Preventing Digital Currency Miners to Launch Attacks Against Mining Pools by a Reputation-Based Paradigm

Mehrdad Nojoumian and Arash Golchubian
Department of Computer & Electrical Engineering and Computer Science
Florida Atlantic University, Boca Raton, FL
{mnojoumian,agolchub}@fau.edu

Laurent Njilla and Kevin Kwiat
Cyber Assurance Branch
Air Force Research Lab, Rome, NY
{laurent.njilla,kevin.kwiat}@us.af.mil

Charles Kamhoua
Network Security Branch
Army Research Lab, Adelphi, MD
charles.a.kamhoua.civ@mail.mil

Abstract: The mining process in the Blockchain is very resource intensive; therefore, miners form coalitions to verify each block of transactions in return for a reward where only the first coalition that accomplishes the proof-of-work will be rewarded. This leads to intense competitions among miners and, consequently, dishonest mining strategies such as block withholding attack, selfish mining, eclipse attack and stubborn mining, to name a few. As a result, it is necessary to regulate the mining process to make miners accountable for any dishonest mining behavior. We therefore propose a new reputation-based framework for the proof-of-work computation in the Blockchain in which miners are not only incentivized to conduct honest mining, but also disincentivized to commit any malicious activities against other mining pools. We first illustrate the architecture of our reputation-based paradigm, explain how the miners are rewarded or penalized in our model, and subsequently, we provide game-theoretical analyses to show how this new framework encourages the miners to avoid dishonest mining strategies. In our setting, a mining game is repeatedly played among a set of pool managers and miners where the reputation of each miner or mining ally is continuously measured. At each round of the game, the pool managers send invitations only to a subset of miners based on a nonuniform probability distribution defined by the miners’ reputation values. We show that by using our proposed solution concept, honest mining becomes Nash equilibrium in our setting. In other words, it will not be in the best interest of the miners to employ dishonest mining strategies even by gaining a short-term utility. This is due to the consideration of a long-term utility in our model and its impact on the miners’ utilities over time.

Keywords: Proof-of-work; reputation systems; game theory.

I. INTRODUCTION

Security games are mainly designed and used to model interaction between attackers and defenders (Roy et al. 2010; Liang and Xiao 2013). In these models, two-player games—extendable to any number of players—are proposed in which both attackers and defenders try to
maximize the utility that each can gain. For instance, the defenders will be able to provide value to the system and, as a result, gain utility by enabling features, shifting the attack surface, and reducing the attack surface measurement. Likewise, the attackers will be able to gain utility if features are disabled or the attack surface measurement is increased.

In the majority of existing security games, attackers and defenders play the game by choosing various actions from the action profiles based on their strategies in each round of the game. For instance, the defenders can modify the setting of the targeted system to shift the attack surface, whereas the attackers can manipulate the system to disable some features. After each round of the game, the game moves to a new state and the players receive their rewards based on some utility functions.

One of the fascinating research areas where the security games can be used is the verification of transactions in the context of digital currencies, e.g., Bitcoin (Nakamoto 2008), or similar paradigms. The mining operation is very resource intensive. As a result, players form different coalitions to verify each block of transactions in return for a reward. This leads to intense competition among competitors since only the first coalition that accomplishes the mining process will be rewarded.

To address what issues this competition may cause, different strategies are analyzed in the literature. Rosenfeld (2011) introduced the block withholding attack, where a dishonest player only reveals a partial solution of the verification problem whenever he has the complete solution to act in favor of another competing coalition. As a result, the dishonest miner shares the revenue obtained by the entire coalition without any contribution. Eyal and Sirer (2014) introduced selfish mining, where the players of a coalition keep their discovered blocks private and continue to verify more blocks privately until they get a subchain that its length is threatened. As a result, selfish players receive the reward. Johnson et al. (2014) look at the malicious activity of the players from another perspective. The authors compare an honest approach with a dishonest strategy, i.e., players of a coalition can invest to acquire additional computing resources, or launch distributed denial-of-service attacks against other competing coalitions. The authors provide game-theoretical analyses by exploring the trade-off between these two strategies when two groups of varying sizes are involved. More attacks were introduced recently, for example, eclipse attack (Heilman et al. 2015) that makes a node invisible in the network, or stubborn mining as a generalization of the selfish mining (Nayak et al. 2016).

We therefore propose a new reputation-based framework in which miners not only are incentivized to conduct honest mining, but also disincentivized to commit any malicious activities against other mining pools, such as block withholding attack, selfish mining, eclipse attack and stubborn mining, to name a few. We first illustrate the architecture of our reputation-based paradigm, explain how miners are rewarded or penalized in our model, and subsequently, we provide game-theoretical analyses to show how this new framework encourages the miners to avoid dishonest mining strategies.

The rest of this chapter is organized as follows. Section II provides preliminary materials on digital currencies and game theory. Section III reviews the existing digital currency literature where game theory is used. Section IV illustrates our model. Section V explains how our reputation-based scheme works. Finally, Section VII concludes with final remarks.
II. PRELIMINARIES

A. Digital Currencies: Terminologies and Mechanics

In the digital currency frameworks, specifically Bitcoin, transactions are grouped in blocks to be verified by a subset of nodes in the network, known as miners. The mining process, named proof-of-work, is computationally intensive with a specific difficulty factor that is increased over time as the computational power of hardware systems grows. Therefore, nodes form mining pools under the supervision of pool managers to accomplish the mining task. In some technical articles, the mining process of the Bitcoin (or even other digital currencies) is referred to as the miners’ mathematical puzzle.

