Incentivizing Blockchain Miners to Avoid Dishonest Mining Strategies by a Reputation-Based Paradigm

M. Nojoumian[1(✉)], A. Golchubian[1], L. Njilla[2], K. Kwiat[2], and C. Kamhoua[3]

[1] Department of CEECS, Florida Atlantic University, Boca Raton, FL, USA
{mnojoumian,agolchub}@fau.edu

[2] Cyber Assurance Branch, Air Force Research Lab, Rome, NY, USA
{laurent.njilla,kevin.kwiat}@us.af.mil

[3] Network Security Branch, Army Research Lab, Adelphi, MD, USA
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 not only are incentivized to conduct honest mining but also disincentivized to commit to 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 non-uniform 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 overtime.

Keywords: Proof-of-work · Reputation systems · Game theory

DISTRIBUTION A. Approved for public release: distribution unlimited. Case Number: 88ABW-2017-4327, Dated 08 Sept 2017.

K. Arai et al. (Eds.): SAI 2018, AISC 857, pp. 1118–1134, 2019.
https://doi.org/10.1007/978-3-030-01177-2_81

1 Introduction

Security games are mainly designed and utilized to model interaction between attackers and defenders [1,2]. 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 in order to shift the attack surface whereas the attackers can manipulate the system in order 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 utilized is the verification of transactions in the context of digital currencies, e.g., Bitcoin [3], 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 competitions 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 [4] introduces 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 [5] introduce *selfish mining* where the players of a coalition keep their discovered blocks private and continue to verify more blocks privately until they get a sub-chain that its length is threatened. As a result, selfish players receive the reward. Johnson et al. [6] 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. Recently, more attacks were introduced, e.g., *eclipse attack* [7] that makes a node invisible in the network, or *stubborn mining* as a generalization of the selfish mining [8].

We therefore propose a new reputation-based framework in which miners not only are incentivized to conduct honest mining but also disincentivized to commit to 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 paper is organized as follows. Section 2 provides preliminary materials on digital currencies and game theory. Section 3 reviews the existing digital currency literature where game theory is utilized. Section 4 illustrates our model. Section 5 explains how our reputation-based scheme works. Finally, Sect. 6 concludes with final remarks.

2 Preliminaries

2.1 Digital Currencies: Terminologies and Mechanics

In the digital currency frameworks, specifically Bitcoin, transactions are grouped in blocks in order 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 overtime 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.

The high-level idea of the proof-of-work, verification, or mining is shown in Fig. 1. 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's 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.

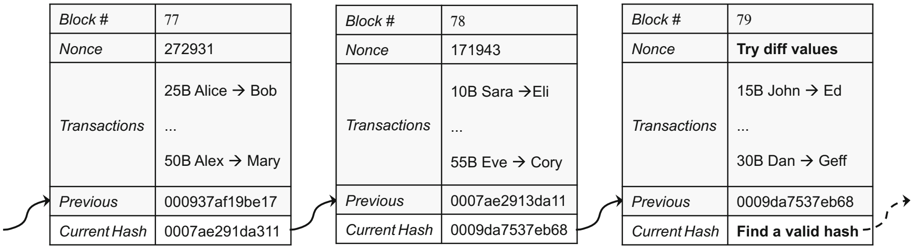


Fig. 1. Blockchain and mining.

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.

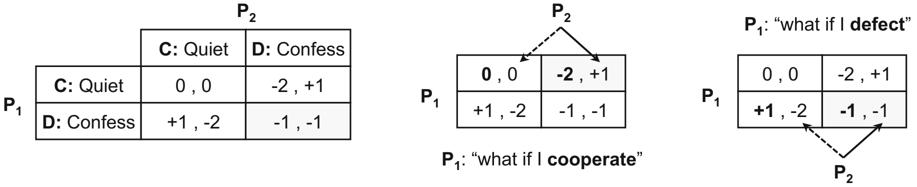
2.2 Game Theory: Basic Notions and Definitions

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 incentivised by agreement. However, in *non-cooperative games*, the players cannot form any agreement to coordinate their behaviors. In other words, any cooperation among the players must be self-enforcing.

