The Gap Game

Itay Tsabary
Technion
sitay@campus.technion.ac.il

Ittay Eyal
Technion
ittay@technion.ac.il

ABSTRACT

Blockchain-based cryptocurrencies secure a decentralized consensus protocol by incentives. The protocol participants, called miners, generate (mine) a series of blocks, each containing monetary transactions created by system users. As incentive for participation, miners receive newly minted currency and transaction fees paid by transaction creators. Blockchain bandwidth limits lead users to pay increasing fees in order to prioritize their transactions. However, most prior work focused on models where fees are negligible. In a notable exception, Carlsten et al. [17] postulated that if incentives come only from fees then a mining gap would form — miners would avoid mining when the available fees are insufficient.

In this work, we analyze cryptocurrency security in realistic settings, taking into account all elements of expenses and rewards. To study when gaps form, we analyze the system as a game we call *the gap game*. We analyze the game with a combination of symbolic and numeric analysis tools in a wide range of scenarios.

Our analysis confirms Carlsten et al.'s postulate; indeed, we show that gaps form well before fees are the only incentive, and analyze the implications on security. Perhaps surprisingly, we show that different miners choose different gap sizes to optimize their utility, even when their operating costs are identical. Alarmingly, we see that the system incentivizes large miner coalitions, reducing system decentralization. We describe the required conditions to avoid the incentive misalignment, providing guidelines for future cryptocurrency design.

CCS CONCEPTS

• **Security and privacy → Distributed systems security**; **Security protocols**; **Economics of security and privacy**; • **Computer systems organization → Peer-to-peer architectures**;

KEYWORDS

Blockchains; Cryptocurrency; Mining Gap; Centralization; Game Theory

ACM Reference Format:
Itay Tsabary and Ittay Eyal. 2018. The Gap Game. In *2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18), October 15–19, 2018, Toronto, ON, Canada.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3243734.3243737

CCS '18, October 15–19, 2018, Toronto, ON, Canada

ACM ISBN 978-1-4503-5693-0/18/10…$15.00
https://doi.org/10.1145/3243734.3243737

1 INTRODUCTION

Since their introduction in 2008 [51], blockchain protocols are securing rapidly increasing amounts of money in the form of so-called cryptocurrencies. As of today, the market cap of the first cryptocurrency, Bitcoin, is estimated at $160B [10, 18], and the market cap of all cryptocurrencies, most of which are secured with blockchain protocols, is estimated at $350B [10, 18].

Cryptocurrencies facilitate users transactions of a currency internal to the system. The underlying protocol, the blockchain, is operated by independent principals called *miners*. Miners collect transactions in blocks and append them to the blockchain, forming a globally-agreed order of transactions. Instead of relying on a central control, the most prominent blockchain-based cryptocurrencies [7, 16, 46, 51, 59] rely on utilizing incentives to secure the system. They use *proof of work* (*PoW*) [24, 39, 51], requiring participants to solve moderately-difficult cryptographic puzzles to generate blocks. The idea is that to successfully attack the system, one would need to control resources proportional in amount to those of all participating miners. To motivate miner participation, cryptocurrencies incentivize them with *block rewards* comprising *subsidy*, newly minted currency created at the generation of each new block, and *transaction fees*, paid by the transactions. Preliminaries on cryptocurrencies and blockchain protocols are in Section 2.

In the dominant operational cryptocurrency systems [16, 51], the subsidy is the substantial part of the incentive as of today. And indeed, despite the breadth of research on blockchain security [14, 20, 50, 61], and despite the significance of incentives for blockchain security, most prior work studied the incentives scheme when the reward comes only from subsidy [29, 31, 33, 44, 53, 58, 60]. However, as a cryptocurrency gains traction, the incoming load of transactions increases [12, 28]. Since transaction bandwidth is limited, a fee market forms – users offer higher fees to motivate miners to place their transaction quickly [13, 27]. Moreover, in Bitcoin and several other cryptocurrencies minting rate decays over time. Hence, fees are on the path to become a substantial part of cryptocurrency rewards.

Carlsten et al. [17] postulated that in a certain scenario, a *mining gap* would form. Their model assumes only operational expenses and no subsidy, and that block size is unbounded, so miners place all pending transactions when mining a block. Therefore, once a block is generated, there are no unclaimed transactions and therefore no unclaimed fees, and so no incentive to mine the next block until sufficient fees have accrued, resulting in a gap in mining period. In Section 3 we review previous work.

In this work, we analyze the incentives and equilibrium of blockchain-based PoW cryptocurrency systems, taking into account rewards from both subsidy and fees, and both capital and operational expenses. We present our model in Section 4.

This model gives rise to a game we call *the gap game* (Section 5). It is played among the miners, which compete on finding blocks – the

first to find a block gets rewarded, while all suffer expenses. It is a one-shot game, where the miners decide when to start their mining rigs, and strive to optimize their average revenue, maximizing the difference between their income and expenses.

To study the game properties we first develop some tools (Section 6). We develop expressions for the average time to find a block and the average revenue of a miner given the start times of all players. Our results match scenarios analyzed in prior art for subsidy-only rewards [51, 53] and fee-only rewards [17], and an independent simulation for scenarios not previously analyzed, where miners choose different mining gaps. We then proceed to derive the utility function of each player, and a numerical analysis tool to find equilibria in the game. Since the expressions for miner utility do not lend themselves to symbolic analysis, we use numerical analysis to find ε-Nash Equilibria over a wide range of parameters.

Our analysis reveals several things (Section 7). As predicted [17], a mining gap does form when subsidy is sufficiently small and operational expenses are large. Unexpectedly, we show that mining gaps varies between miners based on size, even if their per-rig properties are identical. Additionally, we show that by forming coalitions miners increase their gains. The implication is that in a system where fees are sufficiently large miners are incentivized to form coalitions, leading centralization and defeating the basic premise of a blockchain system. We therefore find the required system parameters to ensure the avoidance of a mining gap and its detrimental effects. These values can be used to inform the design of incentive mechanisms in current and future cryptocurrency systems. We conclude by estimating when Bitcoin will be prone to these effects.

We conclude in Section 8 with a discussion on the implications of our results on operational cryptocurrencies and on future cryptocurrency design. Note these results apply for active cryptocurrency systems, such as Bitcoin, Zcash, Litecoin and so forth, while also very much relevant for the design of new systems.

In summary, we make the following contributions:

- Derive expressions for miner revenue with gaps,
- Define the gap game, played among miners,
- Analyze equilibria in a variety of settings,
- Find that gaps differ among miners,
- Find that miners profit by forming coalitions,
- Estimate when Bitcoin will be affected, and
- Show how to prevent those predicaments.

2 BACKGROUND

Blockchain-based cryptocurrency systems [16, 46, 51, 59] allow users to exchange currency via in-system transactions, without the verification of a centralized authority. Such systems use a public distributed ledger, named the blockchain, to record all internal transactions performed. When a user creates a transaction, it is propagated across the cryptocurrency network, and eventually all other users are familiar with it. The ledger is composed of blocks, a set of transactions grouped together. Participants who run the blockchain protocol, named miners, add new blocks to the blockchain.

The aforementioned cryptocurrency systems operate in a *permissionless* setting, allowing any participant to join or leave the network. A challenge of operating in this setting is to keep security and fairness, as malicious participants can join the network and might deviate from honest behavior. Hence, as part of their protocol, systems use different methods to ensure the desired honest behavior of their participants. A popular method is proof of work [24, 39] that requires a miner to solve a moderately-difficulty cryptographic puzzle in order to create a valid block. By solving such puzzle, a miner proves she invested computational work. Systems that use proof of work rely on the assumption that at least 50% of computational work invested on mining is by honest participants [51]. If malicious users control more than 50% of the computational power, they can employ double-spending attacks [40]. The system is designed to assure that participants are incentivized to follow the protocol rules, and failing to do so will result in decreased profit.

The mining process for a new block goes as follows. A miner groups a set of transactions to be included in the new block and validates them using the blockchain. Then, she looks for a solution for a cryptographic puzzle, which is based on the of selected transactions, the last block added to the blockchain and the cryptocurrency protocol. Attached a valid solution, the block is propagated in the network and other miners agree to add it to their blockchains. When a miner adds a new block to her blockchain, she restarts the mining process with respect to that new block.

In a permissionless setting, computational power may join and leave the system. Therefore, the block time interval might vary, which is undesired. To avoid this predicament, the system's protocol defines a fixed block time interval, and adjusts a difficulty parameter, which determines the difficulty of the cryptographic puzzles. If blocks are created at higher (lower) rate than desired, the protocol sets the difficulty to increase (decrease) the time required for solving the cryptographic puzzle.

Miners attempt to create a valid block by iteratively guessing solutions for the cryptographic puzzle. The process of guessing a solution can be modeled as a Bernoulli trial — a solution is guessed, randomly resulting in a 'success' if the solution fits (and then a valid block is created), or by a 'failure' if it doesn't. The success rate of each trial is fixed and determined by the aforementioned difficulty parameter. Observing a series of such trials, the required number of trials for a success result is geometrically distributed. Therefore, the time required for a successful result is drawn from the exponential distribution.

Note that both the geometric and the exponential distributions are memoryless, so the number of previously failed trials or the time that already passed do not change future probability of success. As a result, miner's chance of finding a valid solution is not changed by how many solutions it had attempted previously[1]. Hence, if a miner re-picks the set of transactions to be included in the block, and by doing so restarts the mining process, her chances of mining the next block are not decreased.

Mining blocks comes with a cost. Mining rigs, the machines used for the mining process, require electricity for their operation [22, 23]. Hardware maintenance, network connection and real-estate, all are required to operate rigs and all carry expenses for miners [15, 21, 62]. To incentivize participants to mine, systems offer rewards in the form of currency. The rewarded currency comes from two sources — newly minted currency that's created as a part of a

[1]This holds for any practical matter as the solutions space is practically infinite.

valid block, and transaction fees paid by transactions included in the block. The amount of minted currency is determined by the cryptocurrency protocol, and the amount of fees is determined by the set of transactions the miner included in the block. Each allocated transaction may offer a different fee, and miners get to pick which transactions they want to include in their blocks.

In two of the most popular cryptocurrency system nowadays, Ethereum and Bitcoin, the reward is dominant by the minted currency. In Ethereum, roughly $20k$ new Ethers are minted daily [26] while fees pay about $2k$ daily [27]. In Bitcoin, the expected daily subsidy is $฿12.5 \cdot 24 \cdot 6 = ฿1800$, as average of 6 blocks are generated every hour, each minting ฿12.5. The daily paid fees varies and averages around a few hundreds BTCs a day [13].

This trend will eventually change, as allocated transaction fees are on the rise. Blocks are bounded, and miners have to pick the set of transactions to include. Many transactions end up not being picked at all. Users who wish to get their transactions picked by miners increase the paid fees to incentivize picking their transactions. Another cause for the expected trend change is that many cryptocurrencies, including Bitcoin, are designed to mint a finite supply of currency. The monetary idea of the finite supply is to prevent inflation. In Bitcoin for example, approximately every four years, the amount of of newly minted coins from new blocks is halved. The expected number of total Bitcoins is estimated to be roughly 21 millions [8, 19, 57].

