# Sustaining the Trust in Bitcoin by Incentivizing Solo Mining

Önder Gürcan and Mustafa Safa Özdayi

Laboratory for Trustworthy, Smart, Self-Organizing Information Systems,
Software and Systems Engineering Department,
CEA LIST, 91191 Gif-sur-Yvette, France
{onder.gurcan,mustafa.ozdayi}@cea.fr

**Abstract.** Bitcoin is an open peer-to-peer system where participants, that can join or leave at any time, collectively build a *trusted ledger* called blockchain for performing transactions in-between without neither needing to *trust each other* nor having a *trusted third party*. *User participants* create and broadcast transactions across the network for being confirmed. *Miner participants* try to confirm them as a block by solving a computational puzzle (mining). The successful miner broadcasts its block to the network to be chained to the blockchain and he is awarded for his success. In addition, all participants validate all the data (transactions and blocks) broadcast across the network. However, decreased participation and/or concentration of participants (e.g., mining pools), decrease the trust in the network. In this paper, we study the impact of rewarding mechanisms on incentivizing solo mining to sustain trust over time. We provide two metrics that allow analyzing such incentive: *fairness* and *return rate*. We then analyze the role of existing rewarding mechanisms (introduced in Bitcoin and in Fruitchain) on such incentive using simulations. Our results show that, while Bitcoin and Fruitchain are equivalent in terms of fairness, the return rate of Fruitchain is better than Bitcoin. Consequently, it can be said that Fruitchain is more incentivizing for solo mining, thus more suitable for sustaining the trust.

**Keywords:** multi-agent, trust, incentives, fairness, return rate, reward

## 1 Introduction

Since its genesis in late 2008 [**?**], Bitcoin had a rapid growth in terms of participation, number of transactions and market value. This success is mostly due to innovative use of existing technologies for building a *trusted ledger* called blockchain. In this system, user participants sign transactions with their private keys and broadcast them on an open peer-to-peer network. These transactions are then confirmed (i.e., totally ordered and cryptographically linked to the blockchain) by miner participants and broadcast across the network. Moreover, both transactions and blocks that are broadcast are validated (applying the predefined rules) and diffused by each peer (i.e., participant) in the network, and

invalid ones are discarded. This way, the participants collectively build a *trusted ledger* of transactions where they are *confident* about the balances of each other.

### 1.1 Motivation

The trust in Bitcoin is proportional to the participation of miners and users, i.e. more *confirmers* and more *validators*. Such participation is not trivial and is driven by the *incentives* that the system provides to its participants.

Users are willing to issue transactions in-between without neither needing to *trust each other* nor having a *trusted third party*. Consequently, they expect the system to provide a *trusted* transactional service at a reasonable cost, speed and acceptable quality. Hence, a trend on a growing number of unconfirmed transactions may create a service degradation in Bitcoin, and may result decreased participation of users [**?**]. And if no user stays in the system, miners will have no transactions to confirm[1] and thus the system will be confined to end[2].

Miners, on the other hand, are willing to make profit from (or at least compensate) their computational efforts for confirming transactions (mining). Since the objective is to maintain the security of the blockchain, mining is designed to be hard, which makes it very costly as well. Consequently, miners are incentivized by a reward for each successful mining. However, the expected time and variance of receiving payouts can be quite large for miners. Such a situation disincentivizes the miners with relatively lower computation powers. Hence, such miners either leave the system or combine their resources by creating mining pools. Mining pools may lead to the centralization of the computation power in the network, which may make the entire network to be controlled by a small number of mining pools. Such a situation will decrease the *trust* to the underlying blockchain, and may result, decreased user participation that reduces the *trust* even more.

Based on this observation, our motivation is to focus on incentives for solo mining that can be provided by Bitcoin, to sustain the *trust*.