The *prisoner's dilemma*, as illustrated in Fig. 2, is an example of non-cooperative games. In this setting, two possible actions are considered: $\mathscr{C}$: *keep quiet* (cooperation) and $\mathscr{D}$: *confess* (defection). In the pay-off (utility) matrix, $+1, 0, -1$, and -2 denote freedom, jail for one year, jail for two years, and jail for three years, respectively. The outcome of this game will be $(\mathscr{D}, \mathscr{D})$ because of the *Nash equilibrium* concept, while the ideal outcome is $(\mathscr{C}, \mathscr{C})$. To better understand the notion of Nash equilibrium, and consequently, why the game has such an outcome, consider the following two possible scenarios:

(1) If player P_1 selects $\mathscr{C}$ (1st row), then P_2 will select $\mathscr{D}$ (2nd column) since $+1 > 0$.

(2) If player P_1 selects $\mathscr{D}$ (2nd row), then P_2 will select $\mathscr{D}$ (2nd column) since $-1 > -2$.



		P2	
		C: Quiet	**D:** Confess
P1	**C:** Quiet	0 , 0	-2 , +1
	D: Confess	+1 , -2	-1 , -1

Fig. 2. Nash equilibrium in Prisoner's dilemma.

In other words, regardless of whether player P_1 cooperates or defects, player P_2 will always defect. Since the pay-off matrix is symmetric, P_1 will also defect regardless of whether P_2 cooperates or defects. In fact, since the players are not able to coordinate their behavior, the final outcome will be $(\mathscr{D}, \mathscr{D})$.

We briefly review some well-known game-theoretic concepts [9] for our further analyses and discussions.

Definition 1: Let $\mathscr{A} \overset{\text{def}}{=} \mathscr{A}_1 \times \mathscr{A}_2 \times \cdots \times \mathscr{A}_n$ be an action profile for n players, where $\mathscr{A}_i$ denotes the set of possible actions of player P_i. A *game* $\Gamma = (\mathscr{A}_i, u_i)$ for $1 \leq i \leq n$, consists of $\mathscr{A}_i$ and a utility function $u_i : \mathscr{A} \mapsto \mathbb{R}$ for each player P_i. We refer to a vector of actions $\vec{a} = (a_1, \ldots, a_n) \in \mathscr{A}$ 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 $\vec{a}$ to $\vec{a}'$ iff $u_i(\vec{a}) > u_i(\vec{a}')$, and he *weakly prefers* outcome $\vec{a}$ to $\vec{a}'$ if $u_i(\vec{a}) \geq u_i(\vec{a}')$.

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

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

Definition 4: Let $\mathscr{S}_{-i} \overset{\text{def}}{=} \mathscr{S}_1 \times \cdots \times \mathscr{S}_{i-1} \times \mathscr{S}_{i+1} \times \cdots \times \mathscr{S}_n$. A strategy $\sigma_i \in \mathscr{S}_i$ (or an action) is *weakly dominated* by $\sigma'_i \in \mathscr{S}_i$ (or another action) with respect to $\mathscr{S}_{-i}$ if:

1. For all $\vec{\sigma}_{-i} \in \mathscr{S}_{-i}$, it holds that $u_i(\sigma_i, \vec{\sigma}_{-i}) \leq u_i(\sigma'_i, \vec{\sigma}_{-i})$.
2. There is a $\vec{\sigma}_{-i} \in \mathscr{S}_{-i}$ such that $u_i(\sigma_i, \vec{\sigma}_{-i}) < u_i(\sigma'_i, \vec{\sigma}_{-i})$.

This means player P_i can never improve its utility by playing σ_i, and he can sometimes improve it by not playing σ_i. A strategy $\sigma_i \in \mathscr{S}_i$ is *strictly dominated* if player P_i can always improve its utility by not playing σ_i.

