Security Bounds for Bitcoin Under Network Delay

Mustafa Doger Sennur Ulukus
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
doger@umd.edu *ulukus@umd.edu*

***Abstract*—We improve security-latency bounds of Nakamoto consensus by analyzing the race between adversarial and honest chains in three different phases: pre-mining, confirmation and post-confirmation. We find the probability distribution of the length of the adversarial chain and the rigged adversarial chain under jumper models during the confirmation interval. We analyze certain properties of this race to model pre-mining and post-confirmation phases with random walks that provide tighter bounds than existing results. Combining all three phases provides novel upper and lower bounds for blockchains with small $\lambda\Delta$.**

I. Introduction

Introduced by Nakamoto in 2008 [1], Bitcoin enables public users to maintain a ledger in a distributive manner. Transactions listed in the public ledger are secured by the longest chain protocol using a Proof of Work (PoW) approach. Honest miners who follow the protocol extend the longest chain of blocks containing the transactions. A new block is required to contain a nonce that satisfies difficulty requirements of the chain, making it valid. However, even if a block is at the tip of the longest chain and satisfies all requirements, it cannot be immediately confirmed due to a phenomenon called "forking." The forking phenomenon occurs due to the network delays and adversarial activities.

Assuming all miners work on the same longest chain, the first one to mine a new block sends the new block to its peers immediately. However, due to the finite speed of light and network constraints, the peers receive the new block with some delay. During this delay, they might be able to mine a new block themselves with different content at the same height, hence the name "forking." Forks also happen when adversaries do not follow the longest chain protocol and try to undermine the ledger. As a result, the deeper a transaction is in the longest chain, the more secure its content will be. This observation hints at the trade-off between security and latency of transactions. Following the Bitcoin whitepaper [1], the first to study the latency-security problem of blockchains is Garay et. al. [2]. Further studies have extended the analysis to practical non-lockstep protocols, see e.g., [3].

Recently, several studies have given extensive bounds on the security-latency trade-off under network delay Δ [4]–[6]. They investigate how secure a block is at any point during the execution of the protocol under a network delay of Δ. Specifically, [4] considers races between Poisson and renewal processes to give upper and lower bounds on how secure a block is after it is confirmed. [6] analyzes a dynamic programming algorithm to numerically bound the safety violation probability. While [4] considers confirmation rules that treat latency in terms of time units, [5] considers the case where the committing rule is defined in terms of how deep a block is in the current longest chain, and outperforms the results of [6] in small $\lambda\Delta$ regimes, such as Bitcoin where $\lambda\Delta \approx 1/60$.

Our work focuses on the approach studied in [5], and we tighten the lower and upper bounds of security guarantees significantly. We use the "rigged" model introduced in [5] and modify the region of interest where adversarial and honest chains are racing when the target transaction enters the system. By doing so, we present significant improvements, and by orders of magnitude, for certain parameter regimes. We provide a formula to calculate the probability of achievable and converse results. We present our results for Bitcoin and Ethereum settings with various fractions of adversarial presence.

For Bitcoin, where a block is mined approximately every 10 minutes and assuming network delay is at max 10 seconds, under widely adopted 6-block confirmation rule, safety violation probability is narrowed down to between 0.112% and 0.173% under 10% adversarial presence. For comparison, the previous best-known results [5] were between 0.106% and 0.353%.

II. System Model

A. Protocol: Honest Nodes and Adversaries

We assume that the reader has familiarity with blockchain protocols. Here, we abstract out the basics of the protocol together with the blockchain data structure and validity constraints. In this abtract system, n nodes participate in a network to maintain a distributed ledger which initially consists solely of the genesis (zeroth) block. Honest nodes, who make up α fraction of all nodes, stick to the protocol, i.e., they try to mine a new block at the tip of the longest chain they have seen so far. Whenever a block is mined by an honest miner, the block is shared and assumed to be seen by all miners within a Δ amount of time. Adversarial miners are allowed to deviate from the protocol, i.e., they are not required to mine at the tip of their longest chain and can decide not to share their blocks. However, their blocks should contain a valid PoW (or Proof of Stake (PoS) depending on the model).