### 1.2 Related Work

To analyze mining in Bitcoin-like blockchains [**?**], several formal studies have been conducted so far [**?**,**?**,**?**,**?**,**?**]. Garay et al. [**?**] showed that, assuming that all miners follow the protocol, the number of blocks created by miners is proportional to their fraction of computation powers. Eyal et al. [**?**] were the first to formally show that Bitcoin protocol is not incentive compatible by presenting a deviant mining strategy called *selfish mining*. Their main result states that a miner whose hash rate is %33 of network can generate %38.4 of the blocks by employing this strategy. This consequently shows that Bitcoin protocol is not

[1] Technically the miners can create empty blocks and get block rewards. But this is not the purpose of blockchain systems.

[2] Although it is open system and we can expect that participants may come back in the future, once they lose their trust it is harder to expect this.

*fair*, i.e., fraction of blocks contributed by a miner to the blockchain can deviate significantly from his hash rate fraction. Work of Sapirshtein et al. [**?**] further analyze and optimize this strategy. Carlsten et al. [**?**] shows that selfish mining becomes even more profitable in a setting where there is no fixed block reward, i.e. miners earn their revenue solely through transaction fees. In addition to that, they show it is likely for Bitcoin to become unstable due to miners bribing each other to fork the chain.

To tackle such problems, a solution proposed by Fruitchain [**?**] is to incentivize solo mining by using a novel rewarding scheme. Fruitchain is a blockchain protocol first introduced by Pass and Shi in [**?**]. It seeks to remedy some issues that Bitcoin currently faces such as centralization due to mining pools and unfairness due to selfish mining. Main novelty of the protocol is to introduce a new structure named *fruit* to the Bitcoin protocol whose mining difficulty is lower than blocks. Basically, a *fruit* mined by some miner can be included in the block of an another miner. Due to this, miners whose hash rate is not sufficient to mine blocks can prove their participation and consequently get rewarded by mining *fruits*. In a sense, *fruits* play the role of share of mining pools in a decentralized manner. In our work, we particularly focus on the rewarding scheme of Fruitchain which is introduced in a follow-up work of authors [**?**].

### 1.3 Objectives

Based on the motivation and related work, in this study, our objective is to study the impact of rewarding mechanisms on incentivizing solo mining to sustain trust over time. To this end we provide two metrics, *fairness* and *return rate*, that allow us to observe the incentivization of solo mining under a dedicated system model. We then analyze the role of existing rewarding mechanisms (introduced in Bitcoin and in Fruitchain) on such incentive using simulations.

### 1.4 Contributions

The contribution of this paper is as follows:

- A formal definition of two metrics, namely *fairness* and *return rate*, for analyzing the impact of rewarding mechanisms on incentivizing solo mining thus sustaining the trust;
- A simulation analysis of the existing Bitcoin rewarding mechanisms, namely Bitcoin and Fruitchain, in terms of sustainability of the trust.

### 1.5 Organization

The paper is organized as follows. Section **??** provides the system model of the Bitcoin protocol and the existing rewarding mechanisms. Section focus on the metrics to analyze incentives for solo mining and provides their definitions. Section analyzes the role the rewarding mechanisms introduced in Section **??** on incentives introduced in Section **??** using simulations. Section provides a concludes the paper.

## 2 The Bitcoin Protocol

In this section, we provide a high-level Bitcoin protocol description based on the high-level description given in [**?**]. Since rewarding mechanisms are directly related to the blocks, they will be represented inside the dedicated subsections about different block types.

### 2.1 Network

We model the network as a dynamic directed graph $G = (N, E)$ where $N$ denotes the dynamic node (vertex) set, $E$ denotes dynamic directed link (edge) set. A node $n$ can enter and leave $G$ by using its $join(G)$ and $leave(G)$ actions respectively.

Each node $n$ has a memory pool $\Theta_n$ in which it keeps unconfirmed transactions that have input transactions, an orphan pool $\bar{\Theta}_n$ in which they keep unconfirmed transactions that have one or more missing input transactions (orphan transactions) and a blockchain ledger $B_n$ in which they keep confirmed transactions where $\Theta_n \cap \bar{\Theta}_n = \emptyset$, $\Theta_n \cap B_n = \emptyset$ and $\bar{\Theta}_n \cap B_n = \emptyset$ always hold. Nodes can play two distinct but complementary roles in the network: *user* and *miner*.