3 Literature Review

Even though the concept of Blockchain is relatively new, introduced by an unknown author or authors in 2008 [10], 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.

Johnson et al. [6] 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 [11] that shows 17.1% of small mining pools have been 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. [12] 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 [13] 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 et al. [14] study Bitcoin as a consensus game and consider the economics of Bitcoin from the mining perspective to determine whether there exists any incentive 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 [15] 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. [16] 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.

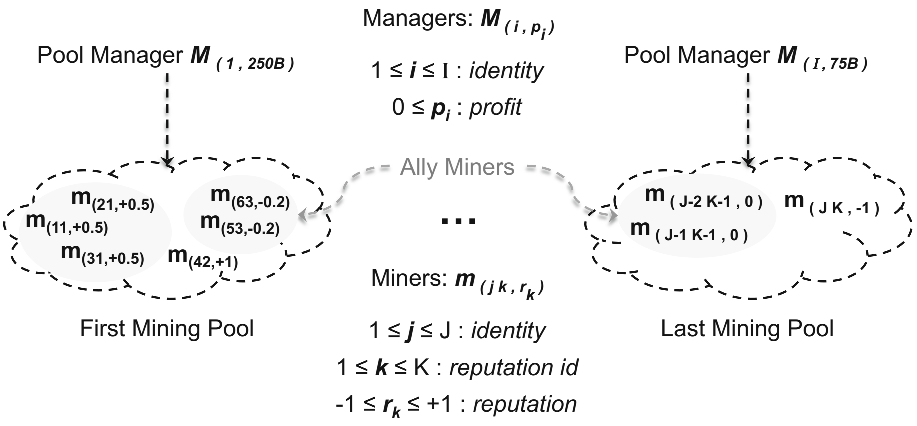
Luu et al. [17] scrutinize the block withholding attack on mining pools, introduced by Rosenfeld [4]. They show that the attack always has incentive when looking at a long term operation, but it may not be profitable for short term duration. Eyal [18] studies the same subject and concludes that when two pools attack each other, it results in a version of the prisoner's dilemma, named the *Miner's Dilemma*. Lewenberg et al. [19] introduce a modification to the Blockchain protocol to allow for inclusion of forked blocks with the aim of increasing the rate of operation. The authors then provide a game theoretic model of the competition for fees between the nodes under the new protocol.

4 Our Reputation-Based Mining Model and Setting

As illustrated in Fig. 3, 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 $m_{(jk,r_k)}$ 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 $\mathscr{D}$: *dishonest mining*, or conduct the proof-of-work honestly, denoted by $\mathscr{H}$: *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 honest 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

i	1	2	3	...	I − 1	I
p_i	250B	125B	0B	...	200B	75B


Managers: $M_{(i, p_i)}$
Pool Manager $M_{(1, 250B)}$
$1 \leq i \leq$ I : identity
$0 \leq p_i$: profit
Pool Manager $M_{(I, 75B)}$
Ally Miners
$m_{(21,+0.5)}$
$m_{(11,+0.5)}$
$m_{(31,+0.5)}$
$m_{(63,-0.2)}$
$m_{(53,-0.2)}$
$m_{(42,+1)}$
...
$m_{(J\text{-}2\ K\text{-}1,\ 0)}$
$m_{(J\text{-}1\ K\text{-}1,\ 0)}$
$m_{(J K, -1)}$
Miners: $m_{(j k, r_k)}$
First Mining Pool
Last Mining Pool
$1 \leq j \leq$ J : identity
$1 \leq k \leq$ K : reputation id
$-1 \leq r_k \leq +1$: reputation

k	1	2	3	...	K − 1	K
j	1, 2, 3	4	5, 6	...	J-2 , J-1	J
r_k	+0.5	+1	-0.2	...	0	-1

Fig. 3. Architecture of our reputation-based setting.

partial proof-of-work, detection of any malicious activity such as collusion with other miners, selfish-mining, or contribution to a distributed denial-of-service 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, while members of a coalition can build reputation all together through r_k by collaborations overtime, they are all responsible for malicious activities triggered even by a single member of their coalition.