A widely adopted model for building a blockchain data structure is to assume that new blocks arrive (i.e., are mined) according to a Poisson process. Hence, the interarrival-times of mined blocks are independent exponentially distributed random variables with mean $1/\lambda$, and a block is honest with probability (w.p.) α by Poisson splitting. The fraction of adversarial miners in the system is denoted by $\beta = 1 - \alpha$,

which is assumed to satisfy $\beta < \frac{1-\beta}{1+(1-\beta)\lambda\Delta}$ [7]. We further assume that the entire adversarial power is concentrated in the hands of a single entity and adopt the convention of strong adversaries in the literature, i.e., the adversary controls the network delay as long as any introduced delay is at most Δ and the ties are broken in the adversary's favor.

B. Confirmation Rule

A block and transactions within that block are considered to be confirmed according to a k-block confirmation rule, if it is part of the longest chain and there are at least k blocks mined on top of it, in the view of an honest miner. The aim of the adversary is to spend the same resource on more than one transaction, i.e., double spend. Hence, we say a confirmed transaction is discarded if and only if a block containing the transaction is confirmed, and later, another block containing a conflicting transaction on the same height is confirmed.

In this paper, we are interested in a certain "target transaction" tx which enters the transaction pool at time τ. We assume that honest miners try to mine a new block ("target block") containing tx at the tip of their longest chain if possible. We would like to calculate lower and upper bounds on the probability of discarding tx after confirmation. In this terminology, anything that the adversary has the ability to do is considered achievable (lower bound). For the upper bound, as explained in [5], we use a "rigged" model which makes the adversary strictly more powerful than physically possible, hence an unachievable scenario.

III. Lower Bound

There are different achievable adversarial strategies. The strategy considered in [5] makes use of $\Delta = 0$ which is a lower bound to non-zero delay but depending on the magnitude of $\lambda\Delta$, this bound may actually be quite off from reality. Here, we deploy an adversarial strategy known as *private attack* with Δ delay, where, the adversary delays every honest block by the maximum allowed Δ, in the meanwhile, mining a private chain in order to double spend. In this scenario, the length of the honest chain will be equal to the number of published jumpers from the start of the protocol until the current time. Jumpers are the first honest blocks that are mined at least Δ time after each other starting with the genesis block [4]. Hence, under the k-block confirmation rule, we calculate a lower bound for the probability that the kth jumper after the target block is mined and at some point from then on tx is excluded from the longest chain of an honest miner.

A. Pre-Mining Gain

We start with the pre-mining gain, i.e., the lead L, which is the difference in the heights of the longest chain versus the longest honest chain before τ; this is the adversary's lead. Assuming τ is large enough, the lead can be modeled as the steady state distribution of an extended birth-death process [4], [5]. When the birth-death process is extended to incorporate the Δ delay strategy, we obtain a Markov chain with the following state transition probability matrix,

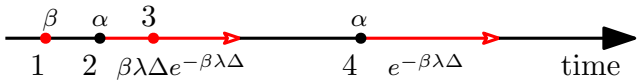

Fig. 1. A sample path of arrivals.

$$P = \begin{bmatrix} \alpha_0 & \alpha_1+\beta & \alpha_2 & \alpha_3 & \alpha_4 & \dots \\ \alpha_0 & \alpha_1 & \alpha_2+\beta & \alpha_3 & \alpha_4 & \dots \\ 0 & \alpha_0 & \alpha_1 & \alpha_2+\beta & \alpha_3 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (1)$$

This Markov chain is constructed as follows: At each "mining event," we toss a coin, the result is either an adversarial block w.p. β or an honest (jumper) block w.p. α, in which case we further consider how many adversarial arrivals there have been in the Δ delay interval of that jumper. Hence, we assume that any honest jumper block arrives together with some number of adversarial blocks which are mined within the Δ delay of the honest block, and that number has a Poisson distribution. In (1), the $(i+1, j+1)$th element of P denotes the probability that the lead goes from i to j. Here, $\alpha_i = \alpha \cdot e^{-\beta\lambda\Delta} \cdot \frac{(\beta\lambda\Delta)^i}{i!}$, which is the probability that an honest jumper block is mined followed by i adversarial blocks are mined within the Δ delay interval.

