Majority is Not Required: A Rational Analysis of the Private Double-Spend Attack from a Sub-Majority Adversary

Yanni Georghiades
ECE Department
UT Austin
Austin, United States
yanni.georghiades@utexas.edu

Rajesh Mishra
ECE Department
UT Austin
Austin, United States
rajeshkmishra@utexas.edu

Karl Kreder
CTO
Dominant Strategies Inc.
Austin, United States
karl@dominantstrategies.io

Sriram Vishwanath
ECE Department
UT Austin
Austin, United States
sriram@utexas.edu

***Abstract*—In Nakamoto-style Proof-of-Work cryptocurrencies, miners compete with each other to append blocks to a blockchain. Typically, the winning miner receives a reward in the form of a fixed block reward and a variable amount of transaction fees. In order to determine the transaction fees, miners are allowed to choose which transactions to reference with their block, and a common strategy for selecting transactions is to simply choose those with the highest fees. This strategy can be problematic if these transactions originate from an adversary with substantial (but less than 50%) computational power, as high-value transactions can present an incentive for the adversary to attempt a double-spend attack even if they are not guaranteed to succeed. In order to discourage double-spend attacks, we argue that miners should limit the amount of transaction value they include in a block.**

To this end, we model cryptocurrency mining as a mean-field game in which we augment the standard mining reward function to simulate the presence of a rational, double-spending adversary. We design and implement an algorithm which characterizes the behavior of miners at equilibrium, and we show that miners who use the adversary-aware reward function accumulate more wealth than those who do not. We show that when the adversary has a significant fraction of the total computational power, the optimal strategy for honest miners is to limit the amount of value which can be transacted in a given block. When miners follow this strategy, there is no incentive for the adversary to attack. In particular, the adversary's incentive to attack is eliminated without requiring any non-mining user or merchant to modify their behavior, either by waiting for extra block confirmations or declining high-value transactions.

***Index Terms*—Proof of Work, Mining, Mean Field, Incentive Design**

I. Introduction

This paper characterizes the behavior of cryptocurrency mining in the presence of a rational adversary who is actively attempting adouble-spend attack. We focus specifically on Nakamoto-style Proof-of-Work cryptocurrencies, in which miners compete to add blocks to the blockchain. A miner who successfully appends a block to the blockchain wins rewards in the form of a constant block reward and a set of transaction fees that correspond to the transactions the miner references in the block.

Commonly, miners seek to maximize the transaction fees they receive by greedily selecting the transactions with the highest fees. However, this strategy is problematic in the presence of a rational adversary with the means to execute a double-spend attack, as the amount of economic value being transferred by a transaction (henceforth referred to as the "transaction value") can significantly impact the adversary's expected profit in attempting such an attack. If high-value transactions which originate from the adversary have the highest transaction fees, then miners can unintentionally render the adversary's double-spend attack profitable by including these transactions within their blocks. To combat this problem, we present a reward function for the miners which accountss for the transaction value in the block and the manner in which it impacts the adversary's incentive to attack. Using this new reward function, we model the mining problem as a mean-field game. We simulate this game in order to compare the behavior and performance of miners utilizing the standard reward function (and greedily selecting the highest fee transactions) with that of miners utilizing our adversary-aware reward function.

A. Motivation

Studying the incentives a rational miner has to attempt a double-spend attack and designing protocols which are resilient to such an attack is critical to the long-term utility of Proof-of-Work cryptocurrencies. However, some would refute such a position, claiming that no rational adversary would be incentivized to execute such an attack because either 1) Bitcoin receives so much mining power that it is not feasible for an adversary to succeed in an attack; or 2) any such adversary must have a significant investment in the cryptocurrency, and attacking the currency would devalue their investment to such an extent that any attack would be ultimately unprofitable.

The first argument is specific to Bitcoin and does not apply to all Proof-of-Work cryptocurrencies, particularly newer and smaller cryptocurrencies which may be easier targets due to

Sriram Vishwanath is also an advisor to Dominant Strategies Inc.

their relatively low network mining power. The second argument, although seemingly persuasive, makes implicit assumptions on the adversary's financial model for the attack which do not hold in all cases. For example, [1] argues that it would be straightforward for attackers to temporarily borrow mining power from other miners through bribery, vastly reducing the investment required to prepare for the attack. Even simpler, adversaries may rent mining power directly from a third party provider such as nicehash.com; indeed, in 2020 this strategy was used to great effect in attacks against Bitcoin Gold and Ethereum Classic, resulting in the theft of assets valued in the millions of dollars [2], [3]. Moreover, a decline in currency value could even be advantageous to a prepared adversary who holds "short"-positions on the value of that currency. Overall, relying on an adversary's aversion to price volatility to refrain from attacks has a limited rational basis, and more careful analysis is needed to understand both the impact and the associated mitigation strategies of such attacks.

B. The Relationship Between Transaction Values and Transaction Fees

In this work, we show that the cost an adversary incurs in attempting a double-spend attack increases with the total network mining power. We also observe that the total network mining power increases as mining rewards increase because miners naturally seek to claim a share of the extra rewards. We argue that, in order to effectively discourage an adversary from attempting a double-spend attack, high-value transactions are attractive targets for adversarial attack and should therefore pay higher fees than low-value transactions. Intuitively, high-value transactions must be associated with a greater network mining power than low-value transactions to sufficiently discourage adversarial attack, so there is conceptual alignment in the notion that senders of high-value transactions pay higher fees to achieve the necessary level of system security to safeguard them.

However, in typical cryptocurrencies, transaction fees are unrelated to transaction value, as fees are typically market-driven based on the demand for transaction settlement. To remedy this issue, we propose a reward model in which transaction fees increase monotonically with transaction values. Specifically, we adopt a simple model in which the transaction fee is a small percentage of the transaction value. Although such a fee model may or may not be optimal, it serves as a good starting point for analysis. We believe the study of transaction fees as they relate to transaction values is critical to the long-term security of Proof-of-Work cryptocurrencies, but we defer further analysis of this topic to future work.