This leads to the notion of *neighborhood-watch* meaning that each member of an alliance is incentivised 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.

Once in a while, 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 non-uniform 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 chose 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 re-entry attack (that is, cheat and come back to the scheme with a new identity j). We utilize the proposed idea of *rational trust modeling* [20] 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 joins 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 also move to a new mining pool when the rearrangement is occurred. This approach not only leads to less managerial overheads for the pool managers but also it creates a framework where practical implementations of preventive and detective protocols become possible.

5 Mining in Our Reputation-Based Model

Since our approach is designed using a reputation-based paradigm, it's necessary to utilize a reputation/trust model that is resistant to the well-known *re-entry 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.

5.1 Prevention of the Re-entry Attack

To deal with the re-entry attack in our reputation-based scheme, we utilize the proposed approach of *rational trust modeling* [20]. We provide a high-level description as how this modeling technique works. Suppose there exist two trust functions as follows. The first function $f_1(\mathscr{T}_i^{p-1}, \alpha_i)$ has two inputs, that is, trust value $\mathscr{T}_i^{p-1}$ of player P_i in period $p-1$ and action α_i (cooperation or defection) selected by player P_i in period $p-1$. This function computes the updated trust value $\mathscr{T}_i^{p}$ of player P_i for the next round p based on these two inputs. However, the second function $f_2(\mathscr{T}_i^{p-1}, \alpha_i, \ell_i)$ has an extra input value that defines the player's lifetime, denoted by ℓ_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 $\alpha_i = \alpha_j = \mathscr{C}$ and their current trust values are equal $\mathscr{T}_i^{p-1} = \mathscr{T}_j^{p-1}$ 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., $\mathscr{T}_i^{p} > \mathscr{T}_j^{p}$. 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"non-defective" versions of the same item with less revenue. If the first sample function f_1 is utilized 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., re-entry attack). However, if the second sample trust function f_2 is utilized, it's 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 has accumulated overtime. Consequently, he loses a 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 a rational trust modeling. In fact, the second sample function uses the lifetime parameter ℓ_i to enforce trustworthiness and prevent the re-entry 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.

5.2 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 non-cooperative actions by miners (e.g., block withholding, selfish mining, distributed denial-of-service 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 be simply 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 [5] who initially introduced the selfish mining, 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. This issue has been investigated by several researchers through experimental analysis[1]. 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 [7], 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 utilized to detect the distributed denial-of-service attack, stubborn mining, etc.

[1] http://scienceblogs.com/builtonfacts/2014/01/11/is-bitcoin-currently-experiencing-a-selfish-miner-attack/.

Besides, 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.

5.3 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 utilized 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 *colluding miner's dilemma*, that may/may not collude with the attacker to disrupt the mining efforts of a targeted mining pool. We further extend this scenario to a 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 1, 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's in their best interest to gain a higher utility. This is a realistic assumption where an attacker with a limited budget tries to disrupts 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 1. Payoff in colluding miner's dilemma

$m_{(jk,r_k)}$ \ $m_{(j'k',r'_k)}$	$\mathscr{H}$: Honest Mining	$\mathscr{D}$: Dishonest Mining
$\mathscr{H}$: Honest Mining	$(\text{฿}0, \text{฿}0)$	$(\text{฿}0, \text{฿}\Omega)$
$\mathscr{D}$: Dishonest Mining	$(\text{฿}\Omega, \text{฿}0)$	$(\text{฿}\frac{\Omega}{2}, \text{฿}\frac{\Omega}{2})$

We approach the colluding miner's dilemma by setting a socio-rational model [21,22] (that is, a repeated game among rational foresighted players with public reputation values where these values directly affect players' utilities overtime) 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 non-uniform probability distribution.

(2) On the other hand, the attacker uses his limited budget to collude with the miners, and consequently, compromise 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, that miner will be selected by the pool managers with a lower probability in the future if his malicious activity is detected. This is due to the reduction of his reputation value, see [23,24] 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 consequently, he will lose many future mining opportunities with a much larger gain.