Fig. 1 shows a sample path of arrivals to visualize this random process. The first coin toss results in an adversarial block 1 w.p. β. The second coin toss, independent of the first one, results in an honest block 2 w.p. α, and we consider the Δ delay interval of this toss (represented by the red arrow) which results in an additional adversarial block w.p. $\beta\lambda\Delta e^{-\beta\lambda\Delta}$. Any honest arrivals during this red interval can be ignored, as they only fork the jumper chain, and do not contribute to the length of the longest chain. After the Δ delay ends, the next coin toss is independent of anything that happened before, hence, we have a memoryless process, represented by a Markov chain.

Since we are finding a lower bound for the lead, we further simplify this strategy by assuming that there are at most two adversarial arrivals during the Δ delay interval of any jumper. Thus, there are three possible outcomes of a coin toss that represents the next arrival: 1) E_1, denoting a jumper honest block arrival (no adversarial arrival during the Δ delay interval) results in honest chain growing by one block. 2) E_2, denoting a jumper honest block arrival together with an adversarial block arrival during the Δ delay interval that results in one block of growth in both honest and private chains. 3) E_3, denoting an adversarial arrival or a jumper honest block arrival together with two adversarial block arrivals during the Δ delay interval. Note that the two possible events listed in E_3 result in net one block growth in the adversarial chain, so we treat them the same. Furthermore, if the lead is zero, for the sake of simplicity, we assume that there can be at most one adversarial arrival during the Δ delay interval of a jumper. Given the above arguments, the following Markov chain is a

lower bound on the lead of the best adversarial strategy,

$$P' = \begin{bmatrix} \alpha_0 & 1-\alpha_0 & 0 & 0 & 0 & \dots \\ \alpha_0 & \alpha_1 & \beta_1 & 0 & 0 & \dots \\ 0 & \alpha_0 & \alpha_1 & \beta_1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{2}$$

where $\beta_1 = 1-\alpha_0-\alpha_1$. Letting π_i denote $\mathbb{P}(L=i)$, we find $\pi_0 = \frac{\alpha_0-\beta_1}{1-\beta_1}$, $\pi_1 = \pi_0 \cdot \frac{1-\alpha_0}{\alpha_0}$, and $\pi_i = \pi_1(\frac{\beta_1}{\alpha_0})^{i-1}$ for $i > 1$. We denote the PMF of L with P_1, CDF of L with F_1, and CCDF of L with $\bar{F}_1 = 1 - F_1$.

B. Confirmation Interval

At time τ, target transaction tx enters the transaction pool of miners and will be included in the next honest (jumper) block by assumption. We call the interval starting at τ and ending when the target block containing tx becomes k-deep in the longest chain of all honest miners, the confirmation interval. Our goal is to find the number of adversarial blocks mined during this confirmation interval. Note that the lead found previously is a steady-state scenario and assumed to be independent of the Poisson arrivals starting at τ. Letting τ_c denote the mining time of the $(k+1)$th jumper block mined starting from τ, we use the interval $[\tau, \tau_c + \Delta]$ as our confirmation interval. Next, we find the adversarial arrival distribution for the confirmation interval defined above under Δ delay strategy.

Lemma 1 *The number of adversarial arrivals during the confirmation interval S has the following distribution,*

$$P_S(s) = \alpha_0^{k+1}\beta^s \sum_{n=0}^{s} \binom{k+n}{n} \frac{(\lambda\Delta(k+1))^{s-n}}{(s-n)!} \tag{3}$$

We denote the PMF of S with P_2. At this point, the adversary wins the race if sum of the lead and confirmation gain is more than k, i.e., $L + S > k$.