3 RELATED WORK

Most of the previous work on cryptocurrencies incentives focused on models where the block reward is composed mostly of subsidy. In the original Bitcoin white paper [51], fees are mentioned briefly and an intuitive reasoning about incentives is presented. Kroll et al. [43] analyze Bitcoin as a consensus game when fees are sufficiently low and conclude their impact is negligible. Eyal and Sirer [33] show a deviant mining strategy named *selfish mining*, by which an attacker increase her relative reward. Sapirshtein et al. [58] and Nayak et al.[52] both show more sophisticated variations of the original selfish mining attack that increase the attacker's reward when applied. Other work by Eyal [29] shows mining pools are incentivized to allocate some of their mining rigs to infiltrate other mining pools. Once an infiltrating rig finds a block for the attacked pool, it withholds rather than publishing it. The work shows that an equilibrium exist where two pools infiltrate one another, in which they both end up losing compared to if they were not attacking to begin with. Kwon et al. [44] combine the infiltration attack with selfish mining. In their work, the infiltrating rig selectively alternates between performing withholding and selfish mining attacks. All these works consider a model where the subsidy is the dominant incentive for mining and expenses are negligible. In this work we use a different model, where miners have expenses that differ according to their mining strategy. We also consider the profit of a miner is comprised of both subsidy and fees.

Babaioff et al. [3] discuss incentives for propagating transactions in a cryptocurrency network. They offer and analyze several reward schemes to incentivize participants to distribute transactions in the network. In this work we analyze systems with the traditional reward scheme, where participants are rewarded for mining blocks. Transaction propagation is not incentivized, as in the classical reward scheme.

Möser and Böhme [49] review and analyze the history of transaction fees in Bitcoin. They conclude that historically miners prefer to follow the protocol rules rather than optimize their gains. They predict such state is sustainable only when fees are a negligible part of the incentive. In our work, we analyze systems where fees are not negligible and show how such systems incentivize participants to undesired behavior.

Carlsten et al. [17] analyze Bitcoin when the mining incentive comes solely from fees, in a model where the number of transactions that can be placed in a block is unbounded. In their model there is no residual fee after block generation as all transactions are included in the previous block, and so the block reward immediately after a block is found is zero. They analyze mining strategies and show how miners are incentivized to fork the main chain, disturbing security and liveness. They also revise selfish mining and show an improved version suited dominant fees incentive. An interesting conjecture briefly presented in their work is of the formation of a mining gap, a period of time in which miners turn their mining rigs off to reduce mining expenses. When such mining gap exists, the mining power utilization of the network is suboptimal. In proof of work scheme the immediate implication is that the system is less resilient to attacks. In this work, we present a model to analyze miners' profits and use it to show that mining gaps do form. Our model holds for both bounded and unbounded blocks, as well as for combinations of subsidy and fees as part of the block reward. In their work, the mining gap conjecture was for a set of identical miners that all stop and start mining simultaneously. In contrast, we show that different miners prefer different mining gaps. We also show that rational miners are ought to form coalitions to increase their gains, leading to a centralized system. We analyze loss of resilience to attacks. We conclude by showing that with sufficient initial block reward, all miners are incentivized to resort to the default mining strategy.

Biais et al. [5] analyze the investment in mining equipment required by miners in proof of work cryptocurrencies. They show that miners require excessive acquisition of mining equipment to stay competitive with other miners. In this work we assume the mining equipment acquired is fixed for the network, yet we consider it as part miners' expenses.

Fruitchain [55] is a protocol that is ϵ-Nash incentive compatible against any minority coalition. It shows that if fees are evenly distributed across different blocks as fees are smeared, the potential increase from deviating from the protocol is bounded. Hybrid Consensus [54], Sleepy consensus [56] and Solida [1] are all newer protocols for implementing distributed consensus with blockchains. They presume an altruistic majority of participants and do not consider incentives. Algorand [36] is another such protocol that use *proof of stake* instead of proof of work. It explicitly does not consider incentives, which call for a different definition in the proof of stake scheme. Ouroboros Praos [42] is also a proof of stake blockchain protocol. It uses a new reward mechanism aimed to mitigate block withholding attacks. Bitcoin-NG [32] is a new protocol with the intentions of scaling Bitcoin. It utilizes proof of work for picking a leader, who creates microblocks to validate transactions. Rewards are distributed by consecutive leaders, yet it also

assumes both negligible fees and miner expenses. Lavi et al. [45] considers two new bidding schemes for Bitcoin's fees market, while focusing on incentivizing miners to offer their true bids rather than strategically bid. Our results focus on Bitcoin-like cryptocurrency protocols and with Bitcoin's current incentive scheme, and do not trivially apply to these other protocols.

4 MODEL

We present a realistic model of cryptocurrency systems that we use throughout the rest of this work. As commonly done in blockchains analysis [17, 33, 34, 55], we model systems in a quasi-static state. That means no miners join or leave [37, 43], existing miners maintain their behavior and the system reached equilibrium. Therefore, in our model the system comprises a fixed set of miners and a fixed set of mining rigs. Each miner controls at least one rig and each rig is controlled by exactly one miner. We assume for simplicity that mining rigs are identical [17]. Rigs have two states — off, the default state, and on. Each miner assigns a *start time* for each of her controlled rigs, in which the rig is turned on. We often refer to a turned-on rig as an *active* rig.

Once a rig is turned on, the time it takes to find a valid block is exponentially distributed with a fixed rate parameter, which is shared among all rigs [33, 51, 52, 58]. Therefore the time to find the first block by any of the rigs is the minimum of all finding times by all different rigs. The value of the rate parameter is determined by the cryptocurrency protocol such that the expected block time interval is of a constant value that is also determined by the protocol. The rate parameter represents the difficulty of the cryptographic puzzle, and we use the terms difficulty and rate interchangeably. The assigned start times of rigs by miners affect the value of the rate parameter. If blocks are found too fast (too slow), then the difficulty parameter value is changed by the protocol to decrease (increase) the rate of each individual rig. In equilibrium, the rate parameter is of a fixed value.

The rig that finds the block first awards its controlling miner the block reward, which is comprised of two parts. The first part is *fees reward* that comes from aggregation of newly introduced transactions to the system. This reward is time-dependent, as the time progresses there are more pending transactions in the system, and the potential fees reward grows. The second part is a subsidy we refer to as *base reward*, which to the contrary of the fees reward is fixed over time. This reward is comprised of the minting of new currency with the creation of each block, as well as the expected reward from transaction fees considering the expected initial set of pending transactions. Note that the finding of a new block does not reward any other miners except the miner who found it.

To participate in the system miners expend resources, and we differentiate two types of such resources. First, miners have capital expenses (capex), which are for owning a rig [21, 62] and apply whether the rig is active or not. Miners also have operational expenses (opex), which are paid for having a rig actively mining [22, 23] i.e. owning an active rig. Note that these expenses apply for all miners and not just on those who manage to successfully mine blocks.

Once a block is found, all miners move on to find the next block. This process is repeated indefinitely. The profit of a miner for each block is the difference between her total expenses and her total reward. Rational miners strive to maximize their profits, giving rise to a game.

5 THE GAP GAME

The repeated search for the blocks becomes a series of independent one-shot competitions, in each only one miner gets the reward but all miners pay expenses. To reason about expected revenues, rather than considering the individual iterations we consider a one-shot game played by the miners. A player's strategy is the choice of start times of all of her rigs – when each rig is turned on. The choice of start times are made a-priori by all players. We define the utility of a player to be her expected profit, which is her expected income minus her expected expenses.

The system comprises k mining rigs controlled by n players. Player i controls the set of rigs with indices R_i. Note that $\forall R_i : R_i \neq \emptyset, \forall i \neq j : R_i \cap R_j = \emptyset$ and $\bigcup_{i=1}^{n} R_i = \{1, 2, \ldots, k\}$. Denote the expected block time interval achieved by the protocol by *Block_Interval*. The start time of each rig j is s_j, and we denote the normalized start time $\widetilde{s}_j = \frac{s_j}{Block\_Interval}$. Once a rig is turned on, the time it requires to find a block is exponentially distributed with a rate parameter $\mu(\bar{s})$.

For simplified writing in the following section, we denote $\bar{s}$ as the vector of increasing order k rigs' start times.

All rigs are identical – Each mining rig costs c_{cap} per time unit for ownership and c_{op} per time unit if it is turned on.

The utility of player i, denoted $utility_i$, is its expected profit, namely its expected rewards minus its expected expenses. We derive an expression for the utility function in the following section.

The utility is affected by the strategies of all players. As is common in miner behavior analysis [30, 33, 43, 44, 58], our solution concept is myopic — each player chooses her best-response strategy based on the current strategies of other players. Players do not take into consideration how other players will adapt based on their new choice of strategy.

Strategy space. Note that the strategy space does not include turning rigs off, as this is an irrational behavior. Block finding time of an active rig is drawn from the exponential distribution, which is memoryless. That means the probability for a rig to find the block in some time interval is not affected by how much time had already passed since that rig began mining. Therefore, a single rig's chances of finding a block are not decreasing over time. Recall that the total reward also increase over time. Hence, if at some point in time the reward justified turning a rig on, then this justification holds from that time until the block is found.

5.1 Parameters Analysis

The parameters values are affected by a wide range of factors, stemming from different sources. The fees are affected by the system users and the market [6, 10, 13, 25, 41, 49]. The base reward is also affected by systems user and market, as these affect the residual fees, but also by the minting rate, which is defined by the cryptocurrency protocol. Capex is affected by factors such as technological advancements of mining rigs efficiency [5], personnel wages, and real estate costs [21, 47, 63]. Opex is affected primarily by the electricity costs [15, 21–23, 47] for operating the mining rigs. That includes both the actual puzzle solving process as well as cooling

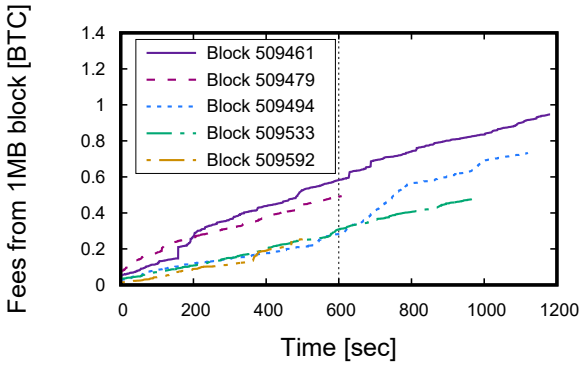


Figure 1: Fees in most rewarding 1MB block accumulation in Bitcoin's mempool.

expenses. These parameters are therefore not only difficult to estimate, but they vary between different currencies, and also over time for the same currency. Hence, we analyze the system for a range of parameters values to make general observations, focusing on trends that are robust across the parameter range.

We begin by analyzing how fees accumulate in the system, and then move towards determining parameter values which we'll be used throughout the rest of this work.

5.1.1 Fees Reward Accumulation over Time. Accurately predicting the fees accumulation function of the pending transactions is out of the scope of this work, and we resort to an educated approximation. We measure how fees accumulate over time in the Bitcoin network and apply our findings to the general model.

We conducted the following measurement at February 2018. Using a Bitcoin node connected to the Bitcoin network, we monitor the pending transactions awaiting to be included in blocks. At fixed time intervals of one second, we find the most rewarding set of transactions to include in a valid 1 MB Bitcoin block. We record the fees that transactions in this set offer. The values recorded correspond to blocks 509426 up to 509605 and span about 30 hours of measurements.

In Figure 1 we present the potential fees reward as a function of time, during the time it took to mine a specific block, for some arbitrary measured blocks. The vertical dashed line shows the expected block time interval, which is 600 seconds in Bitcoin. As expected, some blocks required more (less) time than the expected interval.