5.4 Repeated Mining Game

We use a trust model that is resistant to the re-entry 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 non-cooperative action and *honest/not collude* as a cooperative action, similar to [25].

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 $\alpha_j \in \{\mathscr{H}, \mathscr{D}, \perp\}$, where $\mathscr{H}$ and $\mathscr{D}$ denote *honest mining* and *dishonest mining*, respectively, and $\perp$ indicates miner $m_{(jk,r_k)}$ has not been selected by any pool manager $M_{(i,p_i)}$ 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 non-uniform probability distribution over types of miners, i.e., honest, dishonest and new miners. Each pool manager $M_{(i,p_i)}$ 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/alliance's behavior using a reputation system.

5.5 Colluding Miners' Preferences

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

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

5.6 Colluding Miners' Utilities

In our setting, the long-term utility function u_i is computed based on the utility that each miner $m_{(jk,r_k)}$ 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 in Sect. 5.5.

Let φ_j be the reward factor that is determined by each pool manager $M_{(i,p_i)}$ based on r_k of each miner $m_{(jk,r_k)}$, and let $\delta_j(\vec{a}) = r_k^{\vec{a}}(p) - r_k^{\vec{a}}(p-1)$ be the difference of two consecutive reputation values. Note that $\tau_j = |\delta_j(\vec{a})| / \delta_j(\vec{a})$ is positive if the selected action in period p is $\mathscr{H}$: *honest mining*, and it is negative, if it is $\mathscr{D}$: *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 $u_j(\vec{a})$ through the following linear combination:

$$u_j(\vec{a}) = \Omega\Big(\tau_j\varphi_j + d_j(\vec{a}) + \frac{d_j(\vec{a})}{\Delta(\vec{a})+1}\Big). \tag{1}$$

Note that the actual utility $u'_j(\vec{a})$ only consists of the second and third terms, that is, $u'_j(\vec{a}) = \Omega(d_j(\vec{a}) + d_j(\vec{a})/(\Delta(\vec{a})+1))$. The first term of the utility function denotes miner $m_{(jk,r_k)}$ gains or loses φ_i units of utility in the future games due to his behavior as reflected in r_k. This is due to τ_j that depends on the miner's reputation value r_k. The second term illustrates miner $m_{(jk,r_k)}$ 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.[2]

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

Theorem 2: In a (n,n)-game among n miners, honest mining $\mathscr{H}$ strictly dominates dishonest mining $\mathscr{D}$ when we use the utility function $u_j(\vec{a})$, as defined in (1).

6 Concluding Remarks

In this paper, 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 re-entry attack. At each round of the game, pool managers send invitations only to a subset of miners based on a non-uniform 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

[2] The proofs of both theorems and the related mathematical analyses will be provided in the complete version of this paper.

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 overtime. As our future work, we are interested in implementing our proposed game through a simulation-based approach using real data from the Bitcoing network.

Acknowledgment. Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-18-1-0483. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

