Available online at www.sciencedirect.com

ScienceDirect

Blockchain: Research and Applications 00 (2022)

Blockchain: Research and Applications

www.elsevier.com/locate/procedia

Analysis of Reputation-Based Mining Paradigm Under Dishonest Mining Attacks

Pouya Pourtahmasbi[1], Mehrdad Nojoumian[1]

Florida Atlantic University
Department of Electrical Engineering and Computer Science
777 Glades Road, Boca Raton, FL 33431

Abstract

Since the introduction of Bitcoin, numerous studies on Bitcoin mining attacks have been conducted, and as a result, many countermeasures to these attacks have been proposed. The reputation-based mining paradigm is a comprehensive countermeasure solution to this problem with the goal of regulating the mining process and preventing mining attacks. This is accomplished by incentivizing miners to avoid dishonest mining strategies using reward and punishment mechanisms. This model was validated solely based on game theoretical analyses and the real-world implications of this model are not known due to the lack of empirical data. To shed light on this issue, we designed a simulated mining platform to examine the effectiveness of the reputation-based mining paradigm through data analysis. We implemented block withholding attacks in our simulation and ran the following three scenarios: Reputation mode, non-reputation mode, and no attack mode. By comparing the results from these three scenarios, interestingly we found that the reputation-based mining paradigm decreases the number of block withholding attacks, and as a result, the actual revenue of individual miners becomes closer to their theoretical expected revenue. In addition, we observed that the confidence interval test can effectively detect block withholding attacks; however, the test also results in a small number of false positive cases. Since the effectiveness of the reputation-based model relies on attack detection, further research is needed to investigate the effect of this model on other dishonest mining strategies.

Keywords: Blockchain; cryptocurrency; mining attacks; block withholding attack; trust models; reputation systems.

1. Introduction

Bitcoin [1] is an electronic cash system that is built upon a peer-to-peer [2] network communication, Proof-of-Work (PoW) mechanism and cryptographic verification algorithms. These state-of-the-art technologies ensure a consistent and reliable cash system that is also decentralized and publicly accessible. The Bitcoin network functions based on the condition that no single entity holds 50% or more of the total computational power of the network. As long as this condition is upheld, the system is able to sustain attacks and it will remain functional and consistent. While this basic assumption is supported by mathematical proofs, it is also widely accepted that no system is fully secure and immune to attacks. In the case of Bitcoin and other similar cryptocurrencies, a number of vulnerabilities and attack scenarios have been studied over the past

Email addresses: ppourtah@fau.edu (Pouya Pourtahmasbi), mnojoumian@fau.edu (Mehrdad Nojoumian)

several years and are pervasive in the cryptocurrency literature. These articles [3, 4] cover the various vulnerability and attack scenarios on the Bitcoin network. There are also several solutions and countermeasures available in the literature. A portion of these solutions address a particular vulnerability by implementing a new security layer on top of the current Bitcoin's framework [5, 6, 7], while other proposed solutions [8, 9] do not require a major modification to the current Bitcoin consensus.

In our earlier work [10, 11], we proposed a reputation-based mining paradigm that was designed to effectively reduce the number of mining attacks by introducing a reward or punishment-based mechanism. The proposed solution was based on a game theoretical solution concept and it was aimed to incentivize the miners to avoid malicious activities and commit to honest mining strategies. However, our proposed reputation-based model, as well as the vast majority of other proposed countermeasures [12], are predominantly based on mathematical analysis and game theoretical notions. Even though the theoretical arguments are still scientific and valid, due to the lack of sufficient empirical data, the real-world implications of these theoretical solutions are not known. We therefore intend to address this issue to fill the gap in the literature.

1.1. Our Motivation and Contribution

When real-world experimentation with a theory-driven scenario is not achievable, a computer simulation can bring new insights into both the problem and the hypothetical solution. Therefore, to evaluate the effectiveness of the reputation-based paradigm on mining attacks, we designed and implemented a Bitcoin mining simulation environment with the goal of deriving empirical data. From this empirical data, we then perform data analysis on the results to deduce whether the proposed solutions are significant and confirm the real world benefits of the reputation-based paradigm solution.

The reputation-based mining model is designed to effectively reduce all kinds of mining attacks, unlike other mining attack solutions, which are aimed to target one particular attack scenario. Our goal for this simulation is to determine the effectiveness of the reputation-based model therefore any attack and its attack detection solution can be implemented to test this. However, for our simulation, we examine the effect of the reputation-based model only on block withholding attacks. As with any punishment-reward scheme, the key element that predominantly contributes to the effectiveness of the scheme, is the violation detection success rate. For various mining attacks, different detection solutions are available in the literature. The reason we chose to implement the block withholding attack scenario is due to the relative simplicity of its detection method. This detection method is based on the confidence interval test and relies on the theory of probability. Other mining attacks such as selfish mining, eclipsing, etc, could have been used in our experiment, but they are more sophisticated thus their detection methods are more complex. The complex nature of these attacks requires a more sophisticated simulation environment and should be the subject of future research projects. Our experiment with block withholding attack detection and reputation-based mining brings us a deeper insight into the performance and effectiveness of the reputation-based mining paradigm as an effective tool to counteract attacks and also its limitations, which are all discussed in this paper. In addition, by observing and analyzing the simulation data, we are able to evaluate the performance and usability of the block withholding attack detection method.

1.2. Organization of the Article

The rest of the article is organized as follows. Section 2 provides preliminary concepts on cryptocurrency mining, block withholding attack, and an overview for the reputation-based model architecture. Section 3 briefly reviews the existing literature. Section 4 explains the fundamentals of our simulation protocols and settings. Section 5 presents our technical results. Finally, section 6 concludes this article with final remarks.

2. Preliminaries

In this section, we briefly review the fundamental concepts that our simulation program is built upon. First, we review the fundamentals of the mining mechanism and the advantage of pool mining over solo mining. Next, we briefly explain the concept of a block withholding attack, its conditions and the involved entities. Lastly, we review the fundamentals of the reputation-based model, its attributes and settings.

2.1. Proof-of-Work Mechanism

The poof-of-work mechanism is one of the key aspects of Bitcoin security that ensures the functionality of the system. To verify and save the recent transactions, a new block must be generated by a network node. Generating a new block requires solving an extremely computational intensive mathematical puzzle. The network nodes who participate in this task are known as miners. Once a miner successfully provides the proof-of-work, a new block is generated and added to the blockchain and the recent unverified transactions are recorded in this newly generated block. Miner is then rewarded with Bitcoin and all other miners are informed of the new block. Now all miners start the process of finding PoW for the next block.

2.2. Pool Mining Vs. Solo Mining

Since the mining game is extremely competitive, the expected time intervals between each PoW might be very long. This issue becomes more severe as the ratio of the miner's hash power to the whole network hash power becomes smaller. Therefore the miners may need to wait a long period of time until they receive their mining rewards. Since miners must pay for their power consumption, the accumulation of the power costs for a long period of time may become too large to be affordable. To overcome this problem, a group of miners often form a coalition known as a mining pool. Once a miner from the coalition finds the PoW, he will then send the solution to the pool manager. The pool manager publishes the new block on behalf of the pool and subtracts the pool's fee from the reward. The rest of the reward is distributed among all member miners by the pool manager. Each miner receives a portion of the reward equal to his hash power ratio to the entire pool's hash power. The expected revenue from pool mining is slightly smaller than the revenue from solo mining due to the pool fee. But mining in pools will reduce the expected time interval between each reward and as a result, the income from pools is more consistent and predictable than the revenue from solo mining.