C. Post-Confirmation Race

After the confirmation interval, if the adversary is behind in the race, it still has a chance to win. Let $D = k + 1 - L - S$ denote the deficit of the adversary right after the confirmation interval. We can represent this part of the race with a random walk that starts from the origin and moves according to a three-way coin toss with events E_1 (moving one step to the left), E_2 (net zero movement), and E_3 (moving one step to the right). We can denote the ith toss with $W_i \in \{-1, 0, 1\}$. Note that we trim the adversarial arrivals during the Δ delay interval to at most two as we did in the pre-mining gain stage. Let T_i denote the current position of this random walk after i three-way coin tosses, i.e., $T_i = \sum_{j=1}^{i} W_j$. If the random walk ever reaches D, i.e., $\max_{i\geq 1} T_i \geq D$, then, the adversary wins. Moreover, if $\max_{i\geq 1} T_i = D-1$ and E_2 happens while the random walk is at $D-1$ at any point of the process, the adversary wins due to the ability of publishing adversarial blocks mined during the Δ delay interval before the jumper. Thus, combining these two possibilities, we denote the event that the adversary catches the honest chain with $\max_{i\geq 1} T'_i = \max_{i\geq 1}(T_i + \mathbb{1}_{W_i=0}) \geq D$. We denote the PMF of $\max_{i\geq 1} T'_i$ with P_3. In Lemma 2 we find the probability of this event.

Lemma 2 *Consider a sequence of i.i.d. random variables denoted by W_i, $i \geq 1$, and $\mathbb{P}(W_i = -1) = \mathbb{P}(E_1) = \alpha_0$, $\mathbb{P}(W_i = 0) = \mathbb{P}(E_2) = \alpha_1$, $\mathbb{P}(W_i = 1) = \mathbb{P}(E_3) = \beta_1$. Let $T'_i = \mathbb{1}_{W_i=0} + \sum_{j=1}^{i} W_j$. Then, for $a \geq 1$,*

$$\mathbb{P}\left(\max_{i\geq 1} T'_i \geq a\right) = \left(\frac{\beta_1}{\alpha_0}\right)^{a-1}\left(\frac{1-\alpha_0}{1-\beta_1}\right) \tag{4}$$

Finally, we put everything together, namely, the analysis of the steady state (i.e., pre-mining gain), confirmation interval, and post-confirmation race in the following theorem.

Theorem 1 *Given mining rate λ, honest fraction α, delay bound Δ and confirmation depth k, a confirmed transaction can be discarded w.p. at least:*

$$\bar{F}_1(k) + \sum_{i=0}^{k} P_1(i)\bar{F}_2(k-i) + \sum_{i+j\leq k} P_1(i)P_2(j)\bar{F}_3(k-i-j) \tag{5}$$

IV. Upper Bound

To find an upper bound on the security-latency guarantee, one has to know the best adversarial strategy. It is proven in [7] that there is no deterministic adversarial strategy that outperforms other strategies for all possible arrivals. However, it is also proven in [5] that *private attack* is the best attack under the condition that all honest blocks are on different heights (we denote this condition in short as AHBODH). To make use of this fact, [5] uses a rigged model where some honest arrivals are converted into adversarial ones. The bound performs well for small $\lambda\Delta$ and k, however, as these parameters grow, the gap between the upper and lower bounds grows significantly. We improve this model by considering two arrivals at a time during pre-mining and post-confirmation intervals instead of one arrival at a time. We also use the Pascal distribution idea during the confirmation interval which gives an improvement.