C. Mean-Field Games

Mean-field games (MFGs) were introduced in [4], [5] as an efficient mechanism to characterize the interdependent behavior of agents in a system with a large number of participants. They involve aggregating the behavior of all agents into an associated *mean-field state* that is then used by individual agents to make rational decisions. Cryptocurrency mining can be thought of as a system where the decisions of each miner are affected by the behavior of all of the other miners. Although miners may not know the precise mining power of each of their peers, the total mining power employed by the network to mine each block can be approximated by observing block times. Consequently, we model the average mining power employed by each miner as a mean-field state, allowing miners to approximate the network mining power without knowledge of any individual miner's contribution to that amount. In Section III, we present a mean-field game model for cryptocurrency mining in the presence of a rational adversary.

In Section IV, we provide an algorithm which simulates our game and can be used to determine the MFG equilibrium policies for miners. We implement this algorithm in Python3 and use it to compare the mining performance for miners that do account for the adversary in their reward function and those that do not. We find that if miners use an adversary-aware reward function, the optimal equilibrium policies render double-spend attacks unprofitable for the adversary. On the other hand, if miners ignore the threat of adversarial attack and select high transaction values, they experience a significant reduction in profitability when the adversary does attack.

D. Our Contributions

In summary, the key contributions of this paper are:

- We provide a framework that models the costs and rewards of a rational adversary who has the ability to execute double-spend attacks.
- We characterize the mining game in the presence of a rational adversary as a mean-field game and define the associated reward functions for mining.
- We provide detailed pseudo-code for an algorithm which solves for the equilibrium strategies of agents playing this game.
- We show that by using our adversary-aware reward function, miners can ensure that a double-spend attack is unprofitable for a rational adversary.

II. Related Work

There is a considerable body of work in understanding blockchains using mechanism design and game theory. For a game theory perspective, we have a variety of different approaches [6]–[16]. Orthogonal to our analysis of transaction fees, there is a rich body of literature on using mechanism design to implement incentive compatible and collusion-resistant transaction fee mechanisms [17]–[21]. Below we discuss work more closely related to ours.

A. Game Theoretic Analysis of Double-Spend Attacks

In the original Bitcoin whitepaper [22], Nakamoto first estimates the probability of a successful double-spend attack with an infinite time horizon. Nakamoto models the number of blocks the attacker can mine in a given time period as a Poisson distribution and then applies a gambler's ruin argument to find the probability that the attacker can ever catch up to or overtake the honest chain. Importantly, Nakamoto only examines the probability of success and does not model the economic cost of the attack. Rosenfeld [23] later enhances the accuracy Nakamoto's analysis by modeling the number of blocks mined by the attacker with a negative binomial distribution. Rosenfeld briefly examines the economics of the double-spend attack, suggesting heuristics that merchants can use to determine whether or not it is safe to accept a payment.

Bissias *et al.* [24] and Sompolinsky *et al.* [25] further examine double-spend attacks from an economic perspective. They model the profitability of the double-spend attack, and they also consider an adversary that can leverage additional attack vectors to enhance the effectiveness of their attack. Recently, Jang *et al.* [26] propose a more finely-grained model of the attacker's expected profits using a Markov decision process, allowing the attacker to re-evaluate at each time step whether or not it is profitable to continue the attack.

While most of these works examine the double-spend attack in terms of the profitability of the adversary, the only defense mechanisms suggested are to be carried out by the merchant. The merchant can either decline large transactions or require extra block confirmations before the goods exchange hands. In contrast, we consider a scenario in which the onus of defense is placed on the miners, allowing the merchant to remain agnostic to the inner workings of the blockchain. This aligns with the incentives of the "honest" miners (which we still presume to be economically motivated), as double-spend attacks are costly for miners who lose out on block rewards as a result of the attack. We show that it is in the miners' best interests to implement our suggested defense of limiting the transaction value made vulnerable to attack. In so doing, we absolve merchants from the responsibility of defending the system against attacks and place that responsibility in the hands of miners who are far better equipped for the task.

B. The Rational Protocol Design Framework

A separate line of work studying this problem has its foundations in the Rational Protocol Design (RPD) framework proposed by Garay *et al.* [27] as a means to analyze cryptographic protocols under the assumption that participants are rational rather than honest or corrupt. The RPD framework considers a two-party game between the protocol designer and an attacker in which the protocol designer specifies a protocol and the attacker specifies a polynomial-time attack strategy to subvert the protocol. The attacker gains utility by violating security guarantees of the protocol but must pay to corrupt protocol participants, and the protocol is secure if the adversary cannot propose any attack strategy which yields positive utility.

Badertscher *et al.* [28] later extend the RPD framework in order to study Bitcoin from a rational perspective. They find that, even if the majority of miners are not assumed to be honest, Bitcoin is incentive compatible (meaning all parties follow the protocol) if the transaction fees available to miners do not vary significantly between blocks. This result does not apply to our model, as their utility functions explicitly ignore the transaction value (which is a focal point in our analysis).

More recently, Badertscher *et al.* [29] expand on their previous work to study the problem of a 51% double-spend attack. To this end, they devise a utility function which captures the incentives of a double-spending attacker with the ability to take over a majority share of the network mining power for a period of time. Using this utility function, they characterize a range of attack payoffs that the attacker would have to receive in order for the attack to be profitable. Finally, they show that by increasing the number of block confirmations required for a transaction to be considered finalized, the incentive for a rational adversary to attempt a 51% double-spend attack can be eliminated. Out of all works discussed, our model is most similar to theirs in that both analyze the incentives and costs for an attacker to attempt a private double-spend attack and both suggest a mechanism by which that attack can be mitigated. However, their model assumes that the attacker is able to acquire a majority share of mining power for a period of time long enough to guarantee that the attack is successful, whereas we consider the scenario in which the attacker does not control a majority share and is not guaranteed to succeed. The structure of our games are also dramatically different, as we define a dynamic mean-field game in which each round corresponds to a block being mined, while they consider a Stackelberg game of two rounds (the protocol designer proposes a protocol and then the adversary proposes an attack). As a result, our reward functions and analysis methods vary significantly from theirs, where we define exact reward functions to simulate the attacker's decision on whether or not to attack and they provide bounds on their utility functions under which the attacker is incentivized to attack. Finally, our proposed defense mechanism is a limitation on the value which can be transacted in a given block, while theirs is a modification to the number of block confirmations required for a transaction to be finalized. Each of these mechanisms has practical implications which might make one more appealing than the other depending on application requirements, but we think it likely that both would be effective at mitigating attacks in either model.