1. Roy, S., Ellis, C., Shiva, S., Dasgupta, D., Shandilya, V., Wu, Q.: A survey of game theory as applied to network security. In: 43rd Hawaii International Conference on System Sciences (HICSS). IEEE, pp. 1–10 (2010)
2. Liang, X., Xiao, Y.: Game theory for network security. IEEE Commun. Surv. Tutor. **15**(1), 472–486 (2013)
3. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
4. Rosenfeld, M.: Analysis of bitcoin pooled mining reward systems. arXiv preprint arXiv:1112.4980 (2011)
5. Eyal, I., Sirer, E.G.: Majority is not enough: Bitcoin mining is vulnerable. In: International Conference on Financial Crypto and Data Security, pp. 436–454. Springer (2014)
6. Johnson, B., Laszka, A., Grossklags, J., Vasek, M., Moore, T.: Game-theoretic analysis of DDoS attacks against bitcoin mining pools. In: International Conference on Financial Cryptography and Data Security, pp. 72–86. Springer (2014)
7. Heilman, E., Kendler, A., Zohar, A., Goldberg, S.: Eclipse attacks on bitcoin's peer-to-peer network. In: USENIX Security Symposium, pp. 129–144 (2015)
8. Nayak, K., Kumar, S., Miller, A., Shi, E.: Stubborn mining: generalizing selfish mining and combining with an eclipse attack. In: First European Symposium on Security and Privacy (EuroS&P), pp. 305–320. IEEE (2016)
9. Osborne, M.J., Rubinstein, A.: A Course in Game Theory. MIT press, Cambridge (1994)
10. Satoshi, N.: Bitcoin: a peer-to-peer electronic cash system. Bitcoin. org, [cit. 2014-11-13] (2008). https://bitcoin.org/bitcoin.pdf
11. Vasek, M., Thornton, M., Moore, T.: Empirical analysis of denial-of-service attacks in the bitcoin ecosystem. In: International Conference on Financial Cryptography and Data Security, pp. 57–71. Springer (2014)
12. Babaioff, M., Dobzinski, S., Oren, S., Zohar, A.: On bitcoin and red balloons. In: 13th ACM Conference on Electronic Commerce, pp. 56–73. ACM (2012)
13. Antonopoulos, A.M.: Mastering Bitcoin: Unlocking Digital Cryptocurrencies. O'Reilly Media Inc, Sebastopol (2014)
14. Kroll, J.A., Davey, I.C., Felten, E.W.: The economics of bitcoin mining, or bitcoin in the presence of adversaries. In: Proceedings of WEIS, vol. 2013 (2013)

15. Barber, S., Boyen, X., Shi, E., Uzun, E.: Bitter to better-how to make bitcoin a better currency. In: International Conference on Financial Crypto and Data Security, pp. 399–414. Springer (2012)
16. Carlsten, M., Kalodner, H., Weinberg, S.M., Narayanan, A.: On the instability of bitcoin without the block reward. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 154–167. ACM (2016)
17. Luu, L., Saha, R., Parameshwaran, I., Saxena, P., Hobor, A.: On power splitting games in distributed computation: the case of bitcoin pooled mining. In: Computer Security Foundations Symposium (CSF), 2015 IEEE 28th, pp. 397–411. IEEE (2015)
18. Eyal, I.: The miner's dilemma. In: 2015 IEEE Symposium on Security and Privacy (SP), pp. 89–103. IEEE (2015)
19. Lewenberg, Y., Sompolinsky, Y. Zohar, A.: Inclusive block chain protocols. In: International Conference on Financial Crypto and Data Security, pp. 528–547. Springer (2015)
20. Nojoumian, M.: Rational trust modeling. Arxiv, 2017, p. 12 https://arxiv.org/abs/1706.09861
21. Nojoumian, M., Stinson, D.R.: Socio-rational secret sharing as a new direction in rational cryptography. In: 3rd International Conference on Decision and Game Theory for Security (GameSec), LNCS, vol. 7638, pp. 18–37. Springer (2012)
22. Nojoumian, M.: Generalization of socio-rational secret sharing with a new utility function. In: 12th IEEE Annual International Conference on Privacy, Security and Trust, pp. 338–341 (2014)
23. Nojoumian, M., Lethbridge, T.C.: A new approach for the trust calculation in social networks. In: E-business and Telecommunication Networks: 3rd International Conference on E-Business, Best Papers. CCIS, vol. 9, pp. 64–77. Springer (2008)
24. Nojoumian, M.: Novel secret sharing and commitment schemes for cryptographic applications. Ph.D. dissertation, Department of Computer Science, UWaterloo, Canada (2012)
25. Nojoumian, M., Golchubian, A., Saputro, N., Akkaya, K.: Preventing collusion between SDN defenders and attackers using a game theoretical approach. In: Infocom: Adv in Software Defined & Context Aware Cognitive Radio Net. IEEE (2017). p. 6