of length l at $\mathcal{M}$, then $\mathcal{M}$ ignores any alternative chain $\mathcal{C}^*$ of length l that could be

Algorithm 1 Mining algorithm

1: Upon an update of $\mathcal{C}$: (1) fetch a block $B \in \Pi$; (2) create a special string tx that includes an identifier of $\mathcal{M}$; (3) run task **SolvePuzzle**$(\mathcal{C}, B, tx)$ (and stop previous **SolvePuzzle** task if any).
2: Task **SolvePuzzle**$(\mathcal{C}, B, tx)$:
- Increment a counter $\mathcal{N}$ until $\mathcal{N}$ satisfies

$$\mathcal{H}(h_{\mathcal{C}}||B||tx||\mathcal{N}) < d.$$

- Assign $CB := h_{\mathcal{C}}||B||tx||\mathcal{N}$ and $\mathcal{C} := \mathcal{C}, CB$.
- Send $\mathcal{C}$ to every other miner.

3: Upon receiving an alternative chain $\mathcal{C}^*$: if $\mathcal{C}^*$ is longer than $\mathcal{C}$, then assign $\mathcal{C} := \mathcal{C}^*$, and send $\mathcal{C}$ to every other miner.

received later. We say that $\mathcal{M}$ *extends* $\mathcal{M}$'s chain to length l to mean that $\mathcal{M}$ updates $\mathcal{C}$ such that its length becomes l. If $\mathcal{M}$ updates $\mathcal{C}$ after the return of task **SolvePuzzle**, we say that $\mathcal{M}$ *creates* a new block on $\mathcal{C}_{old}$, the old chain of length $l-1$; we also index this new block by l and say that $\mathcal{M}$ creates the lth block. If $\mathcal{M}$ updates $\mathcal{C}$, due to the reception of an alternative chain $\mathcal{C}^*$ from some miner $\mathcal{M}^*$, we say that $\mathcal{M}$ *adopts* $\mathcal{M}^*$'s chain $\mathcal{C}^*$ (or sometimes simply $\mathcal{C}^*$). When $\mathcal{M}$ creates a new block CB, CB includes a string tx that identifies $\mathcal{M}$, and a reference $h_{\mathcal{C}}$ to $\mathcal{M}$'s chain $\mathcal{C}$; thus each block is unique.[2] We say that a block CB is *committed* when every miner's chain includes CB; we say that $\mathcal{M}$ commits CB if $\mathcal{M}$ has created CB and CB is committed.

2.2 Mining as a Poisson process

Task **SolvePuzzle**$(\mathcal{C}, *, *)$ is the proof-of-work used in the classic Bitcoin implementation [5]. Performing such task is called *mining*, or more specifically mining on chain $\mathcal{C}$. We model mining as a Poisson process.[3] For any miner $\mathcal{M}_i$, for any string s, the time to find $\mathcal{N}$ such that $\mathcal{H}(s||\mathcal{N}) < d$ can be modelled as a continuous random variable $X_{i,s}$. For the same miner $\mathcal{M}_i$, any two different strings s_1 and s_2, X_{i,s_1} and X_{i,s_2} are independently and identically distributed with an exponential distribution. Let X_i be the random variable for the common distribution. X_i has an exponential distribution: the cumulative density function of X_i is $F_i(x) = P(X_i \leq x) = 1 - e^{-\lambda_i x}, x \geq 0$, and the probability density function is $f_i(x) = \lambda_i e^{-\lambda_i x}, x \geq 0$, where parameter λ_i is called the *hashrate* of $\mathcal{M}_i$. Intuitively, the hashrate implies that, for any $\mathcal{M}_i$, it takes $\frac{1}{\lambda_i}$ units of time on average to find an appropriate $\mathcal{N}$.

[2] In the classic Bitcoin implementation [2], string tx is actually the coinbase transaction that creates and pays bitcoins to an address of $\mathcal{M}$. Although two miners may use the same address to receive bitcoins, for the sake of establishing our results, we assume tx to be unique for each miner. This is not a restriction on our main result.

[3] Decker and Wattenhofer's experiment [17] on blockchain supported such a model, as we discuss in Section 6. The model was also adopted in the analysis of selfish mining [11, 9].

We assume, w.l.o.g., that compared with the time spent on mining and communication, other tasks at a miner take negligible time. We define the sequence of time instants at which a new block is created (by any miner) as the mining process. If there is only one miner with hashrate λ_1, mining is a Poisson process with rate λ_1. If $u = 0$ and there are m miners with hashrates $\lambda_1, \lambda_2, \ldots, \lambda_m$, then mining is also a Poisson process with rate $\lambda = \sum_{i=1}^{m} \lambda_i$. Intuitively, a new block is created every $\frac{1}{\lambda}$ units of time on average. Rate λ depends on the difficulty level d. In the classic Bitcoin implementation [18, 6], d is selected such that $\frac{1}{\lambda} = 600$ seconds.[4]

3 Unfairness of Blockchain

We establish here our main result. We prove the unfairness of blockchain in the case of two honest miners, $\mathcal{M}_1$ and $\mathcal{M}_2$: if $\mathcal{M}_1$ has a *higher* hashrate than $\mathcal{M}_2$, then in expectation, $\mathcal{M}_1$ can commit *many more* blocks than $\mathcal{M}_2$, in a way disproportionate to their hashrates, formalized in Theorem 1. Consider any $a \in \mathbb{Z}^+$, let $\mathcal{C}_{i,a}$ be $\mathcal{M}_i$'s chain when $\mathcal{M}_i$ extends $\mathcal{M}_i$'s chain to length a for $i \in \{1, 2\}$. Denoted by $N_{i,a}$, the random variable for the number of blocks committed by $\mathcal{M}_i$ in $\mathcal{C}_{1,a}$ and $\mathcal{C}_{2,a}$, i.e., the number of blocks created by $\mathcal{M}_i$ which both $\mathcal{C}_{1,a}$ and $\mathcal{C}_{2,a}$ include.

Theorem 1 (Ratio of probabilities of committed blocks). *If* $\lambda_1 > \lambda_2$, *then*

$$\frac{E(N_{1,a})}{E(N_{2,a})} \geq \frac{e^{\lambda_1 u}}{e^{\lambda_2 u}} \cdot \frac{\lambda_1}{\lambda_2} > \frac{\lambda_1}{\lambda_2}.$$

Proof outline of Theorem 1 To prove Theorem 1, by the linearity of expectation, we examine the expectation of the event that $\mathcal{M}_i$ creates CB as $\mathcal{M}_i$'s kth block for $k \in \mathbb{Z}^+$ (and later commits CB) for each $i \in \{1, 2\}$, the probability of which depends on when the two miners' chains grow to $k - 1$. To this end, we show a fundamental unfairness property on the growth of the blockchain (Section 3.2). We then show unfairness on the success probability in committing CB as the kth block and hence unfairness on the expected number of committed blocks (Section 3.3). To formalize the growth of the blockchain and the success probability, we first introduce some definitions and terminologies in Section 3.1 below.

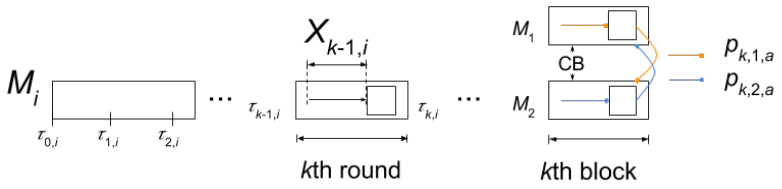
3.1 Definitions and terminologies

We model here with random variables the growth (in length) of the blockchain as well as the events of success of both miners.

[4] As the computational power of CPUs increases, the value of d is changed from time to time such that $\frac{1}{\lambda} = C = 600$ seconds always holds [6]. As the relation $\frac{1}{\lambda} = C$ is always true for the same constant C, in this paper, we consider hashrates $\lambda_1, \ldots, \lambda_m$ as constants.

Definition 1 (Growth in length of the blockchain). *For each miner $\mathcal{M}_i, i \in \{1,2\}$, let $\{\tau_{k,i} | k = 1,2,\ldots\}$ be the sequence of time instants such that at $\tau_{k,i}$, $\mathcal{M}_i$ extends $\mathcal{M}_i$'s chain to length k. W.l.o.g., we define $\tau_{0,i} = 0$ (i.e., the two miners start at the same time 0).*

As defined in Definition 1 and illustrated in Figure 1a, each of the two miners maintains a local chain which grows in length by one. When a miner has a chain of length $k-1$, we say that the miner starts its kth round. For $\mathcal{M}_i$, the kth round is thus $[\tau_{k-1,i}, \tau_{k,i}]$. At $\tau_{k-1,i}$, $\mathcal{M}_i$ starts mining a new block. We denote by $X_{k-1,i}$ the random variable of the time which $\mathcal{M}_i$ spends on mining when $\mathcal{M}_i$ is alone (without a second miner sending an alternative chain). Two miners can have different chains. Given length a, it is unknown whether two miners have the same chain or not. However, if they have the same kth block CB in their chains respectively when $a \geq k$, the kth block remains there. We define the success probability $p_{k,i,a}$ according to which miner mines CB in Definition 2, illustrated in Figure 1b. A prerequisite of $p_{k,i,a} > 0$ is that $\mathcal{M}_i$ completes its mining at the kth round (even with the other miner), which we denote by $W_k \in \{0, i\}$. When $W_k = 0$, both miners complete their mining at the kth round, as defined in Definition 3, and can commit one of the two blocks mined later.



