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

M. Nojoumian and A. Golchubian
Department of CEECS
Florida Atlantic University, Boca Raton, FL
f mnojoumian,agolchubg@fau.edu

L. Njilla and K. Kwiat
Cyber Assurance Branch
Air Force Research Lab, Rome, NY
f laurent.njilla,kevin.kwiatg@us.af.mil

C. 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 rst 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, sel sh 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 rst 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 de ned by the miners' reputation values. We show that by using our proposed solution concept, honest mining becomes

The rest of this paper 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 utilized. Section IV illustrates our model. Section V explains how our reputation-based scheme works. Finally, Section VI concludes with final remarks.

II. Preliminaries

A. 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 Figure 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.

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 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 Figure 2, is an example of non-cooperative games. In this setting, two possible actions are considered: C: keep quiet (cooperation) and 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 (D, D) because of the *Nash equilibrium* concept, while the ideal outcome is (C, 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 C (1st row), then P_2 will select D (2nd column) since $-1 > 0$.
2) If player P_1 selects D (2nd row), then P_2 will select D (2nd column) since $-1 > -2$.

In other words, regardless of whether player P_2 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 (D, D).

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

Definition 1: Let $A \stackrel{def}{=} A_1 \times A_2 \times \cdots \times A_n$ be an action profile for n players, where A_i denotes the set of possible actions of player P_i. A game $G = (A_i, u_i)$ for $1 \le i \le n$, consists of A_i and a utility function $u_i : A \mapsto \mathbb{R}$ for each player P_i. We refer to a vector of actions $a = (a_1, \ldots, a_n) \in A$ 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 a to a' iff $u_i(a) > u_i(a')$, and he weakly prefers outcome a to a' if $u_i(a) \ge u_i(a')$.

To allow the players to follow randomized strategies, we define s_i as a probability distribution over A_i for a player P_i. This means he samples $a_i \in A_i$ according to s_i. A strategy is said to be a *pure-strategy* if each s_i assigns probability 1 to a certain action, otherwise, it is said to be a *mixed-strategy*. Let $s = (s_1, \ldots, s_n)$ be the vector of players' strategies, and let $(s_i', s_{-i}) = (s_1, \ldots, s_{i-1}, s_i', s_{i+1}, \ldots, s_n)$, where P_i replaces s_i by s_i' and all the other players' strategies remain unchanged. Therefore, $u_i(s)$ denotes the expected utility of P_i under the

strategy vectors $\mathbf{s}$. A player's goal is to maximize $u_i(\mathbf{s})$. In the following definitions, one can substitute action $a_i \in A_i$ with its probability distribution $s_i \in S_i$, or vice versa.

Fig. 2. Nash Equilibrium in Prisoner's Dilemma.

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.

IV. Our Reputation-Based Mining Model and Setting

As illustrated in Figure 3, our model consists of a set of pool managers $M_{(i;p_i)}$ who form coalitions for the proof-of-work computations, for $1 \quad i \quad I$, where $0 \quad p_i$ 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 $1 \quad j \quad J$ and $1 \quad k \quad K$, where $1 \quad r_k \quad +1$ 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 D: dishonest mining, or conduct the proof-of-work honestly, denoted by 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 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 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 altogether 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). 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

We emphasize that this is just an example of a rational trust modeling. In fact, the second sample function uses the lifetime parameter τ_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.

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 non-cooperative actions by miners (e.g., block withholding, sel sh 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.

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 de ned 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.

D. 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 utilityfunction. 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 de ned in our setting, i.e., dishonest/colluded as a non-cooperative action and honest/not colluded 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: $-1 \le r_k \le +1$. In addition, each miner's action $a_j \in \{H, D, ?\}$, where H and D denote honest mining and dishonest mining respectively, and $?$ 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 veri cation, 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 pro le, 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.