C. Mean-Field Games

Mean-field games have proven effective in modeling large population games with interdependent agents. Recently, [30] model the cryptocurrency mining problem as a mean-field game and derive equilibrium policies in terms of the mining power and wealth of the miners. They provide both an analytical framework and numerical algorithm to derive equilibrium policies for miners with infinite and finite wealth, respectively. We build upon their model to include an adversary and study how the behavior of miners is affected as a result. Our proposed algorithm is also derived from the work by authors in [31] where they solve for equilibrium policies in mean-field games through a sequential decomposition algorithm.

Separately, Bertucci *et al.* [32] study Bitcoin mining as a mean-field game. Their analysis differs from that of [30] in that it focuses on hashrate stability in response to changing environmental variables such as technological progress, electricity costs, and currency conversion rates. While interesting, we do not consider such variables in our model.

III. Model

We model the mining game as a finite, sequential mean-field game of m agents. Our model includes a static adversary that controls a β-fraction of the network hash power at all times. We refer to non-adversarial miners as either "honest miners" or "miners." Honest miners behave rationally within their action space, but that action space does not include the ability to attempt a double-spend attack of their own. The adversary, on the other hand, does not participate in mining unless actively executing a double-spend attack[1]. In other words, there is a clear delineation in allowable actions between the adversary (who does not mine unless attacking) and the honest miners (who do not attack).

At each time step, we define the mean-field variable $\overline{\alpha}_t$ as the average mining power used by all agents in that time step. Miners use $\overline{\alpha}_t$ in their decision-making process as an approximation of the competition they expect from the other miners.

For ease of analysis, we assume a synchronous network model for mining with no communication delays. Mining is performed in a series of rounds, where there is exactly 1 block mined per round. Miners have limited mining resources (referred to as "wealth") which carry forward as state between time steps. The amount of wealth a miner has at a particular round limits the amount of mining power they can contribute to that block, and after each round all wealths are updated in accordance with the actions taken and the expected rewards they would receive as a result. In a given round, the adversary (if they are attempting an attack) mines the block with probability β and some honest miner mines the block with probability $1 - \beta$.

A. The Adversary

We consider a rational, myopic, risk-neutral adversary. This means that the adversary's sole objective is to maximize their expected reward at each time step in the game without consideration of future rewards. The adversary makes only one choice at each time step: they can attempt a private double-spend attack against the system or do nothing. The adversary only attacks if their expected reward from attempting the attack is greater than their expected cost[2].

1) The Double-Spend Attack: For our purposes, a private double-spend attack is performed as follows:

- The adversary submits a transaction to the blockchain in payment for some asset X which is external to the system (for example, X could be a physical asset such as a yacht, or X could be a digital asset such as a token from another cryptocurrency). Privately, the adversary begins mining an alternate private chain in which they instead send that payment to an address that they control.
- After k confirmation blocks are published to the public blockchain, the adversary receives the asset X.
- The adversary publishes their private chain (if it is longer than the public chain), retaining X and also regaining control of the funds they originally used to purchase X.

The adversary is successful in their attack if and only if they are able to mine $k + 1$ blocks before the honest miners are able to mine $k + 1$ blocks. In other words, the adversary makes a single decision in each time step of whether or not to attempt an attack, and if they do attempt the attack then they carry it out precisely until either the adversary or the honest miners reach $k + 1$ blocks. Although of significant interest, we defer the study of more complex adversarial strategies to future work.

In terms of the amount of transaction value made vulnerable to attack, we consider a worst-case scenario in which *all* of the transactions contained within a block can be simultaneously double-spent by a single adversary. While this may not always be achievable, a sophisticated adversary could ostensibly execute such an attack with foreknowledge of the honest miners' transaction selection strategies.

The adversary's decision on whether or not to attack is represented by the function A, which evaluates to 1 if the expected profit of an attack is positive and 0 otherwise.

$$A(T, \overline{\alpha}) = \begin{cases} 1 & \text{if } R_{adv}(T, \overline{\alpha}) > 0 \\ 0 & \text{else} \end{cases} \tag{1}$$

In order to characterize the adversary's expected attack cost and probability of success, $C(\overline{\alpha}, \beta)$ and $P(\beta)$, we model the double-spend attack at a particular block as the 2-D Markov chain depicted in Figure 1. Each state is a tuple (B_H, B_A), where B_H is the number of blocks the honest agents have mined and B_A is the number of blocks the adversary has mined. From an initial state (b_H, b_A), the transition to $(b_H, b_A + 1)$ occurs with probability β and the transition to $(b_H + 1, b_A)$ occurs with probability $1 - \beta$. Then $P(\beta)$ is defined simply as

$$P(\beta) = \sum_{b_H=0}^{k} \beta Pr[(b_H, k)], \tag{2}$$

where $\beta Pr[(b_H, k)]$ is the probability of reaching state (b_H, k) and then transitioning into state $(b_H, k + 1)$. In other words, $P(\beta)$ is the probability that an adversary with a β fraction of the network mining power achieves $k + 1$ blocks while the honest agents have achieved at most k blocks.

We define $C(\overline{\alpha}, \beta)$ similarly as

$$\begin{aligned} C(\overline{\alpha}, \beta) = &\beta m \overline{\alpha} c\big((2k + 1)(1 - P(\beta)) + \\ &\sum_{b_H=0}^{k} \beta Pr[(b_H, k)](k + 1 + b_H)\big), \end{aligned} \tag{3}$$

[1] see Section VI for additional discussion on this assumption

[2] In order to smooth the reward function, in our simulation there is technically a negligible but non-zero probability that the adversary attacks in spite of a negative expected profit. This does not impact the results.

where $\beta m\overline{\alpha}c$ is the mining cost the adversary pays per time step ($m\overline{\alpha}$ is the total network mining power and the adversary controls a β fraction of it), and the remainder of the expression is the expected number of blocks the adversary attempts to mine as they execute the attack.