2.3. Block Withholding Attack

Block withholding [13, 14] is an attack that can be performed by a malicious miner against the ally pool. Let's assume that a miner is the member of a pool. If a miner finds a PoW and decides not to provide it for the pool, then the miner has performed a block withholding attack against the pool. Although the miner still receives his share of the reward from the pool when other miners provide the PoW, he does not contribute his true hash power to the pool. This will result in revenue loss for the pool, and consequently, all miners will lose revenue as well. As stated in [13], the miners may withhold blocks solely for the purpose of sabotage without gaining any reward. However, the author also presents a more sophisticated scenario when a miner withholds blocks in favor of another pool. In return, the miner receives a percentage of the reward as a bribe from that pool. To be justifiable, this bribe must be greater than the percentage of the reward that the miner could receive from his own pool. Block withholding attack can be practiced by one miner repeatedly or it can be practiced by more than one miner. In fact, a pool may have a secret alliance with multiple miners. This can result in even more severe revenue loss for the pool, and on the opposite side, the other pool will gain significantly more revenue than its theoretical expected revenue.

2.4. An Overview of the Reputation-Based Model

The reputation-based model includes a set of pool managers $M_{(i,p_i)}$ who form mining coalitions for $1 \leq i \leq I$, where $0 \leq p_i$ shows the profit that pool manager has accumulated so far; a set of miners $m_{(jk,r_k)}$ who perform mining, for $1 \leq j \leq J$ and $1 \leq k \leq K$ where $-1 \leq r_k \leq 1$ represents the reputation value of a miner. In the current Bitcoin framework, each miner is given a unique identity i In the reputation-based model, along with i, each miner is also given a public reputation value. The value of r_k shows how reputable the miner has been so far and r_k is updated in specific time intervals based on the miner's commitment to honest mining within that time period. Honest mining denoted by $\mathcal{H}$ will result in an increase in and likewise Dishonest mining denoted by $\mathcal{D}$ will result in a decrease in r_k.

In the reputation-based model, the pool managers evaluate their pool members after a certain period of time. They send invitations to miners based on their reputation value. The reputable miners are more likely

to receive invitations compared to the miners who are not trustworthy. The miners who have received multiple invitations will have the option to join the pool they prefer whereas, the miners who have not received any invitations cannot participate in pool mining. The reputation-based model relies on the detection success rate. This means, the goal of the reputation-based model is only accomplished if effective attack detection solutions are incorporated into the mining scheme. Also, the reputation-based model must be immune to re-entry attacks. This means that the dishonest miner cannot exit out of the system and come back with a new reputation value. To accomplish this goal, the proposed model utilizes the approach of rational trust modeling [15]. In this model, a permanent reputation parameter is linked to the identity of the miner and it will be preserved over time. This ensures the system is immune to re-entry attack.

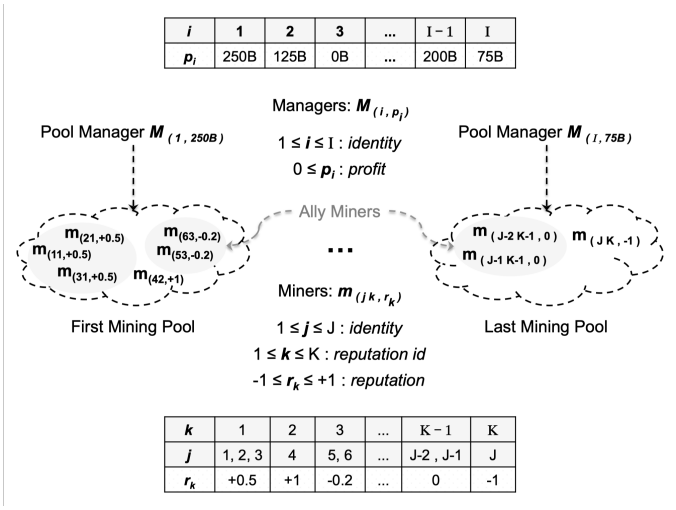

Fig. 1: Architecture of the reputation-based mining [10].

3. Literature Review

The deployment of a reputation-based paradigm is known to be a robust approach in controlling malicious activities on various platforms and environments. Particularly in decentralized platforms, a trust measurement scheme [16, 17] can compensate for the lack of the moderator entity [18]. In this section, we briefly review similar reputation-based solutions that are designed to be implemented on top of the blockchain-based platforms such as [19, 20].

Freeman et al. [9] proposed a scoring mechanism that relies on a machine learning algorithm. This mechanism can help users to identify the risky and potentially fraudulent transactions. The user can interpret the reputability of other users in the network thus, deciding which users are trustworthy to do trades with. The scoring system is implemented in three steps. First, the users who have a history of theft and other malicious activities are blacklisted. In the second step, the honest users are differentiated from the adversary users. In the third step, based on the classification, a risk score is calculated. This score would represent the user's trust factor based on the involvement in fraudulent transactions. Carboni [21] argued that a distributed and decentralized feedback-based reputation system can be implemented on top of the cryptocurrency blockchain. Then the author proposed a feedback mechanism in which a cryptographic link is established between the translations and the services. This link then is recorded in the blockchain. The author acknowledges that the proposed model is not formally proven to be resistant to various attacks but it is still robust and reasonable when it is compared against the current feedback paradigm that exists on many popular online platforms such as eBay.

Zhuang et al. [22] proposed a reputation-based consensus mechanism, named Proof-of-Reputation (PoR). In this framework, all nodes have a reputation. This reputation is developed based on the node's transaction activities, assets and participation. In PoR, the consensus is processed in a round that includes the selection of the leader node who has the highest reputation. Then, the leader node generates and publishes the block. The verification process is done by other higher reputation nodes through voting for the block. The top twenty percent of the nodes in the sorted reputation list are considered high reputation nodes. If the sum of the reputation value of the nodes that voted consent is greater than the sum of the reputation values of the two thirds of the highest reputation nodes, the block is verified and it will be added to the blockchain. Otherwise, the block is invalid and it will be disregarded. When a new block is added to the blockchain, the reputation value of the nodes are updated.

Yu et al. [23] proposed a reputation scheme, named RepuCoin. The authors claim that this proposed mechanism can withstand an attack even if the attacker has temporarily obtained more than 50% of the network's hash power. Also, the proposed framework is claimed to have throughput of 10000 transactions per second. This is done through the proof-of-reputation process, but the rate of voting power growth of the whole system is limited. In this framework, rather than hash power, a miner's power is their reputation, which is the work the miner has done over the entire life of the blockchain.

Do et al. [24] proposed a reputation ranking mechanism, named Delegated Proof-of-Reputation (DPoR), that can be implemented in many systems such as blockchain consensus, credit, social reputation and rating. This consensus mechanism is designed to be scalable and secure with a reasonable decentralization.

Abdo et al. [25] proposed a permission-less hybrid reputation/proof-of-reputation-X consensus algorithm. In order to bring compatibility with permissionless blockchain, this mechanism replaces the trusted identity database in proof-of-reputation-X with a new admission process. The authors compared the performance of the new scheme with proof-of-reputation-X and they demonstrated that the new scheme decreases the number of blocks generated by malicious miners.