A. Pre-Mining Gain

In [5] every honest arrival that is a tailgater, i.e., arrivals within Δ of any other arrival, is converted to an adversarial block. However, this need not be true for AHBODH to hold. For example, an honest arrival need not be converted if all arrivals within the preceding Δ are already adversarial. More specifically, consider the honest chain to be consisting of only honest jumper blocks and all other blocks that are mined belong to the adversarial chain, thus, AHBODH holds, and the transition matrix during pre-mining is as follows,

$$\bar{P} = \begin{bmatrix} \bar{\alpha}_0 & \bar{\alpha}_1+\beta & \bar{\alpha}_2 & \bar{\alpha}_3 & \bar{\alpha}_4 & \dots \\ \bar{\alpha}_0 & \bar{\alpha}_1 & \bar{\alpha}_2+\beta & \bar{\alpha}_3 & \bar{\alpha}_4 & \dots \\ 0 & \bar{\alpha}_0 & \bar{\alpha}_1 & \bar{\alpha}_2+\beta & \bar{\alpha}_3 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{6}$$

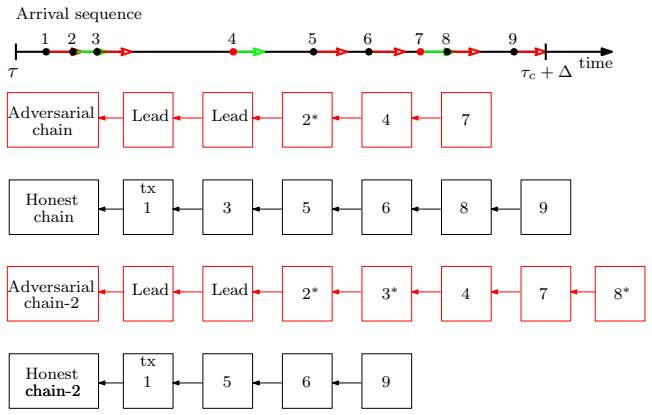

Fig. 2. A sample paths of arrivals and rigged chains

where $\bar{\alpha}_i = \alpha \cdot e^{-\lambda\Delta} \cdot \frac{(\lambda\Delta)^i}{i!}$. Note the difference between P and $\bar{P}$: When we upper bound the lead, we toss a coin, if it lands honest, each arrival during Δ delay interval that follows this block is considered as adversarial, whereas, we discarded honest blocks arriving in the Δ delay interval when we lower bounded the lead. Hence, all blocks except jumpers belong to the adversarial chain.

As an example, consider the arrival sequence shown in Fig. 2 when confirmation interval begins with τ and ends with $\tau_c + \Delta$ (for the honest chain on top) where $k = 5$. Black dots represent honest arrivals and red dots represent adversarial arrivals. Red arrows represent Δ delay intervals of jumper honest blocks and green arrows represent Δ delay intervals of any other type of block. If we only consider arrivals happening in red arrows to be rigged by the adversary, we get the adversarial chain and honest chain shown on top (block 2 gets rigged). If we further consider arrivals in green regions to be rigged as well (like it was done in [5]), then, we get the two chains on the bottom (block 3 and 8 get rigged in addition to block 2). Clearly, adversary wins in the second scenario but not in the first, hence the potential improvement.

We will use this idea in the confirmation interval, however, we will not calculate the lead with this method as calculations become intractable. Instead, we will consider two arrivals at a time during pre-mining which brings improvement to the lead considered in [5]. In the rigged model of [5], an arrival is i.i.d. Bernoulli, and coin toss lands honest w.p. $\alpha e^{-\lambda\Delta}$. Let (H, A) denote the growth of honest and adversarial chains after two arrivals. If we consider two consecutive i.i.d. coin tosses separately and sum their results, then, 1) $\bar{E}'_1$ denoting $(2, 0)$ happens w.p. $(\alpha e^{-\lambda\Delta})^2$; 2) $\bar{E}'_2$ denoting $(1, 1)$ happens w.p. $2(\alpha e^{-\lambda\Delta})(1-\alpha e^{-\lambda\Delta})$; and 3) $\bar{E}'_3$ denoting $(0, 2)$ happens w.p. $(1-\alpha e^{-\lambda\Delta})^2$. In our analysis $\bar{E}'_1$ stays the same but we decrease the probability of $\bar{E}'_3$ by considering two consecutive coin tosses together instead of considering them separately.