(a) Growth of blockchain in length **(b)** Block k

Fig. 1: Definitions and terminologies

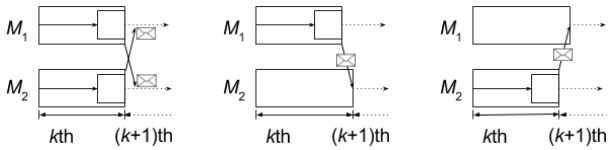
Definition 2 (Success of the kth block). *Consider any $a \in \mathbb{Z}^+$, let $\mathcal{C}_{i,a}$ be $\mathcal{M}_i$'s chain when $\mathcal{M}_i$ extends $\mathcal{M}_i$'s chain to length a for $i \in \{1,2\}$. For any $k \in \mathbb{Z}^+, a \geq k$, we define $p_{k,i,a}$, the probability of the event that (1) $\mathcal{M}_i$ creates CB at the kth round, and (2) both $\mathcal{C}_{1,a}$ and $\mathcal{C}_{2,a}$ have CB.*

Definition 3 (Creation of the kth block). *For each miner $\mathcal{M}_i, i \in \{1,2\}$, let $W_k = i$ if $\mathcal{M}_{3-i}$ adopts the alternative chain (in $\mathcal{M}_{3-i}$'s perspective) from $\mathcal{M}_i$ at the end of the kth round, and $W_k = 0$ when both miners create their kth blocks respectively and fork the blockchain.*

3.2 Unfairness on the growth of the blockchain

We first determine the probability distribution of when the kth round ends (i.e., the blockchain grows to length k) in Lemma 1 and then present a property of this probability distribution in Lemma 2.

It is important to know when the blockchain grows to a certain length k at the two miners, since the difference (in time) at the two miners can give one of them a head start. The calculation of probability distribution is actually straightforward. For each $k = 1, 2, \ldots$, consider $\tau_k = (\tau_{k,1}, \tau_{k,2})$. Suppose that the probability distribution of τ_{k-1} is known. As illustrated in Figure 2, there are three possibilities of how the blockchain grows to k: (a) that the blockchain forks; (b) that $\mathcal{M}_1$ mines a block and $\mathcal{M}_2$ receives this block before mining one; (c) that $\mathcal{M}_2$ mines a block and $\mathcal{M}_1$ receives this block before mining one. Since the probability distribution of $X_{k-1,1}, X_{k-1,2}$ is known, the probability distribution of $\delta_k = \tau_{k+1} - \tau_k$ can be calculated as well as τ_{k+1}. The base case where $k = 0$ is simpler: $\tau_0 = (0, 0)$ is assumed and thus $\tau_1 = \delta_0$. As a result, the probability distribution of τ_k includes a recursive equation D_k, which can be evaluated to a function of s, t alone if λ_1, λ_2, u are specified. The expression $Pr(\tau_k = (s, t))$ in Lemma 1 represents the following probability: $Pr(s \leq \tau_{k,1} \leq s + ds, t \leq \tau_{k,2} \leq t + dt)$ where ds and dt are the infinitesimals.[5] The full proofs of Lemma 1 and Lemma 2 follow from the calculation above and are deferred to the appendix.



(a) $W_k = 0$ **(b)** $W_k = 1$ **(c)** $W_k = 2$

Fig. 2: Three possibilities of round k

Lemma 1 (The growth of the blockchain). *For $s > 0, t > 0$, let $D_0(s, t) = 1$ and for each $k = 1, 2, \ldots$, let*

$$\begin{aligned} D_k(s,t) = \lambda_1\lambda_2 \int_{\substack{|y-z|<u,\\ 0<y<s,\\ 0<z<t}} D_{k-1}(y,z)dydz \\ + \lambda_1 \int_{\substack{y-z=-u,\\ 0<y<s,\\ 0<z<t}} D_{k-1}(y,z)dy \\ + \lambda_2 \int_{\substack{y-z=u,\\ 0<y<s,\\ 0<z<t}} D_{k-1}(y,z)dz. \end{aligned}$$

[5] The probability distribution of τ_k cannot be expressed by a cumulative distribution function because the random variable $\tau_k - \tau_{k-1}$ is of mixed type: neither continuous nor discrete.

Then the probability of τ_k is

$$Pr(\tau_k = (s,t)) = \begin{cases} 0 & |s-t| > u; \\ \lambda_1\lambda_2 e^{-\lambda_1 s - \lambda_2 t} D_{k-1}(s,t) ds dt & |s-t| < u; \\ \lambda_1 e^{-\lambda_1 s - \lambda_2 t} D_{k-1}(s,t) ds & s - t = -u; \\ \lambda_2 e^{-\lambda_1 s - \lambda_2 t} D_{k-1}(s,t) dt & s - t = u. \end{cases}$$

Lemma 2. *For $s > 0, t > 0, |s-t| \leq u$ and $u > 0$, let $D_k(s,t)$ be defined as in Lemma 1. If $\lambda_1 > \lambda_2$ and $s > t, s > u$, then $D_k(s,t) < D_k(t,s), \forall k \in \mathbb{Z}^+$; if $u > s > t$, then $D_k(s,t) = D_k(t,s), \forall k \in \mathbb{Z}^+$.*

Lemma 2 implies a property of τ_k: for any length k, the probability that the miner with a higher hashrate has a chain of length k earlier than the other is higher, which is the intuition behind the proof of inequality in Theorem 1. Lemma 1 can be easily extended to any number of miners. A result similar to Lemma 2 follows: for any number of miners, between any two miners, for any length k, the probability that the miner with a higher computational power has a chain of length k earlier than the other is higher.

3.3 Unfairness on the success of the kth block

We prove Theorem 1 by showing a lower bound on the ratio between the success probability $p_{k,i,a}$ of $\mathcal{M}_i$ in committing $\mathcal{M}_i$'s kth block, for $i \in \{1,2\}$.

Since W_k is the random variable that captures whether, at the end of the kth round, some miner adopts a chain from the other and if so, whose chain is adopted, then the event defined for $p_{k,i,a}$ is equivalent to the union of the following two events: (S1) $W_k = i$, and (S2) $W_k = 0, \ldots, W_{j-1} = 0, W_j = i$. We thus have the success probability $p_{k,i,a}$, for each $i \in \{1,2\}$:

$$p_{k,i,a} = Pr(W_k = i) + \sum_{j=k+1}^{a} Pr(W_k = 0, \ldots, W_{j-1} = 0, W_j = i).$$

We determine a lower bound for each of the two possibilities (S1) and (S2) in Lemma 3 by (1) determining $Pr(W_k = i)$ and $Pr(W_k = 0, \ldots, W_{j-1} = 0, W_j = i)$ based on Lemma 1 and (2) applying the inequality in Lemma 2. The intuition behind the exponential term in Lemma 3 is that (1) at round k when a block is committed, there is a gap of u (the message delay) between $\tau_{k,1}$ and $\tau_{k,2}$, when the two miners end round k respectively, and (2) by the Poisson process, the probability of a block created during the gap is exponential in u. The full proof of Lemma 3 is deferred to the appendix for the space limitation.

Lemma 3. *For any $j, k, a \in \mathbb{Z}^+$ and $a \geq k+1$, if $\lambda_1 > \lambda_2$, then*

$$\begin{aligned} \frac{Pr(W_k = 1)}{Pr(W_k = 2)} &> \frac{e^{\lambda_1 u}}{e^{\lambda_2 u}} \cdot \frac{\lambda_1}{\lambda_2}, \quad \forall k \geq 2; \\ \frac{Pr(W_k = 1)}{Pr(W_k = 2)} &= \frac{e^{\lambda_1 u}}{e^{\lambda_2 u}} \cdot \frac{\lambda_1}{\lambda_2}, \quad k = 1; \end{aligned} \tag{1}$$

and

$$\forall j, k+1 \leq j \leq a,$$
$$\frac{Pr(W_k = 0, \ldots, W_{j-1} = 0, W_j = 1)}{Pr(W_k = 0, \ldots, W_{j-1} = 0, W_j = 2)} \geq \frac{e^{\lambda_1 u}}{e^{\lambda_2 u}} \cdot \frac{\lambda_1}{\lambda_2}. \quad (2)$$

Following Lemma 3, we sketch the proof of Theorem 1 here. Let $n_{k,i,a}$ be the random variable of whether $\mathcal{M}_i$ creates $\mathcal{M}_i$'s kth block and both $\mathcal{C}_{1,a}$ and $\mathcal{C}_{2,a}$ include the kth block. Then as defined in Theorem 1, $N_{i,a} = \sum_{k=1}^{a} n_{k,i,a}$. Since $E(N_{i,a}) = \sum_{k=1}^{a} p_{k,i,a}$, then if $\forall k \in \mathbb{Z}^+, a \geq k$, $\frac{p_{k,1,a}}{p_{k,2,a}}$ is lower bounded by $\frac{e^{\lambda_1 u}}{e^{\lambda_2 u}} \cdot \frac{\lambda_1}{\lambda_2}$, then $\frac{E(N_{1,a})}{E(N_{2,a})}$ is lower bounded by the same. We remark that the lower bound is supported by the inequality between $D_k(s,t)$ and $D_k(t,s)$ in Lemma 2, implying that the lower bound $\frac{e^{\lambda_1 u}}{e^{\lambda_2 u}} \cdot \frac{\lambda_1}{\lambda_2}$ is not yet tight.

Trade-off between the mining speed and fairness. Theorem 1 highlights the fragility of tentative implementations that would reduce the time spent on mining (today set to 600 seconds on average) to improve the throughput of transactions [20, 21]. Recall from Section 2 that this would reduce the difficulty level, which would in turn increase every miner's hashrate proportionally. As a result, fairness could be further undermined: the proportional increase of λ_1 and λ_2 results in a larger gap in the exponential factor $\frac{e^{\lambda_1 u}}{e^{\lambda_2 u}}$, which highlights a trade-off between the speed to mine a block and the fairness of blockchain.

Extension to any number of miners. Similar to Lemma 1 and Lemma 2, Equation 1 in Lemma 3 can be easily extended to any number of miners. In other words, for any number of miners, if we only compare the probability of a miner committing its block immediately between two miners, then the ratio between the two probabilities is also greater than the ratio between hashrates and at least exponential in delay u. Yet it is unclear whether Equation 2 can be extended to any number of miners, which can be a future direction of our work.