**User Node** A node $n$ is said to be a user node if it creates transactions to spend its coins. We model a transaction as $tx = \langle ¢, f_{tx}, m \rangle$ where $¢$ is the amount of coins ($¢ > 0$) paid to $m \in N$ and $f_{tx}$ is the fee to be paid for $tx$.

**Miner Node** A node $n$ can turn to be a miner node if it chooses to create blocks for confirming the transactions (mining) in its memory pool $\Theta_m$. The set of miner nodes is then denoted by $M$ where $M \subseteq N$. In order to be able to mine, $n \in M$ has to solve a cryptographic puzzle (i.e. Proof of Work) using its hashing power[3] $q_n$ where $q_n > 0$. The cryptographic puzzle is tried be solved by using the cryptographic hash function $\mathcal{H}^D(\cdot)$ where $D$ is the difficulty. The more difficult the cryptographic puzzle is, the more hashing power is need to be able to solve as fast as possible. The successful miners are awarded by a block reward (see Section **??** for details) via a coinbase transaction. The coinbase transaction $tx_c$ is special transaction that collects and spends any transaction fees paid by transactions included in a block. It is the first transaction in a block and can only be created by a miner.

### 2.2 Blockchain

We model the blockchain ledger of a node $n$ as a dynamic append-only tree $B_n = \{b_0 \xleftarrow{r_0} b_1 \xleftarrow{r_1} ... \xleftarrow{r_{h-1}} b_h\}$ where each block $b_i$ $(0 < i \leq h)$ contains a

[3] Hashing power is proportional to computation power and nodes may change this power by time.

cryptographic reference $r_{i-1}$ to its previous block $b_{i-1}$, $h = |B_n|$ is the depth of $B_n$, $b_0$ is the root block which is also called the *genesis block* and $b_h$ is the furthest block from the genesis block which is referred to as the *blockchain head*.

### 2.3 Bitcoin Block

A Bitcoin block contains the cryptographic hash code of the previous block ($\mathcal{H}^D(\mathtt{b}_{i-1})$) and the set of candidate transactions $\theta_m \subseteq \Theta_m$ where $\Theta_m$ is the memory pool of the miner (Figure **??**(a)). One of these transactions is the coinbase transaction $tx_c$ that awards the miner node $m$ the block reward

$$R_i = \mathtt{F} + \Sigma f$$

for its work (where $\mathtt{F}$ is the static block reward, and $\Sigma f$ is the total fees of the transactions included in this block), $\theta_m \subseteq \Theta_m$ is the set of candidate transactions chosen for block $i$.

### 2.4 Fruitchain Block

In Fruitchain [**?**], in addition to Bitcoin, the block contains a set fruits $\mathcal{F}_i$ where each fruit $\mathcal{H}^d(\mathtt{h}_j) \in \mathcal{F}_i$ is a cryptographic hash code of a previous block $j$ a miner selected in a $k$-length window with a difficulty of $d < D$ where $0 \leq i - k \leq j < i$ (Figure **??**(b)). Since $d < D$, it is easier to mine fruits compare to mining blocks and miners are allowed to mine as many fruits as they want. When a fruit is created, it is broadcast to the network. Just as transactions, they are included in blocks by miners. The miners get rewards regarding to block mining and fruit mining.

More in detail, Fruitchain has the parameters $k, c_1, c_2$ and $c_3$ that are called *window length*, *direct reward proportion*, *fruit tax* and *fruit freshness bonus* respectively. The miner of the block $b_i$ with a $c_1$ fraction of the total block reward $R_i = \mathtt{F} + \Sigma f$ (as in Section **??**) where $0 \leq c_1 \leq 1$, i.e. $R_i \cdot c_1$ is the reward of the block miner[4]. The remaining of the block reward which is $R_i \cdot (1 - c_1)$ is distributed the miners of fruits. Fruit tax $c_2$, on the other hand, aims encourage miners to put fruits of others in their blocks. Depending on its value, miners might include or exclude fruits of others. Lastly, fruit freshness bonus $c_3$ is used for encouraging miners to mine fresh fruits: i.e. the closer the distance between the block containing the fruit and the block that fruit refers to, the higher the bonus.