The first mining pool that accomplishes the proof-of-work is rewarded a certain amount of freshly mined Bitcoins as an incentive for miners’ works. That is why this process is also known as mining. As soon as a block is verified, it is attached to the list of existing verified blocks, known as Blockchain. Immediately after that, all miners stop the mining process of the already verified block and start working on the next block.

Each block consists of a block number, a nonce value, list of transactions, the hash value of the previous block (address of the previous block), and the hash value of the next block (address of the next block). During the mining process, the miners try to generate a valid hash value of a block that is less than a threshold, i.e., it starts with a certain number of zeros. They will conduct this process by trying different nonce values. It is clear that generating a hash value that starts with, say 5 zeros, is harder than a hash value that begins with 4 zeros; this is what we call the difficulty factor of mining.

The hashing rate, $h_r$, also known as mining power, is the total number of hashes that a miner can calculate during a specific time interval. Therefore, the average time to find a valid hash value, also known as full proof-of-work, correlates to a miner’s hashing rate. In fact, the pool manager sends different templates of the current block to his miners so that they can find a valid hash value by changing the nonce value. If a miner accomplishes the full proof-of-work, he will then send it to his pool manager. Consequently, the pool manager publishes the legitimate block on behalf of the entire pool. He will then distribute the revenue among miners based on their mining powers. Note that new coins are put explicitly in the block by the miner(s) who created it.

To estimate each miner’s power, the pool manager determines a partial target for each miner, much easier than the actual target of the system. For instance, instead of calculating a hash value that starts with, say 5 zeros, a hash value with a single zero is sufficient. Note that this is just a simple example for the sake of clarification. Therefore, each miner is instructed to send a valid hash value according to the partial target. This partial target is defined in such a way that a partial solution can be calculated frequently enough so that the manager can fairly estimate the miners’ powers because, as we stated earlier, the revenue is distributed based on the miners’ powers.

B. Game Theory: Basic Notions and Definitions

A game consists of a set of players, a set of actions and strategies (strategy is the way that each player selects actions), and finally, a utility function that is used by each player to compute how much benefit he obtains by choosing a certain action. In cooperative games, the players collaborate and split the aggregated utility among themselves, that is, cooperation is incentivized
by agreement. However, in noncooperative games, the players cannot form any agreement to coordinate their behaviors. In other words, any cooperation among the players must be self-enforcing. We briefly review some well-known game-theoretic concepts (Osborne and Rubinstein 1994) for our further analyses and discussions.

**Definition 1:** Let \( \mathcal{A} \equiv \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n \) be an action profile for \( n \) players, where \( \mathcal{A}_i \) denotes the set of possible actions of player \( P_i \). A game \( \Gamma=(\mathcal{A},u) \) for \( 1 \leq i \leq n \), consists of \( \mathcal{A}_i \) and a utility function \( u_i: \mathcal{A} \rightarrow \mathbb{R} \) for each player \( P_i \). We refer to a vector of actions \( \bar{a} = (a_1, \ldots, a_n) \in \mathcal{A} \) as an outcome of the game.

**Definition 2:** Utility function \( u_i \) illustrates the preferences of player \( P_i \) over different outcomes. We say \( P_i \) prefers outcome \( \bar{a} \) to \( \bar{a}' \) if \( u_i(\bar{a}) > u_i(\bar{a}') \), and he weakly prefers outcome \( \bar{a} \) to \( \bar{a}' \) if \( u_i(\bar{a}) \geq u_i(\bar{a}') \).

To allow the players to follow randomized strategies, we define \( \sigma_i \) as a probability distribution over \( \mathcal{A}_i \) for a player \( P_i \). This means he samples \( a_i \in \mathcal{A}_i \) according to \( \sigma_i \). A strategy is said to be a pure-strategy if each \( \sigma_i \) assigns probability 1 to a certain action, otherwise, it is said to be a mixed-strategy. Let \( \bar{\sigma} = (\sigma_1, \ldots, \sigma_n) \) be the vector of players’ strategies, and let \( (\sigma_i', \bar{\sigma}) = (\sigma_1, \ldots, \sigma_{i-1}, \sigma_i', \sigma_{i+1}, \ldots, \sigma_n) \), where \( P_i \) replaces \( \sigma_i \) by \( \sigma_i' \) and all the other players’ strategies remain unchanged. Therefore, \( u_i(\bar{\sigma}) \) denotes the expected utility of \( P_i \) under the strategy vector \( \bar{\sigma} \). A player’s goal is to maximize \( u_i(\bar{\sigma}) \). In the following definitions, one can substitute action \( a_i \in \mathcal{A}_i \) with its probability distribution \( \sigma_i \in \mathcal{A}_i \), or vice versa.

**Definition 3:** A vector of strategies \( \bar{\sigma} \) is Nash equilibrium if, for all \( i \) and any \( \sigma_i' \neq \sigma_i \), it holds that \( u_i(\sigma_i', \bar{\sigma}_{-i}) \leq u_i(\bar{\sigma}) \). This means no one gains any advantage by deviating from the protocol as long as the others follow the protocol.