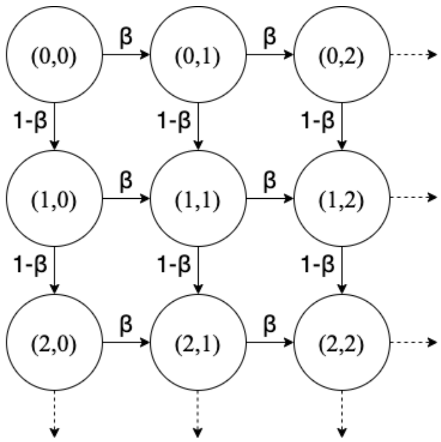

Figure 1. The 2-D Markov chain used to approximate the probability that the adversary can succeed in a double-spend attack and the expected number of blocks that the attack takes. States are represented by the tuple (B_H, B_A), where B_H is the number of blocks mined by the honest miners since the attack began and B_A is the number of blocks mined by the adversary. B_A is incremented with probability β, and B_H is incremented with probability $(1-\beta)$.

2) The Adversary's Reward Function: Recall that T is the amount of value being transacted within a particular block. The adversary's expected reward when attempting a double-spend attack is

$$R_{adv}(T, \overline{\alpha}) = P(\beta)\big((k+1)b + T + f(T)\big) - C(\overline{\alpha}, \beta), \quad (4)$$

where $(k+1)b$ corresponds to the $k+1$ block rewards the adversary wins, and T and $f(T)$ are the transaction value and transaction fees, respectively, that they recuperate as a result of the double-spend attack.

Note that because the adversary has a fixed strategy and does not carry state forward throughout the game (they are assumed to have a β fraction of the mining power at their disposal at all times), they can more or less be thought of as a part of the environment.

B. Honest Miners

Miners are assumed to be rational[3], risk-neutral, and forward-thinking. At each time step, miners select the actions that maximize the sum of their expected reward at that time step and the future rewards they expect to receive over the remainder of the game.

Honest miners choose two actions: (1) the amount of mining power α that they contribute towards mining a block, and (2) the total amount of transaction value T that they choose to reference in the block, where the transaction fees they will receive upon mining a block are determined by a fee function $f(T)$. We abstract T and $f(T)$ in this way because attempting to simulate transaction pools and more complex transaction selection mechanisms is needlessly cumbersome. Notably, we constrain the action space of each miner such that the miners cannot select an α that exceeds their total wealth, as no miner can have infinite wealth. We define a system parameter T_{max} to be the maximum total transaction value that can be referenced by a block, and we assume that this value is always available to miners (i.e., the amount of available transaction value does not change between rounds).

[3]Honest miners are rational in the sense that they select the best action within their action space. We leave examination of a model in which miners can "switch sides" and attempt a double-spend attack of their own to future work.

1) Reward Function: We define two reward functions for the honest miners: the "naive" reward in which miners are not aware of or not responsive to the threat of adversarial attack, and the actual reward which does account for the threat of attack. All reward functions in this work are expectations taken over the probability of successfully mining a block.

We first define the probability that the miner wins a block and receives a reward as

$$\Gamma_{naive}(\overline{\alpha}, \alpha, T) = \frac{\alpha}{\alpha + m\overline{\alpha}}. \quad (5)$$

Recall that α is the amount of mining power a miner decides to contribute to mining a particular round, so $\alpha + m\overline{\alpha}$ is the miner's best estimate on the total amount of mining power contributed by the rest of the network.

Then the naive reward for honest miners is

$$R_{naive}(\overline{\alpha}, \alpha, T) = \Gamma_{naive}(\overline{\alpha}, \alpha, T)\big(b + f(T)\big) - \alpha c, \quad (6)$$

where b is the constant block reward and c is the mining cost per unit of mining power. Combined with transaction fees, $b + f(T)$ is the total reward the miner receives for winning the block, and αc is the total cost to mine in that round. Recall that $f(T)$ increases monotonically with T, meaning R_{naive} also increases with T. The best choice of T under this reward function is therefore always T_{max}, which intuitively aligns with the current status quo strategy (greedily select the transactions with the highest fees).

On the other hand, if we consider the threat of adversarial attack, the probability that the miner receives the reward is instead

$$\Gamma(\overline{\alpha}, \alpha, T) = \big(1 - P(\beta)A(\overline{\alpha}, T)\big)\frac{\alpha}{\alpha + m\overline{\alpha}}, \quad (7)$$

Intuitively, if the adversary attempts an attack $\big(1 - P(\beta)A(\overline{\alpha}, T)\big)$ can be simplified to $\big(1 - P(\beta)\big)$, which simply corresponds to the probability that the attack fails. If the attack were to succeed, the honest miner would receive no rewards for the block.

The adversary-aware reward function is then

$$R(\overline{\alpha}, \alpha, T) = \Gamma(\overline{\alpha}, \alpha, T)\big(b + f(T)\big) - \alpha c, \quad (8)$$

In contrast with R_{naive}, R does not necessarily increase with T, because a higher transaction value increases the likelihood of adversarial attack through Γ.

C. Transaction Fees

The transaction fee function $f(T)$ is monotonically increasing in T, i.e., transactions with higher transaction values must pay out higher fees. We model $f(T)$ like this for two reasons.

The first is that, as mentioned in Section I, the incentive for the adversary to attack increases with transaction value. Higher fees can be used to offset the risk imposed by higher value transactions because they serve as incentives for miners to contribute additional mining power to each block (thereby increasing the adversary's cost to attack).

The second reason is that defending against double-spend attacks is trivial if high value transactions do not pay the highest fees. In this case, there would be no reason for an honest miner to include a high value transaction in their block, as they could receive the same or greater rewards without risk by including only low value transactions. By defining $f(T)$ to be increasing in T, we impose a tradeoff between the rewards received for mining and the risk of adversarial attack. Honest miners want to maximize the transaction fees they include in the block, but in order to receive higher fees they must include transactions of higher value. In practice, this tradeoff would necessarily exist, as the adversary would ensure that their transactions paid the highest fees to ensure inclusion in the block.

In our implementation, we define $f(T)$ as

$$f(T) = \lambda T, \tag{9}$$

where λ is typically 0.01 in our experiments (i.e., a 1% transaction fee). While there are many different fee functions that we could have chosen, a "percentage-based" fee is both computationally convenient and easy to understand. We defer further exploration of alternate transaction fee functions to future work.

IV. Implementing the Mining Game

In this section, we detail the algorithm that we use to solve for equilibrium behavior of miners in our game.

A. Miner Wealth

We model miner wealth at time t using a density function w_t, where $w_t(x)$ is the fraction of miners with wealth x at time t (and $\sum_x w_t(x) = 1$ for all t).