More concretely, for each fruit $\varphi$, that are included in blocks in a $k$-length sliding window ($b_{i-k} \cdots b_{i-1}$), the fruit miners are awarded as

$$\frac{R_i \cdot (1 - c_1)}{\sum_{h=i-k}^{i-1} |\mathcal{F}_h|} \cdot (1 - c_2 + c_3 \cdot (1 - \frac{l_\varphi}{k-1}))$$

[4] Note that, when $c_1 = 1$ the rewarding scheme is the same as Bitcoin.

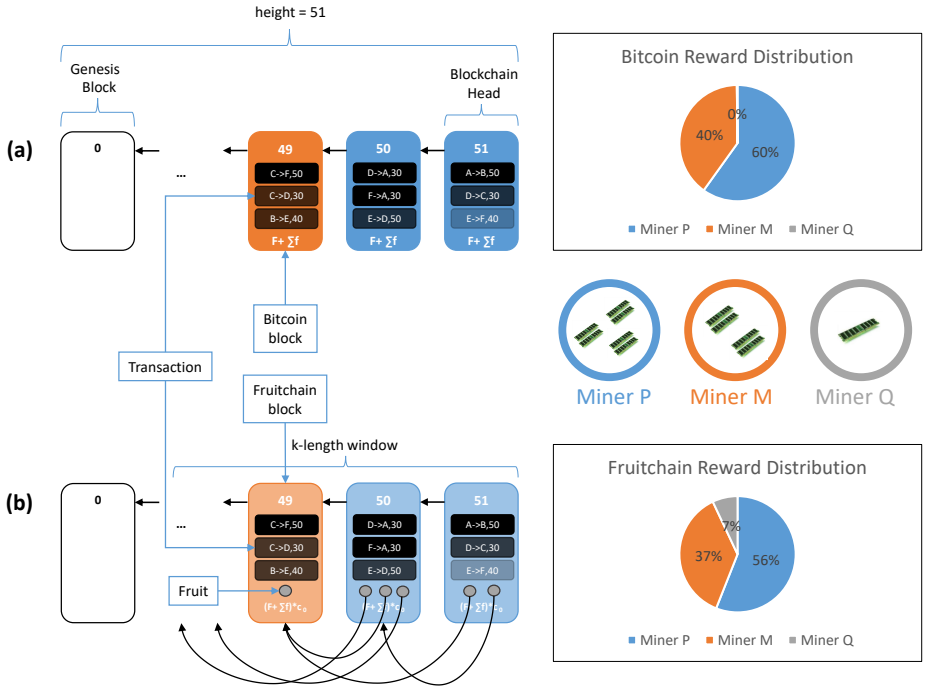



Fig. 1: Illustration of the Bitcoin (a) and Fruitchain (b) rewarding mechanisms where there are three miners P, M and Q with proportional hashing powers 6, 4 and 1 respectively. While the Bitcoin blocks contain only transactions and the rewards are calculated according to a fixed block reward plus the total fees, the Fruitchain blocks contain also fruits and the rewards are calculated by taking into account the fruit miners also. It is claimed that Fruitchain distributes rewards more fairly by up until now there is no quantitative study that shows so.

and the block miners are awarded as

$$\frac{R_i \cdot (1 - c_1)}{\sum_{h=i-k}^{i-1} |\mathcal{F}_h|} \cdot (c_2 - c_3 \cdot (1 - \frac{l_\varphi}{k-1})$$

where $|\mathcal{F}_h|$ is the number of fruits in $b_h$, $c_2$ is the *fruit tax*[5], $c_3$ is the *freshness bonus*[6] and $l_\varphi$ $(0 \leq l_\varphi \leq k-1)$ is the number of blocks between the one that contains fruit $\varphi$ and the one that $\varphi$ hangs from [**?**].