**Definition 4:** Let \( \mathcal{A}_i \equiv \mathcal{A}_i \times \cdots \times \mathcal{A}_{i-1} \times \mathcal{A}_{i+1} \times \cdots \times \mathcal{A}_n \). A strategy \( \sigma_i \in \mathcal{A}_i \) (or an action) is weakly dominated by \( \sigma_i' \in \mathcal{A}_i \) (or another action) with respect to \( \mathcal{A}_i \) if: For all \( \bar{\sigma}_{-i} \in \mathcal{A}_{-i} \), it holds that \( u_i(\sigma_i, \bar{\sigma}_{-i}) \leq u_i(\sigma_i', \bar{\sigma}_{-i}) \). There is a \( \bar{\sigma}_{-i} \in \mathcal{A}_{-i} \) such that \( u_i(\sigma_i, \bar{\sigma}_{-i}) < u_i(\sigma_i', \bar{\sigma}_{-i}) \). This means player \( P_i \) can never improve its utility by playing \( \sigma_i \), and he can sometimes improve it by not playing \( \sigma_i \). A strategy \( \sigma_i \in \mathcal{A}_i \) is strictly dominated if player \( P_i \) can always improve its utility by not playing \( \sigma_i \).

**III. LITERATURE REVIEW**

Even though the concept of Blockchain is relatively new, introduced by an unknown author or authors in 2008, it has gained considerable attention from the computer science and economics communities because of its unique approach in decentralizing verification of transactions related to a digital currency, and its inherent security because of this decentralized nature. However, the body of work that is focused on the study of Blockchain through the use of game-theoretic methods is limited. In this section, related research works to game theory and Blockchain are reviewed.

The authors Johnson et al. (2014) study the incentives for a mining pool to carry out a distributed denial-of-service (DDoS) attack against another mining pool. The authors scrutinize this problem from an economic point of view where the incentive for an attack is to increase one’s own probability of successfully verifying the next block of transactions, and hence, earning the Bitcoin rewards from this mining operation. They conclude that there is a greater incentive to
attack a large mining pool rather than a small pool. The authors point out that this finding is consistent with statistics reported by Vasek, Thornton, and Moore (2014) that shows 17.1% of small mining pools have suffered from DDoS attacks, whereas 62.5% of large pools have been affected by such attacks. The authors make two other interesting observations as well. First of all, the ability to mitigate the DDoS attacks will increase the market threshold for the size at which a pool becomes vulnerable to the DDoS attack. This makes intuitive sense since the ability to mitigate such attacks will decrease the attacker’s utility. Second, the cost of these attacks will keep small pools out of the DDoS market since the incentive for attacking such pools is relatively low.

Babaioff et al. (2012) look at a different problem that is present in the Bitcoin protocol. In fact, this problem will intensify once the mining reward is ended in the Bitcoin network. In the current design, the nodes that authorize a transaction are rewarded through two separate methods. The first is through the generation of new Bitcoins for every new block that is added to the Blockchain, and the second method is through a transaction fee. The maximum number of Bitcoins is limited to about 21 million (Antonopoulos 2014) and the creation of new Bitcoins becomes exponentially smaller until the maximum limit is reached. The transaction fee will be the only resource to incentivize the miners when the maximum threshold is reached. At that point, miners are incentivized to keep the information of a possible transaction secret as there will be no new Bitcoins to be mined from the efforts of mining, that is, there is only the transaction fee that is given to the verifier of transactions. This incentive to keep information secret can potentially cripple the Bitcoin system as the time for confirming a transaction will be long when there is only one node attempting to verify the transaction.

Kroll, Davey, and Felten (2013) study Bitcoin as a consensus game and consider the economics of Bitcoin from the mining perspective to determine whether any incentive exists for rational players to deviate from the mining protocol. The authors show that there is a Nash equilibrium outcome for which all players cooperate with the Bitcoin reference implementation. However, there are infinitely many equilibria where the players can behave otherwise. The authors show that a motivated adversary may be capable of crashing the currency; as a result, governance structures will be necessary.

Even though the authors in Barber et al. (2012) don’t refer to any game-theoretic models, they detail several possible vulnerabilities within the Blockchain protocol that are great candidates for game-theoretic study such as deflationary spiral, the History Revision attack, and delayed transaction confirmation. Carlsten et al. (2016) study the issues of Bitcoin and Blockchain when the last block reward is collected. The authors show that once the mining reward is removed from the protocol, leaving only the transaction fees, the incentive for defection increases.

Luu et al. (2015) scrutinize the block withholding attack on mining pools, introduced by Rosenfeld (2011). They show that the attack always has incentive when looking at a long-term operation, but it may not be profitable for short-term duration. Eyal (2015) studies the same subject and concludes that when two pools attack each other, it results in a version of the prisoner’s dilemma, named the Miner’s Dilemma. Lewenberg, Sompolinsky, and Zohar (2015) introduce a modification to the Blockchain protocol to allow for inclusion of forked blocks with the aim of increasing the rate of operation. They then provide a game-theoretic model of the competition for fees between the nodes under the new protocol.
IV. OUR REPUTATION-BASED MINING MODEL AND SETTING

As illustrated in Figure 1, our model consists of a set of pool managers $M_{(i,p_i)}$ who form coalitions for the proof-of-work computations, for $1 \leq i \leq I$, where $0 \leq p_i$ denote profits that pool managers have so far accumulated; a set of miners/ally miners $m_{(j,k,r_k)}$ who perform proof-of-works, for $1 \leq j \leq J$ and $1 \leq k \leq K$, where $-1 \leq r_k \leq +1$ denote the reputation value of a miner/ally miners. In our model, miners/ally minors may commit to malicious activities through direct attacks (e.g., DDoS attack) or collusion attacks (e.g., block withholding) to disrupt the proof-of-work computations of certain mining pools. As such, two actions are considered in the miners’ action profile, that is, commit to malicious activity to disrupt computations of mining pools, denoted by $\mathcal{D}$: dishonest mining, or conduct the proof-of-work honestly, denoted by $\mathcal{H}$: honest mining.