All miners begin with the same wealth, and a miner's actions are limited by their current wealth just as they would be in a real system. Recall that c is the cost per unit of mining power, then the maximum α a miner with wealth x can play is x/c, and any miner reaching wealth 0 will no longer be able to participate. For this reason, miners cannot simply optimize their expected reward at each time step; instead, they must also consider the rewards they might receive at future time steps as a function of their resulting wealth.

B. Value Function

In order to optimize over future expected rewards, we introduce a value function into the miner's decision-making process. Slightly informally, the value function V_t is defined as

$$V_t(x) = R(\overline{\alpha}, \alpha, T) + \sum_{x'} V_{t+1}(x') Pr[\text{wealth at } t+1 = x' | \text{wealth at } t = x], \tag{10}$$

where $Pr[\text{wealth at } t+1 = x' | \text{wealth at } t = x]$ is the probability that a miner has wealth x' at time $t+1$ given that they have wealth x at time t. Essentially, the value function describes the total reward a miner expects to gain over the remainder of the game as a function of their current wealth.

C. Approximating the Adversary's Decision Function

In the model we have defined, there is a discontinuity in the mining reward function which arises from the adversary's 'attack' or 'do not attack' decision. For the sake of computational efficiency, we approximate $A(T, \overline{\alpha})$ using the sigmoid

$$A'(T, \overline{\alpha}) = \frac{1}{1 + e^{-R_{adv}}}. \tag{11}$$

This approximation is computationally convenient and does not materially impact our results.

D. Finding the Equilibrium

Algorithm 1 solves for the optimal equilibrium policies for miners of varying wealth. It consists of three main computational steps which are executed in a loop until $\overline{\alpha}$ converges, meaning an equilibrium is reached.

1) Compute optimal actions and associated value functions (lines 2-7): We know that the value function at time $\tau + 1$ is 0 because the game concludes after τ rounds. This allows for the use of backwards recursion to solve for the optimal actions and value functions at each prior time step.

Starting at time $t = \tau$, we compute the optimal $\alpha^*(x)$ and $T^*(x)$ as a function of wealth x. Using these actions, we can compute the expected reward a miner receives at time t, their possible next wealth (x_{win} or x_{lose}, depending on whether they win or lose the block) at time $t+1$, and the probability with which each next wealth is realized. This allows us to compute $V_t(x)$ as the summation of $R(\overline{\alpha}, \alpha^*(x), T^*(x))$, $V_{t+1}(x_{win})$, and $V_{t+1}(x_{lose})$ weighted over the probability of winning or losing the block. We repeat this until reaching $t = 0$.

2) Compute wealth distribution (lines 8-13): At $t = 0$ the wealth distribution is known. At each time step, we compute the probability that a miner with wealth x wins or loses the next block when playing their best actions, $\alpha_{t,n}(x)$ and $T_{t,n}(x)$. For all x, the wealth distribution at time $t+1$ at wealth x_{win} and x_{lose} is incremented by the fraction of miners with wealth x multiplied by their probability of winning and losing, respectively.

Algorithm 1: Compute α, T at equilibrium

Input:
m: Number of miners
c: Mining cost per unit of mining power
b: Block reward per block
γ: Momentum parameter
τ: The number of time steps
$\overline{\alpha}_0$: Initial mean mining power
w_0: Initial mean wealth
Output: α_N, T_N

1 **for** *each* n *until* N **do**
2   **for** $t = \tau, \tau - 1, \ldots, 0$ **do**
3     $\forall x \in \mathcal{X}\ \alpha^{\star}_{t,n}(x), T^{\star}_{t,n}(x) = \arg\max_{\alpha,T} R\left(\overline{\alpha}_{t,n}, \alpha, T\right) + \Gamma(\overline{\alpha}_{t,n}, \alpha, T) V_{t+1,n}\left(x + R\left(\overline{\alpha}_{t,n}, \alpha, T\right)\right) + \big(1 - \Gamma(\overline{\alpha}_{t,n}, \alpha, T)\big) V_{t+1,n}\left(x - \alpha c\right)$
4     $\forall x \in \mathcal{X}\ V_{t,n}\left(x\right) = R\left(\overline{\alpha}_{t,n}, \alpha^{\star}(x), T^{\star}(x)\right) + \Gamma(\overline{\alpha}_{t,n}, \alpha^{\star}(x), T^{\star}(x)) V_{t+1,n}\left(x + R\left(\overline{\alpha}_{t,n}, \alpha^{\star}(x), T^{\star}(x)\right)\right) + \big(1 - \Gamma(\overline{\alpha}_{t,n}, \alpha^{\star}(x), T^{\star}(x))\big) V_{t+1,n}\left(x - \alpha c\right)$
5     $\forall x \in \mathcal{X}\ \alpha_{t,n}\left(x\right) \leftarrow \alpha^{\star}(x)$
6     $\forall x \in \mathcal{X}\ T_{t,n}\left(x\right) \leftarrow T^{\star}(x)$
7   **end**
8   **for** $t = 0, \ldots, \tau$ **do**
9     $w_{t+1,n} \leftarrow$ *zeros*
10     $\forall x \in \mathcal{X}$
11      $x_{win} \leftarrow x + R\left(\overline{\alpha}_{t,n}, \alpha_{t,n}\left(x\right), T_{t,n}\left(x\right)\right)$
12      $x_{lose} \leftarrow x - \alpha_{t,n} c$
13      $w_{t+1,n}\left(x_{win}\right) += \Gamma(\overline{\alpha}_{t,n}, \alpha_{t,n}\left(x\right), T_{t,n}\left(x\right)) w_{t,n}\left(x\right)$
14      $w_{t+1,n}\left(x_{lose}\right) += \big(1 - \Gamma(\overline{\alpha}_{t,n}, \alpha_{t,n}\left(x\right), T_{t,n}\left(x\right))\big) w_{t,n}\left(x\right)$
15   **end**
16   **for** $t = 0, \ldots, \tau$ **do**
17     $\overline{\alpha}_{t+1,n+1} = \gamma \overline{\alpha}_{t,n} + \sum_x \alpha_{t,n}\left(x\right) w_{t,n}\left(x\right)$
18   **end**
19   **if** $\overline{\alpha}_{n+1} == \overline{\alpha}_n$ **then**
20   break
21   **end**
22 **end**

3) Update $\overline{\alpha}$ (lines 14-16): The final step is to compute the new values of $\overline{\alpha}_t$ for each time step. These values are used in the next iteration of the outer loop until the algorithm converges. Fortunately, $\overline{\alpha}_t$ is simple to compute, as it is just the average of each α played weighted by the fraction of miners who played it. Similar to that used in [30], we make use of a momentum parameter $\gamma \in [0, 1)$ which slows the rate at which $\overline{\alpha}_t$ changes in order to prevent oscillatory behavior between iterations.