4 Application to Selfish Mining

We show here how our result can be used to generalize one of the main results in selfish mining [11]. Selfish mining is an attack for a minority of miners to commit more blocks in expectation than its fair share. In a model of two miners, one selfish and one honest, with message delay $u = 0$, Eyal and Sirer [11] showed that, it is sufficient for $\alpha > \frac{1}{3} \triangleq th$ to launch selfish mining.[6] We generalize

[6] Eyal and Sirer [11] showed a lower bound (on α) as a function of parameter γ which represents the percentage of miners in the honest majority $\mathcal{M}_1$ that adopt the selfish minority $\mathcal{M}_2$'s chain, and did not consider exact message delay u. The sufficient condition in [11] is obtained when $\gamma = 0$, which implies $u = 0$.

the threshold th to a realistic setting with $u > 0$.[7] We first recall below the selfish mining algorithm and some of its results from [11]. For the classic Bitcoin blockchain implementation, we show that blockchain is not as vulnerable as previously believed against selfish mining.

4.1 Selfish mining

We recall the main idea underlying the selfish mining algorithm below, assuming a selfish miner $\mathcal{M}_2$ and an honest miner $\mathcal{M}_1$. (More details can be found in [11].) When both $\mathcal{M}_1$ and $\mathcal{M}_2$ mine on chain $\mathcal{C}$ and $\mathcal{M}_2$ succeeds in creating a block CB on $\mathcal{C}$, $\mathcal{M}_2$ continues to mine on $\mathcal{C}_{mi} = \mathcal{C}, CB$ (instead of sending $\mathcal{C}_{mi}$ to $\mathcal{M}_1$ as in Algorithm 1). Then $\mathcal{M}_2$ maintains two chains $\mathcal{C}_{mi}$ and $\mathcal{C}_{ma}$ locally. The latter is initialized to $\mathcal{C}$. Miner $\mathcal{M}_2$ updates $\mathcal{C}_{mi}$ when it creates a new block on $\mathcal{C}_{mi}$; $\mathcal{M}_2$ updates $\mathcal{C}_{ma}$ when $\mathcal{M}_1$ sends an alternative chain. The goal of $\mathcal{M}_2$ is to commit all blocks created by $\mathcal{M}_2$ on $\mathcal{C}_{mi}$ (those blocks of $\mathcal{C}_{mi}$ after prefix $\mathcal{C}$). There are two scenarios where $\mathcal{M}_2$ commits by sending $\mathcal{C}_{mi}$ to $\mathcal{M}_1$:

1. After an update of $\mathcal{C}_{ma}$, $\mathcal{C}_{mi}$ and $\mathcal{C}_{ma}$ have the same length but differ at the last block (denoted event E_1), and now $\mathcal{M}_2$ creates a new block on $\mathcal{C}_{mi}$ and updates $\mathcal{C}_{mi}$; and
2. After an update of $\mathcal{C}_{ma}$, $\mathcal{C}_{mi}$ has exactly one more block than $\mathcal{C}_{ma}$ (denoted event E_2).

The state machine of selfish mining is illustrated in Figure 3, where $k, k \geq 0, k \in \mathbb{Z}$ in each state represents that C_{mi} is k-block longer than C_{ma} at $\mathcal{M}_2$ and $0'$ is the resulting state of commit in E_2. Then N_2 and N_1, the expected numbers of blocks which $\mathcal{M}_2$ and $\mathcal{M}_1$ commit respectively, can be calculated from the state machine in terms of α. Let $R_0 = N_2/(N_1 + N_2)$. The threshold th is the solution α^* to $R_0(\alpha) = \alpha$.

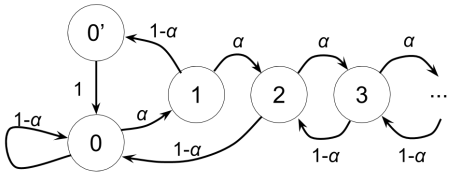


Fig. 3: State machine of selfish mining without message delay

[7] Here parameter γ as in [11] is set to 0. If we consider γ and u as two parameters of the lower bound, then when $u > 0$, for all γ, the case where $\gamma = 0$ requires the highest computational power th from the selfish minority and as a result, $\alpha > th$ is again a sufficient condition to launch selfish mining.

4.2 Upper bound on the number of committed blocks

Let R be the proportion of the expected number of blocks committed by $\mathcal{M}_2$ among all committed blocks. We show that R is upper bounded in Theorem 2 in a realistic setting where $u > 0$. The key is that due to non-zero message delays, when $E_j, j \in \{1, 2\}$ occurs, $\mathcal{M}_2$ only *tries* to commit by sending $\mathcal{C}_{mi}$. There are several possibilities following $\mathcal{M}_2$'s sending $\mathcal{C}_{mi}$: $\mathcal{M}_2$ commits and then $\mathcal{M}_2$ has additional $2u$ units of time as a head start, or $\mathcal{M}_2$ fails. The state machine of selfish mining thus changes. In Figure 4, N_{2u} represents how many blocks $\mathcal{M}_2$ can find during $2u$, ρ_1 and ρ_2 represent upper bounds on the probability of $\mathcal{M}_2$ committing $\mathcal{C}_{mi}$. The full proof of Theorem 2 is deferred to the appendix for space limitation.

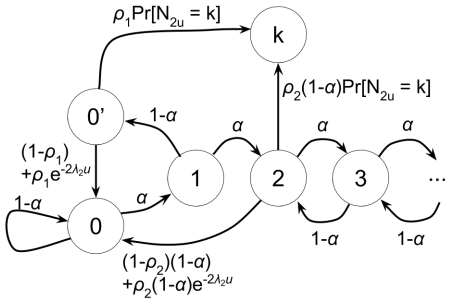


Fig. 4: State machine of selfish mining with message delay ($k \in \mathbb{Z}^+$)

Theorem 2. *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two miners with hashrates λ_1 and λ_2 respectively. Let $\alpha = \frac{\lambda_2}{\lambda_1+\lambda_2}$ and*

$$\mathcal{U} = \frac{2P_1\rho_1(1-\alpha) + 2P_2\rho_2(1-\alpha) + P_{k>2}\rho_2(1-\alpha)}{2P_1(1-\alpha) + 2P_2(1-\alpha) + P_0(1-\alpha) + P_{k>2}(1-\alpha) + P_2(1-\rho_2)(1-\alpha)}$$

where $\rho_1 = \alpha e^{-2\lambda_1 u} + \alpha(1 - e^{-2\lambda_1 u})r_2$, $\rho_2 = e^{-2\lambda_1 u} + (1 - e^{-2\lambda_1 u})[\alpha + (1 - \alpha)(1 - e^{-2\lambda_2 u})r_2]$ and $r_2 = \frac{1}{r+1}, r = \frac{e^{\lambda_1 u}}{e^{\lambda_2 u}} \cdot \frac{\lambda_1}{\lambda_2}$ and $P_{0'}$ and $P_k, k \geq 0, k \in \mathbb{Z}$ are the probabilities of each state in the state machine of Figure 4.

If $\lambda_1 > \lambda_2$ and $\mathcal{U} > \alpha$, then $R \leq \mathcal{U}$.

4.3 Lower bound on the threshold

We are now ready to find a lower bound on the threshold assuming message delay $u > 0$. Recall that the threshold should be the solution to $R(\alpha) = \alpha$. According to Theorem 2, since the relative number of blocks committed by the selfish miner is upper bounded by $\mathcal{U}$, the solution $\mathcal{L}$ to $\mathcal{U}(\alpha) = \alpha$ is a lower bound on the threshold.

Table 1: Lower bound on the threshold

u	1	10	20	100
$\mathcal{L}$	0.334	0.339	0.343	0.371
$\mathcal{U}(\mathcal{L})$	0.33418	0.33999	0.34361	0.37087

Assuming $\lambda = \frac{1}{600}$, and $u = 1, 10, 20, 100$ (where the units of time are seconds), which are taken from the classic Bitcoin implementation [18, 6, 17], we show the numeric calculation of $\mathcal{L}$ in Table 1, which is greater than $\frac{1}{3}$. In addition, for small $u > 0$ such that $Pr[N_{2u} = k], \forall k \in \mathbb{Z}^+$ is negligible, $\mathcal{L} > \frac{1}{3}$ is always true in theory. In either way, we obtain that $\mathcal{L} > \frac{1}{3} = th$, suggesting that with reasonable message delay $u > 0$, blockchain is not as *unfair* as previously believed when some dishonest miner performs selfish mining.

5 Application to Clusters of Miners

We consider in our main result (Theorem 1) two miners with message delay u. We now consider a model of $m > 2$ miners that can be divided into two sets S_1 and S_2 such that (1) within S_1 or S_2, there is no message delay, and (2) between every miner in S_1 and every miner in S_2, there is message delay u.

Corollary 1 below is obtained from Theorem 1. While Theorem 1 shows that, in the case of two miners, a miner with a higher hashrate has an exponential advantage over the other, Corollary 1 shows that in a system of more than two miners, between two miners with the same hashrate, one miner can still have an exponential advantage over the other as long as the former one is very close (such that the message delay is 0) to some other miner with a high hashrate.

Corollary 1 (Lower bound with more than two miners). *Let $\mathcal{M}_1 \in S_1$ and $\mathcal{M}_2 \in S_2$ be any two miners with the same hashrate λ_0. Assume message delay u between S_1 and S_2. Suppose that the sum of all miners' hashrates in S_1 is $m_1\lambda_0$ and that of S_2 is $m_2\lambda_0$. Recall that for $i \in \{1, 2\}$, $N_{i,a}$ denotes the number of blocks committed by $\mathcal{M}_i$ before the end of round a. Assuming $\mathcal{M}_1$ and $\mathcal{M}_2$ start with the same chain at the same time, if $m_1 > m_2$, then*

$$\frac{E(N_{1,a})}{E(N_{2,a})} \geq \frac{e^{m_1\lambda_0 u}}{e^{m_2\lambda_0 u}} > 1.$$

6 Related Work

6.1 Theoretical analyses of blockchain

Satoshi Nakamoto [18] was credited for the proposal of Bitcoin and its underlying distributed protocol, blockchain. With the popularity of Bitcoin, a lot of work has been devoted to formalizing blockchain and verifying its claimed properties

[8, 22, 12, 23]. We discuss below some approches, and contrast our main result with the properties obtained from previous theoretical analyses.