Note that in the current setting of digital currencies, each miner is defined by a unique identity $j$. However, in our proposed framework, each miner is also assigned a public reputation value, $r_k$, where $k$ is the index of this value. In fact, the reputation value reflects how well the miner has so far performed in the system in terms of mining performance as well as hostile or malicious activities (i.e., a history of behavior). This public reputation value $r_k$ is updated after a specific period of time based on different criteria, e.g., the ratio of full proof-of-work over partial proof-of-work, detection of any malicious activity such as collusion with other miners, selfish mining, or contribution to a DDoS attack. Moreover, each pool manager $i$ is also assigned a parameter $p_i$ that defines the profit that he has so far accumulated through his pool. As $p_i$ reflects how well a manager is performing, it can be interpreted as his reputation.

In our setting, a subset of miners who highly trust each other (due to partnerships, personal relationships, common nationality, or even geographical proximity) can form an alliance, named ally miners, and request a single reputation value $r_k$ even though they each have a separate identity $j$. This means that while members of a coalition can build reputation all together through $r_k$ by collaborations over time, they are all responsible for malicious activities triggered by even
a single member of their coalition. This leads to the notion of neighborhood-watch, meaning that each member of an alliance is incentivized to monitor his allies. For instance, members can agree to execute a randomized algorithm to monitor each other through various methods, that is, cybersecurity detection techniques or transparency policies to make sure no one has ever received any bribe from other mining pools due to any sort of collusion attacks. As a result, the pool manager doesn’t need to have any concern for every single member of his mining pool. Furthermore, if a member decides to launch an attack, he may need to convince all his coalition members or act solo, which might be caught by his allies through randomized monitoring before it can even affect the mining procedure.

Occasionally the pool managers rearrange their groups to form new coalitions for the proof-of-work. They send invitations (i.e., an invitation-based approach) to miners/ally miners based on a nonuniform probability distribution that is defined by the reputation values $r_k$. In other words, the miners/ally miners who are more reputable have a higher chance to be invited to the mining pools and those who are not trustworthy have a lower chance to receive invitations. The miners/ally miners can also choose to whom they would like to join if they receive multiple invitations, that is, a mutual merit-based setting for both miners and managers.

Since this public reputation system is sustained over time, it will be in the best interests of the miners/ally miners to become reputable (or sustain their high reputation) to maximize their long-term utility. This will incentivize the miners/ally miners to avoid any dishonest behavior even if it has a short-term utility. Note that the underlying reputation system must be immune against reentry attack (that is, cheat and come back to the scheme with a new identity $j$). We use the proposed idea of rational trust modeling (Nojoumian 2017) to make sure our proposed mining paradigm is not vulnerable to these sorts of attacks against reputation systems.

Furthermore, in our proposed model, while ally miners are incentivized to form larger coalitions to sustain a high reputation value and consequently gain more revenue, they are not incentivized to admit any new miner to their alliance unless they fully trust the newcomer. This is due to the fact that a single miner can harm the entire coalition. Moreover, it is worth mentioning that, although ally miners only have a single reputation identity $r_k$, a miner cannot commit to malicious activities in a set and then simply join another alliance because each miner still has a unique identifier $j$.

Our proposed model can be seen as a global community where each mining pool represents a federal authority and each alliance represent a state authority. Therefore, each alliance is responsible to detect malicious activities inside the coalition in a smaller scale. In addition, each alliance can be changed in size and move to a new mining pool when the rearrangement occurs. This approach not only leads to less managerial overheads for the pool managers, but it also creates a framework where practical implementations of preventive and detective protocols become possible.

V. MINING IN OUR REPUTATION-BASED MODEL

Since our approach is designed using a reputation-based paradigm, it is necessary to use a reputation/trust model that is resistant to the well-known reentry attack, that is, corrupted players return to the scheme using new identities. Otherwise our approach cannot be utilized properly. We will discuss this in the next section.
A. Prevention of the Reentry Attack

To deal with the reentry attack in our reputation-based scheme, we use the proposed approach of rational trust modeling (Nojoumian 2017). We provide a high-level description how this modeling technique works. Suppose there exists two trust functions as follows. The first function \( f_1(T_i^{p-1}, \alpha_i) \) has two inputs, that is, trust value \( T_i^{p-1} \) of player \( P_i \) in period \( p - 1 \) and action \( \alpha_i \) (cooperation or defection) selected by player \( P_i \) in period \( p - 1 \). This function computes the updated trust value \( T_i^p \) of player \( P_i \) for the next round \( p \) based on these two inputs. However, the second function \( f_2(T_i^{p-1}, \alpha_i, \ell_i) \) has an extra input value that defines the player’s lifetime, denoted by \( \ell_i \). This extra input determines how long a player with a reasonable number of interactions exists in a reputation-based scheme, for instance, in our proposed reputation-based mining framework.

Using the second function, the reputation-based scheme should then be designed in a way that a player with a longer lifetime can be rewarded (penalized) more (less) than a player with a shorter lifetime, assuming that the other two inputs (i.e., current trust value and the action) are the same. In this setting, “reward” means gaining a higher trust value/becoming more trustworthy, and consequently, receiving a higher utility, and “penalty” means otherwise. In other words, if two players \( P_i \) and \( P_j \) both cooperate \( \alpha_i = \alpha_j = C \) and their current trust values are equal \( T_i^{p-1} = T_j^{p-1} \) but their lifetime parameters are different, say \( \ell_i > \ell_j \), the player with a higher lifetime parameter gains a higher trust value for the next round, i.e., \( T_i^p > T_j^p \). This helps player \( P_i \) to accumulate more utility/revenue in the targeted reputation-based framework.

