Poster: Timestamp Verifiability in Proof-of-Work

Tzuo Hann Law*
tzuohann@gmail.com
Unaffiliated
Boston, MA, USA

Selman Erol
erol@cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Lewis Tseng*†
lewistseng@acm.org
Clark University
Worcester, MA, USA

ABSTRACT

Various blockchain systems have been designed for *dynamic networked systems*. Due to the nature of the systems, the notion of "time" in such systems is somewhat subjective; hence, it is important to understand how the notion of time may impact these systems. This work focuses on an adversary who attacks a Proof-of-Work (POW) blockchain by selfishly constructing an alternative longest chain. We characterize *optimal* strategies employed by the adversary when a difficulty adjustment rule alà Bitcoin applies.

CCS CONCEPTS

• **Theory of computation → Distributed algorithms**.

KEYWORDS

Blockchain, longest chain attack, difficulty adjustment

ACM Reference Format:
Tzuo Hann Law, Selman Erol, and Lewis Tseng*. 2023. Poster: Timestamp Verifiability in Proof-of-Work. In *The Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '23), October 23–26, 2023, Washington, DC, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3565287.3617934

1 INTRODUCTION

Permissionless Proof-of-Work (POW) systems feature a network of peer-to-peer nodes that add blocks containing certain information. Since every node prefers blocks with information specific to their own benefit, there will be no consensus unless a mechanism is used to determine which blocks to be added. Nakamoto [5] consensus is one such mechanism. Its rule for determining which block to add is quite simply a proof that work (for generating a block) was done. Other nodes who receive this information then check if the newly received information contains the more *cumulative* and *valid* work than their own leading block as measured by the same metric. If it is the case, the receiving node would accept the newly minted block and build on top of it. This process is referred to as *mining* and amounts to an arms race. Miners who control more computation power and have access to cheap energy are able to do more work more quickly. As a result, these miners add more blocks in their own favor.

Difficulty Adjustment. Without additional safeguards, such a POW design implies that an increase in the mining capacity would result in a higher rate of token generation. To stabilize it, blockchains typically specify a *difficulty adjustment* protocol. The difficulty is adjusted so that the token generation rate is steered towards some target rate as defined in the protocol. There are many different difficulty adjustment rules being used with this same overarching mandate. In this paper, we consider one that is modeled after the protocol used in Bitcoin [5].

Ideally, a difficulty adjustment rule would adjust the difficulty level according to the mining capacity of the network since that is the primary determinant of the block-finding rate. However, the mining capacity of the network is unobservable and particularly so in permissionless systems. Instead, the difficulty adjustment algorithms utilize the time taken for successive blocks to be created as an estimate of mining capacity and this is in turn "**proxied**" by the timestamps reported in each block.

We emphasize the word "proxied" because the real wall-clock time of an action is itself unobservable. Miners can report any timestamp they please so long as the reported timestamp conforms to some protocol which in turn ensures its acceptance by other nodes. In addition, nodes can successfully mine a block and not report their success until a later time (e.g., selfish mining). Since there are various protocols for accepting/rejecting timestamps, there is substantial variation in the flexibility nodes have for timestamp reporting across different POW blockchains.

Timestamp Verifiability. We investigate how this timestamp flexibility in relation to the difficulty adjustment rule influences the "*optimal strategy*" that an adversary employs when mounting a longest-chain attack. We say that timestamps are *verifiable* (or with very small flexibility), if the timing of the adversary's actions are observable by the honest miners. By this, we do not mean that honest miners can see everything the adversary does. We have in mind a situation where it is essentially impossible to falsify time due to the presence of a well-designed accept/reject protocol. It is also possible by adopting some hardware technology like Intel SGX which provides verifiable timestamps. When timestamp is *unverifiable*, if the adversary can choose an arbitrary timestamp without being caught by honest miners.

Main Result. We characterize the optimal strategies in a POW blockchain. Our main finding is that difficulty adjustment rules offer substantial protection against longest-chain attacks provided timestamps are accurate relative to the frequency of the difficulty adjustment. Our result indicates that an adversary who faces a

*Work partially done when authors were with Boston College.
†This material is based upon work partially supported by the National Science Foundation under Grant CNS-2238020.

MobiHoc '23, October 23–26, 2023, Washington, DC, USA

ACM ISBN 978-1-4503-9926-5/23/10.
https://doi.org/10.1145/3565287.3617934

Table 1: Time taken to mount an attack of N blocks and the corresponding maximum lead A the adversary can overcome for $M_a = 3$ and $M_a = 99$.