Garay et al. [8] proposed the *q-bounded synchronous setting* to model blockchain. Given that the mining algorithm increments a counter to find the preimage of some hash function, this q-bounded synchronous setting assumes that, in each round, any miner can increment at most q times the counter [8]. [8] Pass et al. [12] studied blockchain in an asynchronous network where message delays can be arbitrary. Both Garay et al. [8], as well as Pass et al. [12] verified the *chain quality* property, in their models respectively, considering dishonest miners. The (μ, ℓ)-chain quality property identifies an upper bound on the proportion of dishonest miners' committed blocks among any ℓ consecutive blocks for some ℓ [8, 12]. This property differs from our notion of fairness, in that ours considers the proportion of the expected number of committed blocks. (Later Pass and Shi [23] strengthened chain quality property as fairness and used the term, eventual fairness, for our notion of fairness.) Pass et al. [8], as well as Garay et al. [12], derived chain quality property with μ higher than the proportion of dishonest miners' hashrates, which thus does not imply the (un)fairness of blockchain (as considered in this paper).

Kosba et al. [22] proposed a cryptographic model of blockchain, assuming ideal functionality [22]; i.e., they assumed that blockchain satisfies certain idealized properties, which do not fit our analysis here.

6.2 Unfairness of blockchain

We are the first to theoretically prove the unfairness of blockchain among honest miners. Previous work either observed the unfairness (among honest miners) via simulation [13, 14], or proposed attacks to break fairness (with dishonest miners) [11, 16, 24, 9].

Lewenberg et al. [14], as well as Eyal et al. [13], simulated blockchain (among honest miners), and observed indeed that some miner's reward can be lower than its fair share; yet neither work provided a theoretical explanation of such unfairness. Both presented an exponentially descending curve for the proportion of the victim's committed blocks with increasing speed to mine a block [13, 14], which our Theorem 1 explains. To mitigate the fairness issue, Lewenberg et al. [14] and Eyal et al. [13] proposed alternatives. However, both alternatives rely on blockchain and thus still suffer from the unfairness we highlight in this paper.[9] Lewenberg et al. [25] presented a formula on the proportion of one miner's committed blocks between two miners and omitted the proof. To compare with,

[8] This setting neglects the variable relation between mining and time, e.g., when a message is delayed, a miner may have more time (more increments) to mine a block.

[9] Lewenberg et al. [14] proposed *inclusive blockchain* as an alternative, which stores all possible blockchains in a directed acyclic graph, and then still observed disproportionate rewards among honest miners. Eyal et al. [13] proposed *Bitcoin-NG*, where a leader (a special miner that is entitled to include transaction blocks) is elected based on blockchain (and its companion mining algorithm); the leader election could still suffer from the unfairness of blockchain as shown in this paper.

we model the growth of blockchain, which can be extended to any number of miners and may be of independent interest, and prove our result based on the model of growth, independently from [25].

Eyal and Sirer [11] proposed a very interesting attack, called *selfish mining*, for a minority of miners to commit more blocks in expectation than its fair share. (Sapirshtein et al. [9], as well as Gervais et al. [16] and Nayak et al. [26], optimized selfish mining.) Such unfairness resulting from selfish (dishonest) miners does not imply our result. Assuming no message delay, Eyal and Sirer [11] determined a threshold th on the proportion α of the minority's hashrate for selfish mining to be feasible, while Sapirshtein et al. [9] established a lower threshold by optimizing selfish mining for each value of α. As an application of our main result, we extend Eyal and Sirer's threshold in a model with message delays. In Section 4, we show that the classic Bitcoin blockchain implementation can tolerate selfish mining more than previously believed: under reasonable message delays, a threshold for selfish mining to be feasible is greater than th. Sapirshtein et al. [9] additionally showed that, assuming message delays, any miner can commit more blocks by being dishonest, but did not study the effect of message delays on selfish mining. Unlike previous work, Gervais et al. [16] modelled selfish mining with Markov Decision Processes parameterized by the stale block rate (which intuitively captures message delays as well as hashrates) yet did not provide a threshold. Nayak et al. [26] composed network-level attacks (eclipse attacks) with generalized selfish mining, which however focused on isolating miners instead of concrete message delays.

Heilman et al. [24] presented *eclipse attacks* on Bitcoin, which enforce some miners to connect only to an attacker (which may control multiple miners). As these honest miners are fed with selected transactions and blocks, a dishonest miner's reward can be higher than its fair proportion [24]. Eclipse attacks can also increase message delays [17, 24], and can then transform Theorem 1 into an attack against the miner with a low hashrate.

In addition, to address unfairness resulting from dishonest miners, Pass and Shi [23] proposed fruitchain, which mines another data structure called fruits, as well as blocks. They proved that the proportion of dishonest miners' committed fruits is upper bounded by their proportional hashrate and in this sense, is fair [23]. Since fruitchain takes a different approach from the blockchain considered in this paper, our result and theirs are incomparable.

6.3 Message delays in blockchain

Our assumptions on message delays (level of seconds in Section 4) as well as our model of mining (as a Poisson process throughout the paper) have already been discussed in the literature.

Our assumptions on message delays are justified by the study of Decker and Wattenhofer [17], as well as Croman et al. [20], on block propagation delays in Bitcoin network. Decker and Wattenhofer [17] showed that in 2013, the median time for a node (not necessarily a miner) to receive a block is 6.5 seconds, whilst the mean is 12.6 seconds [17]. Croman et al. [20] repeated the measurement of

block propagation in 2014 and 2015, and found a median time of 8.7 seconds. Our model of mining is justified by Decker and Wattenhofer [17]'s measurement on the probability of the time to create a block in the Bitcoin network. The measured distribution fits the exponential distribution [17], which justifies the mining Poisson process widely used in this paper, as well as in the literature [2, 11, 9].

Assuming message delays, Natoli and Gramoli [27] observed a *blockchain anomaly*, which can be considered as an extreme case of our Theorem 1. The classic Bitcoin implementation stipulates a value $k = 6$ such that, if an honest miner $\mathcal{M}$'s chain has at least k blocks after a certain block CB, then $\mathcal{M}$ considers CB confirmed. Natoli and Gramoli [27] exhibited an attack that only delays messages (arbitrarily) in a system of two miners, against any k: as long as the attacker's hashrate is higher than $\mathcal{M}$, for any k, a confirmed block can be later removed from $\mathcal{M}$'s chain. Theorem 1 explains such anomaly: with the message delay approaching infinity, an attacker commits almost all blocks with a high probability, while $\mathcal{M}$ commits (or confirms) nearly none. In this sense, our unfairness result can be viewed as a generalization of the observation of [27].

7 Concluding Remarks

In this paper, we show that Bitcoin blockchain is actually unfair. We prove that even in a distributed system of two honest miners, under non-instantaneous message delivery, the expected number of committed blocks between two miners is lower bounded by a function exponential in the product of the message delay and the difference between the two miners' hashrates. Possible future work includes quantifying the unfairness of blockchain in a distributed system Ω of more than two honest miners. In a realistic scenario, as suggested by Lemma 1 and Lemma 2, the miner $\mathcal{M}$ with the highest hashrate starts a round earlier than any other miner (in expectation) and thus blockchain must have a head start in Ω. Yet it is not clear whether the proportion of the expected number of committed blocks is also exponential in the product of the difference between hashrates and the message delay.

Acknowledgement

This work has been supported in part by the European ERC Grant 339539 - AOC

Bibliography

[1] "100+ companies that accept bitcoins as payment," Online, 2015, http://www.ebay.com/gds/100-Companies-That-Accept-Bitcoins-As-Payment-/10000000206483242/g.html.

[2] Bitcoin community, "Bitcoin," Januray 2016, https://en.bitcoin.it/wiki/Bitcoin.

[3] J. Davidson, "No, big companies aren't really accepting bitcoin," Online, 2015, http://time.com/money/3658361/dell-microsoft-expedia-bitcoin/.

[4] R. McMillan, "IBM bets on bitcoin ledger," February 2016, https://www.wsj.com/articles/ibm-bets-on-bitcoin-ledger-1455598864.

[5] Bitcoin community, "Proof of work," May 2016, https://en.bitcoin.it/wiki/Proof_of_work.

[6] ——, "Protocol rules," October 2016, https://en.bitcoin.it/wiki/Protocol_rules.

[7] E. Felten, "Bitcoin research in princeton cs," Online, november 2013, https://freedom-to-tinker.com/2013/11/29/bitcoin-research-in-princeton-cs/.

[8] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in *EUROCRYPT 2015*, E. Oswald and M. Fischlin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 281–310.

[9] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," *CoRR*, vol. abs/1507.06183, 2015. [Online]. Available: https://arxiv.org/abs/1507.06183

[10] Bitcoin community, "Majority attack," July 2015, https://en.bitcoin.it/wiki/Majority_attack.

[11] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *FC 2014*, N. Christin and R. Safavi-Naini, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 436–454.

[12] R. Pass, L. Seeman, and abhi shelat, "Analysis of the blockchain protocol in asynchronous networks," Cryptology ePrint Archive, Report 2016/454, 2016, http://eprint.iacr.org/2016/454.

[13] I. Eyal, A. E. Gencer, E. G. Sirer, and R. V. Renesse, "Bitcoin-ng: A scalable blockchain protocol," in *NSDI 2016*. USENIX Association, 2016, pp. 45–59.

[14] Y. Lewenberg, Y. Sompolinsky, and A. Zohar, "Inclusive block chain protocols," in *FC 2015*, R. Böhme and T. Okamoto, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 528–547.

[15] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in bitcoin," in *FC 2015*, pp. 507–527.

[16] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *CCS 2016*. New York, NY, USA: ACM, 2016, pp. 3–16.