Assume that we start from time zero and consider groups of two arrivals in the rigged model. Note that the genesis block is the zeroth jumper that arrives at time zero. For $\bar{E}_2$ to happen, i.e., $(H, A) = (1, 1)$, there are 3 cases: 1) The first arrival, arriving at $t_1 > \Delta$, is honest and the second arrival, arriving at t_2 is not an honest block or $t_2 - t_1 < \Delta$ (gets rigged if honest). This case is treated in the same way as it is done in [5] and has probability $\alpha e^{-\lambda\Delta}(1 - \alpha e^{-\lambda\Delta})$. 2) The first arrival, arriving at $t_1 > \Delta$, is adversary and the second arrival is honest. This case, has probability $\alpha\beta e^{-\lambda\Delta}$. 3) The first arrival arrives at $t_1 < \Delta$ (gets rigged if honest), the second arrival is honest and $t_2 > \Delta$. This case has probability $\alpha\lambda\Delta e^{-\lambda\Delta}$. We note that, in all these cases, the honest block has to be on different height than all previous (non-rigged) honest blocks hence, satisfying AHBODH. Moreover, this improves the results of [5] in certain scenarios. For example, if the first arrival with $t_1 > \Delta$ is adversary and the second arrival is honest with $t_2 - t_1 < \Delta$, then, [5] converts the second arrival to adversary, we do not.

After the second arrival at t_2, by the memorylessness property of exponential arrivals, same arguments will hold for the next group of two arrivals and so forth. Hence, each group is identically distributed and independent from each other. Note that, $\mathbb{P}(\bar{E}_2) = \alpha e^{-\lambda\Delta}(1 + \lambda\Delta + \beta - \alpha e^{-\lambda\Delta}) \geq \mathbb{P}(\bar{E}'_2)$, hence the improvement. Putting these observations into a random walk starting from zero and moving with (A, H), we obtain the following Markov chain,

$$\bar{P}' = \begin{bmatrix} 1-\rho-\bar{\beta}^2 & \rho & \bar{\beta}^2 & 0 & 0 & 0 & 0 & \dots \\ \bar{\alpha}^2 & \rho' & 0 & \bar{\beta}^2 & 0 & 0 & 0 & \dots \\ \bar{\alpha}^2 & 0 & \rho' & 0 & \bar{\beta}^2 & 0 & 0 & \dots \\ 0 & \bar{\alpha}^2 & 0 & \rho' & 0 & \bar{\beta}^2 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{7}$$

where $\mathbb{P}(\bar{E}_1) = \bar{\alpha}^2$, $\mathbb{P}(\bar{E}_2) = \rho'$ and $\mathbb{P}(\bar{E}_3) = \bar{\beta}^2$. Note that similar to birth-death process, we cannot move left while at state zero, and $\rho = \alpha\beta e^{-\lambda\Delta}$ which is a subevent of case 1 in $\bar{E}_2$. The resulting steady state distribution is $\bar{\pi}_0 = \frac{\bar{\alpha}^2 - \bar{\beta}^2}{\bar{\alpha}^2 + \rho}$, and

$$\bar{\pi}_{i+2} = \bar{\pi}_i \left(\frac{\bar{\beta}}{\bar{\alpha}}\right)^2 \tag{8}$$

which is the distribution of the lead, $\bar{L}$, whose PMF is P'_1.

B. Confirmation Interval

Similar to the case of the lower bound we find the number of blocks mined before each honest jumper is published. Keep in mind that, in addition to delaying the publication of jumpers by Δ, all honest blocks mined during this delay are rigged. Hence, all blocks but jumpers are adversarial in this model.