To exemplify, consider a situation in which sellers, in a reputation-based e-commerce setting, have options to sell the “defective” versions of an item with more revenue or the “nondefective” versions of the same item with less revenue. If the first sample function \( f_1 \) is used in the scheme, it might be tempting for a seller to sell the defective items with more revenue and then he returns to the e-commerce framework with a new identity (i.e., reentry attack). However, if the second sample trust function \( f_2 \) is used, it is no longer in a seller’s best interest to sell the defective items because if he returns to the community with a new identity, his lifetime indicator becomes zero and he loses all the credits that he accumulated over time. Consequently, he loses huge potential revenue that he could gain because of his lifetime parameter, i.e., buyers always prefer a seller with a longer lifetime (longer existence with a reasonable number of transactions) over a seller who is a newcomer.

We emphasize that this is just an example of rational trust modeling. In fact, the second sample function uses the lifetime parameter \( \ell_i \) to enforce trustworthiness and prevent the reentry attack. It is worth mentioning that different parameters can be incorporated into trust functions/reputation systems based on the context (e-commerce, mining in Blockchain, etc.), and consequently, different attacks can be prevented.

B. Technical Discussion on Detection Mechanisms

Detection mechanisms are required to reward or penalize miners in our reputation-based setting. In this section, we provide technical discussions and mechanisms by which noncooperative actions by miners (e.g., block withholding, selfish mining, DDoS attack, eclipse attack, stubborn mining, or upcoming attacks that are unknown) can be detected.
A mining pool can detect if it is under a block withholding attack with a relatively high accuracy. In fact, calculation of the partial proof-of-work is much easier than calculation of the full proof-of-work. Therefore, a mining pool can simply estimate its expected mining power in addition to its actual mining power. As a result, any difference between the expected and actual mining powers, which is above a certain threshold, can be an indication of a block withholding attack.

To determine which registered miner is the perpetrator, there are two possibilities. First, if the mining power of a miner/ally miners is high enough, the ratio of the full proof-of-work over the partial proof-of-work can indicate whether the miner/alliance is committing to the block withholding attack. Second, if the mining power is not high, the frequency of success to find the full proof-of-work is very low, and statistically, we may not be able to define if a miner is really committing to the block withholding attack. However, the latter case has a negligible (close to zero) impact on the mining process and can simply be ignored, i.e., block withholding attack by a single miner or miners with a low mining power cannot negatively affect the fair mining process.

As suggested by Eyal and Sirer (2014), an increase in the number of orphaned blocks can be an indication of selfish mining in the Blockchain. Furthermore, the amount of time taken to release consecutive blocks in the Blockchain can potentially provide evidence of selfish mining. Several researchers have investigated this issue through experimental analysis. In other words, two blocks in close succession should be a very rare incident when miners are honest, and this is more common when a miner/a group of miners quickly releases selfishly mined blocks to overcome the honest miners. As a result, it’s not hard to detect which miners are committing to the selfish mining.

As stated in Heilman et al. (2015), the eclipse attack has several signatures and properties that make it detectable, e.g., a flurry of short-lived incoming TCP connections from diverse IP addresses. Moreover, an attacker that suddenly connects a large number of nodes to the Bitcoin network could also be detected. Therefore, anomaly detection software systems that look for similar behaviors can be helpful to detect the attacker. Likewise, there are many other techniques in the security literature that can be used to detect the DDoS attack, stubborn mining, and so on.

Other methods might be used to detect bribes and illegal money exchanges among registered miners in the transparent network of Bitcoin (unless they exchange bribes outside of the Bitcoin network). This is how the government agencies usually detect money laundering/illegal money exchanges in the traditional banking system. In other words, detection of these bribes might be an indication of collusion—why miners from two competing pools should frequently exchange money with a certain amount.

C. Colluding Miner’s Dilemma

In this section we consider a scenario in which two miners (independent or from two different alliances) have to decide whether to collude with an attacker to disrupt another mining pool’s effort or not. Two collusion scenarios can be considered, i.e., a single miner colludes with the attacker, or multiple miners form a coalition with the attacker. We consider the latter case as it is the general case of the first scenario. It is worth mentioning that game-theoretical paradigms are usually used to analyze interaction between honest parties and attackers. However, we intend to model collusion between miners and an attacker in the context of Blockchain’s proof-of-work. In
our setting, we initially consider a 2-miner game, named \textit{colluding miner’s dilemma}, which may or may not collude with the attacker to disrupt the mining efforts of a targeted mining pool. We further extend this scenario to an \( n \)-miner game that is played repeatedly among all the miners of the Blockchain network for an unknown number of rounds.

In the 2-miner setting, shown in Table I, if both miners collude with the attacker, they each gain a half-unit of utility. In other words, the attacker’s budget will be equally shared between both miners. However, if one miner colludes with the attacker but the other one acts honestly, the colluding miner will receive one unit of utility from the attacker. As a result of this dilemma, collusion is Nash equilibrium, meaning that miners always collude because it is in their best interest to gain a higher utility. This is a realistic assumption where an attacker with a limited budget tries to disrupt the proof-of-work computation of a mining pool in favor of another alliance. Note that the budget is limited because mining reward is fixed in the Blockchain network.