Xue et al. [26] study the incentive mechanisms for the miners who mine for mining pools. The authors highlight that in the existing mining pool models, the cost and the strategy of the individual miners are not considered. Therefore, they present two incentive mechanisms for miners considering the cost differences among the miners. The authors consider two mining models: a public cost model and a private cost model for pool mining. For the public cost model, a Stackelberg game is used and they show that there exists a unique Stackelberg Equilibrium for the mining game and the game is rational and profitable for individual miners. For the private cost model, Budget Feasible Reward Optimization is formulated. In this model, the goal is to maximize the reward function with the consideration of the budget limit. The authors show that their model is efficient, individually rational and budget feasible.

Singh et al. [27] consider a dynamic game model for Bitcoin mining. In this model, the authors discuss two approaches that miners can have to maximize their profit. The first approach is a cooperative strategy (Social Optimum) in which all miners consider a fixed amount of electricity for the mining game. In return, they receive Bitcoin at market price as a reward so they can jointly maximize their reward and the reward is shared among them equally. The second approach (Nash equilibrium and myopic Nash equilibrium) is non-cooperative in which the individual miners behave selfishly to maximize their profit in the Bitcoin system. The authors demonstrate that in the cooperative way, the miners gain a higher total profit over time compared to the non-cooperative way. Also, a greedy Nash equilibrium strategy will result in the depletion of the electricity resources, while in the social optimum strategy, the electricity resources remain sustainable.

4. Our Protocols and Implementations

We design our mining simulation to be as realistic as possible. We consider many parameters, conditions and random events. There are several global parameters for our simulated mining game including: the total hash power of the system, the price of the cryptocurrency, mining profitability, and time intervals between each PoW. Also the entities, including the miners and the pools, are programmed in a way that each entity has its own unique set of parameters and characteristics. This approach mimics the real environment where a number of various parameters would affect the outcomes for both the entire system and the individual entities. In this section, we define the fundamental protocols and settings for our simulation program.

4.1. Miners and Pools

The miners in our simulation are a set of entities who participate in the mining game to gain profit. Each miner is given a constant hash power and a power consumption cost rate. These values are generated randomly based on a defined normal distribution property. Since the mining game is essentially an economic activity, each miner is given a negative loss tolerance threshold (LTT). This value defines the amount of loss that the miner can tolerate. A miner whose profit becomes less than or equal to LTT, will leave the mining game. The miner's population is a variable throughout the life of the simulation. The simulation starts with a fixed population n_0 and population growth is determined by a set of sigmoid functions. Note that the mentioned functions only contribute to the growth of the population. The decline in the population is the result of miners falling below their LTT thus leaving the system. Miners can mine solo or they can join a pool and share the rewards. The miners can determine the expected number of rounds it would take to provide a PoW. Let i be the current round and $i + k$ is the round that the miner is expected to provide a PoW. The expected profit at round $i + k$ is calculated as follow:

$$E\left[P_{i+k}\right] = P_i - \frac{\Delta C}{p(x)}$$

In the above equation, P_i denotes the profit until mining round i , ΔC is the average power cost per round and $p(x)$ is the miner's hash power ratio to the whole network's hash power. If $P_{i+k} \leq LTT$, then there is a high probability that the miner's profit will fall below LTT before he can earn the reward. Therefore, the miner will join a pool to reduce the expected number of mining rounds for each reward thus reducing the risk of falling below LTT. Unlike miners, the number of pools in our simulation is constant and it is set to 8. Each pool is given a constant fee rate. Pools subtract their fee from the reward before distributing it among miners. In our simulation, the miners select pools based on a non-uniform probability distribution that is defined by the pools' fee rate. This means that the pools with a lower fee rate are more likely to be joined by the miners than the pools with a higher fee rate.

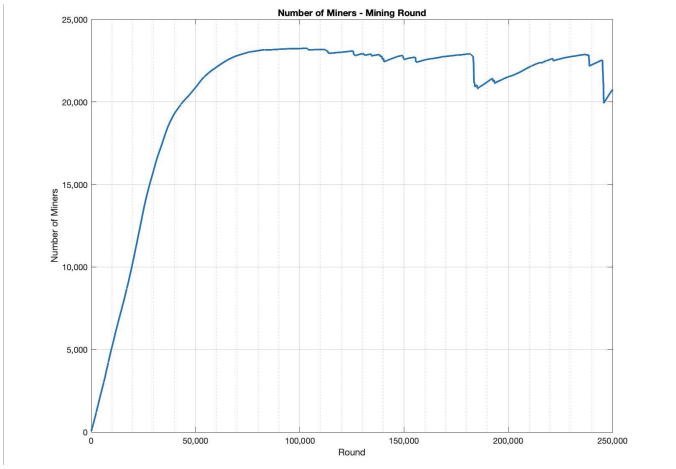

Fig. 2: An example of miner population change in our simulation environment.

4.2. Mining Game

The outcome of the mining game in our simulation is determined by a probabilistic algorithm. The probability distribution of the random variable that is generated by this algorithm, is similar to the probability distribution of a real mining environment. Let $M = \{m_1, m_2, m_3, ..., m_n\}$ be the set of all miners and $H = \{h_1, h_2, h_3, ..., h_n\}$ be the set of the corresponding hash powers. The hash power for the total network is:

$T_H = \sum_{j=1}^{n} h_j \quad \text{where} \quad n = |M|$

The value of h_j/T_H defines the probability that the miner m_j provides a PoW at round i. To simulate the mining game, we generate a random number that is uniformly distributed in the range $[1, T_H]$ and by the pseudocode in Algorithm 1, a random miner can be selected. Note that since the value of T_H may change for each round, the probability of providing a PoW for each miner also changes.

Algorithm 1 Mining game pseudocode.

```
1: procedure MiningGame(M, H)
2:     let l be a random number between 1 and T_H
3:     Let M' be the list containing a random permutation of all elements in M
4:     let H' be the list containing all elements in H corresponding to M'
5:     sum = 0
6:     for j = 1 to n do
7:         sum = sum + h_j
8:         if sum ≥ l then return m_j
```

After each round of the game, the mining power costs are updated for all miners. The time between each mining round is calculated randomly using a pre-defined normal distribution property. Next, the cost for round i is calculated and added for all miners. This cost value is calculated based on each miner's power cost rate and the time duration of round i.

4.3. Attack Setup

In our simulation, the block withholding attack always originates from a dishonest pool. As mentioned earlier, there are 8 pools in our system; 3 pools are dishonest and 5 pools are honest. A dishonest pool only attacks an honest pool. This setting will result in a more appropriate data set for the purpose of highlighting the effects of a block withholding attack on the revenue of honest and dishonest entities separately. The block withholding attack is conducted in two phases as follow:

1. Initialization: Let D be the set of dishonest (suspect) pools and H be the set of honest (victim) pools. After each round, a random dishonest pool d from D and a random honest pool h from H is selected. Then pool d will try to find a malicious miner from pool h. d searches for a malicious miner who also has a high hash power. If such a malicious miner is found, then the entities are set and the initialization is successful. Otherwise the initialization will terminate.

2. Process: Let m be the malicious miner from pool h who is committed to withholding one or more PoW from pool h and instead, provides PoWs for pool d for k times. Once miner m finds a PoW, it is delivered to pool d and in return, pool d will pay miner m a percentage of the PoW reward as bribe. This activity is repeated for k times under the condition that m finds a total k PoW.
   In this model, pool d would distribute the reward among the member miners the same way that it would have distributed it if the reward was earned honestly. The values for k and the bribe percentage are random based on a set of defined normal distribution properties.