An equilibrium is reached when $\overline{\alpha}_t$ converges for all t (lines 17-19).

V. Results

In this section, we examine the behavior and performance (in terms of wealth gained) of miners in the presence of adversaries of varying strength. We show that the miners using R as a reward function are far more profitable when a powerful adversary is present than those using R_{naive}.

A. Miners Using R_{naive}

We first examine the scenario in which an adversary is ready to attack, but miners do not account for the adversary in their reward function (i.e., they choose actions based on R_{naive}). Recall that miners using R_{naive} always select a transaction value of T_{max} in order to maximize the transaction fees they might receive upon winning the block. Under these conditions, Figure 2 shows the probability with which the adversary attacks at each time step[4]. When the adversary is weak, their probability of success is too low for the attack to be profitable under any of the experiment parameters we tested. However, for $\beta = .35$ and $\beta = .45$, the adversary attacks at all time steps because miners select α and T without consideration for adversarial attack.

Figure 3 shows the result: for high values of β, mining performance is dramatically impacted, whereas for low values

[4]Note that $\beta = 0$ is excluded from this figure, as an adversary with no mining power is not able to attack.

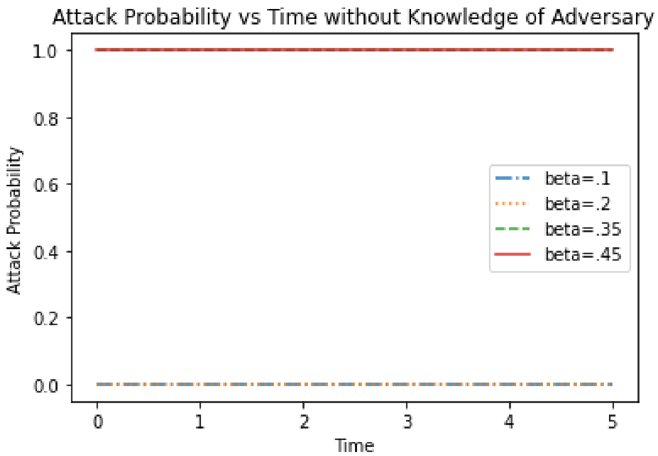

Figure 2. The adversary attack probability over time for different β values. Note that the lines for $\beta = 0.1$ and $\beta = 0.2$ are overlapping, as are the lines for $\beta = 0.35$ and $\beta = 0.45$.

of β, mining performance is completely unaffected. Miners still remain profitable (albeit less so) against an adversary with 35% of the mining power, but mining performance continues to degrade as β increases, and at $\beta = 0.45$ miners are actually losing wealth in each time step.

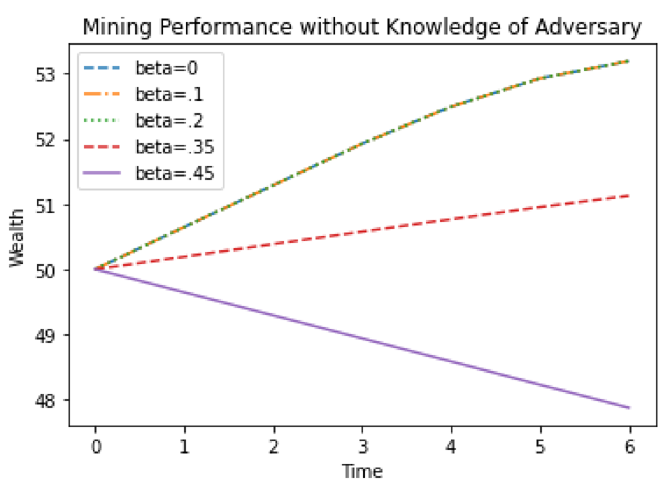

Figure 3. Evolution of average wealth over time for miners without knowledge of the adversary. Each line depicts mining performance of honest miners in the presence of adversaries of differing strength. Note that the lines for $\beta = 0, \beta = 0.1, \beta = 0.2$ are identical and therefore overlap one another.

B. Miners with Knowledge of the Adversary

Next we examine the scenario in which an adversary is waiting to attack and miners *do* account for it in their reward function. In this case, instead of selecting T_{max}, miners select a transaction value $T^\star$ which corresponds to the highest transaction value for which the adversary has no incentive to attack. Figure 4 can be used to visualize the shape of the expected reward for honest miners as a function of the transaction value they select.

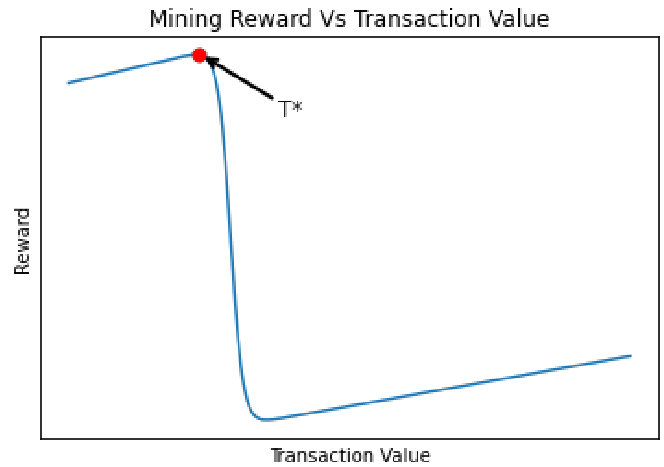

Figure 4. The honest mining reward as a function of transaction value selected when $\beta = 0.4$. The point at which mining reward is maximized is $T^\star$, and the reward decreases sharply if a transaction value higher than $T^\star$ is selected because in that regime it is profitable for the adversary to attack. Note that according to our model, there should be a discontinuity in the plot at $T^\star$. See Section IV for details on why we smooth out this discontinuity in our implementation.

Under the same experimental parameters as those used in the previous section, Figure 5 shows that the adversary never attacks, as there is no time step during which the attack would be profitable.