Lemma 3 *The number of blocks mined during the confirmation interval (except jumpers) under the rigged model $\bar{S}$ has the following distribution*

$$P_{\bar{S}}(s) = \bar{\alpha}_0^{k+1} \sum_{n=0}^{s} \binom{k+n}{n} \frac{(\lambda\Delta(k+1))^{s-n}}{(s-n)!} \beta^n \tag{9}$$

We denote the PMF of $\bar{S}$ with P'_2.

C. Post-Confirmation Race

Clearly, if $\bar{L} + \bar{S} > k$, then, the adversary wins the race, otherwise, the deficit of the adversary is $\bar{D} = k + 1 - \bar{L} - \bar{S}$, which it has to make up during the post-confirmation race. Here, we go back to the simplified (A, H) random walk model

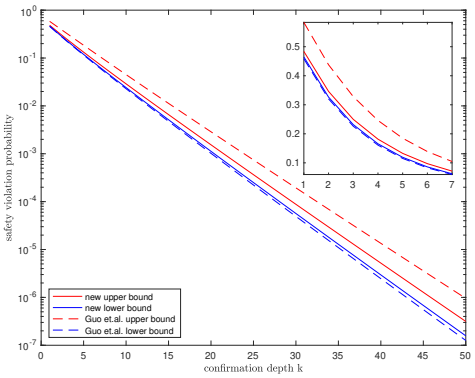

Fig. 3. Bitcoin safety violation with $\alpha = 0.75$.

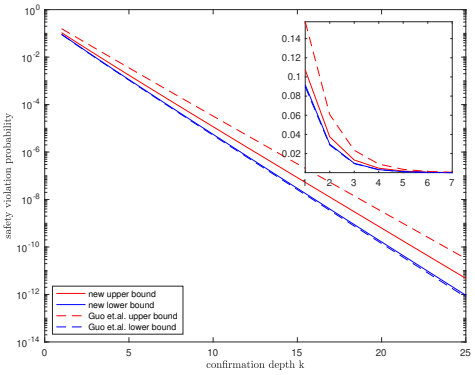

Fig. 4. Bitcoin safety violation with $\alpha = 0.90$.

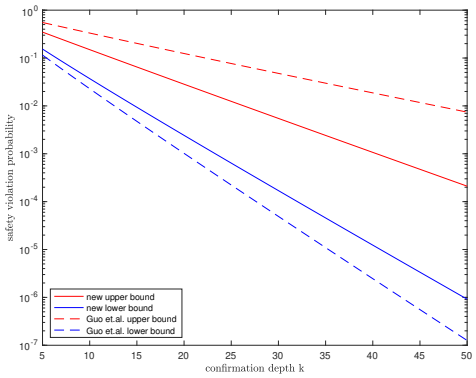

Fig. 5. Ethereum safety violation with $\alpha = 0.75$.

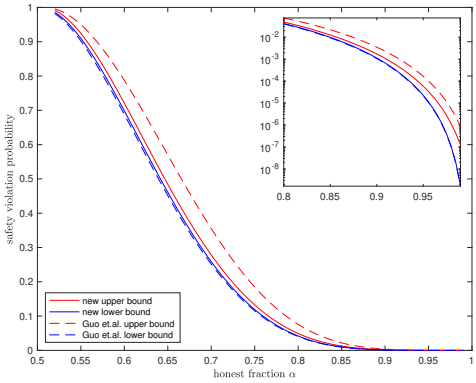

Fig. 6. Bitcoin safety violation vs α under 6-block confirmation rule.

and consider two arrivals at a time as we did for the lead. It is straightforward to show that a random walk which starts at the origin and moves two steps at a time, left or right with $\mathbb{P}(\bar{E}_1) = \bar{\alpha}^2$ and $\mathbb{P}(\bar{E}_3) = \bar{\beta}^2$, respectively, can reach the point $2m$ with probability

$$\mathbb{P}(\bar{M} \geq 2m) = \left(\frac{\bar{\beta}}{\bar{\alpha}}\right)^{2m} \tag{10}$$