4.4. Attack Detection

The attack detection method for a block withholding attack is relatively simple and relies on basic statistical analysis. As mentioned in part 4.2, the probability that miner m provides a PoW in round i is simply h_m/T_{H_i}, where T_{H_i} is the total hash power at round i. Consequently, the expected number of PoWs after playing number of rounds can be calculated as follow:

$E[x] = \sum_{i=1}^{n} \frac{h_m}{T_{H_i}}$

The difference between $E[x]$ x and may be significant for small values of n, but for sufficiently large n, it is expected that $E[x] \approx x$. Therefore, after a sufficient number of rounds, the pool manager can perform a statistical test on the individual miners to determine whether the difference between their actual and expected PoW is significant or not. If the difference is indeed significant for one or more miners, particularly for miners with higher hash power, the pool manager can conclude that the pool is under a block withholding attack. To examine whether the difference between their actual and expected PoW is statistically significant, we use the confidence interval (CI) test. First we calculate the ratio between $E[x]$ and x as follow:

$\hat{x} = \frac{E[x]}{x}$

Then, the confidence interval is calculated as follow:

$$CI = \left(\hat{x} - z_c \sqrt{\frac{\hat{x}(1-\hat{x})}{x}} \ , \ \hat{x} + z_c \sqrt{\frac{\hat{x}(1-\hat{x})}{x}} \right)$$

For this experiment, different confidence levels such as 0.95 , 0.98 or 0.99 can be used. A lower confidence interval will result in a higher attack detection rate, but the drawback is that the number of false positive cases are expected to increase. Therefore, for this simulation we select 0.98 confidence level which provides the optimal results.

4.5. Miner's Attack Utility

In our earlier work [10], we showed that when an attack detection mechanism as well as an appropriate punishment measurements exist in a system, attacking is no longer Nash equilibrium. Therefore it is expected that the players act rationally and they avoid dishonest behaviors. When a miner m is given an attack opportunity, he can determine whether it is in his best interest to commit to the attack or not. The key to this determination is the rate of attack detection. Miners are incentivized to be honest as more dishonest miners in the system are detected and punished. In this case, the punishment is a lower level of reputation and possible chance of losing mining in mining pools. The miner's attack utility function that is used in our simulation is based on this concept. Therefore, the rate of detection defines the extent that miners act honestly. The pseudocode for miner's attack utility is given in Algorithm 2 where, r is the attack detection ratio and, $0 \leq r \leq 1$ and d is a dishonest parameter that is unique to the miner where $0 < d < 1$. The higher the value of d , the more probable that the miner will accept the attack offer. The value for $targetProfit$ represents the amount of bribe that is large enough so the miner is willing to accept the attack offer and neglect his reputation.

Algorithm 2 Attack utility.

```
1: procedure AcceptDishonestRevenue(dishonestReward, r)
2:     if dishonestReward + profit ≥ targetProfit then return FALSE
3:     α = a random number between 0 and 1
4:     t = (1 − r)^(1/d)
5:     if α < t then return TRUE
6:     return FALSE
```

4.6. Simulation Modes

In order to evaluate the effectiveness of the reputation-based paradigm, we need to make a comparison between the reputation-based and the non-reputation-based mining environments. We also require a third scenario where no attack takes place. This simulation mode resembles the ideal environment where no dishonest activity takes place. The results from this mode will provide a reference point for the evaluation of the

reputation-based model. The data similarity between Reputation mode and No Attack mode, demonstrates a high level of success and effectiveness for the reputation-based paradigm.

We run the simulation program for 250,000 rounds of mining in each mode and the data for each mode is recorded individually. The probability distribution attributes for all random parameters are the same for all three modes. For example, the value for the hash power is generated randomly for all miners. These random hash power values are distributed normally and the defined mean and the standard deviation parameters are equivalent in our three modes. This will ensure that the distribution of hash power in all three modes are statistically similar. This setting is true for all other random parameters in our simulation. Therefore the differences between the results in each mode is guaranteed to be influenced by the reputation-based paradigm in the system and not the consequence of randomness. In addition, the settings for each pool including the pool's fee rate and the state of honesty or dishonesty are equivalent in all three modes.

Non-Reputation Mode: This mode resembles the current setting of the Bitcoin network where there is no mining attack detection scheme and the reputation-based paradigm is in place. In this setting, attacking is Nash Equilibrium since there is no potential consequence for block withholding attack. Miners are always engaged in attack if the opportunity is available. In this mode, the miners can join pools freely and the pool managers always allow any miner in the network to join their pool. The actual distribution of rewards among miners is expected to be significantly different from the theoretical probability distribution because the dishonest miners gain more rewards while the honest miners lose.

Reputation Mode: In this mode, all miners are given a reputation value that is initially zero and is periodically updated by their pool managers based on their level of their commitment to honest mining. In reputation mode, attacking is not Nash Equilibrium. If a pool manager M detects an attack, conducted by a miner m, the pool manager will apply a defect function on the reputation value of the miner, and subsequently, he will expel the miner from the pool. The short-term utility function is the same as it is in the Non-Reputation mode. The long-term utility function, however, considers the long-term consequences of committing an attack and it is used whenever an opportunity for an attack exists. The miner will determine whether the attack is profitable in the long term. In this mode, the miners can only join a pool if they received an invitation from that pool. The actual distribution of rewards among miners is expected to be different from the theoretical probability distribution, but not as drastically different as in the Non-Reputation mode.

No Attack Mode: This mode represents the ideal situation where all miners are committed to honest mining at all times. This setting is similar to the other two modes with the exception that block withholding opportunities are not available to the miners thus no attacks will take place. In this mode, the pool joining mechanism is identical to Non-Reputation mode where miners join pools based on a probability distribution that is defined by the pools' fee percentage. The actual distribution of rewards among miners is expected to be the closest to the theoretical probability distribution.

5. Our Technical Results

In this section, we provide the results of our simulation program in all three modes explained in section five. First we provide a summary of the results in tables and figures and then we present the statistical analysis for the results.

5.1. Summary of the Results

The summary for 250,000 rounds of mining in each scenario is shown in Table 2. Note that the values for distributed rewards and costs are shown in dollars. The reward values are calculated based on the price of the cryptocurrency at the end of round 250,000 and the changes in price over time for all 3 scenarios are shown in Figure 4. The value for miners' costs is the accumulation of all miners' costs until the last mining round. The active miners are the miners whose profit value is still above LTT and the inactive miners are the miners whose profit value fell below LTT at some point during the game and as a result they have left the system. All the statistical results that are shown in tables and graphs, include the data for both active and inactive miner groups.

The number of block withholding cases indicate the number of times an individual attack case is processed. For each attack case, the miner who attacks may withhold blocks more than once. The number of

Table 1: Differences between the simulation modes.

	Non-Reputation	**Reputation**	**No Attack**
Pool Joining Mechanism	Miners can join pools freely	Miners require invitation from the pools they wish to join	Miners can join pools freely
Implemented Attack	-	Block Withholding	-
The consequence of Attack	-	- The reputation value will be effected negatively - The attacker miner will be expelled from the pool	-
Utility Functions	Pool Joining	Pool Joining, Attack	Pool Joining

Table 2: Summary of the simulation in each mode at the end of round 250,000.