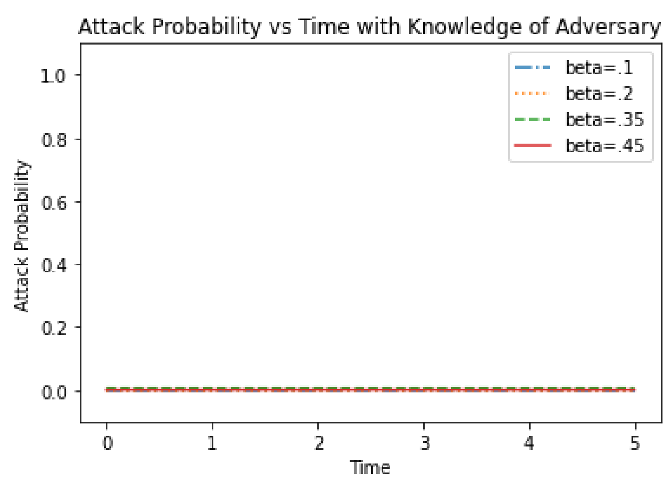

Figure 5. The adversary attack probability over time for different β values. When the honest miners defend against attack, it is never profitable for the adversary to do so.

Figure 6 shows the performance of miners in this scenario. Unsurprisingly, the performance is identical for values of $\beta \leq 0.2$, as the adversary is simply too weak to affect mining performance, so $T^\star = T_{max}$. However, in contrast with miners using R_{naive}, miners using R as a reward function are able to effectively counter a strong adversary controlling a significant fraction of the mining power. In this scenario, miners achieve positive wealth in each time step, albeit with a slight reduction in performance resulting from the lower transaction fees collected.

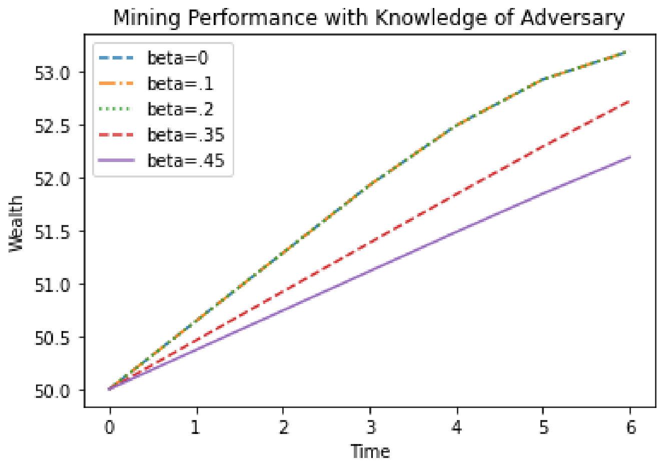

Figure 6. Evolution of average wealth over time for miners with knowledge of the adversary. Each line depicts mining performance of honest miners in the presence of adversaries of differing strength. Note that the lines for $\beta = 0, \beta = 0.1, \beta = 0.2$ are identical and therefore overlap one another.

C. Mining Performance Comparison with No Adversary

There is an inherent tradeoff between using R and R_{naive}, as miners must deliberately accept lower transaction fees in order to remove the incentive for the adversary to attack. In order to understand this tradeoff, we also consider the case that the honest miners optimize against the threat of an adversarial attack but no adversary is present. Figure 7 shows the result of this experiment.

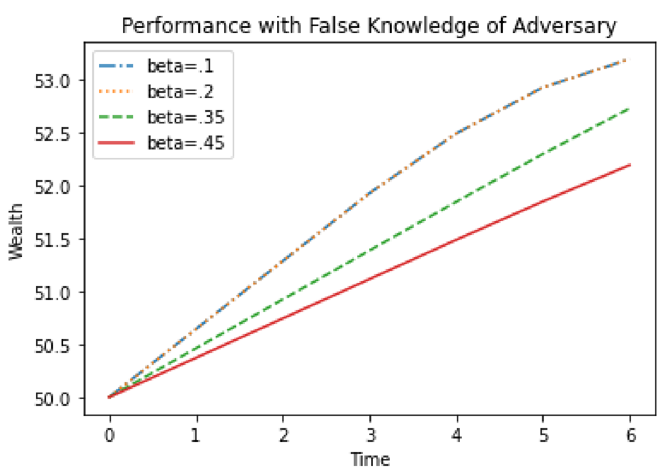

Figure 7. Evolution of average wealth over time for miners selecting their actions under the presumed threat of attack when no adversary is actually present. In this figure, β values correspond to the adversary strength the miners are defending against, and the (overlapping) plots for $\beta = .1$ and $\beta = .2$ can be used as a baseline to determine mining performance without consideration for the adversary.

Interestingly, Figure 7 is identical to Figure 5 despite the fact that these experiments were run in different settings. This is expected, as the purpose of optimizing against R is to remove the ability for the adversary to profit from the attack. From the perspective of the miners, there is no difference between the scenario in which an adversary could attack but does not and that in which there is no adversary.

The takeaway from Figure 7 is that, in the absence of an adversary, mining can be profitable for miners using R as a reward function, although less so than for those using R_{naive}. The degree to which performance is impacted depends on specific parameters of the system, and we note that our model does not capture the broader incentive miners might have to prevent attacks from being perpetrated against the system. We leave a more comprehensive characterization of this tradeoff to future work.

VI. Discussion and Future Work

In this section, we discuss the implications of this work and possible extensions of our model.

A. The Adversary

The adversary we present is somewhat limited in its strategy, as it can only decide whether or not to attack at each time step and has no ability to call off an attack early (if the honest miners pull too far ahead) or continue an attack for longer if they are not very far behind. Additionally, we currently treat the adversary as being external to the system prior to launching an attack. This is a plausible modeling decision under the assumption that the adversary can quickly reallocate their mining power amongst different cryptocurrencies, either by renting computational power or swapping between different cryptocurrencies which use the same mining function. However, understanding the case where an "honest" miner could opportunistically attempt a double-spend is also critical to the long-term security of these systems.