[17] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *IEEE P2P 2013*, 2013, pp. 1–10.

[18] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008, https://bitcoin.org/bitcoin.pdf.
[19] Bitcoin community, "Block chain download," January 2016, https://en.bitcoin.it/wiki/Block_chain_download.
[20] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gün Sirer, D. Song, and R. Wattenhofer, "On scaling decentralized blockchains," in *FC 2016*, J. Clark, S. Meiklejohn, P. Y. Ryan, D. Wallach, M. Brenner, and K. Rohloff, Eds., 2016, pp. 106–125.
[21] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," in *iNetSec 2015*, J. Camenisch and D. Kesdoğan, Eds. Cham: Springer International Publishing, 2015, pp. 112–125.
[22] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *SP 2016*, 2016, pp. 839–858.
[23] R. Pass and E. Shi, "Fruitchains: A fair blockchain," Cryptology ePrint Archive, Report 2016/916, 2016, http://eprint.iacr.org/2016/916.
[24] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *SEC 2015*. Berkeley, CA, USA: USENIX Association, 2015, pp. 129–144.
[25] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein, "Bitcoin mining pools: A cooperative game theoretic analysis," ser. AAMAS 2015. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 919–927.
[26] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *EuroS&P 2016*, 2016, pp. 305–320.
[27] C. Natoli and V. Gramoli, "The blockchain anomaly," in *NCA 2016*, 2016, pp. 310–317.

A Proof of Lemma 1

According to the mining algorithm, there is a recurrence relation between τ_k and τ_{k-1}. Using W_k, the recurrence relation between τ_k and τ_{k-1} can be written as follows:

$$\tau_{k,1} = \begin{cases} \tau_{k-1,1} + X_1 & \text{if } W_k \neq 2, \\ \tau_{k-1,2} + X_2 + u & \text{otherwise.} \end{cases} \tag{3}$$

$$\tau_{k,2} = \begin{cases} \tau_{k-1,2} + X_2 & \text{if } W_k \neq 1, \\ \tau_{k-1,1} + X_1 + u & \text{otherwise.} \end{cases} \tag{4}$$

We prove the statement of Lemma 1 by induction. We first prove for the base case where $k = 1$. According to the mining algorithm, $|\tau_{1,1} - \tau_{1,2}| \leq u$; thus $Pr(\tau_1 = (s,t)) = 0$ when $|s - t| > u$. When $|s - t| < u$, $W_1 = 0$; thus by

Equations (3) and (4), and $\tau_0 = (0, 0)$,

$$\begin{aligned} Pr(\tau_1 = (s,t)) &= Pr(X_1 = s, X_2 = t) \\ &= Pr(X_1 = s) \cdot Pr(X_2 = t) \\ &= f_1(s) f_2(t) ds dt \\ &= \lambda_1 \lambda_2 e^{-\lambda_1 s - \lambda_2 t} D_0(s,t) ds dt. \end{aligned}$$

When $s - t = -u$, $W_1 \neq 2$; thus by Equations (3) and (4), and $\tau_0 = (0, 0)$,

$$\begin{aligned} Pr(\tau_1 = (s,t)) &= Pr(X_1 = s, X_2 \geq t) \\ &= Pr(X_1 = s) \cdot Pr(X_2 \geq t) \\ &= f_1(s) ds \int_t^\infty f_2(x) dt \\ &= \lambda_1 e^{-\lambda_1 s - \lambda_2 t} D_0(s,t) ds. \end{aligned}$$

Similarly, when $s - t = u$, $W_1 \neq 1$; thus by Equations (3) and (4), and $\tau_0 = (0, 0)$,

$$\begin{aligned} Pr(\tau_1 = (s,t)) &= Pr(X_1 \geq s, X_2 = t) \\ &= Pr(X_1 \geq s) \cdot Pr(X_2 = t) \\ &= f_2(t) dt \int_s^\infty f_2(x) ds \\ &= \lambda_2 e^{-\lambda_1 s - \lambda_2 t} D_0(s,t) dt. \end{aligned}$$

As a result, the statement is true for the base case where $k = 1$.

Now assuming that the statement is true for an arbitrary $k \in \mathbb{Z}^+$, we prove the statement is true for $k + 1$. Then for $\tau_{k+1} = (s, t)$, according to the mining algorithm, $|s - t| \leq u$; thus $Pr(\tau_{k+1} = (s,t)) = 0$ when $|s - t| > u$. When $|s - t| \leq u$, we calculate probability $Pr(\tau_{k+1})$ using our assumption on $Pr(\tau_k)$:

$$\begin{aligned} & Pr(\tau_{k+1} = (s,t)) \\ = & \int_{\substack{|y-z|<u, \\ 0<y<s, \\ 0<z<t}} Pr(\tau_{k+1} = (s,t) | \tau_k = (y,z)) \cdot Pr(\tau_k = (y,z)) \\ + & \int_{\substack{y-z=-u, \\ 0<y<s, \\ 0<z<t}} Pr(\tau_{k+1} = (s,t) | \tau_k = (y,z)) \cdot Pr(\tau_k = (y,z)) \\ + & \int_{\substack{y-z=u, \\ 0<y<s, \\ 0<z<t}} Pr(\tau_{k+1} = (s,t) | \tau_k = (y,z)) \cdot Pr(\tau_k = (y,z)) \end{aligned} \tag{5}$$

where the equality is justified by the following: for any event E, any random variable Y, any area $A \subseteq \mathbb{R}$ and any function f such that $Pr(Y \in A) = \int_{y \in A} f(y) dy = 1$,

$$Pr(E) = Pr(E, Y \in A) = \int_{y \in A} Pr(E | Y = y) f(y) dy.$$

The probability distribution for $Pr(\tau_k)$ can be replaced following the assumption on k.

The rest calculates $Pr(\tau_{k+1} = (s,t)|\tau_k = (y,z))$. When $|s-t| < u$, we have

$$\begin{aligned} & Pr(\tau_{k+1} = (s,t)|\tau_k = (y,z)) \\ =& Pr(X_1 = s-y, X_2 = t-z) \\ =& \lambda_1\lambda_2 e^{-\lambda_1(s-y)-\lambda_2(t-z)} dsdt; \end{aligned} \tag{6}$$

hence, when $|s-t| < u$, we substitute Equation (6) in Equation (5) and obtain

$$\begin{aligned} & Pr(\tau_{k+1} = (s,t)) \\ =& \lambda_1\lambda_2 e^{-\lambda_1 s-\lambda_2 t} D_k(s,t) dsdt. \end{aligned}$$

For the other two cases, we can calculate $Pr(\tau_{k+1} = (s,t)|\tau_k = (y,z))$:

$$Pr(\tau_{k+1} = (s,t)|\tau_k = (y,z)) = \begin{cases} \lambda_1 e^{-\lambda_1(s-y)-\lambda_2(t-z)} ds & \text{when } s-t = -u; \\ \lambda_2 e^{-\lambda_1(s-y)-\lambda_2(t-z)} dt & \text{when } s-t = u; \end{cases}$$

and $Pr(\tau_{k+1} = (s,t))$ (similarly to the case where $|s-t| < u$):

$$Pr(\tau_{k+1} = (s,t)) = \begin{cases} \lambda_1 e^{-\lambda_1 s-\lambda_2 t} D_k(s,t) ds & \text{when } s-t = -u; \\ \lambda_2 e^{-\lambda_1 s-\lambda_2 t} D_k(s,t) dt & \text{when } s-t = u. \end{cases}$$

Thus the statement for $k+1$ is proved based on the assumption for k. As a result, the statement of Lemma 1 is true for any $k \in \mathbb{Z}^+$.

B Proof of Lemma 2

We prove the statement by induction. For $k = 1$, we expand $D_1(s,t)$:

$$\begin{aligned}
D_1(s,t) =& \lambda_1\lambda_2 \int_{\substack{|y-z|<u,\\ 0<y<s,\\ 0<z<t}} dydz + \lambda_1 \int_{\substack{y-z=-u,\\ 0<y<s,\\ 0<z<t}} dy \\
&+ \lambda_2 \int_{\substack{y-z=u,\\ 0<y<s,\\ 0<z<t}} dz \\
=& \lambda_1\lambda_2 \int_0^s \int_{\max(0,y-u)}^{\min(t,y+u)} dzdy \\
&+ \lambda_1 I(t \geq u) \cdot \int_0^{\min(s,t-u)} dy \\
&+ \lambda_2 I(s \geq u) \cdot \int_0^{\min(t,s-u)} dz \\
=& \lambda_1\lambda_2 [st + I(s \geq u)(su - \frac{1}{2}u^2 - \frac{1}{2}s^2) \\
&+ I(t \geq u)(tu - \frac{1}{2}u^2 - \frac{1}{2}t^2)] \\
&+ \lambda_1 I(t \geq u)(t-u) + \lambda_2 I(s \geq u)(s-u)
\end{aligned}$$

where $I(P)$ is the indicator of whether a predicate P is true or not, and the last equality comes from the fact that $|s-t| \leq u$. Therefore, the difference between $D_1(s,t)$ and $D_1(t,s)$ is:

$$\begin{aligned}
& D_1(s,t) - D_1(t,s) \\
=& [\lambda_1 I(t \geq u)(t-u) + \lambda_2 I(s \geq u)(s-u)] \\
& - [\lambda_1 I(s \geq u)(s-u) + \lambda_2 I(t \geq u)(t-u)] \\
=& (\lambda_1 - \lambda_2)[I(t \geq u)(t-u) - I(s \geq u)(s-u)]
\end{aligned}$$

Since $s > t$, there are only three cases for the possible values of $I(t \geq u)$ and $I(s \geq u)$: (1) $I(t \geq u) = 1, I(s \geq u) = 1$; (2) $I(t \geq u) = 0, I(s \geq u) = 1$ and (3) $I(t \geq u) = 0$ and $I(s \geq u) = 0$. For case (1), $D_1(s,t) - D_1(t,s) = (\lambda_1 - \lambda_2)(t-s) < 0$; for case (2), $D_1(s,t) - D_1(t,s) = (\lambda_1 - \lambda_2)(u-s) < 0$; for case (3), $D_1(s,t) - D_1(t,s) = (\lambda_1 - \lambda_2) \cdot 0 = 0$. Thus the statement is true for $k = 1$.