Description	**Non-Reputation**	**Reputation**	**No Attack**
Time in Seconds	154,092,032	154,090,378	154,332,223
Total Hash Power	61,270,349	67,394,318	67,822,864
Number of Active Miners	20,744	22,801	22,938
Number of Inactive Miners	9,298	7,257	8,042
Percentage of Pool Miners	93.7%	93.1%	94.4%
Price of the Cryptocurrency	$1,168.28	$1,375.37	$1,237.23
Total Distributed Rewards among All Miners	$5,840,080,000	$6,870,660,000	$6,186,100,000
Total Power Costs for All Miners	$6,312,700,000	$6,290,500,000	$6,374,900,000
Percentage of Miners with Positive Profit	31.3%	49.9%	36.4%
Total Block Withholding Cases	1,128	557	0
Detected Block Withholding Cases	0	382	0
False Detected Block Withholding Cases	0	23	0

Table 3: Summary of pools at the end of round 250,000 in non-reputation mode.

Pool	BYT-728	KRM-664	MME-935	RLC-061	SJN-888	UFR-774	VPK-703	WSQ-559
Hash %	19.7%	4%	3.56%	4.74%	5.45%	16.6%	17.3%	17.1%
E-PoW	41,509	10,153	8,712	8,729	10,608	37,499	39,670	38,802
A-PoW	45,838	8,275	7,003	12,877	14,805	34,044	36,749	35,845

Table 4: Summary of pools at the end of round 250,000 in reputation mode.

Pool	BYT-728	KRM-664	MME-935	RLC-061	SJN-888	UFR-774	VPK-703	WSQ-559
Hash %	21.98%	2.76%	2.8%	3.3%	3.99%	17.3%	17.1%	16.97%
E-PoW	47,251	7,063	7,027	7,298	8,680	40,070	37,988	38,635
A-PoW	48,336	6,750	6,718	8,672	10,044	38,755	37,124	37,749

Table 5: Summary of pools at the end of round 250,000 in no attack mode.

Pool	BYT-728	KRM-664	MME-935	RLC-061	SJN-888	UFR-774	VPK-703	WSQ-559
Hash %	17.99%	4.26%	4.8%	3.22%	4.84%	18.12%	18.38%	18.28%
E-PoW	39,232	9,064	9,553	7,340	10,682	39,318	40,173	39,909
A-PoW	39,266	9,004	9,545	7,330	10,756	39,257	40,019	39,860

Table 6: Summary of the calculated statistics for miners.

Description		Non-Reputation	Reputation	No Attack
Square Error for Actual PoW and expected PoW for All Miners		26.3	10.1	8.1
Revenue for All Miners from all Pools (Including Dishonest Revenue)	Mean	139.2	143.8	132.1
	Median	11.7	11.1	10.7
	Standard Deviation	222.9	222.8	209.7
Revenue for All Honest Miners from the Victim Pools	Mean	107.5	123	131.8
	Median	9.2	9.7	10.5
	Standard Deviation	162.2	184.9	209.2
Dishonest Revenue for Block Withholder Miners from the Victim Pools	Mean	141.8	46.4	-
	Median	105.6	39.8	-
	Standard Deviation	109	35.7	-
Revenue for All Miners from the Suspect Pools	Mean	168.4	149.7	132.8
	Median	13.7	12.2	10.8
	Standard Deviation	256.7	232.7	10.8

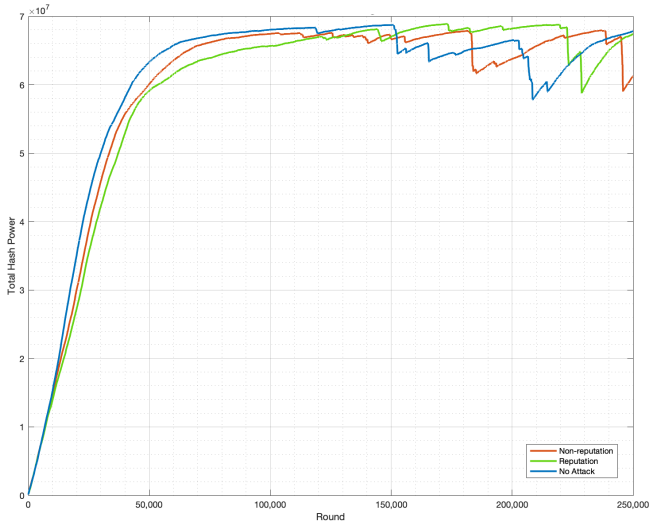

Fig. 3: Change of total hash power.

block withholding instances for each attack is set when the attack is initialized. The results for each pool are shown in Table 3-5. Note that the percentage of hash share for each pool, shown in Table 3-5 and Figure 5, varies throughout the life of the simulation. The given results represent the distribution at the end of round 250,000. The hash power distribution in Non-Reputation and No Attack modes are very close, but it is slightly different in Reputation mode. This is due to the fact that the pool joining mechanism in Reputation mode is invitation based and explains the slightly higher percentage of solo miners in the Reputation mode. The pool joining mechanism in No Attack and Non-Reputation modes are the same therefore, the insignificant differences between the two modes must be the result of randomness in the simulation.

The differences between the actual number of PoWs and the expected number of PoWs shown in Figure 8, indicates a significant difference between each mode. The high values for the Non-Reputation mode are

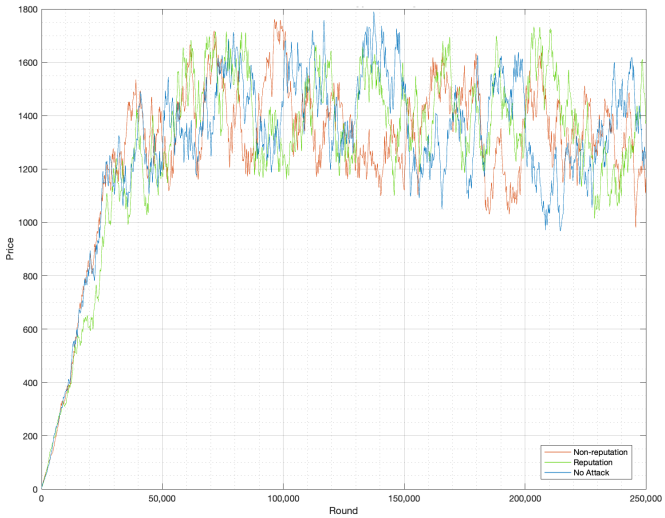

Fig. 4: Change of cryptocurrency price.

the consequences of block withholding attacks where the actual number of PoWs are much higher than the expected number of PoWs for all suspect pools. The opposite is true for all victim pools. We observe that in the Reputation mode, this similar pattern still exists but at a much lower magnitude. In contrast, the small differences between the actual PoW and the expected PoW in the No Attack mode, is the natural result of the probability outcome and it does not follow any pattern.

Figure 7 shows the number of block withholding attacks in groups of 50,000 rounds. In both modes the number of attacks are in decline for the first 100,000 rounds. At first glance this pattern may seem unusual, but a deeper analysis demonstrates that as the number of miners increase in the system, the probability that a single block withholder miner provides a PoW decreases. This pattern is followed by a decrease in the success rate of block withholding attacks. Figure 3 shows that the total hash power of the system reaches its peak at approximately round 100,000. While the attack initialization rate remains relatively constant, this pattern perfectly correlates with the decline in the number of attacks until round 100,000.

Figure 8 demonstrates the ratio between the detected attacks and all attacks in the reputation mode. The bar heights in Figure 9 correlate with the attack detection success ratio shown in Figure 8. It is evident that as the detection ratio increases, the number of attacks also decreases. From a different perspective, Figure 11 demonstrates that as the number of attacks decrease, the growth rate of the total amount of bribes received by the block withholding miners drops. The basic statistical analysis for the revenue of all miners are presented in Table 6. The first row of the table demonstrates the square error between the actual PoW and the expected PoW for all miners in each mode. The rest of the table shows the values for mean, median and standard deviation for all miners, honest miners from the victim pools, dishonest miners from the victim pools and the miners from the suspect pools respectively.