[5] An incentive for miners to include fruits of other miners. The fruit tax can be specified individually for each fruit $\varphi$ by its miner, i.e. $\hat{c_2}(\varphi)$, $c_2$ is just the default value.
[6] Freshness bonus aims to encourage miners to release their fruits early

## 3 Incentivization of Solo Mining

In this section, we provide a formal definition of two metrics, namely *fairness* and *return rate*, for analyzing the impact of rewarding mechanisms on incentivizing solo mining thus sustaining the trust.

### 3.1 Fairness

Fairness is simply defined as proportional distribution of rewards with respect to the hash-rate of miners. It is an important property that a rewarding mechanism should satisfy in order to promote participation in the mining process. Formally, we define fairness as follows.

**Definition 1.** *The fairness of a rewarding mechanism. Let $m \in N$ be a miner with a hash-rate $q_m$ and let $q_N$ be the total hash-rate of the network. Let $\Sigma R_i^m$ denote the total reward $m$ gains and let $\Sigma R_i^N$ denote the total reward gained in the network until the block $i$. We call a reward scheme fair if for every miner $m$, we have $\frac{E[\Sigma R_i^m]}{E[\Sigma R_i^N]} = \frac{q_m}{q_N}$ where $E[\cdot]$ is the expected value function.*

The more a rewarding mechanism is fair, the more it can incentivize solo mining. This is because even if the miner has a low computation power, it knows that it can at least compensate its cost in a fair system.

### 3.2 Return rate

Miners should make regular payments to keep their businesses running (e.g., electric bills). This means that receiving their rewards regularly incentivizes them. Due to this, *return rate* is an important property that a reward scheme should satisfy. Formally, we define the *return rate* of a miner as follows.

**Definition 2.** *The return rate of a miner. Let $m \in N$ be a miner, $\Sigma R_i^m$ be the total reward of $m$ until block $i$ and $T_m$ denote its average reward gap[7]. Then, we define return rate of $m$ for as $\frac{E[\Sigma R_i^m]}{E[T_m]}$ where $E[\cdot]$ is the expected value function.*

The higher a rewarding mechanism's return rate, the more it can incentivize solo mining. This is because, especially for the ones that have low computation power, the miners will get the return of their investment more quickly.

## 4 Simulations and Results

In this section, we analyze the role the rewarding mechanisms introduced in Section **??** on incentives introduced in Section **??** using simulations.

[7] Number of blocks between two instants in which $m$ gains its rewards. For example, if $m$ gains a reward for block 2 and 4, its reward gap for this interval is given by 4-2+1 = 3 (end points are inclusive). If $m$ does not gain any reward during this period, its reward gap is equal to the height of the blockchain.

### 4.1 Simulations

We implemented a simulator based on the system model given in Section **??** using Mesa agent-based modeling framework [**?**]. We considered a round-based synchronous no-delay reliable[8] setting, as in Garay et al. [**?**]. At each round $i$, the simulator is selecting one miner for mining the block $r$ and several miners for mining fruits with the probabilities $p$ and $p_f$ respectively ($1 > p_f > p > 0$). The parameters of the simulator are as follows:

- $p$: Probability of mining a block at a round.
- $p_f$: Probability of mining a fruit at a round.
- $r$: Total number of rounds.
- $n$: Total number of miners.
- $h$: List of hashrate fractions such that $h_i$ is the hashrate fraction of miner $i$.

During the simulations, we simultaneously record how much reward each miner earns under Bitcoin and Fruitchain.

For Fruitchain, we use the default parameters, i.e. $c_1 = 0.01$, $c_2 = 0.1$, $c_3 = 0.01$, $k = 16$, proposed by [**?**].

Further, we assume every block has accumulated an amount of 12.5 BTC in transactions fees and ignore the fixed reward as Fruitchain only takes fees into account.