Now assume that the statement is true for an arbitrary $k \geq 1$. Then for $k+1$, by Lemma 1, the difference between $D_{k+1}(s,t)$ and $D_{k+1}(t,s)$ is:

$$\begin{aligned}
&D_{k+1}(s,t) - D_{k+1}(t,s) \\
=&\lambda_1\lambda_2\Big(\int_{\substack{|y-z|<u,\\ 0<y<s,\\ 0<z<t}} D_k(y,z)dydz - \int_{\substack{|y-z|<u,\\ 0<y<s,\\ 0<z<t}} D_k(y,z)dydz\Big) \\
&+ \Big(\lambda_1 \int_{\substack{y-z=-u,\\ 0<y<s,\\ 0<z<t}} D_k(y,z)dy - \lambda_2 \int_{\substack{y-z=u,\\ 0<y<s,\\ 0<z<t}} D_k(y,z)dz \\
&+ \lambda_2 \int_{\substack{y-z=u,\\ 0<y<s,\\ 0<z<t}} D_k(y,z)dz - \lambda_1 \int_{\substack{y-z=-u,\\ 0<y<s,\\ 0<z<t}} D_k(y,z)dy\Big) \\
\triangleq&\Delta_1 + \Delta_2;
\end{aligned}$$

where after some simplification,

$$\Delta_1 = \lambda_1\lambda_2 \int_{|y-z|<u, t<y<s, 0<z<t} [D_k(y,z) - D_k(z,y)]dydz;$$

$$\Delta_2 = -\int_{z-y=u, t<z<s, 0<y<t} [\lambda_1 D_k(y,z) - \lambda_2 D_k(z,y)]dy.$$

Recall our assumption that the statement is true for k, i.e., $D_k(y,z) - D_k(z,y) < 0$ if $\lambda_1 > \lambda_2$, $y > z$ and $y > u$ and $D_k(y,z) - D_k(z,y) = 0$ if $\lambda_1 > \lambda_2$, $u > y > z$. Then $\Delta_1 < 0$ if $s > t$ and $s > u$ and $\Delta_1 = 0$ if $u > s > t$. Moreover, if $s < u$, $\Delta_2 = 0$; otherwise, if $s > u$, $s > t$ and $\lambda_1 > \lambda_2$, then $\Delta_2 < 0$. Thus, based on the assumption for k, the statement for $k+1$ is proved. Since we prove the result by induction, we can safely conclude that the statement of Lemma 2 is true for any $k \in \mathbb{Z}^+$.

C Proof of Lemma 3

We first prove the correctness of Equation (1). Before proving Equation (1), we calculate $Pr(W_k = 1)$. It is easy to calculate the probability of the event $W_k = 1$ conditioned on some given τ_{k-1}.

$$\begin{aligned}
&Pr(W_k = 1 | \tau_{k-1} = (s,t)) \\
=&Pr(\tau_{k-1,1} + X_{k-1,1} \leq \tau_{k-1,2} + X_{k-1,2} + u \text{ and} \\
&\tau_{k-1,1} + X_{k-1,1} + u < \tau_{k-1,2} + X_{k-1,2} | \tau_{k-1} = (s,t)) \\
=&Pr(s + X_{k-1,1} + u < t + X_{k-1,2}) \\
=&\frac{\lambda_1}{\lambda_1 + \lambda_2} \cdot e^{-\lambda_2(u+s-t)}, \text{ if } u + s - t \geq 0
\end{aligned}$$

where the last equality comes from the fact that $X_{k-1,i}$, $i \in \{1,2\}, \forall k \geq 2$ has a common distribution X_i. When $k = 1$, $\tau_{k-1} = (0,0)$ and thus $Pr(W_k = 1) =$

$\frac{\lambda_1}{\lambda_1+\lambda_2} \cdot e^{-\lambda_2 u}$. When $k \geq 2$, the probability distribution of τ_k is considered, and then by Lemma 1,

$$
\begin{aligned}
& Pr(W_k = 1) \\
=& \int_{\substack{s-t=-u, \\ s>0,t>0}} Pr(W_k = 1|\tau_{k-1} = (s,t)) \cdot Pr(\tau_{k-1} = (s,t)) \\
+& \int_{\substack{s-t=u, \\ s>0,t>0}} Pr(W_k = 1|\tau_{k-1} = (s,t)) \cdot Pr(\tau_{k-1} = (s,t)) \\
+& \int_{\substack{|s-t|<u, \\ s>0,t>0}} Pr(W_k = 1|\tau_{k-1} = (s,t)) \cdot Pr(\tau_{k-1} = (s,t)) \\
=& \frac{\lambda_1}{\lambda_1+\lambda_2} e^{-\lambda_2 u} \cdot \\
& [\int_{\substack{|s-t|<u, \\ s>0,t>0}} e^{-s(\lambda_1+\lambda_2)} \cdot \lambda_1\lambda_2 D_{k-2}(s,t) ds dt \\
& + \int_{\substack{s-t=-u, \\ s>0,t>0}} e^{-s(\lambda_1+\lambda_2)} \cdot \lambda_1 D_{k-2}(s,t) ds \\
& + \int_{\substack{s-t=u, \\ s>0,t>0}} e^{-s(\lambda_1+\lambda_2)} \cdot \lambda_2 D_{k-2}(s,t) dt].
\end{aligned}
$$

By symmetry, we can have similar formulas (when $k = 1$ and when $k \geq 2$) for $Pr(W_k = 2)$.

We then verify the correctness of Equation (1). For $k = 1$, by the formulas of $Pr(W_k = 1)$ and $Pr(W_k = 2)$, it is easy to verify the correctness of Equation (1). For $k \geq 2$, let

$$
\begin{aligned}
\Delta_1 =& \int_{\substack{|s-t|<u, \\ s>0,t>0}} e^{-s(\lambda_1+\lambda_2)} \cdot \lambda_1\lambda_2 D_{k-2}(s,t) ds dt \\
& - \int_{\substack{|s-t|<u, \\ s>0,t>0}} e^{-t(\lambda_1+\lambda_2)} \cdot \lambda_1\lambda_2 D_{k-2}(s,t) ds dt, \\
\Delta_2 =& [\int_{\substack{s-t=-u, \\ s>0,t>0}} e^{-s(\lambda_1+\lambda_2)} \cdot \lambda_1 D_{k-2}(s,t) ds \\
& + \int_{\substack{s-t=u, \\ s>0,t>0}} e^{-s(\lambda_1+\lambda_2)} \cdot \lambda_2 D_{k-2}(s,t) dt] \\
& - [\int_{\substack{s-t=-u, \\ s>0,t>0}} e^{-t(\lambda_1+\lambda_2)} \cdot \lambda_1 D_{k-2}(s,t) ds \\
& + \int_{\substack{s-t=u, \\ s>0,t>0}} e^{-t(\lambda_1+\lambda_2)} \cdot \lambda_2 D_{k-2}(s,t) dt].
\end{aligned}
$$

Then by the formula of $Pr(W_k = 1), k \geq 2$ and that of $Pr(W_k = 2), k \geq 2$ (by symmetry), we have

$$\begin{aligned}&\frac{\lambda_1+\lambda_2}{\lambda_1}e^{\lambda_2 u}Pr(W_k=1)-\frac{\lambda_1+\lambda_2}{\lambda_2}e^{\lambda_1 u}Pr(W_k=2)\\=&\Delta_1+\Delta_2, \forall k \geq 2.\end{aligned}$$

Clearly, if $\Delta_1 + \Delta_2 > 0$, then Equation (1) is correct for $k \geq 2$. After some simplification, we obtain the following equations.

$$\begin{aligned}\Delta_1 =&\lambda_1\lambda_2 \cdot \int_{\substack{|s-t|<u,\\ s>t,s>0,t>0}} [e^{-s(\lambda_1+\lambda_2)} - e^{-t(\lambda_1+\lambda_2)}]\\ &\cdot [D_{k-2}(s,t) - D_{k-2}(t,s)]dsdt;\\ \Delta_2 =&\int_{\substack{t-s=-u,\\ s>0,t>0}} [e^{-s(\lambda_1+\lambda_2)} - e^{-t(\lambda_1+\lambda_2)}]\\ &\cdot [\lambda_2 D_{k-2}(s,t) - \lambda_1 D_{k-2}(t,s)]dt.\end{aligned}$$

Since by Lemma 2, $\Delta_1 \geq 0$ and $\Delta_2 > 0$, Equation (1) is also correct for $k \geq 2$. As a result, the correctness of Equation (1) holds for any $k \in \mathbb{Z}^+$.