The scatter plots presented in Figure 10 through 17, show the distribution of the revenue for individual miners throughout the life of the simulation. The scatter plots are presented for miners from each pool individually. These plots highlight the changes in revenue for the whole group of miners in all 3 modes. The straight line in these plots represents the linear trend in No Attack mode. The same line on the other two modes highlights the revenue shift resulting from block withholding attacks conducted by dishonest miners.

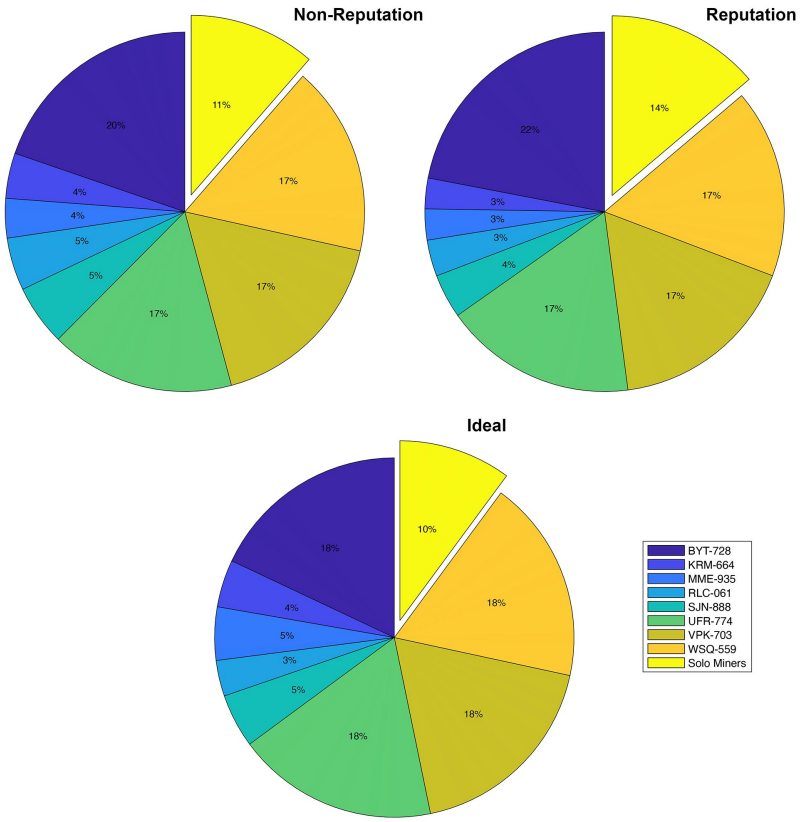

Fig. 5: Distribution of hash power among pools in each simulation mode.

5.2. Analysis of the Results

A) The effect of the Reputation-based model on the distribution of PoW: As demonstrated in Figure 5 through 7, the number of block withholding attacks are reduced as the detection ratio increases, and consequently, the dishonest revenue for the entire network reduces. Besides, Figure 8 shows the difference between the actual and the expected number of PoWs for all pools and how the revenue for all pools becomes more stable and predictable. As the attack detection success ratio improves, the miners are more incentivized to commit to honest mining strategies. As shown in Figure 6, the attack detection success ratio is stabilized around round 100,000. This is due to the fact that as the number of mining rounds increases, the predictability of potential attacks improves. This stability also has a relationship with the selected confidence intervals. A large number of trials are required for an effective and precise attack detection.

B) The effect of the Reputation-based model on the distribution of Revenue: The statistical measurements presented in Table 6, show that the reputation-based model significantly improved the revenue of the honest miners significantly. It is also evident that the dishonest revenue is significantly decreased by the reputation-based model. However, when we observe the measurements for all miners, it shows slight fluctuations in the median and mean values and no significant change in the standard deviation. This indicates that the reputation-based model does not significantly change the statistical attributes for the revenue of the whole system. When we compare the No Attack mode with the other two modes, it is evident that the distribution of revenue is slightly more balanced in the No Attack mode as its standard deviation value is smaller. Note that the large standard deviation values for the revenue are predominantly the result of an unbalanced distribution of hash power among the miners. In our simulation, a small percentage of miners

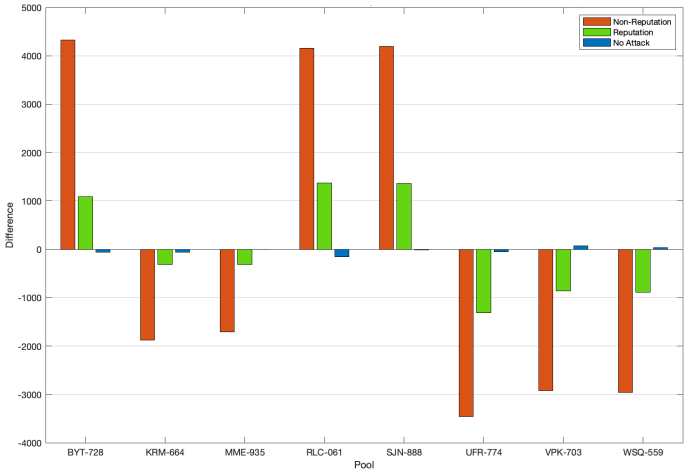

Fig. 6: Differences between actual and expected number of PoW for all 8 pools after 250,000 rounds of mining.

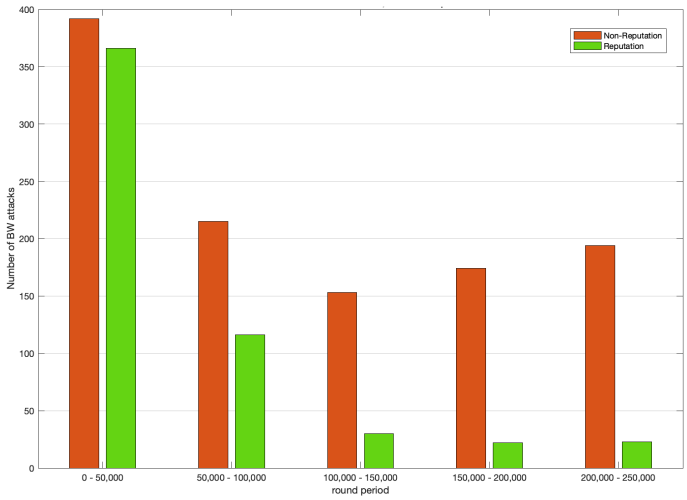

Fig. 7: Number of BW attacks in the periods of 50,000 rounds.

own a relatively large percentage of the total hash power. This will directly cause an unbalanced distribution of revenue among miners regardless of block withholding attacks. In figure 10 to 17, it is observable that the reputation-based model significantly improves the predictability and reliability of the revenue distribution for both suspect and victim pools as the actual distribution of revenue among miners resembles a similarity between Reputation and No-Attack modes. On the other hand, the revenue distribution in the Non-Reputation mode is predominantly above the straight line for the suspect pool and below the straight line for the victim pools.

Fig. 8: Attack detection ratio in the reputation mode.