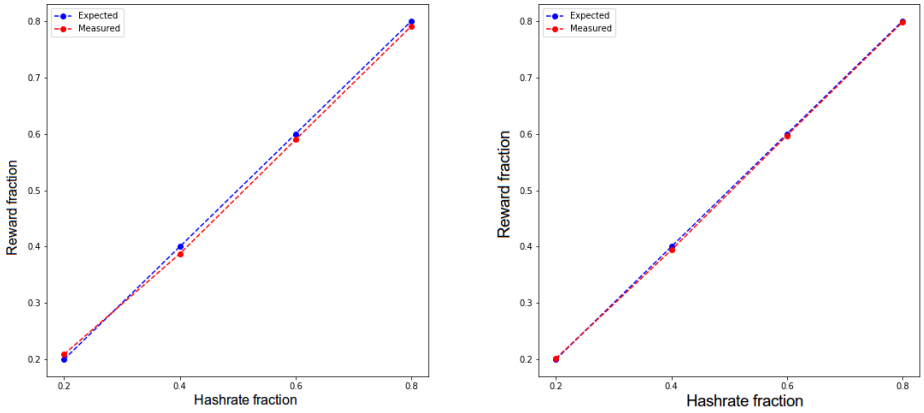
### 4.2 Results

We made several experiments for analyzing the rewarding mechanisms based on *fairness* and *return rate* metrics under different settings.

**Fairness Experiments** We conducted an experiment to compare fairness of Bitcoin and Fruitchain under the following settings: we fix $r = 10^5, p = 0.01, p_f = 1, n = 2$ and vary $h_2$ from 0.2 to 0.8 by incrementing it 0.2 between simulations.

Figure **??** shows the results obtained from this simulation by plotting reward fractions for different hash-rate fractions for miner 2. . The results show that the reward fractions of miners are equal to their hash-rate fractions in expectation for both Bitcoin and Fruitchain.

[8] Reliable means when a message (transaction, fruit and/or block) is sent all agents get it.



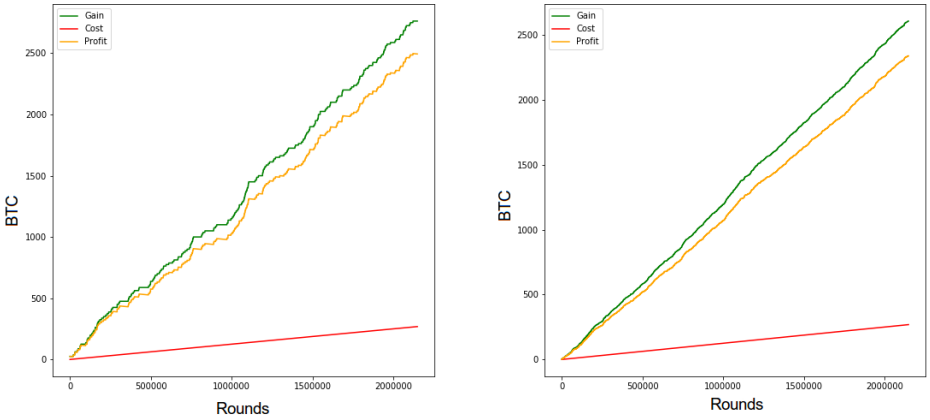

(a) Fairness of Bitcoin. (b) Fairness of Fruitchain.

Fig. 2: Fairness of Bitcoin and Fruitchain for miner 2.

**Return rate Experiments** We now attempt to highlight the difference between Bitcoin and Fruitchain by explicitly showing a miner's *return rate* under them. Basically, we investigate for what hashrate fractions *return rate* of Fruitchain becomes visible. To this end, we plot *return* of a miner against his hashrate fraction for both Bitcoin and Fruitchain.