Next we prove the correctness of Equation (2). Before proving Equation (2), we calculate $Pr(W_k = 0, \ldots, W_{j-1} = 0, W_j = 1)$ by calculating the probability of $\tau_k, \ldots, \tau_{j-1}, \tau_j$ falling into the specific ranges which satisfy the event F_1 that $W_k = 0, \ldots, W_{j-1} = 0, W_j = 1$. Let $|t_{ix,1} - t_{ix,2}| < u, k \leq ix \leq j-1, ix \in \mathbb{Z}^+$ and $t_{j,1} - t_{j,2} = -u$. Since according to the mining algorithm, for any j,

$$\begin{aligned}&Pr(\tau_j = (t_{j,1}, t_{j,2})|\\ &\tau_k = (t_{k,1}, t_{k,2}), \ldots, \tau_{j-1} = (t_{j-1,1}, t_{j-1,2}))\\ =&Pr(\tau_j = (t_{j,1}, t_{j,2})|\tau_{j-1} = (t_{j-1,1}, t_{j-1,2})),\end{aligned}$$

therefore, we have

$$\begin{aligned}p \triangleq&Pr(\tau_k = (t_{k,1}, t_{k,2}), \ldots, \tau_j = (t_{j,1}, t_{j,2}))\\ =&Pr(\tau_k = (t_{k,1}, t_{k,2}))\\ &\cdot \prod_{ix=k}^{j-1} Pr(\tau_{ix+1} = (t_{ix+1,1}, t_{ix+1,2})|\tau_{ix} = (t_{ix,1}, t_{ix,2}))\end{aligned}$$

By Lemma 1 and the fact that random variables $X_{k,i}$, $X_{k+1,i}, \ldots, X_{j-1,i}, i \in \{1, 2\}$ are identically distributed with common distribution X_i, after some simplification, we have

$$\begin{aligned}p =&\lambda_1(\lambda_1\lambda_2)^{j-k}e^{-\lambda_1 t_{j,1}-\lambda_2 t_{j,2}}\\ &\cdot D_{k-1}(t_{k,1}, t_{k,2}) \prod_{ix=k}^{j-1} dt_{ix,1}dt_{ix,2} \cdot dt_{j,1}.\end{aligned}$$

Then the probability of the event F_1 is

$$Pr(F_1) = \int_{\substack{t_{j,1}-t_{j,2}=-u,\\ t_{j,1}>0,\\ t_{j,2}>0}} \prod_{ix=j-1}^{k} \int_{\substack{|t_{ix,1}-t_{ix,2}|<u,\\ 0<t_{ix,1}<t_{ix+1,1},\\ 0<t_{ix,2}<t_{ix+1,2}}} p.$$

Then after some simplification,

$$Pr(F_1) = \lambda_1(\lambda_1\lambda_2)^{j-k}e^{-\lambda_2 u}\Delta_1,$$

where

$$\Delta_1 = \int_{\substack{t_{j,1}-t_{j,2}=-u,\\ t_{j,1}>0,\\ t_{j,2}>0}} \prod_{ix=j-1}^{k} \int_{\substack{|t_{ix,1}-t_{ix,2}|<u,\\ 0<t_{ix,1}<t_{ix+1,1},\\ 0<t_{ix,2}<t_{ix+1,2}}} e^{-t_{j,1}(\lambda_1+\lambda_2)} \cdot D_{k-1}(t_{k,1}, t_{k,2}) \prod_{ix=k}^{j-1} dt_{ix,1}dt_{ix,2} \cdot dt_{j,1}.$$

Similarly, for $W_k = 0, \ldots, W_{j-1} = 0, W_j = 2$ denoted event F_2, $Pr(F_2) = \lambda_2(\lambda_1\lambda_2)^{j-k}e^{-\lambda_1 u}\Delta_2$, where

$$\Delta_2 = \int_{\substack{t_{j,2}-t_{j,1}=-u,\\ t_{j,1}>0,\\ t_{j,2}>0}} \prod_{ix=j-1}^{k} \int_{\substack{|t_{ix,1}-t_{ix,2}|<u,\\ 0<t_{ix,1}<t_{ix+1,1},\\ 0<t_{ix,2}<t_{ix+1,2}}} e^{-t_{j,2}(\lambda_1+\lambda_2)} \cdot D_{k-1}(t_{k,1}, t_{k,2}) \prod_{ix=k}^{j-1} dt_{ix,1}dt_{ix,2} \cdot dt_{j,2}.$$

We then verify the correctness of Equation (2). For any $b > 0$, let

$$\Omega_1 = \int_{\substack{|t_{j-1,1}-t_{j-1,2}|<u,\\ 0<t_{j-1,1}<b,\\ 0<t_{j-1,2}<b+u}} \prod_{ix=j-2}^{k} \int_{\substack{|t_{ix,1}-t_{ix,2}|<u,\\ 0<t_{ix,1}<t_{ix+1,1},\\ 0<t_{ix,2}<t_{ix+1,2}}} D_{k-1}(t_{k,1}, t_{k,2}) \prod_{ix=k}^{j-1} dt_{ix,1}dt_{ix,2}$$

$$\Omega_2 = \int_{\substack{|t_{j-1,1}-t_{j-1,2}|<u,\\ 0<t_{j-1,1}<b+u,\\ 0<t_{j-1,2}<b}} \prod_{ix=j-2}^{k} \int_{\substack{|t_{ix,1}-t_{ix,2}|<u,\\ 0<t_{ix,1}<t_{ix+1,1},\\ 0<t_{ix,2}<t_{ix+1,2}}} D_{k-1}(t_{k,1}, t_{k,2}) \prod_{ix=k}^{j-1} dt_{ix,1}dt_{ix,2}.$$

Then we have

$$\begin{aligned}
&\frac{\lambda_1+\lambda_2}{\lambda_1}e^{\lambda_2 u}Pr(F_1) - \frac{\lambda_1+\lambda_2}{\lambda_2}e^{\lambda_1 u}Pr(F_2)\\
=&(\lambda_1\lambda_2)^{j-k}(\Delta_1 - \Delta_2)\\
=&(\lambda_1\lambda_2)^{j-k}\int_{b>0} e^{-b(\lambda_1+\lambda_2)}(\Omega_1 - \Omega_2)db,
\end{aligned}$$

where after some simplification,

$$\begin{aligned}
&\Omega_1 - \Omega_2 \\
&= \int_{\substack{|t_{j-1,1}-t_{j-1,2}|<u,\\ 0<t_{j-1,1}<b,\\ b<t_{j-1,2}<b+u}} \prod_{ix=j-2}^{k} \int_{\substack{|t_{ix,1}-t_{ix,2}|<u,\\ 0<t_{ix,1}<t_{ix+1,1},\\ t_{ix+1,1}<t_{ix,2}<t_{ix+1,2}}} \\
&\quad [D_{k-1}(t_{k,1}, t_{k,2}) - D_{k-1}(t_{k,2}, t_{k,1})] \prod_{ix=k}^{j-1} dt_{ix,1} dt_{ix,2}
\end{aligned}$$

Since by Lemma 2, $\Omega_1 - \Omega_2 \geq 0$, therefore Equation (2) is correct.

D Proof of Theorem 2

As a first step to prove Theorem 2, we generalize Theorem 1. In general, the two miners may start at arbitrary time instants T_1 and T_2 respectively. Then with a similar proof to that of Lemma 3, we can generalize Lemma 3 as well as Theorem 1 to arbitrary $T_1, T_2 > 0$ in Corollary 2.[10]

Corollary 2. *Following Definitions 1, 2 and 3, except for $\tau_0 = (T_1, T_2)$ for some $T_1, T_2 > 0$ here, for any $j \in \mathbb{Z}^+$ and $a \geq k+1$, if $\lambda_1 > \lambda_2$, then*

$$\frac{p_{k,1,a}}{p_{k,2,a}} > r, \quad \forall k \geq 2, \quad \textit{where } r = \frac{e^{\lambda_1 u}}{e^{\lambda_2 u}} \cdot \frac{\lambda_1}{\lambda_2}.$$

For $i \in \{1, 2\}$, denote by $p_{k,i,\infty}$ the success probability $p_{k,i,a}$ when a goes to infinity. For $k = 1$, if $\lambda_1 > \lambda_2$ and $|T_1 - T_2| \leq u$, then

$$\begin{aligned}
\frac{p_{k,1,\infty}}{p_{k,2,\infty}} &\geq \frac{Pr(W_k = 1) + [1 - Pr(W_k = 1) - Pr(W_k = 2)] \cdot r_1}{Pr(W_k = 2) + [1 - Pr(W_k = 1) - Pr(W_k = 2)] \cdot r_2}, \\
\textit{where } r_1 = \frac{r}{1+r},\ r_2 = \frac{1}{1+r},\ &\textit{and } Pr(W_k = 1) = \frac{\lambda_1}{\lambda_1 + \lambda_2} \cdot e^{-\lambda_2(u+T_1-T_2)}, \\
Pr(W_k = 2) &= \frac{\lambda_2}{\lambda_1 + \lambda_2} \cdot e^{-\lambda_1(u+T_2-T_1)}.
\end{aligned}$$

Corollary 2 also covers the case where some miner, for example, $\mathcal{M}_1$ crashes and later recovers at time T_1. By the stationary property of the Poisson process, $\mathcal{M}_1$ and $\mathcal{M}_2$ can be considered as starting with the same chain at different time instants: T_1 and T_2 for some $T_2 < T_1$ such that $|T_1 - T_2| < u$. As a result, Corollary 2 shows the unfairness of blockchain among honest miners in general, regardless of crashes. Following Corollary 2, we expand the expression and restate Theorem 2 as follows.

[10] We delay any message msg (that may arrive before a miner has started) to arrive after the receiver of msg starts.

Theorem 3. *Consider the selfish mining algorithm in [11] with $\mathcal{M}_1$, the set of honest miners and $\mathcal{M}_2$, the pool of selfish miners. W.l.o.g., $\mathcal{M}_1$ and $\mathcal{M}_2$ are two miners with hashrates λ_1 and λ_2 respectively. Let*

$$\mathcal{U} = \frac{2P_1\rho_1(1-\alpha) + 2P_2\rho_2(1-\alpha) + P_{k>2}\rho_2(1-\alpha)}{2P_1(1-\alpha) + 2P_2(1-\alpha) + P_0(1-\alpha) + P_{k>2}(1-\alpha) + P_2(1-\rho_2)(1-\alpha)}$$

where $\rho_1 = \alpha e^{-2\lambda_1 u} + \alpha(1-e^{-2\lambda_1 u})r_2$, and $\rho_2 = e^{-2\lambda_1 u} + (1-e^{-2\lambda_1 u})[\alpha + (1-\alpha)(1-e^{-2\lambda_2 u})r_2]$ with $\alpha = \frac{\lambda_2}{\lambda_1+\lambda_2}$, r_1 defined in Corollary 2, and P_1, P_2, $P_{k>2}$ are one solution to the following equations:

$$\begin{aligned}
\alpha P_0 &= [(1-\rho_1) + \rho_1 e^{-\lambda_2 \cdot 2u}]P_{0'} + [(1-\rho_2)(1-\alpha) + \rho_2(1-\alpha)e^{-\lambda_2 \cdot 2u}]P_2 \\
P_1 &= \alpha P_0 + \rho_1 e^{-\lambda_2 \cdot 2u}\lambda_2 \cdot 2uP_{0'} + \rho_2(1-\alpha)e^{-\lambda_2 \cdot 2u}\lambda_2 \cdot 2uP_2 \\
\forall k \geq 2, P_k &= \alpha P_{k-1} + (1-\alpha)P_{k+1} \\
&\quad + \rho_2(1-\alpha)(e^{-\lambda_2 \cdot 2u}\frac{(\lambda_2 \cdot 2u)^k}{k!})P_2 + \rho_1(e^{-\lambda_2 \cdot 2u}\frac{(\lambda_2 \cdot 2u)^k}{k!})P_{0'} \\
P_{0'} &= (1-\alpha)P_1.
\end{aligned}$$

Now to prove Theorem 3, we upper bound the probability p_j of $\mathcal{M}_2$ committing $\mathcal{C}_{mi}$ conditioned on E_j in Lemma 4. (In contrast, Eyal and Sirer [11] calculated exact formula $p_1 = \alpha$ and $p_2 = 1$ assuming $u = 0$.) Then we count N_2 and N_1, and prove Theorem 3.