Thus, if $\bar{D}$ is even, then we have $\mathbb{P}(\bar{M} \geq \bar{D}) = \mathbb{P}(\bar{M}' \geq \bar{D})$. If odd, however, there can be cases of unobservable adversarial win at the end of two tosses. To avoid this complication, we simply consider a single toss (using the rigged model of [5]) initially to make sure deficit becomes even before using (10): $\mathbb{P}(\bar{M}' \geq \bar{D}) = \bar{\alpha}_0 \mathbb{P}(\bar{M} \geq \bar{D}+1) + (1-\bar{\alpha}_0)\mathbb{P}(\bar{M} \geq \bar{D}-1)$. We denote PMF of $\bar{M}'$ with P'_3.

Theorem 2 *Given mining rate λ, honest fraction α, delay bound Δ and confirmation depth k, a confirmed transaction cannot be discarded w.p. greater than:*

$$\bar{F}'_1(k) + \sum_{i=0}^{k} P'_1(i)\bar{F}'_2(k-i) + \sum_{i+j\leq k} P'_1(i)P'_2(j)\bar{F}'_3(k-i-j) \tag{11}$$

V. Numerical Results

We present our results for Bitcoin in Fig. 3. We choose $\lambda = 1/600$ and $\Delta = 10$ seconds [8] for $\alpha = 0.75$. We compare the tightest result of [5] (Theorem 3) with our tightest result. Note that, lower and upper bounds in our results have the same order of magnitude. As k grows, i.e., the confirmation time becomes longer, and non-jumper honest arrivals grow, which in turn get rigged and increase the adversarial win probability. Hence, the difference in the slopes of upper and lower bounds. We further note that, the improvements we introduce for pre-mining and post-confirmation regions mostly shift the curve, whereas the improvement we introduce in the confirmation interval changes the slopes. These results narrow down the safety violation probability of Bitcoin under 6-block confirmation rule with $\alpha = 0.75$ to $[0.12, 0.13]$, and with $\alpha = 0.90$ to $[0.00112, 0.00173]$.

We present our results for Ethereum in Fig. 5. We choose $\lambda = 1/13$ and $\Delta = 2$ seconds and $\alpha = 0.75$. Note how increasing $\lambda\Delta$ affects the improvement. As also observed in [5], the safety violation probability is mainly determined by the confirmation interval, hence, modeling the honest chain with jumpers which heavily depends on the value of $\lambda\Delta$, we are able improve the bounds by orders of magnitude.

We present Bitcoin's security bounds for varying honest fraction $\alpha \in [0.52, 0.99]$ under 6-block confirmation rule in Fig. 6. Taking all numerical results into account, we see the following trends: As α grows, the difference between the safety violation probability upper and lower bounds grows in orders of magnitude. We were able to decrease this effect by our improvements on the the upper bound, whereas for small Δ, our improvements on the lower bound are modest. However, as our improvements on the lower bound focus on Δ delay strategy, as $\lambda\Delta$ grows, e.g., Ethereum, then the improvements on the lower bound become more significant.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." https://bitcoin.org/bitcoin.pdf, March 2008.

[2] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in *EUROCRYPT 2015*, April 2015.

[3] R. Pass and E. Shi, "Rethinking large-scale consensus," in *IEEE CSF 2017*, August 2017.

[4] J. Li, D. Guo, and L. Ren, "Close latency-security trade-off for the nakamoto consensus," in *ACM AFT 2021*, November 2021.

[5] D. Guo and L. Ren, "Bitcoin's latency–security analysis made simple," 2022.

[6] P. Gazi, L. Ren, and A. Russell, "Practical settlement bounds for proof-of-work blockchains," in *ACM CCS 2022*, November 2022.

[7] A. Dembo, S. Kannan, E. Tas, D. Tse, P. Viswanath, X. Wang, and O. Zeitouni, "Everything is a race and nakamoto always wins," in *ACM CCS 2020*, November 2020.

[8] "DSN Bitcoin Monitoring." https://dsn.tm.kit.edu/bitcoin/. Accessed: 2021-07-30.