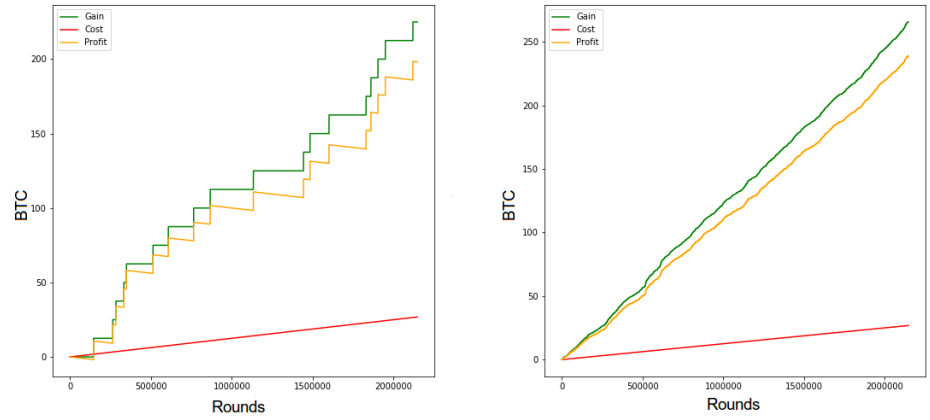
The plots in this section are obtained under the following settings: we fix $p = 0.001, p_f = 1, n = 2$ and adjust the running time of each simulation such that it corresponds to roughly 15 days, i.e. $r = \frac{1}{p} \cdot 144 \cdot 15 = 2.16 \cdot 10^6$ [9]. The cost of mining per round is arbitrarily set as 10% of the expected gain per round. We do 3 simulations for $h_2 = 0.1, 0.01$ and $0.001$ and we do our measurements for miner 2.

[9] Roughly, 144 blocks are created in each day in the Bitcoin network, see `https://en.bitcoin.it/wiki/Confirmation`.



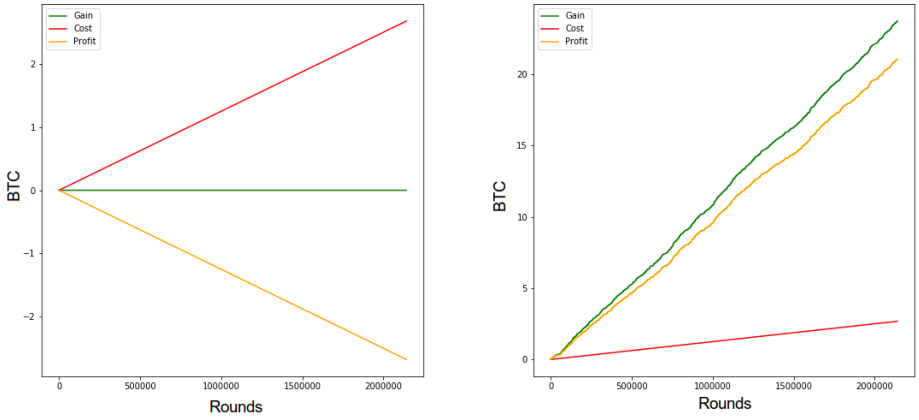

(a) Return rate under Bitcoin. (b) Return rate under Fruitchain.

Fig. 3: Return rate under Bitcoin and Fruitchain with $h_2 = 0.1$.




(a) Return rate under Bitcoin. (b) Return rate under Fruitchain.

Fig. 4: Return rate under Bitcoin and Fruitchain with $h_2 = 0.01$.




(a) Return rate under Bitcoin. (b) Return rate under Fruitchain.

Fig. 5: Return rate under Bitcoin and Fruitchain with $h_2 = 0.001$.

### 4.3 Discussion

We see that Bitcoin and Fruitchain are nearly equivalent in Figure **??**. At the end of the simulation, the *total return* of the miner is roughly 2500 BTC for both under Bitcoin and Fruitchain. In Figure **??**, the miner is better under Fruitchain. Its *total return* is about 225 BTC under Fruitchain and slightly below 200 under Bitcoin. Note that graphs are scaled differently. Finally, the advantage of Fruitchain is obvious in Figure **??**. Under Bitcoin, the *return rate* of miners is negative as it was not able to create any block. However, under Fruitchain, it is positive due to rewards it gained from its fruits.

Our experiments show that *return rate* of a miner is roughly the same for Bitcoin and Fruitchain if it has a relatively large portion of the hashrate ($\approx 0.1$). However, if it happens to have relatively a small portion of the hashrate ($\approx 0.01$), it clearly has more returns under Fruitchain. The main reason is, due to low reward gap, rewards under Fruitchain converge faster to their expected values (recall that the expected reward is the same for Bitcoin and Fruitchain due to our results on fairness).