Using linear regression on all the measured blocks, we calculate the squared correlation value and get an average of $R^2 = 0.96$. We conclude a linear approximation is reasonable and therefore treat the fees reward as if it increases linearly.

We also note that immediately after a block is found, there are still pending transactions awaiting to be included in future blocks. We can consider the expected fees of these pending transactions as if they were part of the fixed base reward out of the total block reward.

Hence, we model the total block reward as a linear function, where the slope is the expected fees accumulation rate, and the intercept is the sum of the newly minted currency and the expected fees available immediately after a block is found. We repeated these measurements at other dates for different periods of time and received similar results. We denote λ_t as the fees accumulation rate and λ_0 as the base reward.

Name	c_{op}	c_{cap}
high_op	0.02	0.00
med_op	0.01	0.01
low_op	0.00	0.02

Table 1: Opex and capex settings.

5.1.2 Analysis Parameters. We denote by *Expected_Total_Fees* the expected total fees accumulating during the expected time to find a block, namely, $Expected\_Total\_Fees = Block\_Interval \cdot \lambda_t$. Denote by *EBRR* the ratio of the expected base reward and the expected accumulated fees, so $EBRR = \frac{\lambda_0}{Expected\_Total\_Fees}$. Throughout the following sections we present results for different values of *EBRR*.

For all experiments we choose the following parameters arbitrarily: Fees increase rate is set to $\lambda_t = 1$, the expected block interval to $Block\_Interval = 10000$, and the number of rigs to $k = 128$.

Recall we analyze systems at a quasi-static state and miners do not join or leave the system. The profit for miners should therefore be slightly more than the interest rate plus associated risk. For simplicity, to avoid introducing unnecessary parameters, we set the expense parameters such that the expected profit of miners will be zero. Therefore, we choose values so $c_{op} + c_{cap}$ is of a fixed value.

The ratio between of the two types of expenses, opex and capex, can vary considerably among cryptocurrencies. Different cryptocurrencies use different proof of work [2, 35, 38, 48, 65] with different computational costs, varying mining technology [5, 21, 62], and varying electricity expenses [15, 47]. Therefore, we use three different settings that are of interest for the ratio of capex and opex parameters values, which are detailed below and summarized in Table 1. Two settings describe extreme cases, where in one all the expenses are opex, and in the other all the expenses are capex. The third setting describes the average case of the first two, where the opex and capex are equal.

The system's properties are determined by the parameters ratios — the ratio of expected fees reward and the base reward, the ratio of opex and opex and so forth. Throughout the rest of this work we cover a wide range of these ratios that demonstrate the important trends. We emphasize that different values satisfying the same ratios yielded the same qualitative results.

6 GAME ANALYSIS

To find the utility of each player, we start by analyzing the block finding time probability distribution. This is a function of the players' selection of start times. We model the block finding time as a random variable denoted B with cumulative distribution function (CDF) and probability density function (PDF) denoted $F_B(t; \bar{s}, \mu(\bar{s}))$ and $f_B(t; \bar{s}, \mu(\bar{s}))$, respectively.

We begin by discussing the difference of the probability distributions in our model distributions from ones considered in prior art. We present three different scenarios of rigs' start time choices and the derived probabilities of the system. Table 2 lists the values used in each scenario and Figure 2 depicts the resultant distributions. Figure 2a shows the ratio of active rigs as a function of time, while Figures 2b, 2c show the PDF and CDF of the block finding time B,

	Rigs Quarter			
Scenario	**Q1**	**Q2**	**Q3**	**Q4**
classical [33, 51, 52, 58]	0	0	0	0
uniform gap [17]	0.5	0.5	0.5	0.5
arbitrary gap	0.2	0.4	0.6	0.8

Table 2: Rigs Start Times.

respectively. In this example the expected block interval is set to be $Block\_Interval = 1$. Each scenario has four equal-size players, each controlling 32 out of the total $k = 128$ rigs in the system.

In the *classical* scenario, all players set their rigs' start times to 0. This is the scenario commonly analyzed in the literature [33, 51, 52, 58]. In Figure 2a we see a constant ratio of 1 as all rigs are set to have $t = 0$. From Figures 2b, 2c we learn that $f_B(t; \bar{s}, \mu(\bar{s})) > 0, F_B(t; \bar{s}, \mu(\bar{s})) > 0$ for all t, which is expected as all rigs are active throughout the entire scenario. As all block finding times of single rigs are exponentially distributed, the block finding time is also exponentially distributed. The rate parameter is $\frac{1}{Block\_Interval}$ such that the expected time will be $Block\_Interval$.

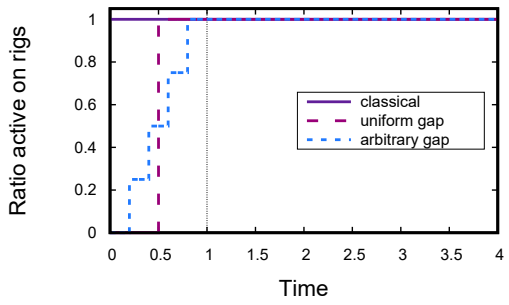
In the *uniform gap* scenario, all players set their rigs' start times to 0.5. This is scenario is analyzed in [17]. In Figure 2a we can see that all the ratio is 0 while $t < 0.5$ and to 1 while $t \geq 0.5$, as all rigs are set to have $s = 0.5$. In Figures 2b, 2c $f_B(t; \bar{s}, \mu(\bar{s})) = 0, F_B(t; \bar{s}, \mu(\bar{s})) = 0$ while $t < 0.5$ as no rigs are active. When $t \geq 0.5$, all rigs are turned on and $f_B(t; \bar{s}, \mu(\bar{s})) > 0, F_B(t; \bar{s}, \mu(\bar{s})) > 0$. In this case, the block finding times of single rigs are shifted-exponentially distributed, the block finding time is also shifted-exponentially distributed. The shift is of 0.5 time units and the rate parameter is doubled $\frac{2}{Block\_Interval}$ to compensate. Notice that the expected block time interval is still $Block\_Interval$.

In the *arbitrary gap* scenario, each player set her rigs' start times to a different value. To the best of our knowledge, this scenario is first analyzed in this work. In Figure 2a we can the ratio increases as time progresses. The spikes occur at the times where rigs are turned on. Notice that for $t < 0.2$ all the rigs are still turned off and the ratio is 0. At $t = 0.2, 0.4, 0.6, 0.8$, the change in the number of turned on rigs causes the CDF in Figure 2c to be semi-differentiable, resulting in the jump discontinuities of the PDF in Figure 2b. As expected, $f_B(t; \bar{s}, \mu(\bar{s})) = 0, F_B(t; \bar{s}, \mu(\bar{s})) = 0$ while $t < 0.2$ as no rigs are turned on. Note that for all scenarios $\lim_{t\to\infty} f_B(t; \bar{s}, \mu(\bar{s})) = 0$, $F_B(0; \bar{s}, \mu(\bar{s})) = 0$ and $\lim_{t\to\infty} F_B(t; \bar{s}, \mu(\bar{s})) = 1$.

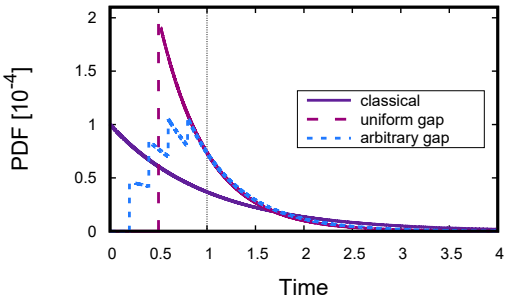
The rest of this section is organized as follows. In Section 6.1 we derive an expression for the distribution based on the selected values of $\bar{s}$, and proceed to derive an expression for $utility_i$ in Section 6.2. Then, in Section 6.3 we present a simulator designed to confirm our theoretical analysis. We conclude in Section 6.4 by presenting an optimizing tool created to find equilibria in the game.

6.1 Distribution Analysis

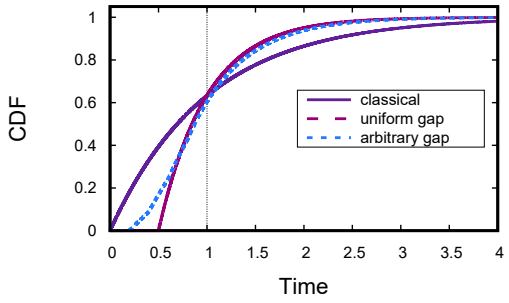
The first step towards analyzing the system is to derive an expression for the distribution, namely $F_B(t; \bar{s}, \mu(\bar{s}))$ and $f_B(t; \bar{s}, \mu(\bar{s}))$, based on players' strategies. We begin by deriving the distribution of a single rig. Observe any single rig j that with start time s_j. Denote the time this rig requires for finding a block as a random variable B_j. Recall that the rate of a single rig is $\mu(\bar{s})$, which is set by



(a) Ratio of turned on rigs.



(b) PDF $-f_B(t; \bar{s}, \mu(\bar{s}))$



(c) CDF $-F_B(t; \bar{s}, \mu(\bar{s}))$

Figure 2: System properties for different scenarios.

the protocol. The value of B_j is drawn from the shifted exponential distribution, with a shift of s_j and rate $\mu(\bar{s})$.

The PDF of B_j is

$$f_{B_j}(t; s_j, \mu(\bar{s})) = \begin{cases} 0, & t \leq s_j \\ \mu(\bar{s}) \cdot \exp\left(-\mu(\bar{s})(t - s_j)\right) & t > s_j \end{cases}$$

and its CDF is

$$F_{B_j}(t; s_j, \mu(\bar{s})) = \begin{cases} 0, & t \leq s_j \\ 1 - \exp\left(-\mu(\bar{s})(t - s_j)\right) & t > s_j \end{cases}.$$

As $F_{B_j}(t; s_j, \mu(\bar{s})) = \Pr(t \geq B_j) = 1 - \Pr(t \leq B_j)$ we get that

$$\Pr(t \leq B_j) = \begin{cases} 1, & t \leq s_j \\ \exp\left(-\mu(\bar{s})(t - s_j)\right) & t > s_j \end{cases}.$$

All rigs are competing on finding the next block. The rig that finds the next block first is the rig with the minimal value of B_j. Therefore, the time required for finding the next block is $B = \min_{j\in\{1,2,\ldots,k\}} B_j$.

We define for any time t and any player i the set $active_i(t)$ to be all player i's rig indices that are active at time t : $active_i(t) = \left\{j \mid j \in R_i \wedge s_j \leq t\right\}$. We define $active(t)$ to be the set of all active rigs at time t. Note that $active(t) = \bigcup_{i=1}^{n} active_i(t)$.