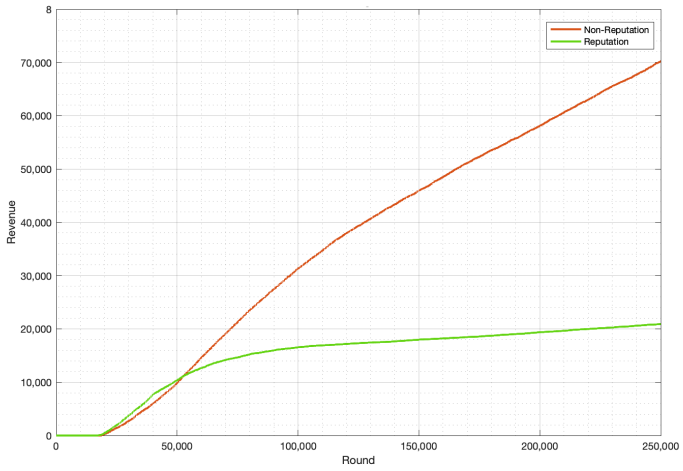

Fig. 9: Accumulation of bribe paid to dishonest miners by the suspect pools.

C) The False Attack Detection Observation: As it is shown in Table 2, there are 23 false positive attack detection cases. In these cases, the miners are falsely detected for block withholding attacks where they had committed no attack. The difference between their expected and actual PoW was only a result of chance. We can reduce this effect by increasing the test confidence parameter and by increasing the number of trials per interval. However, this will negatively affect the detection success rate and the number of undetected attacks will increase. Practically, it is not feasible to rely on this statistic-based test with 100% certainty of not having false positives. If such a goal is desired, other attack detection methods are needed in conjunction with the probability-based methods.

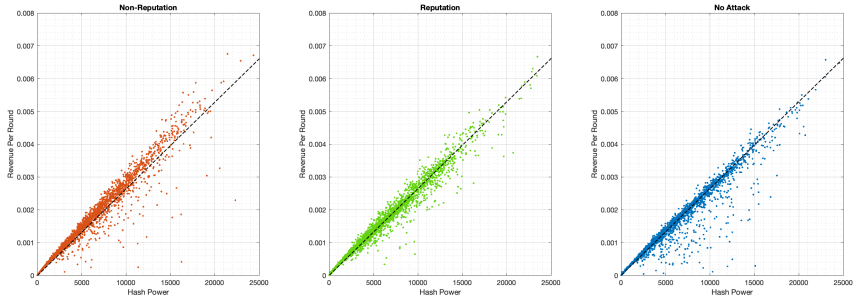

Fig. 10: Average revenue per round for all miners from suspect pool BYT-728.

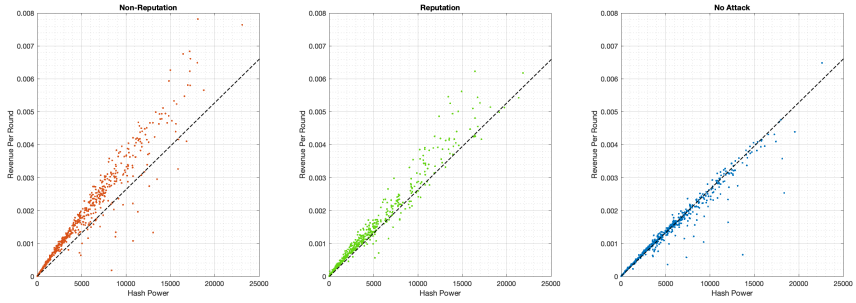

Fig. 11: Average revenue per round for all miners from suspect pool RLC-061.

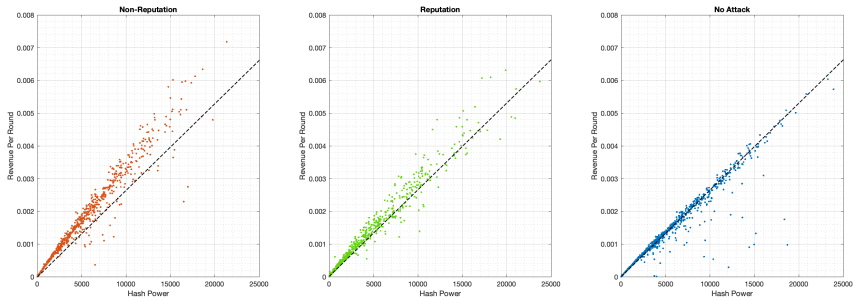

Fig. 12: Average revenue per round for all miners from suspect pool SJN-888.

6. Concluding Remarks

In this paper, we presented our mining simulation in order to evaluate the effectiveness of the proposed reputation-based model. We designed our simulation in a way that resembles the real mining environment. We did this by considering various variable parameters such as total hash power and price of the cryptocurrency and we programmed miners and pools in a way that each individual had a unique set of characteristics. We implemented block withholding attacks in our simulation to determine if the reputation-based model could reduce the number of block withholding attacks. We performed our simulation program in 3 different modes namely, Non-Reputation, Reputation and No-Attack mode, and each mode consisted of 250,000 rounds of mining. The results show that the Reputation-based model can significantly reduce the number of block withholding attacks and consequently, the distribution of revenue among miners becomes

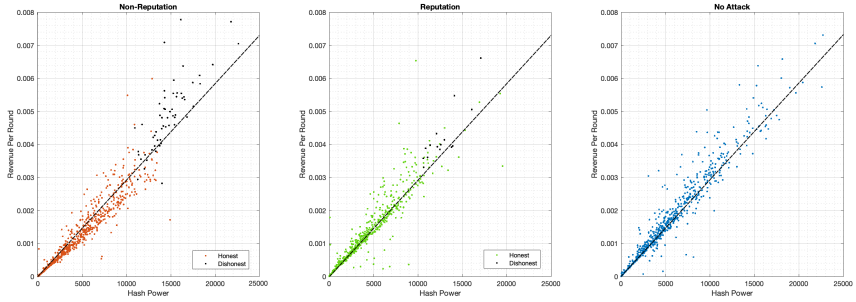

Fig. 13: Average revenue per round for all miners from victim pool KRM-664.

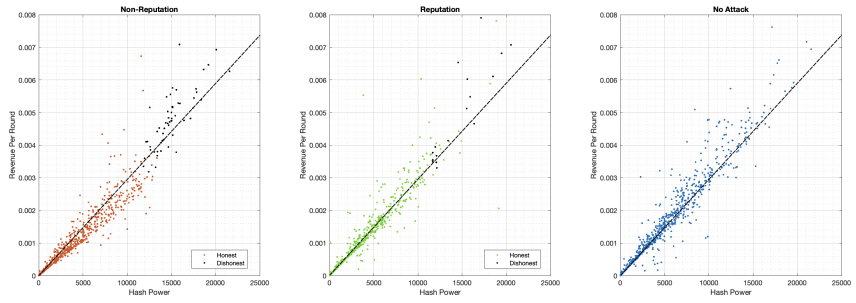

Fig. 14: Average revenue per round for all miners from victim pool MME-935.

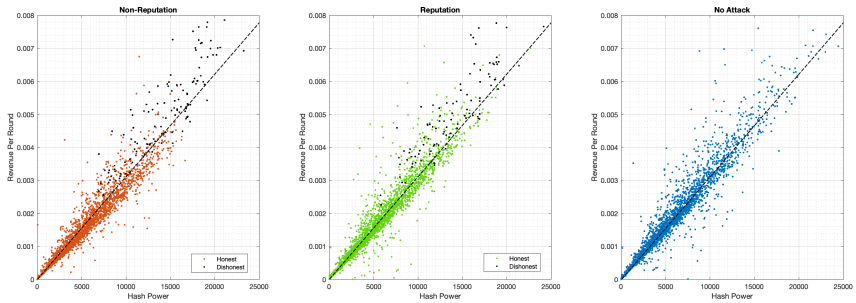

Fig. 15: Average revenue per round for all miners from victim pool UFR-774.

more predictable and reliable. We showed that the statistical method used for the detection of a block withholding attack can result in a small percentage of false positive cases and other detection methods need to be incorporated in order to resolve this issue. Further research is needed in order to examine the effectiveness of the reputation-based model on other mining attacks such as selfish mining and eclipsing.