## 5 Conclusions

Bitcoin is an open and dynamic peer-to-peer system, where participants need to rely on the balances provided by other participants to accomplish their transactions. During this process, participants are exposed to the risk of being exploited by others. Such risks, if not mitigated, can cause serious breakdowns in the operation of Bitcoin and threaten its long-term wellbeing. To protect participants

from the uncertainty in the behavior of their interaction partners, Bitcoin proposes a novel trust management approach[10] where participant collectively build a trusted ledger of transactions. The power of this collective trust management approach is proportional to the participation of miners and users, i.e. more *confirmers* and more *validators*. Such participation is driven by incentives provided by Bitcoin. In this study, we focused on one such incentive that promotes participation of miners: the rewarding mechanism.

Concretely, we tried to capture and highlight the differences between the two existing rewarding mechanisms proposed in Bitcoin [**?**] and Fruitchain [**?**]. To this end, we first introduced a simplified system model that focuses on the rewarding mechanisms of the protocols. Then, we explicitly defined two important properties for miners: *fairness* and *return rate*. Following that, we have analyzed using simulations the performance of Bitcoin and Fruitchain in terms of satisfying these properties. Through our analysis, we have shown that the return rate of Fruitchain is better than Bitcoin and they are equivalent in terms of fairness[11]. Moreover, we showed the return rate of Fruitchain is especially important to miners with small hashrates by comparing their profits under Bitcoin and Fruitchain. Concretely, we showed miners were able to earn profits in a steady manner under Fruitchain. This confirms that computing power is less likely to be centralized under Fruitchain.

In a nutshell, we claim that blockchain systems are complex adaptive systems with intricate network structures exchanging information. Using existing Bitcoin data may provide us with a powerful tool for determining the dynamics of these networks and, consequently, for quantitatively testing our theoretical prediction given in this paper. Thus, as a future work, we plan to gather such data and make more realistic simulations. This will potentially make the simulator a powerful tool for planning and development of blockchain systems.

## Acknowledgements

The authors would like thank Sara Tucci-Piergiovanni (CEA LIST) and Rachid Guerraoui (EPFL) for their contributions.

## References

1. Iddo Bentov, Yuncong Hu, Rafael Pass, Elaine Shi, and Siqiu Yao. Decentralized pooled mining: An implementation of fruitchain. In *Manuscript*.
2. Miles Carlsten, Harry Kalodner, S Matthew Weinberg, and Arvind Narayanan. On the instability of bitcoin without the block reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 154–167. ACM, 2016.

[10] For a survey of trust management approaches see [**?**].
[11] However, regarding fairness, we have not tested our model against every attact yet.

3. Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *International Conference on Financial Cryptography and Data Security*, pages 436–454. Springer, 2014.
4. Juan Garay, Aggelos Kiayias, and Nikos Leonardos. *The Bitcoin Backbone Protocol: Analysis and Applications*, pages 281–310. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
5. Önder Gürcan, Antonella Del Pozzo, and Sara Tucci-Piergiovanni. On the bitcoin limitations to deliver fairness to users. In Hervé Panetto, Christophe Debruyne, Walid Gaaloul, Mike Papazoglou, Adrian Paschke, Claudio Agostino Ardagna, and Robert Meersman, editors, *On the Move to Meaningful Internet Systems. OTM 2017 Conferences*, pages 589–606, Cham, 2017. Springer International Publishing.
6. David Masad and Jacqueline Kazil. Mesa: An agent-based modeling framework. In *Proc. of the 14th Python in Science Conference (SCIPY 2015)*, pages 53–60, 2015.
7. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. `https://bitcoin.org/bitcoin.pdf`.
8. Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the blockchain protocol in asynchronous networks. *IACR Cryptology ePrint Archive*, 2016:454, 2016.
9. Rafael Pass and Elaine Shi. Fruitchains: A fair blockchain. Cryptology ePrint Archive, Report 2016/916, 2016. `http://eprint.iacr.org/2016/916.pdf`.
10. Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. Optimal selfish mining strategies in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 515–532. Springer, 2016.
11. H. Yu, Z. Shen, C. Leung, C. Miao, and V. R. Lesser. A survey of multi-agent trust management systems. *IEEE Access*, 1:35–50, 2013.