The probability that none of the rigs have found a block by time t, $\Pr(t \leq B)$, is the product of $\Pr(t \leq B_j)$ for all j (as rigs are independent of another one). This probability is given by

$$\Pr(t \leq B) = \bigcap_{j\in\{1,2,\ldots,k\}} \Pr(t \leq B_j) = \prod_{j=1}^{k} \Pr(t \leq B_j) = \exp\left(-\mu(\bar{s}) \cdot \sum_{j\in active(t)} (t - s_j)\right).$$

The CDF of B is therefore

$$F_B(t; \bar{s}, \mu(\bar{s})) = 1 - \Pr(t \leq B) = 1 - \exp\left(-\mu(\bar{s}) \sum_{j\in active(t)} (t - s_j)\right) \quad (1)$$

and the derivative is its PDF,

$$f_B(t; \bar{s}, \mu(\bar{s})) = \mu(\bar{s}) \cdot |active(t)| \cdot \exp\left(-\mu(\bar{s}) \cdot \sum_{j\in active(t)} (t - s_j)\right). \quad (2)$$

As expected, when $|active(t)| = 0$ then $\sum_{j\in active(t)}(t - s_j) = 0$ which results in $F_B(t; \bar{s}, \mu(\bar{s})) = 0$ and $f_B(t; \bar{s}, \mu(\bar{s})) = 0$. We can verify that $f_B(t; \bar{s}, \mu(\bar{s}))$ is a valid PDF by checking that $\int_{-\infty}^{\infty} f_B(t; \bar{s}, \mu(\bar{s}))dt = 1$ holds. We prove this is in fact the case in Appendix A. We also find the value of $\mu(\bar{s})$ at equilibrium. This process is presented in Appendix B and utilized when required throughout this work.

6.2 Utility

We are now ready to express $utility_i$. Recall that $utility_i$ is the *expected* profit of player i. We define three new random variables — $Income_i$, $Expenses_i$, $Profit_i$, representing the income, expenses and profit of player i, respectively. Throughout the rest of this section, we assume the value of B is t, and use it to find the expected profit of player i that is denoted as $E\left(Profit_i \mid B = t\right)$. We then use the law of total expectation and the PDF of B from Equation 2 to derive an expression for $E\left(Profit_i\right)$, which is by definition $utility_i$.

6.2.1 Income Function. We model the income function linearly with a slope of λ_t and an intercept of λ_0. Therefore, the total available reward at time t is $\lambda_0 + \lambda_t \cdot t$.

Recall that once a rig is turned on, the time it requires to find a block is drawn from the exponential distribution. The exponential distribution is memoryless, meaning the time that passed does not affect the chances of a rig to find the block. Since the rate parameter $\mu(\bar{s})$ is shared among all rigs, at any given time all the active rigs have the same chance to find the block, regardless of how much time they had been active for already.

Observe the set of active rigs at the time the block is found $active(t)$. The probability of a specific active rig to find the block is one divided by the total number of active rigs. Note that since the block was found at time t, then $\exists j \in \{1, 2, \ldots, k\}$ such that $s_j \leq t$ and therefore $|active(t)| > 0$. Players control many rigs, so the probability that player i controls the rig that found the block is the number of her controlled active rigs divided by the total number of active rigs. We denote the ratio of player i's active rigs out of all the active rigs at time t as $\alpha_i(t) = \frac{|active_i(t)|}{|active(t)|}$. The ratio $\alpha_i(t)$ is therefore the expected factor of player i's portion of the total reward.

We conclude that if a block was found at time t, then the expected income of player i is

$$E(Income_i \mid B = t) = \alpha_i(t)(\lambda_0 + \lambda_t \cdot t). \quad (3)$$

6.2.2 Expenses Function. Recall that players have two kind of expenses. The first, capex, for owning a rig. The second, opex, for having a rig active.

Capex applies for all rigs controlled by the player, whether they are turned on or not. For each rig, the capex it imposes by time t is the product of c_{cap} and t. Recall that R_i is the set of rig indices that player i controls, which totals with $|R_i|$ rigs. The total capex of player i by time t are therefore $c_{cap} \cdot |R_i| \cdot t$.

Opex applies only for active rigs. For each active rig, the expenses it imposes by time t is the product of c_{op} and the time duration this rig is turned on already. At time t, active rig j with s_j has been active for $t - s_j$ time. Summing for all rigs of player i results that by time t the total opex are $c_{op} \cdot \sum_{s\in active_i(t)}(t - s)$.

Combining both of these expenses, if a block was found at time t then the expected expenses of player i are

$$E\left(Expenses_i \mid B = t\right) = c_{cap} \cdot |R_i| \cdot t + c_{op} \cdot \sum_{s\in active_i(t)} (t - s). \quad (4)$$

6.2.3 Profit Function. The expected profit of a player is her expected income minus her expected expenses. Using Equations 3 and 4, we get that if a block was found at time t then the expected profit of player i is

$$E\left(Profit_i \mid B = t\right) = E\left(Income_i \mid B = t\right) - E\left(Expenses_i \mid B = t\right). \quad (5)$$

6.2.4 Utility Function. To get the expected profit of a player, we use the law of total expectation (sometimes referred to as the *smoothing theorem*). We use the PDF of B that from Equation 2. Therefore, the expected profit of player i, which is also defined as her utility, is

$$utility_i = E\left(Profit_i\right) = E\left(E\left(Profit_i \mid B = t\right)\right) = \int_{-\infty}^{\infty} \left(E\left(Profit_i \mid B = t\right) \cdot f_B(t; \bar{s}, \mu(\bar{s}))\right) dt. \quad (6)$$

6.3 Cryptocurrency System Simulator

In addition to the theoretical analysis, we implemented a cryptocurrency system simulator. It is built as an event driven simulation and operates at the continuous time space. It includes a set of miners that control mining rigs. Each miner keeps a private copy of the blockchain and compete with the other miners on finding the next block. We use exponentially distributed random events to simulate

block mining intervals. The rate parameter of the exponential distribution is set such that the mean block time interval is kept at a fixed value. When a miner finds a block, he announces it to the rest of the miners. Each miner sets a-priori a start time for each of her controlled rigs, which refers to required time to pass since the finding of the previous block so this specific rig will become active. Active rigs keep on mining until the next block is found by any rig. Transactions accumulate over time and found blocks include the allocated fees as a reward, as well as a base reward for each block. Miners also pay expenses as a function of their controlled rigs (capex) and the time those rigs were turned on (opex).

We emphasize that the theoretical analysis yields the expected profit for a player from a *single* block, while simulations create a long blockchain containing many blocks mined by all participating miners. Hence, when referring to the results of the simulator, we refer to mean profit of a miner over time.

6.4 System Equilibrium Search

The utility presented in Equation 6 is derived given all players' strategies. If a player changes her strategy, then the utility of all the other players is also affected. We are interested in finding equilibria, i.e. strategies of all players such that no player can improve her utility by changing her strategy.

The utility of a player is infeasible to express in a symbolic manner. It is a function of all player strategies as well as the difficulty parameter, which can be expressed only as an implicit function (in any case where there are at least two distinguished start times). Therefore, we use numerical analysis to find equilibria in the system.

We implemented an equilibrium-search-tool — a tool we use to numerically search for an equilibrium, and that works in the following manner. The equilibrium-search-tool receives as an input the system income and expenses parameters, as well as a list of tuples representing all players' strategies. Each tuple of that list is in the form of $\{i, j, s\}$, where i is a player's index, j is a rig index that are controlled by player i and s is a start time selected for rig j. Note that $\{j_1, \ldots, j_m\} \subset \{1, 2, \ldots, k\}$.

Iteratively, the equilibrium-search-tool chooses at random an input tuple $\{i, j, s\}$, and searches what value of a new start time for rig j will result in maximal utility for player i. This process is repeated until no player increases her utility by changing any of her rigs, meaning an equilibrium is reached.

Note that all equilibria found by such process are only ϵ-Nash-equilibria, as they are limited by the numerical precision of the calculation. To counter that predicament, we repeat the search process with different random start times and different optimizing order. In all conducted experiments, the randomness introduced had no effect on the output equilibrium. That strengthens our analysis of an equilibria.

7 ANALYSIS RESULTS

We study the system behavior in a wide range of scenarios, detailed in Section 5.1 – from the common case in today's operational currencies where subsidy dominates rewards to the extreme case where fees dominate rewards, and with varying expenses distributions. We proceed to verify our analysis tools using the cryptocurrency system simulator, compare it to known results, and observe some predictable trends (Section 7.1).

In Section 7.2 we present the first trend that was not predicted in prior art – even when rig parameters are identical, players of different sizes choose different gap sizes in equilibrium. We also present the utility of a single player as a function of other players' strategies and show players are expected to optimize. Then, in Section 7.3, we analyze the game with equal-size players of varying size, showing how the gap game encourages equilibria that affect the security of the system.

We want to compare the utility of players in systems with different reward schemes. To eliminate the effect of players having high utility as they are in systems that offer high rewards, we instead consider on the utility of players out of the total utility available in the system. We also want to eliminate the effects of bigger players having more utility and therefore we actually consider the utility normalized by size. More formally, we use *normalized* utility of players, that is defined to be the utility presented in Equation 6, normalized by two factors. The first factor is $\lambda_t \cdot Block\_Interval + \lambda_0$ that represents the total expected income from a block in the system. In our experiments this factor varies as a function of λ_0. This normalization allows us to compare systems with different λ_0. The second normalization factor is the number of rigs each player controls, which varies for each player.

7.1 Analysis Tools Validation

We validate our analysis by comparing our theoretical results with both simulated and previously known results. In Section 7.1.1 we compare with the classical scenario discussed in previous work [33, 51, 52, 58], when there are no gaps. Next, in Section 7.1.2, we present and analyze a scenario with arbitrary gaps. For this scenario there is no previous work to compare with, so we compare our theoretical results only against the simulation.

7.1.1 Scenario One – No Mining Gap. We analyze a simple scenario where the system is comprised of two miners, both mine without a mining gap, and with no expenses. We use the analytic expression and the simulator to obtain the *relative* utility of player 1 – the ratio of a her utility out of the utility of all players. We vary the player 1's relative mining power and plot its relative utility. This is the common metric that was used in previous work [33, 51, 52, 58]. In those previous works, reward from fees is negligible, meaning the relative utility is the ratio of blocks mined by a player. It is also the metric used for the scenario where there is no reward from minting, all transactions are identical in their fees, and blocks are unbounded — the relative utility is the ratio of transactions included by blocks mined by the player [17]. In both cases the expected result is for a player with α relative mining power to have a relative utility of α. All of these works neglect the expenses of players. Hence, for comparison purposes we nullify these expenses in this particular scenario by setting $c_{op} = c_{cap} = 0$.

We compare the relative utility according to the game analysis, the simulated results, and the expected result. For the simulated results, we use the average of 10 different runs with different random seeds. We use several values for *EBRR* and, as expected, the results match.

7.1.2 Scenario Two – Arbitrary Mining Gap. We analyze a different scenario with arbitrary mining gaps. The game consists of two players that choose arbitrary start times for arbitrary portions of

Portion of player 1's rigs	Normalized start time
0.2	0.1
0.7	0.3
0.1	0.9

Portion of player 2's rigs	Normalized start time
0.2	0.2
0.4	0.5
0.4	0.6

Table 3: Normalized start times for players' rigs.

their rigs. Each player partitions her controlled rigs into three sets, each with a different start time. We choose the start times arbitrarily, and their values are presented in Table 3. Recall that $\widetilde{s}_j$ is the start time of rig j normalized by the expected block time interval. We use the game analysis and the simulator to obtain the normalized utility of player 1, and plot it as a function of her relative mining power. The values of c_{op} and c_{cap} are presented in Table 1. We repeat the analysis for different values of *EBRR*.

Results are presented in Figure 3. As in the previous experiment, for the simulated results, we use the average of 10 different runs with different random seeds. The error bars show the highest and lowest values.