B. Transaction Fees, Transaction Value, and Resistance to Attack

There is an implicit relationship between the maximum transaction value that can be included in a block, the transaction fees a miner receives, and the anticipated mining power of an adversary. Higher value transactions increase the incentive for an adversary to attempt an attack, greater network mining power can increase the cost of attack, and higher transaction fees can be used to attract additional mining power to the network. Understanding this relationship in greater detail would allow for the design of systems which secure a desired transaction value per block against rational attack without wasting resources on excess mining. We believe this to be an exciting direction for future work.

C. Conclusions

In this work, we model the mining game for Proof-of-Work cryptocurrencies in the presence of a rational, powerful adversary as a mean-field game. We model the reward functions of honest miners under threat of adversarial attack design and implement an algorithm that can efficiently solve for equilibrium policies of this game. We show that miners using our adversary-aware reward function are able to eliminate the threat of attack from a profit-seeking adversary with only a small tradeoff in profitability.

References

[1] J. Bonneau, "Why buy when you can rent?" in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 19–26.

[2] D. Palmer, "Bad actors rent hashing power to hit bitcoin gold with new 51% attacks," https://decrypt.co/41492/rented-hash-power-huge-problem-for-proof-of-work-etc-labs-ceo, 2020.

[3] F. Staff, "Etc labs ceo: Rented hash power is "huge" problem for proof of work," https://decrypt.co/41492/rented-hash-power-huge-problem-for-proof-of-work-etc-labs-ceo, 2020.

[4] M. Huang, R. P. Malhamé, P. E. Caines *et al.*, "Large population stochastic dynamic games: closed-loop mckean-vlasov systems and the nash certainty equivalence principle," *Communications in Information & Systems*, vol. 6, no. 3, pp. 221–252, 2006.

[5] J.-M. Lasry, P.-L. Lions, J.-M. Lasry, and P.-L. Lions, "Mean field games," *Japan. J. Math*, vol. 2, pp. 229–260, 2007.

[6] A. Kiayias, E. Koutsoupias, M. Kyropoulou, and Y. Tselekounis, "Blockchain mining games," in *Proceedings of the 2016 ACM Conference on Economics and Computation*, 2016, pp. 365–382.

[7] C. Ewerhart, "Finite blockchain games," *Economics Letters*, vol. 197, p. 109614, 2020.

[8] R. Singh, A. D. Dwivedi, G. Srivastava, A. Wiszniewska-Matyszkiel, and X. Cheng, "A game theoretic analysis of resource mining in blockchain," *Cluster Computing*, vol. 23, no. 3, pp. 2035–2046, 2020.

[9] G. Yuan, "The framework of consensus equilibria for mining-pool games in blockchain ecosystems," *arXiv preprint arXiv:2003.05067*, 2020.

[10] Z. Liu, N. C. Luong, W. Wang, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "A survey on applications of game theory in blockchain," *arXiv preprint arXiv:1902.10865*, 2019.

[11] G. Goren and A. Spiegelman, "Mind the mining," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, 2019, pp. 475–487.

[12] J. Sun, P. Tang, and Y. Zeng, "Games of miners," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, 2020, pp. 1323–1331.

[13] T. Min and W. Cai, "A security case study for blockchain games," in *2019 IEEE Games, Entertainment, Media Conference (GEM)*. IEEE, 2019, pp. 1–8.

[14] E. Altman, D. Menasché, A. Reiffers-Masson, M. Datar, S. Dhamal, C. Touati, and R. El-Azouzi, "Blockchain competition between miners: a game theoretic perspective," *Frontiers in Blockchain*, p. 26, 2020.

[15] Y. Wang, C. Tang, F. Lin, Z. Zheng, and Z. Chen, "Pool strategies selection in pow-based blockchain networks: Game-theoretic analysis," *IEEE Access*, vol. 7, pp. 8427–8436, 2019.

[16] M. V. Ferreira and S. M. Weinberg, "Proof-of-stake mining games with perfect randomness," in *Proceedings of the 22nd ACM Conference on Economics and Computation*, 2021, pp. 433–453.

[17] R. Lavi, O. Sattath, and A. Zohar, "Redesigning bitcoin's fee market," *ACM Transactions on Economics and Computation*, vol. 10, no. 1, pp. 1–31, 2022.

[18] A. C. chih Yao, "An incentive analysis of some bitcoin fee designs."

[19] T. Roughgarden, "Transaction fee mechanism design," *ACM SIGecom Exchanges*, vol. 19, no. 1, pp. 52–55, 2021.

[20] ——, "Transaction fee mechanism design for the ethereum blockchain: An economic analysis of eip-1559," *arXiv preprint arXiv:2012.00854*, 2020.

[21] H. Chung and E. Shi, "Foundations of transaction fee mechanism design," in *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2023, pp. 3856–3899.

[22] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[23] M. Rosenfeld, "Analysis of hashrate-based double spending," *arXiv preprint arXiv:1402.2009*, 2014.

[24] G. Bissias, B. N. Levine, A. P. Ozisik, and G. Andresen, "An analysis of attacks on blockchain consensus," *arXiv preprint arXiv:1610.07985*, 2016.

[25] Y. Sompolinsky and A. Zohar, "Bitcoin's security model revisited," *arXiv preprint arXiv:1605.09193*, 2016.

[26] J. Jang and H.-N. Lee, "Profitable double-spending attacks," *Applied Sciences*, vol. 10, no. 23, p. 8477, 2020.

[27] J. Garay, J. Katz, U. Maurer, B. Tackmann, and V. Zikas, "Rational protocol design: Cryptography against incentive-driven adversaries," in *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 2013, pp. 648–657.

[28] C. Badertscher, J. Garay, U. Maurer, D. Tschudi, and V. Zikas, "But why does it work? a rational protocol design treatment of bitcoin," in *Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part II 37*. Springer, 2018, pp. 34–65.

[29] C. Badertscher, Y. Lu, and V. Zikas, "A rational protocol treatment of 51% attacks," in *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part III 41*. Springer, 2021, pp. 3–32.

[30] Z. Li, A. M. Reppen, and R. Sircar, "A mean field games model for cryptocurrency mining," *arXiv preprint arXiv:1912.01952*, 2019.

[31] R. K. Mishra, D. Vasal, and S. Vishwanath, "Model-free reinforcement learning for non-stationary mean field games," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 1032–1037.

[32] C. Bertucci, L. Bertucci, J.-M. Lasry, and P.-L. Lions, "Mean field game approach to bitcoin mining," *arXiv preprint arXiv:2004.08167*, 2020.