<table>
<thead>
<tr>
<th>( m_i )</th>
<th>( m_j )</th>
<th>( \mathcal{H} ) Honest Mining</th>
<th>( \mathcal{D} ) Dishonest Mining</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_i )</td>
<td>( m_j )</td>
<td>( (0, 0) )</td>
<td>( (0, \Omega) )</td>
</tr>
<tr>
<td>( \mathcal{H} ): Honest Mining</td>
<td>( \Omega )</td>
<td>( (0, \Omega) )</td>
<td></td>
</tr>
<tr>
<td>( \mathcal{D} ): Dishonest Mining</td>
<td>( (0, 0) )</td>
<td>( \frac{\Omega}{2}, \frac{\Omega}{2} )</td>
<td></td>
</tr>
</tbody>
</table>

\( \Omega \): This payoff is the attacker's budget for disrupting the mining pool.

\( m_i \): This payoff is the miners’ utility for not colluding with the attacker.

\( m_j \): This payoff is the miners’ utility for colluding with the attacker.

TABLE I. PAYOFF IN COLLUDING MINER’S DILEMMA.

We approach the colluding miner’s dilemma by setting a socio-rational model (Nojoumian and Stinson 2012; Nojoumian 2014), i.e., a repeated game among rational foresighted players with public reputation values where these values directly affect players’ utilities over time, in which:

1) Each pool manager sends invitations to miners to form his mining pool for the proof-of-work computation. He not only tries to maximize his pool’s revenue, but also intends to protect his pool against any malicious activity. These invitations are defined based on miners’ trust values using a nonuniform probability distribution.

2) On the other hand, the attacker uses his limited budget to collude with the miners and consequently compromises the proof-of-work computation of a targeted pool.

In this setting, if a miner colludes with the attacker, he may gain some utility in the current round of the game; however, the pool managers will select that miner with a lower probability in the future if his malicious activity is detected. This is due to the reduction of his reputation value. See Nojoumian and Lethbridge 2008 and Nojoumian 2012 for a trust/reputation management system. Therefore, it will be in the best interest of the miners not to collude with the attacker because a malicious miner will lose his public reputation and thus lose many future mining opportunities with much larger gains.
D. Repeated Mining Game

We use a trust model that is resistant to the reentry attack in a repeated game setting. The miners try to maximize their utilities through the proof-of-work computation as well as collusion with the attacker or any dishonest mining strategies. We show that by using our proposed model, cooperation (not colluding with the attacker or committing to any malicious activity) is always Nash equilibrium because of a long-term utility function that we consider in our model in addition to a short-term utility function. Our model not only rewards honest miners but also penalizes colluding/dishonest miners. For the sake of simplicity and without loss of generality, two classes of actions are defined in our setting, i.e., dishonest/collude as a noncooperative action and honest/not collude as a cooperative action, similar to Nojoumian et al. (2017).

The mining game is repeatedly played for an unknown number of rounds. Each miner $m_{(jk, r_k)}$ has a public reputation value $r_k$, where the initial value is zero, and it is bounded as follows: $-1 \leq r_k \leq +1$. In addition, each miner’s action $a_j \in \{\mathcal{H}, \mathcal{D}_1\}$, where $\mathcal{H}$ and $\mathcal{D}$ denote honest mining and dishonest mining respectively, and $\perp$ indicates miner $m_{(jk, r_k)}$ has not been selected by any pool manager $M_{(i, p)}$ in the current round. Finally, each miner calculates two utility functions to select his action, that is, a long-term utility function $u_j$ and an actual utility function $u_j'$. Note that each round of the game consists of a sequence of block verification, for instance, after verifying a constant number of blocks or after a certain amount of time.

1) Suppose we have a nonuniform probability distribution over types of miners, i.e., honest, dishonest and new miners. Each pool manager $M_{(i, p)}$ sends invitations to a subset of miners based on this probability distribution in each round of the game.

2) Each miner $m_{(jk, r_k)}$ computes his long-term utility $u_j$, and then selects a new action from the action profile, i.e., employ honest or dishonest mining strategies.

3) Each $m_{(jk, r_k)}$ receives his short-term utility $u_j'$, i.e., the actual reward that each miner gains, at the end of each round of the game based on the proof-of-works’ outcomes.

4) The reputation values $r_k$ of the selected miners/ally miners are publicly updated based on each miner’s/ally’s behavior using a reputation system.

E. Colluding Miners’ Preferences

Let $u_j(\bar{a})$ denote $m_{(jk, r_k)}$’s long-term utility in outcome $\bar{a}$ by taking into account the current and future games, and let $u_j'(\bar{a})$ denote $m_{(jk, r_k)}$’s short-term utility in outcome $\bar{a}$ of the current game. Also, let $d_j(\bar{a}) \in \{0,1\}$ denote if the miner $m_{(jk, r_k)}$ has employed dishonest mining strategies in the current game, and define $\Delta(\bar{a}) = \sum_i d_j(\bar{a})$, that is, the total number of miners who have used dishonest mining strategies. Let $r_k^a(p)$ denote the reputation of $m_{(jk, r_k)}$ after outcome $\bar{a}$ in period $p$; note that $\bar{a}$ and $\bar{a}'$ are two different outcomes of our repeated game.

The miners’ preferences are as follows: $d_j(\bar{a}) = d_i(\bar{a}') \& r_k^a(p) > r_k^{a'}(p) \Rightarrow u_j(\bar{a}) > u_j(\bar{a}')$, that is, each miner $m_{(jk, r_k)}$ prefers to sustain a high reputation value over time despite employing honest or dishonest mining strategies as he can potentially gain a higher long-term utility; $d_i(\bar{a}) > d_i(\bar{a}') \Rightarrow u_j'(\bar{a}) > u_j'(\bar{a}')$, that is, if a miner $m_{(jk, r_k)}$ uses a dishonest mining strategy, he gains a short-term utility from the attacker, and finally; $d_i(\bar{a}) > d_i(\bar{a}') \&
\( \Delta(\bar{a}) < \Delta(\bar{a}') \Rightarrow u'_j(\bar{a}) > u'_j(\bar{a}') \), that is, if \( m_{(jk, rk)} \) employs dishonest mining strategies and the total number of dishonest miners in \( \bar{a} \) is less than the total number of dishonest miners in \( \bar{a}' \), the miner gains a higher short-term utility in \( \bar{a} \).