This comparison demonstrates the effect of the *EBRR* value. For the low values of *EBRR*, player 1 has negative utility. As player 1 controls more rigs (i.e., has higher relative mining power), her per-mining-rig utility is decreasing with her total mining power. Even though player 1 has higher probability to get rewarded as she controls more mining power, the increase in her expenses is more significant, resulting in lower utility. For the higher values of *EBRR*, the opposite occurs. As player 1 controls more rigs, her per-rig-utility is increasing with her total mining power. The increase in the probability to get rewarded surpasses the increase in expenses, resulting in higher utility. This trend is maintained for all settings of opex and capex ratios.

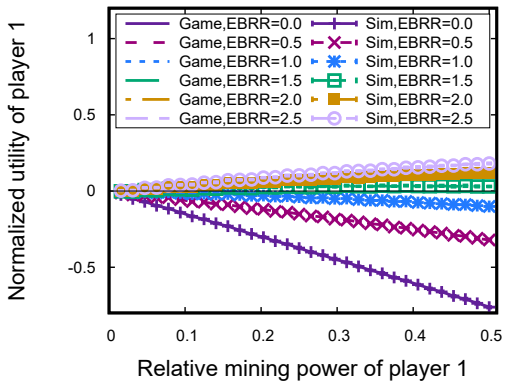
Another interesting result shows the impact of the opex-capex ratio. For any player 1 relative mining power and any *EBRR*, the utility of player 1 where capex is dominant (*low_op*, Figure 3a) is lower than when capex and opex and equal (*med_op*, Figure 3b) and when opex is dominant (*high_op*, Figure 3c). Player 1's choice of start times that are greater than 0 is an optimization. By doing so, she reduces her expected opex as her controlled rigs are expected to be active for less time. The more rigs she controls, the more impactful this effect is. Hence, this suggests that at when opex is at play (*med_op* ,*high_op*), mining gap formation is beneficial.

We conclude the simulations discussion with the following observation. Recall the analysis is for the expected behavior and hence considers the expected pending transaction fees as part of the base reward. The simulations confirm the results predicted by the expected-case analysis, despite the fact the reward varies between individual blocks.

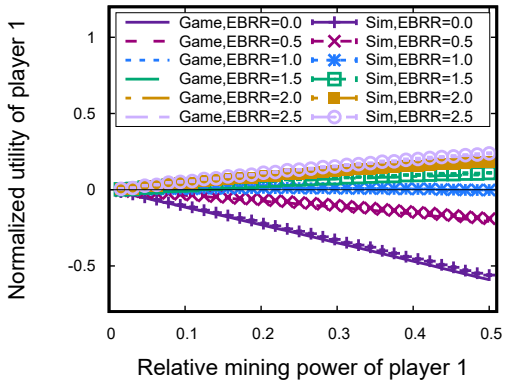
7.2 Case Studies

We present two insights regarding the optimal start time of players. The first reviews, through an example, the effects of other players' start time strategies on the normalized utility of a player. The second reviews optimal start times of players of different sizes.

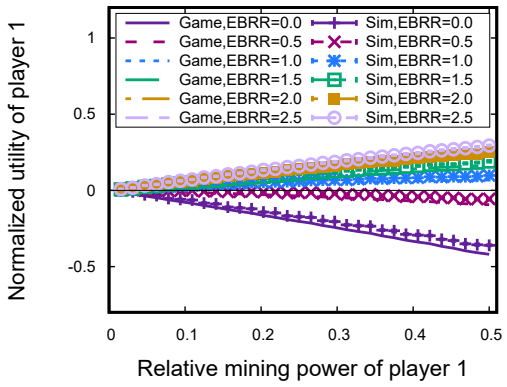
7.2.1 Case Study One — Effects of Other Players' Strategies. In our example we use a game with 8 players, controlling 16 rigs each, where each player selects a single mutual start time for all of her



(a) *low_op*



(b) *med_op*

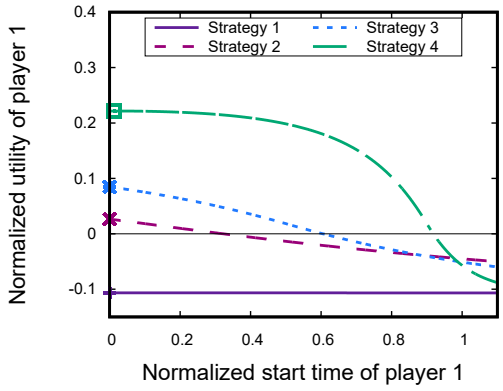


(c) *high_op*

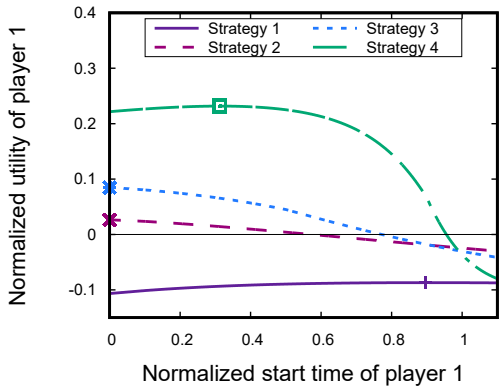
Figure 3: Comparison of the normalized utility of player 1 — game analysis and simulation.

	Number of players		
Normalized start times	**0.1**	**0.5**	**0.9**
Strategy 1	4	0	3
Strategy 2	7	0	0
Strategy 3	0	7	0
Strategy 4	0	0	7

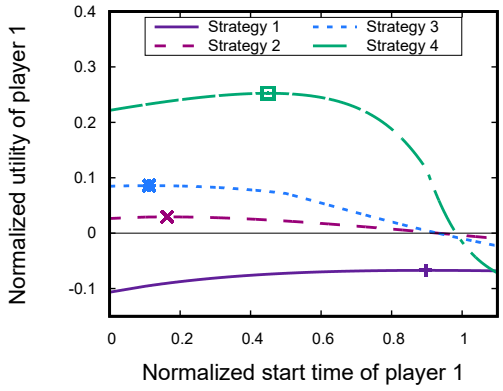
Table 4: Normalized start times of the other players.



(a) *low_op*



(b) *med_op*



(c) *high_op*

Figure 4: Normalized utility of player 1, for different strategies of other players.

rigs. We use $EBRR = 2$ for this example. In Figure 4 we present the normalized utility of player 1 as a function of her rigs start time for different start times of the other players. The maximal value of each curve is marked. Start times strategies of the other players are listed in Table 4. In strategy 1, four of the players choose normalized start time of 0.1 while the remaining three choose 0.9. In strategies 2,3 and 4, all the other seven players choose normalized start times of 0.1, 0.5 and 0.9, respectively.

	Relative Size, Normalized Start Time			
#	Player 1	Player 2	Player 3	Player 4
1	0.125, **0.157**	0.125, **0.157**	0.250, **0.261**	0.5, **0.452**
2	0.250, **0.261**	0.250, **0.261**	0.500, **0.452**	-, -
3	0.125, **0.131**	0.375, **0.350**	0.500, **0.452**	-, -
4	0.125, **0.131**	0.250, **0.261**	0.625, **0.452**	-, -

Table 5: Case study of different size players.

An increase in player's normalized utility is achieved by two means — increasing her chance of being rewarded and therefore increasing her expected reward, and by reducing her expenses. When a player chooses an early start time, she prefers to increase her chance for the reward, at the cost of increasing her expenses. When a player chooses a late start time, she prefers to decrease her expenses, at the cost of lowering her chances to be rewarded.

Notice strategy 4, where all other seven players choose normalized start time of 0.9. At the *low_op* setting, where $c_{op} = 0$, player 1 can increase her chances of being rewarded without an increase in her expenses. Hence, the optimal normalized start time as seen in Figure 4a is zero. At the *med_op* and *high_op* settings, where $c_{op} > 0$, the conflict described above comes in play. Choosing normalized start time of zero will cause unnecessary expenses, resulting in sub optimal normalized utility. Choosing a relatively late normalized start time, such as 0.9, will also result in sub optimal normalized utility, as now player 1 has much lower chances to be rewarded and therefore much lower normalized utility. The optimal normalized start time is therefore a time that balances the two conflicting interests. From Figures 4b and 4c we can learn the optimal normalized start time in this case is in the range of $[0.2, 0.5]$.

At strategy 4, player 1 has relatively long period of time where she was the only player with active rigs. This leads to relatively high chance for her to be rewarded, which she could forfeit to reduce her expenses. When the other players use strategy 2 for example, this privilege doesn't exist anymore, and player 1 shouldn't forfeit any chance she can get to win the reward. Hence, in all settings, her optimal normalized start time is 0. Strategy 3 is in a sense the average case. The other players start at 0.5. This start isn't too early yet not too late, and player 1 can optimize. As expected, the optimal time is also dependent on the opex value.

Strategy 1 demonstrates the opposite case, where player 1 is better off waiting to decrease her expenses. When $c_{op} > 0$, player 1 minimizes her expenses by choosing fairly late start times. When $c_{op} = 0$, player 1 can't reduce expenses by choosing later start times, and therefore the optimal choice is normalized start time of zero.

7.2.2 Case Study Two — Different-Size Players. We now use the equilibrium-search-tool to analyze a scenario with players of different sizes. In this scenario we use the *high_op* setting with $EBRR = 2$. We present the equilibria obtained by the equilibrium-search-tool for some arbitrary sets of players. Sets at examination and the resulting equilibria start times are presented in Table 5.

We note that players with the same size choose same start times, such as player 1 and player 2 in scenario 1. We also note that players with higher relative size choose higher start times. Another result is that the bigger player in each scenario picks the same start time, even when the smaller other players choose different start times.

We now present an intuition for these results. Consider a player of $1 - \epsilon$ relative mining power for some infinitely small ϵ. This player is practically guaranteed to find the block and get the reward, whether she chooses early or late start times for her rigs. Such player will then prefer to cut her expenses by choosing later start times, as her chances of winning are practically unaffected by such choice. Now consider the opposite case, with a player of ϵ relative mining power. This player has low chance to win the reward and she cannot afford to dwindle it any further. Hence, such player will choose start time 0, to maintain what low chances of getting a reward she has. Intuitively, the higher relative power a player controls, the later the start time she prefers for her rigs.

We now move to analyze simpler settings, where all players are of equal sizes.

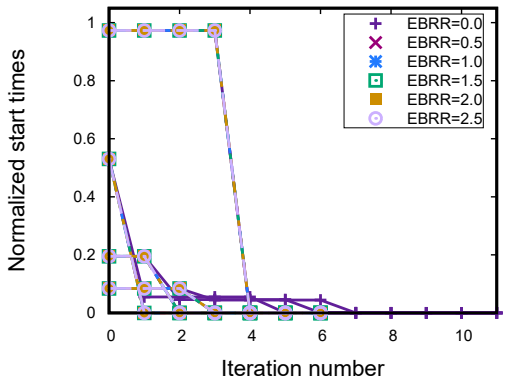
7.3 Equal-Size Miners Equilibria

We proceed to analyze equilibria where all miners are of equal sizes. For a varying number of players, we divide the k mining rigs among the players evenly, creating a set of equal-size players. For the different settings presented in Table 1, and different values of *EBRR*, we use the equilibrium-search-tool to find equilibria start times for the players.