Acknowledgement

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

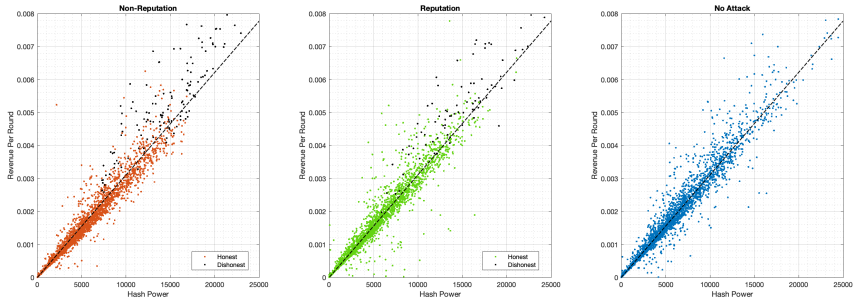

Fig. 16: Average revenue per round for all miners from victim pool VPK-703.

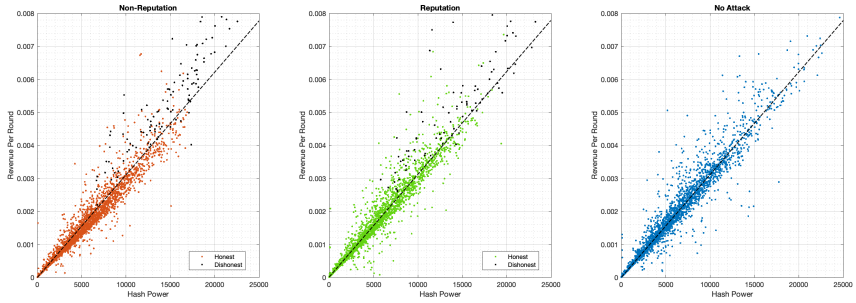

Fig. 17: Average revenue per round for all miners from victim pool WSQ-559.

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] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Decentralized Business Review (2008) 21260.
[2] R. Schollmeier, A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications, in: Proceedings First International Conference on Peer-to-Peer Computing, IEEE, 2001, pp. 101–102.
[3] M. Conti, E. S. Kumar, C. Lal, S. Ruj, A survey on security and privacy issues of bitcoin, IEEE Communications Surveys & Tutorials 20 (4) (2018) 3416–3452.
[4] F. Tschorsch, B. Scheuermann, Bitcoin and beyond: A technical survey on decentralized digital currencies, IEEE Communications Surveys & Tutorials 18 (3) (2016) 2084–2123.
[5] R. Zhang, B. Preneel, Publish or perish: A backward-compatible defense against selfish mining in bitcoin, in: Cryptographers' Track at the RSA Conference, Springer, 2017, pp. 277–292.
[6] I. Eyal, E. G. Sirer, Majority is not enough: Bitcoin mining is vulnerable, in: International conference on financial cryptography and data security, Springer, 2014, pp. 436–454.
[7] M. Bastiaan, Preventing the 51%-attack: a stochastic analysis of two phase proof of work in bitcoin (2015).
[8] E. Heilman, One weird trick to stop selfish miners: Fresh bitcoins, a solution for the honest miner, in: International Conference on Financial Cryptography and Data Security, Springer, 2014, pp. 161–162.
[9] S. Lee, S. Kim, Countering block withholding attack efficiently, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2019, pp. 330–335.
[10] M. Nojoumian, A. Golchubian, L. Njilla, K. Kwiat, C. Kamhoua, Incentivizing blockchain miners to avoid dishonest mining strategies by a reputation-based paradigm, in: Computing Conference (CC), AISC 857, Springer, 2018, pp. 1118–1134.
[11] P. Pourtahmasbi, M. Nojoumian, Impacts of trust measurements on the reputation-based mining paradigm, in: 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), IEEE, 2021, pp. 225–228.
[12] M. Nojoumian, A. Golchubian, N. Saputro, K. Akkaya, Preventing collusion between SDN defenders and attackers using a game theoretical approach, in: Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2017, pp. 802–807.

[13] M. Rosenfeld, Analysis of bitcoin pooled mining reward systems, arXiv preprint arXiv:1112.4980 (2011).
[14] S. Bag, S. Ruj, K. Sakurai, Bitcoin block withholding attack: Analysis and mitigation, IEEE Transactions on Information Forensics and Security 12 (8) (2016) 1967–1978.
[15] M. Nojoumian, Rational trust modeling, in: 9th Conference on Decision and Game Theory for Security (GameSec), LNCS 11199, Springer, 2018, pp. 418–431.
[16] M. Nojoumian, T. C. Lethbridge, A new approach for the trust calculation in social networks, in: E-Business and Telecommunication Networks: 3rd International Conference on E-Business, CCIS, Springer, 2008, pp. 64–77.
[17] M. Nojoumian, Trust, influence and reputation management based on human reasoning, in: 4th AAAI Workshop on Incentives and Trust in E-Communities (WIT-EC), 2015, pp. 21–24.
[18] P. Resnick, R. Zeckhauser, Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system, in: The Economics of the Internet and E-commerce, Emerald Group Publishing Limited, 2002, pp. 127–157.
[19] L. Zamir, A. Shaan, M. Nojoumian, ISRaft consensus algorithm for autonomous units, in: 29th International Conference on Network Protocols (ICNP), IEEE, 2021, pp. 1–6.
[20] L. Zamir, M. Nojoumian, Information sharing in the presence of adversarial nodes using Raft, in: Future Technologies Conference (FTC), LNNS 360, Springer, 2021, pp. 159–172.
[21] D. Carboni, Feedback based reputation on top of the bitcoin blockchain, arXiv preprint arXiv:1502.01504 (2015).
[22] Q. Zhuang, Y. Liu, L. Chen, Z. Ai, Proof of reputation: a reputation-based consensus protocol for blockchain based systems, in: Proceedings of the 2019 International Electronics Communication Conference, 2019, pp. 131–138.
[23] J. Yu, D. Kozhaya, J. Decouchant, P. Esteves-Verissimo, RepuCoin: Your reputation is your power, IEEE Transactions on Computers 68 (8) (2019) 1225–1237.
[24] T. Do, T. Nguyen, H. Pham, Delegated proof of reputation: A novel blockchain consensus, in: Proceedings of the 2019 International Electronics Communication Conference, 2019, pp. 90–98.
[25] J. Bou Abdo, R. El Sibai, J. Demerjian, Permissionless proof-of-reputation-x: A hybrid reputation-based consensus algorithm for permissionless blockchains, Transactions on Emerging Telecommunications Technologies 32 (1) (2021) e4148.
[26] G. Xue, J. Xu, H. Wu, W. Lu, L. Xu, Incentive mechanism for rational miners in bitcoin mining pool, Information Systems Frontiers 23 (2021) 317–327.
[27] R. Singh, A. D. Dwivedi, G. Srivastava, A. Wiszniewska-Matyszkiel, X. Cheng, A game theoretic analysis of resource mining in blockchain, Cluster Computing 23 (3) (2020) 2035–2046.