**F. Colluding Miners’ Utilities**

In our setting, the long-term utility function \( u_i \) is computed based on the utility that each miner \( m_{(jk, rk)} \) potentially gains or loses by considering both current and future games, i.e., taking into account all stated utility preferences. However, the short-term utility function \( u'_i \) is only calculated based on the current gain or loss in a given time interval, i.e., taking into account the last two utility preferences, as mentioned previously.

Let \( \varphi_j \) be the reward factor that is determined by each pool manager \( M_{(i, p)} \) based on \( r_k \) of each miner \( m_{(jk, rk)} \), and let \( \delta_j(\bar{a}) = r_k(\bar{a}) - r_k(\bar{a} - 1) \) be the difference of two consecutive reputation values. Note that \( \tau_j = |\delta_j(\bar{a})|/\delta_j(\bar{a}) \) is positive if the selected action in period \( p \) is \( H \) : honest mining, and it is negative if it is \( D \) : dishonest mining. Also, let \( \Omega > 0 \) be a unit of utility, for instance, $50. To satisfy the miners’ preferences, we compute the long-term utility \( u_j(\bar{a}) \) through the following linear combination:

\[
 u_j(\bar{a}) = \Omega \left( \tau_j \varphi_j + \frac{d_j(\bar{a})}{\Delta(\bar{a}) + 1} \right). \tag{1}
\]

Note that the actual utility \( u'_j(\bar{a}) \) only consists of the second and third terms, that is, \( u'_j(\bar{a}) = \Omega (d_j(\bar{a}) + d_j(\bar{a})/\Delta(\bar{a}) + 1) \). The first term of the utility function denotes miner \( m_{(jk, rk)} \) gains or loses \( \varphi_i \) units of utility in the future games due to his behavior as reflected in \( r_k \). This is due to \( \tau_j \) that depends on the miner’s reputation value \( r_k \). The second term illustrates miner \( m_{(jk, rk)} \) gains one unit of utility if he employs dishonest mining strategies or colludes with the attacker in the current game, and he loses this opportunity otherwise. Finally, the last term results in almost one unit of utility to be shared among all dishonest miners.

**VI. EVALUATION OF OUR MODEL USING GAME-THEORETICAL ANALYSES**

In this section, we evaluate our proposed reputation-based mining paradigm using game-theoretical analyses. We first consider a (2,2)-game that is played between two miners to show honest mining always dominates dishonest mining in our setting. We further extend this analysis to an \((n, n)\)-game that is played among \( n \) miners.

**Theorem 1:** In a (2,2)-game between two miners, honest mining \( H \) strictly dominates dishonest mining \( D \) when we use utility function \( u_j(\bar{a}) \), as defined in Eq. 1.

**Proof:** We compute \( u_j \) of each outcome for \( m_{(jk, rk)} \). Let \( m_{(jk', rk')} \) be the other miner.

1) If both miners employ honest mining strategies, \( \delta_j \) is positive, \( d_j = 0 \), and \( \Delta = 0 \):

\[
 (\delta_j > 0, d_j = 0, \Delta = 0) \Rightarrow u_j^H(\bar{a}^H) = \Omega \varphi_j.
\]
2) If only \( m_{(jk,r_k)} \) uses honest mining strategies, \( \delta_j \) is positive, \( d_j = 0 \) since \( m_{(jk,r_k)} \) has not colluded, and \( \Delta = 1 \) since \( m_{(jk',r_{k'})} \) has used dishonest mining strategies: 
\[
(\delta_j > 0, d_j = 0, \Delta = 1) \Rightarrow u_j(\mathcal{H}; \mathcal{D}) = \Omega \phi_j.
\]

3) If only \( m_{(jk',r_{k'})} \) uses honest mining strategies, \( \delta_j \) is negative, \( d_j = 1 \) since miner \( m_{(jk,r_k)} \) has employed dishonest mining strategies, and \( \Delta = 1 \): 
\[
(\delta_j < 0, d_j = 1, \Delta = 1) \Rightarrow u_j(\mathcal{H}; \mathcal{D}) = \Omega(-\phi_j + 1.50).
\]

4) If both miners employ dishonest mining strategies, \( \delta_j \) is negative, \( d_j = 1 \), and \( \Delta = 2 \) because both miners have colluded: 
\[
(\delta_j < 0, d_j = 1, \Delta = 2) \Rightarrow u_j(\mathcal{H}; \mathcal{D}) = \Omega(-\phi_j + 1.33).
\]

If reward factor \( \phi \geq 1.5 \), which is defined by each pool manager \( M_{(i,p_i)} \), we will have the following payoff inequalities that prove our theorem:

\[
\frac{m_{(jk,r_k)}: \text{honest mining}}{u_j(\mathcal{H}; \mathcal{D})(\bar{d}) = u_j(\mathcal{H}; \mathcal{D})(\bar{d})} > \frac{m_{(jk,r_k)}: \text{dishonest mining}}{u_j(\mathcal{H}; \mathcal{D})(\bar{d}) = u_j(\mathcal{H}; \mathcal{D})(\bar{d})}
\]

Likewise, if we assume \( \phi \) is at least 1.5 (note that the minimum value is defined based on the model’s parameters), the payoff matrix is as follows in Table II:

<table>
<thead>
<tr>
<th>( m_{(jk,r_k)} ): honest mining</th>
<th>( \mathcal{H} ): Honest Mining</th>
<th>( \mathcal{D} ): Dishonest Mining</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_{(jk',r_{k'})} ): Dishonest Mining</td>
<td>( (1.5, 1.5) )</td>
<td>( (1.5, 0) )</td>
</tr>
<tr>
<td>( \mathcal{H} ): Honest Mining</td>
<td>( (0, 1.5) )</td>
<td>( (-0.17, -0.17) )</td>
</tr>
</tbody>
</table>

Table II: (2,2)-Game Between Two Miners.