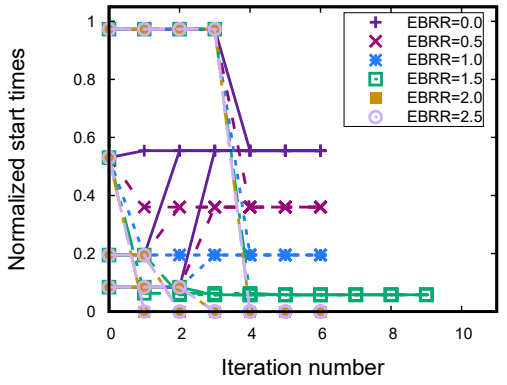
We visualize, as an example, some of the equilibria search processes for a system comprised of 4 equal-size players, controlling 32 rigs each. In Figure 5, for the three different settings and different *EBRR* values, we plot at each iteration of the equilibrium-search-tool the normalized start times of all of the 4 players. We get the same qualitative results for any different numbers of players, and for different random initial start times.

We first notice that for all settings and for all values of *EBRR*, each player eventually converges to the same start time. We conclude symmetry holds. We also notice that some settings require only one iteration before reaching the equilibrium start time, while other settings require a few iterations. This strengthens the analysis result that the start times of other players affect the optimal strategy. Another result is that different settings and values of *EBRR* lead to different optimal start times. We discuss these result thoroughly in the following section.

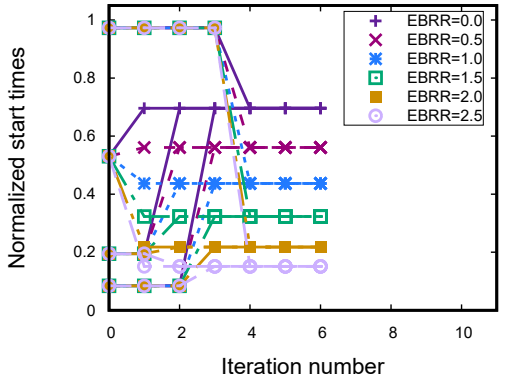
7.3.1 Start Times at Equilibria. In Figure 6 we present the normalized start times at equilibrium of all miners as a function of the number of miners. For the *low_op* setting, presented in Figure 5a, the equilibrium is at start time zero for all values of *EBRR*. This is expected, as $c_{op} = 0$ and players do not suffer an increase in expenses by turning their rigs on earlier. By setting their rigs' start times to zero, the players maximize their probability of getting rewarded, hence increasing their utility. For the *med_op* and the *high_op* settings, presented in Figures 5b and 5c respectively, start times at equilibrium are zero only for the higher values of *EBRR*. When *EBRR* is low, the base reward is not substantial enough to incentivize players to choose start time zero, as they rather turn their rigs on at a later time and decrease their expected expenses. Therefore the expenses prevented due to the optimization are more significant than the loss of potential reward. When *EBRR* is high the base reward becomes more substantial and the opposite optimization takes place. Players prefer start time zero, as the increase in probability to get the reward and therefore the expected reward are more significant than the increase in expenses.



(a) *low_op*



(b) *med_op*

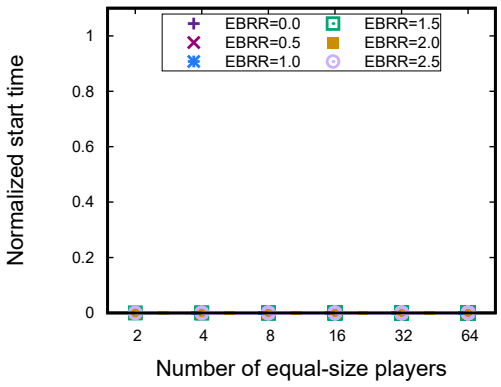


(c) *high_op*

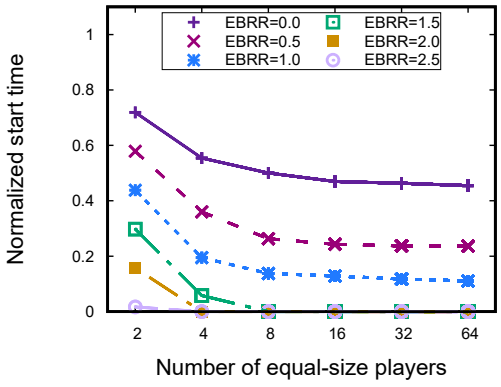
Figure 5: Convergence of normalized start times of equal-size players.

Another interesting result is that players with higher relative power prefer later start times. An intuition for that was presented in Section 7.2. For example, in a system with only 2 players, each player has a relative mining power of 0.5, these players choose the latest start time. For systems with more players, say 16, each has relative mining power of 0.0625 and choose an earlier start time.

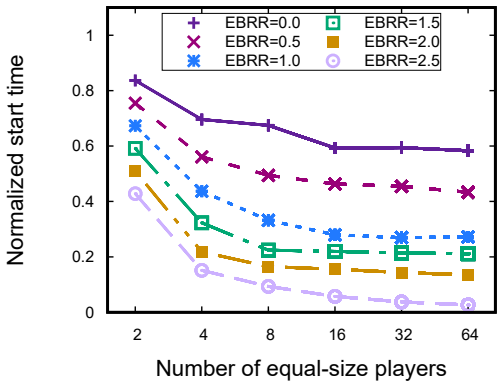
7.3.2 Utility Increase from Optimization. In Figure 7 we present the normalized utility increase of players from optimization. We



(a) *low_op*

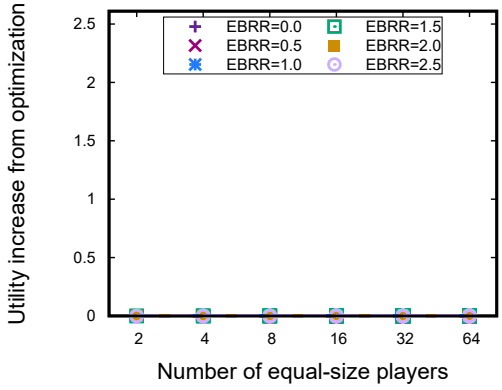


(b) *med_op*

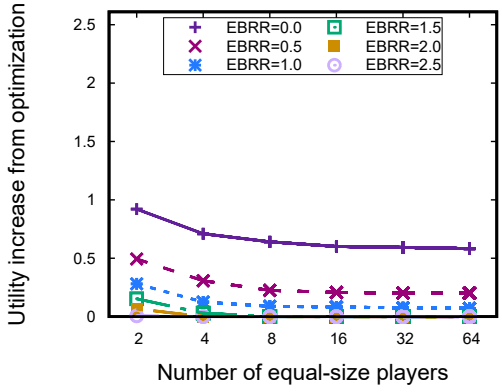


(c) *high_op*

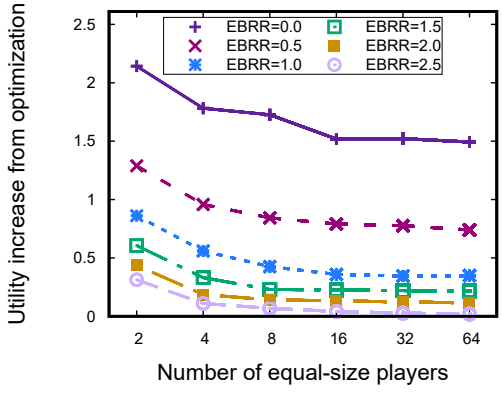
Figure 6: Normalized start times of equal-size players.



(a) *low_op*



(b) *med_op*



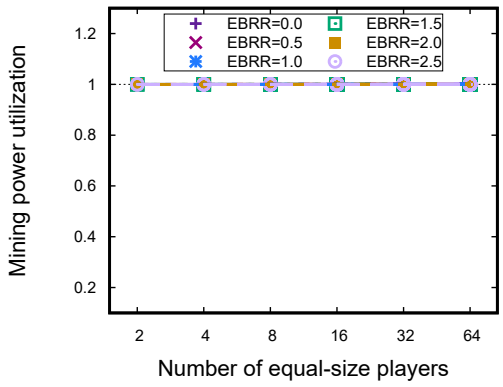
(c) *high_op*

Figure 7: Utility increase from optimization.

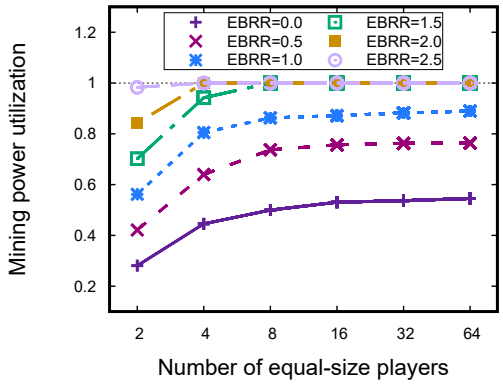
measure the utility of players at the optimal and zero start times and subtract the latter from the former.

Recall that for the *low_op* setting, the equilibrium start time is zero, hence there is no increase in utility. This result is presented in Figure 7a, For the *med_op* and the *high_op* settings, equilibrium start time is zero only for the higher values of *EBRR*. The results of such optimization are presented in Figures 7b and 7c. When players optimize, they gain a substantial increase in their utility. Notice that three factors contribute to an increase in utility — low *EBRR*, high c_{op} and a small number of players. All these three make optimization more profitable, by reducing the expected reward from finding a block and increasing the potential gain of saving expenses.

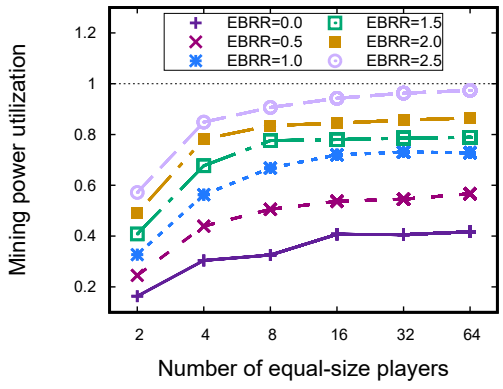
7.3.3 Mining Power Utilization. Equilibria with positive gap sizes negatively affect system security by reducing the amount of resources protecting the system. Recall that in proof of work systems, the security of the system relies on the honest miners' mining power. When less mining power takes part, the system becomes less resilient to attacks, as now attackers require less resources.



(a) *low_op*



(b) *med_op*



(c) *high_op*

Figure 8: Mining power utilization.

The mining power utilization [31] is the ratio of mining power that effectively secures the blockchain out of all mining power in the hands of well-behaved miners. If the mining power utilization is smaller than one, then an attacker can perform a 51% attack with less than 51% of the mining power, and selfish mining becomes easier to achieve. Figure 8 show the mining power utilization in various scenarios. In Figure 8a, when the *low_op* setting applies, all players choose start time zero, and the mining power utilization is not affected. In Figures 8b and 8c, when the *med_op* and the *high_op* settings apply, players use mining gaps, leading to decrease in the mining power utilization. Note that at the most extreme scenario of two players, high opex and low base reward, the mining power utilization drops to about 10%.

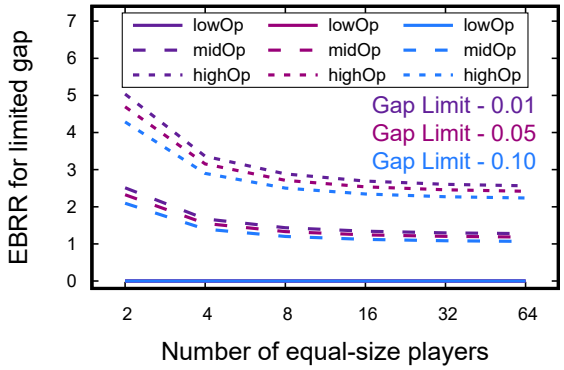


Figure 9: Minimal *EBRR* for a limited gap.