E. Colluding Miners' Preferences

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

The miners' preferences are as follows: $d_i(a) = d_i(a') \;\&\; r^a_k(p) > r^{a'}_k(p) \;) \; u_j(a) > u_j(a')$, 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. $d_i(a) > d_i(a') \;) \; u'_j(a) > u'_j(a')$, that is, if a miner $m_{(jk;r_k)}$ utilizes a dishonest mining strategy, he gains a short-term utility from the attacker, and nally; $d_i(a) > d_i(a') \;\&\; D(a) < D(a') \;) \; u'_j(a) > u'_j(a')$, that is, if $m_{(jk;r_k)}$ employs dishonest mining strategies and the total number of dishonest miners in a is less than the total number of dishonest miners in a', the miner gains a higher short-term utility in a.

F. Colluding Miners' Utilities

In our setting, the long-term utility function u_i is computed based on the utility that each miner $m_{(jk;r_k)}$ potentially gains or

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.

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

Theorem 2: In a (n,n)-game among n miners, honest mining H strictly dominates dishonest mining D when we use the utility function $u_j(\cdot a)$, as defined in Eqn (1).[3]

VI. 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 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.

VII. Acknowledgment

We would like to thank Florida Atlantic University, Air Force Research Lab, and Army Research Lab for supporting this project. We also thank the anonymous reviewers for their constructive feedback and inspiring comments.

References

[1] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu, "A survey of game theory as applied to network security," in 43rd Hawaii Int. Conference on System Sciences (HICSS), pp. 1–10, IEEE, 2010.

[2] X. Liang and Y. Xiao, "Game theory for network security," IEEE Communications Surveys & Tutorials, vol. 15, no. 1, pp. 472–486, 2013.

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[4] M. Rosenfeld, "Analysis of bitcoin pooled mining reward systems," arXiv preprint arXiv:1112.4980, 2011.

[5] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in Int. conf. on financial crypto and data security, pp. 436–454, Springer, 2014.

[6] B. Johnson, A. Laszka, J. Grossklags, M. Vasek, and T. Moore, "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] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network.," in USENIX Security Symposium, pp. 129–144, 2015.

[8] K. Nayak, S. Kumar, A. Miller, and E. Shi, "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] M. J. Osborne and A. Rubinstein, A Course in Game Theory. MIT press, 1994.

[10] N. Satoshi, "Bitcoin: A peer-to-peer electronic cash system," Bitcoin. org, [cit. 2014-11-13]: https://bitcoin.org/bitcoin.pdf, 2008.

[11] M. Vasek, M. Thornton, and T. Moore, "Empirical analysis of denial-of-service attacks in the bitcoin ecosystem," in Int. Conference on Financial Cryptography and Data Security, pp. 57–71, Springer, 2014.

[12] M. Babaioff, S. Dobzinski, S. Oren, and A. Zohar, "On bitcoin and red balloons," in 13th ACM conference on electronic commerce, pp. 56–73, ACM, 2012.

[13] A. M. Antonopoulos, Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.", 2014.

[14] J. A. Kroll, I. C. Davey, and E. W. Felten, "The economics of bitcoin mining, or bitcoin in the presence of adversaries," in Proceedings of WEIS, vol. 2013, 2013.

[15] S. Barber, X. Boyen, E. Shi, and E. Uzun, "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] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "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] L. Luu, R. Saha, I. Parameshwaran, P. Saxena, and A. Hobor, "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] I. Eyal, "The miner's dilemma," in Security and Privacy (SP), 2015 IEEE Symposium on, pp. 89–103, IEEE, 2015.

[19] Y. Lewenberg, Y. Sompolinsky, and A. Zohar, "Inclusive block chain protocols," in Int. Conf. on Financial Crypto and Data Security, pp. 528–547, Springer, 2015.

[20] M. Nojoumian, "Rational trust modeling," in https://arxiv.org/abs/1706.09861, p. 12 pages, Arxiv, 2017.

[21] M. Nojoumian and D. R. Stinson, "Socio-rational secret sharing as a new direction in rational cryptography," in 3rd International Conference on Decision and Game Theory for Security (GameSec), vol. 7638 of LNCS, pp. 18–37, Springer, 2012.

[22] M. Nojoumian, "Generalization of socio-rational secret sharing with a new utility function," in 12th IEEE Annual Int Conf on Privacy, Security and Trust, pp. 338–341, 2014.

[23] M. Nojoumian and T. C. Lethbridge, "A new approach for the trust calculation in social networks," in E-er099 -8.966ula(p.)-350(12)-3gSaxand Ga

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