As shown, honest mining is always Nash equilibrium in our reputation-based mining paradigm. To expand our proof to a case with \( n \) miners, let \( \mathcal{H} \) or \( \mathcal{D} \) denote \( m_{(jk,r_k)} \) employs honest mining strategies (or dishonest mining strategies), and let \( \mathcal{H}_j \) or \( \mathcal{D}_j \) denote, excluding \( m_{(jk,r_k)} \), all other miners use honest mining strategies (or dishonest mining strategies), and finally, let \( \mathcal{M}_{-j} \) denote, excluding \( m_{(jk,r_k)} \), some miners employ honest mining strategies and some of them use dishonest mining strategies.

**Theorem 2:** In an \( (n, n) \)-game among \( n \) miners, honest mining \( \mathcal{H} \) strictly dominates dishonest mining \( \mathcal{D} \) when we use the utility function \( u_j(\bar{d}) \), as defined in Eq. 1.
Proof: We compute the utility of each outcome in different scenarios. Let $n > k \geq 2$.

1) If all miners employ honest mining strategies, or $m_{(jk,rk)}$ and $k - 1$ miners employ honest mining strategies, or only $m_{(jk,rk)}$ conduct honest mining, as a result, $\delta_j$ is positive, $d_j = 0$, and $\Delta \in s = \{0, n - k, n - 1\}$:

$$(\delta_j > 0, d_j = 0, \Delta \in s) \Rightarrow u_j^{(M_\text{h} \text{h})} = u_j^{(M_\text{h} \text{m} \_j)} = u_j^{(M_\text{h} \text{d} \_j)} = \Omega \varphi_j.$$ 

2) If only $m_{(jk,rk)}$ uses dishonest mining strategies, $\delta_j$ is negative, $d_j = 1$ and $\Delta = 1$:

$$(\delta_j < 0, d_j = 1, \Delta = 1) \Rightarrow u_j^{(M_\text{d} \_j)} = \Omega(-\varphi_j + 1.5).$$

3) If $m_{(jk,rk)}$ as well as $k - 1$ miners employ dishonest mining strategies, and the rest of them use honest mining strategies:

$$(\delta_j < 0, d_j = 1, \Delta = k) \Rightarrow u_j^{(M_\text{d} \_j)} = \Omega(-\varphi_j + \frac{k + 2}{k + 1}).$$

4) If all miners use dishonest mining strategies, $\delta_j$ is negative, $d_j = 1$, and $\delta = n$ because no one has conducted honest mining:

$$(\delta_j < 0, d_j = 1, \Delta = n) \Rightarrow u_j^{(M_\text{d} \_j)} = \Omega(-\varphi_j + \frac{n + 2}{n + 1}).$$

Our analysis will be as follows. Let *-j be $M_\text{h} \_j$, $M_\text{d} \_j$ or $M_\text{d} \_j$. It is easy to show that:

$$1.5 > \frac{k + 2}{k + 1} > \frac{n + 2}{n + 1} \text{ when } n > k \geq 2.$$

Likewise, if we assume $\varphi_j$ is at least 1.5, honest mining or not colluding with the attacker is always Nash equilibrium. As a result, it is always in $m_{(jk,rk)}$’s best interest to use honest mining strategies no matter what other miners do:

$$u_j^{(M_\text{h} \_j)}(\bar{a}) > u_j^{(M_\text{d} \_j)}(\bar{a}).$$

VII. CONCLUDING REMARKS

In this chapter, we proposed a new reputation-based mining paradigm for the proof-of-work computation in the Blockchain. We first illustrated the problem of dishonest mining, demonstrated our proposed model, and subsequently, provided a candidate solution concept to the aforementioned problem. Note that, by dishonest mining, we refer to any malicious activity against other mining pools or competitors, such as block withholding attack, selfish mining, eclipse attack, and stubborn mining, to name a few.
Our proposed mining game is repeatedly played among a set of pool managers and miners where the reputation value of each miner or mining ally is continuously measured by a trust management scheme that is resistant to the reentry attack. At each round of the game, pool managers send invitations only to a subset of miners based on a nonuniform probability distribution defined by the miners' reputations. It is worth mentioning that each round of the game consists of a sequence of block verification, for instance, after verifying a constant number of blocks or after a certain amount of time.

We showed that, by using our proposed solution concept, honest mining becomes Nash equilibrium in our setting. In other words, it will not be in the best interest of the miners to disrupt the proof-of-work computation or commit to dishonest mining even by gaining a short-term utility. This is due to the consideration of a long-term utility function in our model and its impact on the miners' utilities over time. As our future work, we are interested in implementing our proposed game through a simulation-based approach using real data from the Bitcoin network.

VIII. ACKNOWLEDGMENT

We would like to thank Florida Atlantic University, US Air Force Research Laboratory, and US Army Research Laboratory for supporting this project.

REFERENCES