7.3.4 EBRR for a Limited Mining Gap. We have seen the implications on security from mining gaps, therefore we explore the question of how to avoid such gaps. We find the minimal *EBRR* to limit the size of a mining gap. Assume we want to limit the start time of players at equilibrium to be a factor of x from the *Block_Interval*. Therefore, we look for the minimal *EBRR* value such that the start time at equilibrium will be less than $x \cdot Block\_Interval$.

We use binary search over a wide range of *EBRR* values and mark the lowest *EBRR* that bounds the gap by $x \cdot Block\_Interval$, for three different factors $\forall x \in \{0.01, 0.05.0.1\}$. We repeat this experiment for various number of players as well as different opex and capex values. We present the results in Figure 9.

As expected, for the *low_op* setting, the start time at equilibrium is zero as mining is free, and even $EBRR = 0$ suffices. As optimization becomes more profitable due to the aforementioned reasons, higher *EBRR* values are required to limit start times at the equilibrium. When the *EBRR* is higher, the total reward from finding the block is higher. Hence, it incentivizes miners to prefer increasing their chances of winning the reward over decreasing their expenses, which ultimately leads to a limited mining gap.

Note that even as the number of players grows, the curves converge to a fixed value of *EBRR*. We deduce that even in a system with many small miners, a gap still forms in the presence of opex.

7.4 Case Study: Bitcoin

We now make an educated estimation to when Bitcoin becomes prone to the undesired effects of mining gaps. There are many operational cryptocurrency systems, all vary in minting, fees, market cap, and expenses. Given such parameters for any cryptocurrency, a similar estimation can be performed using our model. We present a case study of Bitcoin.

We consider the popular mining rig *Antminer s9* [9] with an estimated life expectancy of one year. Its required power is about 1.3kW and average cost about $1000. Electricity cost is about $0.1/kWh [64]. In one year the electricity expenses of one miner sum up to $876, which means the system falls in the area of *med_op*.

In Bitcoin today there are 7 mining pools [11] controlling about 85% of the mining power, while the rest is divided among

many smaller mining pools. Although they vary in size, we approximate that situation by assuming 8 equal-size miners.

Using results of Section 7.3.4, we deduce that $EBRR \approx 1$ is required to maintain a small gap. Currently, the rewards from minting and fees are ₿12.5 and about ₿1, respectively. Therefore currently $EBRR \approx 12.5$, so gaps are not profitable. However, in about ten years the minting reward drop will drop to about 1, which means $EBRR \approx 1$ and the system will be in a state where gaps are profitable.

The ten-year estimate is an optimistic one, as it assumes the reward from fees does not increase. Different mining hardware, change in electricity costs, and changes in the currency market, all might lead to different results. We emphasize that our estimation does not consider incentives external to our model resulting in seemingly altruistic behavior [4].

8 CONCLUSION

We defined and analyzed the gap game exploring how mining gaps form as a function of subsidy and fees, capex and opex. We showed that once fees become significant gaps form, though not uniformly as previous believed, and their effect on blockchain security is significant, decreasing mining utilization by up to 90% in extreme scenario, and leading to centralization incentives.

This means that base rewards are critical for system security, and should be achieved either by subsidy, fee backlogs, or alternative fee schemes [45, 55]. We show that $EBRR \approx 6$ is sufficient to avoid mining gaps in presented scenarios; we expect Bitcoin to drop below this threshold within a decade.

Establishing that gaps occur is an early and important step in the security analysis of cryptocurrency systems. This work is a step in that direction, demonstrating that gap analysis is critical for a more complete security analysis of blockchains. Such analysis can be used to inform the design of future and current cryptocurrencies.

Acknowledgements

This research was supported by the Israel Science Foundation (grant No. 1641/18), the Technion Hiroshi Fujiwara cyber-security research center, and the Israel cyber bureau.

REFERENCES

[1] Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Alexander Spiegelman. 2016. Solida: A Blockchain Protocol Based on Reconfigurable Byzantine Consensus. *arXiv preprint arXiv:1612.02916* (2016).
[2] Joël Alwen, Jeremiah Blocki, and Krzysztof Pietrzak. 2018. Sustained space complexity. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 99–130.
[3] Moshe Babaioff, Shahar Dobzinski, Sigal Oren, and Aviv Zohar. 2012. On Bitcoin and red balloons. In *ACM Conference on Electronic Commerce*. Valencia, Spain, 56–73.
[4] Christian Badertscher, Juan Garay, Ueli Maurer, Daniel Tschudi, and Vassilis Zikas. 2018. But why does it work? A rational protocol design treatment of bitcoin. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 34–65.
[5] Bruno Biais, Christophe Bisiere, Matthieu Bouvard, and Catherine Casamatta. 2018. The blockchain folk theorem. *ssrn id 3108601* (2018).
[6] Will Binns. 2018. How do I calculate my transaction fee? (2018). https://support.earn.com/digital-currency/bitcoin-transactions-and-fees/how-do-i-calculate-my-transaction-fee
[7] Bitcoin Cash community. 2018. Bitcoin Cash Site. https://www.bitcoincash.org/, retrieved May. 2018. (2018).
[8] BitcoinWiki. 2018. Controlled supply. https://en.bitcoin.it/wiki/Controlled_supply, retrieved May. 2018. (2018).
[9] Bitmain.com. 2018. Antminer S9i. https://shop.bitmain.com/product/detail?pid=00020180503144211733Dd3wi9Ez06A0, retrieved May. 2018. (2018).
[10] Blockchain.info. 2018. Bitcoin Market Capitalization. http://blockchain.info/charts/market-cap, retrieved Feb. 2018. (2018).
[11] Blockchain.info. 2018. Bitcoin Mining Pools. https://blockchain.info/pools, retrieved May. 2018. (2018).
[12] Blockchain.info. 2018. Mempool Transaction Count. https://blockchain.info/charts/mempool-count, retrieved Feb. 2018. (2018).
[13] Blockchain.info. 2018. Transaction Fees. https://blockchain.info/charts/transaction-fees, retrieved Feb. 2018. (2018).
[14] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten. 2015. Research perspectives on Bitcoin and second-generation cryptocurrencies. In *Symposium on Security and Privacy*. IEEE, San Jose, CA, USA.
[15] Ryan Browne. 2017. The cheapest and most expensive countries to mine bitcoin. (2017). https://www.cnbc.com/2018/02/15/the-cheapest-and-most-expensive-countries-to-mine-bitcoin.html
[16] Vitalik Buterin. 2013. A Next Generation Smart Contract & Decentralized Application Platform. https://www.ethereum.org/pdfs/EthereumWhitePaper.pdf/, retrieved Feb. 2015. (2013).
[17] Miles Carlsten, Harry Kalodner, S. Matthew Weinberg, and Arvind Narayanan. 2016. On the Instability of Bitcoin Without the Block Reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. ACM, New York, NY, USA, 154–167. https://doi.org/10.1145/2976749.2978408
[18] Coinmarketcap.com. 2018. Cryptocurrency Market Capitalizations. (2018). https://coinmarketcap.com/
[19] Ellery Davies. 2015. Why is there a limited amount of bitcoin available? (2015). https://www.quora.com/Why-is-there-a-limited-amount-of-bitcoin-available
[20] Wouter den Haan, Martin Ellison, Ethan Ilzetzki, Michael McMahon, and Ricardo Reis. 2017. Economists relaxed about Bitcoin: New CFM-CEPR expert survey on cryptocurrencies, the financial system, and economic policy. *VoxEU. org* 21 (2017).
[21] Digiconomist.net. 2017. A Deep Dive in a Real-World Bitcoin Mine. (2017). https://digiconomist.net/deep-dive-real-world-bitcoin-mine
[22] Digiconomist.net. 2018. Bitcoin Energy Consumption Index. (2018). https://digiconomist.net/bitcoin-energy-consumption
[23] Digiconomist.net. 2018. Ethereum Energy Consumption Index. (2018). https://digiconomist.net/ethereum-energy-consumption
[24] Cynthia Dwork and Moni Naor. 1993. *Pricing via Processing or Combatting Junk Mail*. Springer Berlin Heidelberg, Berlin, Heidelberg, 139–147. https://doi.org/10.1007/3-540-48071-4_10
[25] Earn.com. 2018. Predicting Bitcoin Fees For Transactions. (2018). https://bitcoinfees.earn.com/
[26] Etherscan.io. 2018. Ether Supply Growth. https://etherscan.io/chart/ethersupply, retrieved Feb. 2018. (2018).
[27] Etherscan.io. 2018. Ether Transaction Fees. https://etherscan.io/chart/transactionfee, retrieved Feb. 2018. (2018).
[28] Etherscan.io. 2018. Pending Transactions. https://etherscan.io/chart/pendingtx, retrieved Feb. 2018. (2018).
[29] Ittay Eyal. 2015. The miner's dilemma. In *Security and Privacy (SP), 2015 IEEE Symposium on*. IEEE, 89–103.
[30] Ittay Eyal. 2015. The Miner's Dilemma. In *IEEE Symposium on Security and Privacy*. 89–103. https://doi.org/10.1109/SP.2015.13
[31] Ittay Eyal, Adem Efe Gencer, Emin Gun Sirer, and Robbert Van Renesse. 2016. Bitcoin-ng: A scalable blockchain protocol. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. USENIX Association, 45–59.
[32] Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert Van Renesse. 2016. Bitcoin-NG: A Scalable Blockchain Protocol.. In *NSDI*. 45–59.
[33] Ittay Eyal and Emin Gün Sirer. 2014. Majority is not Enough: Bitcoin Mining is Vulnerable. In *Financial Cryptography and Data Security*.
[34] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. 2015. The Bitcoin Backbone Protocol: Analysis and Applications. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. 281–310. https://doi.org/10.1007/978-3-662-46803-6_10
[35] Arthur Gervais, Ghassan O Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. 2016. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 3–16.
[36] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*. ACM, 51–68.
[37] Gur Huberman, Jacob D Leshno, and Ciamac C Moallemi. 2017. Monopoly without a monopolist: An economic analysis of the bitcoin payment system. *ssrn id 3025604* (2017).
[38] Intel. 2018. Sawtooth-core source code (validator). (2018). https://github.com/hyperledger/sawtooth-core/tree/0-7/validator/sawtooth_validator/consensus/poet1 [Online; accessed May-2018].

[39] Markus Jakobsson and Ari Juels. 1999. Proofs of work and bread pudding protocols. In *Secure Information Networks*. Springer, 258–272.

[40] Ghassan Karame, Elli Androulaki, and Srdjan Capkun. 2012. Two Bitcoins at the Price of One? Double-Spending Attacks on Fast Payments in Bitcoin. *IACR Cryptology ePrint Archive* 2012, 248 (2012).

[41] Sudhir Khatwani. 2018. Ethereum: Ether, Ether Gas, Gas Limit, Gas Price and Fees. (2018). https://coinsutra.com/ethereum-gas-limit-gas-price-fees/

[42] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. 2017. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*. Springer, 357–388.

[43] Joshua A Kroll, Ian C Davey, and Edward W Felten. 2013. The Economics of Bitcoin Mining or, Bitcoin in the Presence of Adversaries. In *Workshop on the Economics of Information Security*.