Lemma 4. *Given message delay $u > 0$, for $j \in \{1, 2\}$, $p_j \leq \rho_j$ with ρ_j defined in Theorem 3.*

Proof. For $j \in \{1, 2\}$, let p_j^* be the probability of $\mathcal{M}_2$'s commit conditioned on E_j after which $\mathcal{M}_2$ behaves as an honest miner. We first prove an upper bound on p_j^* and then show that $p_j \leq p_j^*$ so that the upper bound on p_j^* is also an upper bound on p_j.

First we establish an upper bound on p_1^*. If $\mathcal{M}_2$ commits given E_1, then $\mathcal{M}_2$ has created a new block CB on C_{mi} and committed CB. Define 0 as the time when $\mathcal{M}_1$ sends $\mathcal{C}_{ma}$. Then conditioned on E_1, $\mathcal{M}_1$ and $\mathcal{M}_2$ start with blockchains of the same length at time $T_1 = 0$ and $T_2 = u$ respectively, and p_1^* is the success probability $p_{1,2,\infty}$ (defined in Corollary 2) for the first block CB which may be created by $\mathcal{M}_2$. Then Corollary 2 applies to $p_1^* = p_{1,2,\infty}$. As the sum of $p_{1,2,\infty}$ and $p_{1,1,\infty}$ is 1, it is easy to verify that $p_1^* \leq \rho_1$.

We then prove an upper bound on p_2^*. If $\mathcal{M}_2$ commits given E_2, then $\mathcal{M}_2$ commits the last block of $\mathcal{C}_{mi}$. As event E_2 is an update of $\mathcal{C}_{ma}$, then to $\mathcal{M}_1$, between when $\mathcal{M}_1$ sends this $\mathcal{C}_{ma}$ and when $\mathcal{M}_1$ receives $\mathcal{C}_{mi}$, there are $2u$ units of time. As a result, for $\mathcal{M}_2$, to commit (the last block of) $\mathcal{C}_{mi}$ is equivalent to the union of the following two events: (1) $\mathcal{M}_1$ has not created a block on $\mathcal{C}_{ma}$ during these $2u$ units of time, i.e., before $\mathcal{C}_{mi}$ arrives (denoted event E_a); and (2) $\mathcal{M}_1$ has created a block on $\mathcal{C}_{ma}$ before $\mathcal{C}_{mi}$ arrives (denoted E_b) but $\mathcal{M}_2$ manages to create a new block CB on $\mathcal{C}_{mi}$ and commit CB. Conditioned on E_2, the probabilities of E_a and E_b are $e^{-2\lambda_1 u}$ and $1 - e^{-2\lambda_1 u}$ respectively. Define 0

as the time when $\mathcal{M}_2$ sends $\mathcal{C}_{mi}$. Conditioned on E_b as well as E_2, suppose that $\mathcal{M}_1$ creates a block on $\mathcal{C}_{ma}$ at some time $T_1, -u < T_1 < u$. In this case, then $\mathcal{M}_1$ and $\mathcal{M}_2$ are two miners starting with blockchains of the same length at time T_1 and $T_2 = 0$ respectively. As a result, $p_2^* = e^{-2\lambda_1 u} + (1 - e^{-2\lambda_1 u}) \cdot p_{1,2,\infty}$ (with $p_{1,2,\infty}$ defined in Corollary 2). To have an upper bound on p_2^*, we assume $T_1 = u$. Then by Corollary 2, we can similarly verify that $p_2^* \leq \rho_2$.

Finally, we show that $p_j \leq p_j^*, j \in \{1, 2\}$. Clearly, to show the inequality, we focus on how selfish $\mathcal{M}_2$ deviates from the mining algorithm: when $\mathcal{M}_2$ believes $\mathcal{M}_1$ to have adopted $\mathcal{M}_2$'s chain $\mathcal{C}_{mi}$, $\mathcal{M}_2$ keeps some newly created block(s) private. However, such deviation allows $\mathcal{M}_1$ to have more time in creating new blocks, which decreases the probability that $\mathcal{M}_1$ adopts $\mathcal{M}_2$'s (possibly longer) chain. In other words, $p_j \leq p_j^*$. As a result, an upper bound on p_j^*, which we derive above, is also an upper bound on p_j.

Proof (Proof of Theorem 3). If $\mathcal{M}_1$ has created a new block before $\mathcal{C}_{mi}$ arrives, then we say that $\mathcal{M}_1$ and $\mathcal{M}_2$ enter a competition (in committing the blocks they have just created respectively). Recall that after event E_j, $\mathcal{M}_1$ and $\mathcal{M}_2$ might enter a competition which ends at some point where $\mathcal{M}_1$ and $\mathcal{M}_2$ start to mine on some same chain $\mathcal{C}$. We first show that to have an upper bound $\mathcal{U} > \alpha$, we can neglect the expected number of additional blocks (created during the competition and) committed in $\mathcal{C}$. Let CB be the last block of $\mathcal{C}_{mi}$ when E_j occurs (no matter whether $j \in \{1, 2\}$). If any additional block created by $\mathcal{M}_2$ is committed in $\mathcal{C}$, then CB is also committed. Thus by Lemma 4, the probability of $\mathcal{M}_2$ committing any additional block conditioned on E_1 is upper bounded by ρ_1, while the probability of $\mathcal{M}_2$ committing any additional block conditioned on E_2 is upper bounded by $\gamma_2 = (\rho_2 - e^{-2\lambda_1 u}) \frac{1}{1 - e^{-2\lambda_1 u}}$ where $1 - e^{-2\lambda_1 u}$ is the probability of $\mathcal{M}_1$ and $\mathcal{M}_2$ entering a competition conditioned on E_2. As a result, by the linearity of expectation, after event E_1, the proportion of the expected number of these additional blocks created by $\mathcal{M}_2$ is no more than ρ_1. Conditioned on E_2, if E_a (defined in the proof of Lemma 4, as well as E_b) occurs, then neither miner creates any additional block; if E_b occurs, then the proportion of the expected number of these additional blocks created by $\mathcal{M}_2$ is no more than γ_2. Therefore, since $\rho_1 < \alpha$, $\gamma_2 < \alpha$ and $\mathcal{U} > \alpha$, we can neglect these additional blocks when calculating an upper bound on R.

Therefore, we count N_2 and N_1 as follows. Let $P_k, k \in \mathbb{Z}^+$ be the probability of the event S_k that $\mathcal{C}_{mi}$ is k blocks longer than $\mathcal{C}_{ma}$. Let $P_{0'}$ be the probability of the event $S_{0'}$ that $\mathcal{C}_{mi}$ and $\mathcal{C}_{ma}$ have the same length but are different. Let P_0 be the probability of the event S_0 that $\mathcal{C}_{mi}$ and $\mathcal{C}_{ma}$ are the same. Then

$$N_1 = P_{0'} \cdot (1-p_1) \cdot 2 + P_2 \cdot (1-\alpha)(1-p_2) \cdot 3 + P_{k>2} \cdot (1-\alpha)(1-p_2) + P_0 \cdot (1-\alpha),$$

$$N_2 = P_{0'} \cdot p_1 \cdot 2 + P_2 \cdot (1-\alpha) p_2 \cdot 2 + P_{k>2} \cdot (1-\alpha) p_2.$$

Combined with Lemma 4, we obtain

$$\frac{N_2}{N_1 + N_2} \leq U$$

Probabilities $P_{0'}$ and $P_k, k \geq 0$ are calculated below. The state machine of selfish mining with message delay is illustrated in Figure 4. The state machine in Figure 4 does not include the intermediary states during a commit as the blocks created during a commit are not counted as explained above. (It is also due to the fact that the fraction $\frac{N_2}{N_1+N_2}$ relies only on the relative values of the probability of each state and thus there is no need to cover all states.) Clearly, the probabilities of two state transitions, state $S_{0'}$ to state 0 and state S_2 to state S_0, are adjusted according to Lemma 4. In addition, when $\mathcal{M}_2$ succeeds in committing $\mathcal{C}_{mi}$, $\mathcal{M}_2$ has additionally $2u$ units of time in mining. We thus define N_{2u} as the random variable that represents the number of blocks $\mathcal{M}_2$ may find in this time period, and then there are possibilities of additional state transitions: state $0'$ to state k and state 2 to state k, for $k \in \mathbb{Z}^+$. Since $Pr[N_{2u} = k] = e^{-\lambda_2 \cdot 2u} \frac{(\lambda_2 \cdot 2u)^k}{k!}$ for $k \geq 0$, it is easy to verify that the equations listed in Theorem 3 correspond to Figure 4 and $P_{0'}$ and $P_k, k \geq 0$ are one solution to those equations.