N	$M_a = 3$ (75pct of total capacity)				$M_a = 99$ (99pct of total capacity)			
	Verifiable Time		Unverifiable Time		Verifiable Time		Unverifiable Time	
	$T^*(N)$	A_{max}	$T^*(N)$	A_{max}	$T^*(N)$	A_{max}	$T^*(N)$	A_{max}
3	2.08	0.92	0.96	2.04	0.65	2.35	0.03	2.97
5	4.01	0.99	1.43	3.57	1.99	3.01	0.04	4.96
10	8.96	1.04	2.21	7.79	6.32	3.68	0.07	9.93
20	18.93	1.07	3.04	16.96	15.89	4.11	0.09	19.91
100	98.91	1.09	4.37	95.63	95.51	4.49	0.13	99.87

difficulty adjustment rule will find a longest-chain attack very challenging when timestamps are verifiable. POW blockchains with frequent difficulty adjustments relative to time reporting flexibility will be substantially more vulnerable to longest-chain attacks.

2 ANALYSIS AND DISCUSSION

To make key ideas clearer, we work with deterministic mining [2, 4]. The strategies that apply in a deterministic setting continue to apply in the probabilistic setting (in expectation) under the right assumptions about the adversary's preferences. None of our key points and findings depends on the deterministic mining setting. Our technical report [3] presents the theoretical analysis that proves the optimal adversary strategy that adopts the longest-chain attack.

2.1 Numerical Analysis

We present the numerical analysis of the adversary's optimal strategy between the two regimes here. We consider two adversaries, one with 75pct of the mining capacity and the other with 99pct of the mining capacity. While it may be comical to think about a conventional miner with such capabilities, this risk is a lot more tangible when we allow for quantum computing possibilities [1]. Such a risk may also be a lot more conceivable for POW blockchains where the overall hash rate is much lower than of Bitcoin's.

Table 1 reports what values of initial deficits (in terms of number of blocks) an adversary can overcome if it mounts an attack where it selfishly mines N blocks. We also report the time taken $T^*(N)$ to mine the alternative chain. As the adversary mines N blocks in such a fashion, the honest miners would have extended chain the canonical chain by $T^*(N)$ blocks.[1] Therefore, the largest A_{max} the adversary could have overcome would be given by $N - T^*(N)$. (The larger the A_{max}, the easier to launch a longest-chain attack.)

2.2 Discussion: Practical Implications

The key metric that matters in practical systems is how flexible timestamps can be relative to epoch length. For instance, Bitcoin's timestamps can be any time in a 3-hour window and be accepted. Its epoch length is 2016 blocks which will take about 2 weeks to mine. As a ratio, the relative flexibility approaches zero. This means that Bitcoin is probably very close to our setting with verifiable time and it suggests that an adversary controlling 75pct of the mining capacity will be able to start an entire epoch behind the canonical chain and overtake it after 4 epochs have elapsed on the canonical chain. Monero and Bitcoin Cash, recalculate the block adjustment every block using the previous day's worth of blocks. The degree of time reporting flexibility is similar to Bitcoin's. As a ratio, these two blockchains would be further away from perfect time verifiability compared to Bitcoin.

2.2.1 Takeaway 1: Verifiable timestamps diminish the efficacy of M_a. As shown in Table 1, the time it takes to construct more blocks increases with the number of blocks constructed. Since the canonical chain is growing at the same time, the adversary will need to control huge amounts of mining power in order to overcome small leads. In other words, an adversary will find it very difficult to start an alternative chain that is more than a few blocks behind the leading block and overtake it. The crucial insight here is that the *best* strategy the adversary can employ is to scale up its mining efforts following a power law and power law progressions ramp up very quickly. It also sets limits on how far ahead the target chain can be for an adversary with a fixed capacity.

2.2.2 Takeaway 2: Unverifiable timestamps lead to approximately linear attack duration. With unverifiable timestamps, the time taken to construct N blocks is approximately linear in N, which implies that an adversary possessing mining power greater than 51pct can and will catch up any distance provided it continues selfishly mining for long enough.

2.3 Future Work

We solved for the optimal attack an adversary can mount against naive honest miners assuming a limited action set for the adversary. Quantifying other potential actions depending on time verifiability (e.g., chain hopping) is an immediate extension of this paper.

REFERENCES

[1] D. A. Bard et al. Quantum advantage on proof of work. *Array*, 2021.
[2] B. Johnson et al. Game-theoretic analysis of ddos attacks against bitcoin mining pools. In *Financial Cryptography and Data Security*, 2014.
[3] Law et al. Longest-chain Attacks: Difficulty Adjustment and Timestamp Verifiability. https://arxiv.org/abs/2308.15312, arXiv, 2023.
[4] D. Meshkov et al. Short paper: Revisiting difficulty control for blockchain systems. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology - ESORICS 2017.*
[5] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Cryptography Mailing list*, 2009.

[1] In our model with deterministic mining [2, 4], the block finding rate is 1 block per-unit time and each epoch contains one block. We also assume honest miners control 1 mining capacity and do not behave strategically.