[44] Yujin Kwon, Dohyun Kim, Yunmok Son, Eugene Vasserman, and Yongdae Kim. 2017. Be Selfish and Avoid Dilemmas: Fork After Withholding (FAW) Attacks on Bitcoin. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 195–209.

[45] Ron Lavi, Or Sattath, and Aviv Zohar. 2017. Redesigning Bitcoin's fee market. *arXiv preprint arXiv:1709.08881* (2017).

[46] Litecoin Project. [n. d.]. Litecoin, open source P2P digital currency. https://litecoin.org, retrieved Nov. 2014. ([n. d.]).

[47] Samara Malkin. 2018. Cheapest Places Mining Bitcoin. (2018). https://cryptocurrencynews.com/daily-news/cryptocurrency-mining/cheapest-places-mining-bitcoin/

[48] Andrew Miller, Elaine Shi, Ari Juels, Bryan Parno, and Jonathan Katz. 2014. Permacoin: Repurposing Bitcoin Work for Data Preservation. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE, San Jose, CA, USA. http://research.microsoft.com/apps/pubs/default.aspx?id=217984

[49] Malte Möser and Rainer Böhme. 2015. Trends, Tips, Tolls: A Longitudinal Study of Bitcoin Transaction Fees. In *Financial Cryptography and Data Security*, Michael Brenner, Nicolas Christin, Benjamin Johnson, and Kurt Rohloff (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 19–33.

[50] Ujan Mukhopadhyay, Anthony Skjellum, Oluwakemi Hambolu, Jon Oakley, Lu Yu, and Richard Brooks. 2016. A brief survey of cryptocurrency systems. In *Privacy, Security and Trust (PST), 2016 14th Annual Conference on*. IEEE, 745–752.

[51] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. http://www.bitcoin.org/bitcoin.pdf. (2008).

[52] Kartik Nayak, Srijan Kumar, Andrew Miller, and Elaine Shi. 2015. Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack. *IACR Cryptology ePrint Archive* 2015 (2015), 796. http://eprint.iacr.org/2015/796

[53] Rafael Pass, Lior Seeman, and Abhi Shelat. 2017. Analysis of the Blockchain Protocol in Asynchronous Networks. In *Advances in Cryptology – EUROCRYPT 2017*, Jean-Sébastien Coron and Jesper Buus Nielsen (Eds.). Springer International Publishing, Cham, 643–673.

[54] Rafael Pass and Elaine Shi. 2016. Hybrid Consensus: Efficient Consensus in the Permissionless Model. Cryptology ePrint Archive, Report 2016/917. (2016). http://eprint.iacr.org/2016/917.

[55] Rafael Pass and Elaine Shi. 2017. Fruitchains: A fair blockchain. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*. ACM, 315–324.

[56] Rafael Pass and Elaine Shi. 2017. The sleepy model of consensus. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 380–409.

[57] Nathan Reiff. 2017. What Happens to Bitcoin After All 21 Million are Mined? (2017). https://www.investopedia.com/news/what-happens-bitcoin-after-all-21-million-are-mined/

[58] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. 2016. Optimal Selfish Mining Strategies in Bitcoin. In *Financial Cryptography and Data Security*.

[59] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. 2014. Zerocash: Decentralized anonymous payments from bitcoin. In *Security and Privacy (SP), 2014 IEEE Symposium on*. IEEE, 459–474.

[60] Okke Schrijvers, Joseph Bonneau, Dan Boneh, and Tim Roughgarden. 2016. Incentive compatibility of bitcoin mining pool reward functions. In *International Conference on Financial Cryptography and Data Security*. Springer, 477–498.

[61] Bayu Adhi Tama, Bruno Joachim Kweka, Youngho Park, and Kyung-Hyune Rhee. 2017. A critical review of blockchain and its current applications. In *Electrical Engineering and Computer Science (ICECOS), 2017 International Conference on*. IEEE, 109–113.

[62] Jordan Tuwiner. 2017. Bitcoin Mining Hardware. (2017). https://www.buybitcoinworldwide.com/mining/hardware/

[63] Cindy Wang. 2017. A Visit to a Bitcoin Mining Farm in Sichuan, China Reveals Troubles Beyond Regulation. (2017). https://news.bitcoin.com/a-visit-to-a-bitcoin-mining-farm-in-sichuan-china-reveals-troubles-beyond-regulation/

[64] wikipedia.com. [n. d.]. Electricity Pricing. https://en.wikipedia.org/wiki/Electricity_pricing, retrieved May. 2018. ([n. d.]).

[65] Fan Zhang, Ittay Eyal, Robert Escriva, Ari Juels, and Robbert Van Renesse. 2017. REM: Resource-Efficient Mining for Blockchains. *IACR Cryptology ePrint Archive* 2017 (2017), 179.

A VALID PDF PROOF

Denote $R = \left\{(s_1, s_2), \left(s_2, s_3\right), \ldots, \left(s_{k-1}, s_k\right), \left(s_k, \infty\right)\right\}$ the list of all start time intervals. Note that for any interval $(s_l, s_{l+1}) \in R$, no rigs are turned on, meaning $active(t)$ does not change.

Our goal is to show that $\int_{-\infty}^{\infty} f_B(t; \bar{s}, \mu(\bar{s}))dt = 1$. We begin by taking notice that at time s_{l+1} rig $l+1$ becomes active, resulting in $active\left(s_{l+1}\right) \setminus active\left(s_l\right) = \left\{s_{l+1}\right\}$. We get that for any $l \in \{1, 2, \ldots, k\}$:

$$\begin{aligned}
&\sum_{j \in active\left(s_{l+1}\right)} (s_{l+1} - s_j) = \\
&\sum_{j \in active\left(s_l\right)} (s_{l+1} - s_j) + s_{l+1} - s_{l+1} = \\
&\sum_{j \in active\left(s_l\right)} (s_{l+1} - s_j) + s_{l+1} - s_{l+1} = \\
&\sum_{j \in active\left(s_l\right)} (s_{l+1} - s_j) \quad .
\end{aligned} \tag{7}$$

We are now ready to present the full verification process, which is detailed in Equation 8.

B DIFFICULTY PARAMETER VALUE AT EQUILIBRIUM

In this section we show how to find the value of $\mu(\bar{s})$ at equilibrium. The system's protocol dictates that the mean block creation time interval is *Block_Interval*. This is done by setting the value of $\mu(\bar{s})$. At equilibrium, the mean block creation time interval is the expected time to find a block $E[B]$. Hence at equilibrium $E[B] = Block\_Interval$. We use this equality to express a constraint on the system at equilibrium. A known result in probability theory applies here — since B is a non-negative random variable, its expected value $E[B] = \int_0^{\infty} \Pr(t \leq B)dt$. This results in

$$\begin{aligned}
Block_Interval &= \\
&= E[B] \\
&= \int_{-\infty}^{\infty} \left(t f_B(t; \bar{s}, \mu(\bar{s}))\right) dt \\
&= \int_0^{\infty} \left(1 - F_B(t; \bar{s}, \mu(\bar{s}))\right) dt \\
&= \int_0^{\infty} \Pr(t \leq B)dt \, .
\end{aligned}$$

Based on Equation 1, we know that $\Pr(t \leq B)$ is equal to $\exp\left(-\mu(\bar{s}) \sum_{j \in active(t)} (t - s_j)\right)$. We again use R notion that was defined in Appendix A Note that for any interval $(s_l, s_{l+1}) \in R$, no rigs are turned on, meaning $active(t)$ does not change. In Equation 9 we derive an expression for $E[B]$ as a function of $\mu(\bar{s})$ and $\bar{s}$. Note that this is an implicit function with respect to $\mu(\bar{s})$.

$$\int_{-\infty}^{\infty} f_B(t;\bar{s},\mu(\bar{s}))dt =$$

$$\sum_{(s_l,s_{l+1})\in R}\left[\int_{s_l}^{s_{l+1}} f_B(t;\bar{s},\mu(\bar{s}))dt\right] =$$

$$\sum_{(s_l,s_{l+1})\in R}\left[\int_{s_l}^{s_{l+1}} \mu(\bar{s})\cdot|active(t)|\cdot\exp\left(-\mu(\bar{s})\cdot\sum_{j\in active(t)}(t-s_j)\right)\right] =$$

$$\sum_{(s_l,s_{l+1})\in R}\left[\int_{s_l}^{s_{l+1}} \mu(\bar{s})\cdot\left|active\left(s_l\right)\right|\cdot\exp\left(-\mu(\bar{s})\cdot\sum_{j\in active\left(s_l\right)}(t-s_j)\right)\right] =$$

$$\sum_{(s_l,s_{l+1})\in R}\left[\left(-\exp\left(-\mu(\bar{s})\cdot\sum_{j\in active\left(s_l\right)}(t-s_j)\right)\Bigg|_{s_l}^{s_{l+1}}\right)\right] =$$

$$\sum_{(s_l,s_{l+1})\in R}\left[\exp\left(-\mu(\bar{s})\cdot\sum_{j\in active\left(s_l\right)}(s_l-s_j)\right)-\exp\left(-\mu(\bar{s})\cdot\sum_{j\in active\left(s_l\right)}(s_{l+1}-s_j)\right)\right] =$$

Using Equation 7 we get the last expression is a telescopic sum. Substituting in the relevant expressions yields

$$\exp\left(-\mu(\bar{s})\cdot\sum_{j\in active(s_1)}(s_1-s_j)\right)-\exp\left(-\mu(\bar{s})\cdot\sum_{j\in active\left(s_k\right)}(\infty-s_j)\right) =$$

$$\exp(0)-\exp(-\infty) = 1-0 = 1$$

as required. (8)

(a) Verifying the PDF.

$$Block\_Interval = E[B] = \int_0^{\infty}\Pr(t\le B)dt =$$

$$\int_0^{s_1} 1dt+\sum_{(s_l,s_{l+1})\in R}\left[\int_{s_l}^{s_{l+1}}\exp\left(-\mu(\bar{s})\sum_{j\in active(t)}(t-s_j)\right)dt\right] =$$

$$s_1+\sum_{(s_l,s_{l+1})\in R}\left[\int_{s_l}^{s_{l+1}}\exp\left(-\mu(\bar{s})\sum_{j\in active(t)}(t-s_j)\right)dt\right] =$$

$$s_1+\sum_{(s_l,s_{l+1})\in R}\left[\int_{s_l}^{s_{l+1}}\exp\left(-\mu(\bar{s})\cdot\sum_{j\in active\left(s_l\right)}(t-s_j)\right)\right] = \quad (9)$$

$$s_1+\sum_{(s_l,s_{l+1})\in R}\left[\left(-\frac{1}{\mu(\bar{s})\cdot\left|active\left(s_l\right)\right|}\exp\left(-\mu(\bar{s})\cdot\sum_{j\in active\left(s_l\right)}(t-s_j)\right)\Bigg|_{s_l}^{s_{l+1}}\right)\right] =$$

$$s_1+\sum_{(s_l,s_{l+1})\in R}\frac{\exp\left(-\mu(\bar{s})\cdot\sum_{j\in active\left(s_l\right)}(s_l-s_j)\right)-\exp\left(-\mu(\bar{s})\cdot\sum_{j\in active\left(s_l\right)}(s_{l+1}-s_j)\right)}{\mu(\bar{s})\cdot\left|active\left(s_l\right)\right|}.$$

(b) Difficulty parameter $\mu(\bar{s})$ at equilbria constraint.