Incentive Mechanism for Bitcoin Mining Pool Based on Stackelberg Game

Gang Xue, Jia Xu(✉), Hanwen Wu, Weifeng Lu, and Lijie Xu

Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, China
xujia@njupt.edu.cn

Abstract. Bitcoin is the most popular cryptocurrency all over the world. Existing mining pool systems do not consider the cost of miners. In this paper, we propose a novel pool mining mechanism based on Stackelberg game to incentivize the rational miners in Bitcoin mining pool. Through both theoretical analysis and simulations, we demonstrate that the proposed mechanism achieve computational efficiency, individual rationality, and profitability. Moreover, we show that the Stackelberg game has a unique Equilibrium.

Keywords: Bitcoin · Mining pool · Incentive mechanism · Nash equilibrium

1 Introduction

Bitcoin is the world's first decentralized digital currency, which relies on the network of computers that synchronize transactions with a process called mining to find valid blocks. In this way, miners repeatedly compute hashes until one finds a numerical value, which is low enough, and thus get the reward from the block. Small miners participate in the mining pool to achieve large computing power in total, and divide the reward from blocks in order to receive a smaller but steadier stream of income.

Incentive mechanisms are important for many human-involved cooperative systems, such as computation offloading [1], and crowdsourcing [2, 3]. Some research efforts [4–6] focus on designing incentive mechanisms to entice miners to participate in mining pools. However, none of them considers the cost of each miner. Designing an efficient mechanism to incentive the rational miners within the mining pool is a challenging issue.

This paper considers the rational miners with different cost. For example, people living in areas with high electricity bills will have higher mining cost than others. Their mining strategies will be influenced by their cost. To solve this problem, we design an incentive mechanism to motivate the miners to participate in the mining pool. In our incentive mechanism, the mining pool platform has the absolute control over the total payment to the miners affiliated, and miners can determine the mining actions based on the total payment decided by mining pool platform and their cost.

There are two noteworthy properties of our mechanism which are distinguished with most mining mechanisms. First, our mechanism satisfies the property of individual rationality, which can guarantee nonnegative utility for both side of miner and

F. Liu et al. (Eds.): SciSec 2019, LNCS 11933, pp. 1–9, 2019.
https://doi.org/10.1007/978-3-030-34637-9_14

platform. Second, the platform has the absolute control of the pool, and takes all risk of the miners. This means that the platform needs to pay to the miners no matter whether the pool finds a valid block, and the miners always have steady income.

2 System Model and Problem Formulation

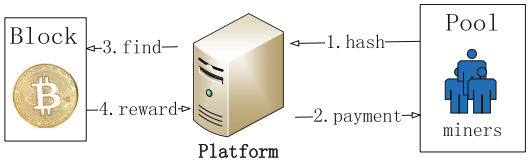

Fig. 1. A mining pool system

We use Fig. 1 to illustrate the mining pool system. The system consists of a mining pool platform and a mining pool which contains a set $M = \{1, 2, \ldots, n\}$ of rational miners, where $n \geq 2$. The Miners provide hash quantity by consuming their computing power with different unit cost. Therefore, these rational miners expect the payment to compensate for their cost. Each miner makes its own mining strategy, which is the hash quantity, and then submits it to the platform. After collecting the mining strategies from miners, the platform sends the payments to the miners. The miners with positive hash quantity send the hash to the platform. If they are lucky enough, through the integrated efforts of the miners in the pool, the mining pool will find a valid block and receive the reward from the block. On the other side, if any miner outside the pool finds a valid block, the platform can't get reward. Overall, the platform absorbs all the variance for the miners in the pool.

The platform is only interested in maximizing its own utility. Since computing power is owned by different individuals, it is reasonable to assume that miners are selfish and rational. Hence each miner only wants to maximize its own utility and won't participate in mining pool unless there is sufficient incentive.

For mining a block, the platform announces a total payment $P > 0$, motivating miners to participate in the mining pool. Each miner decides its mining strategy of participation based on the payment. The mining strategy of any miner $i \in M$ is represented by h_i, $h_i \geq 0$, the hash quantity he is willing to provide. Specifically, if $h_i = 0$, miner i indicates that he will not participate in the mining pool. The mining cost of miner i is $k_i h_i$, where $k_i > 0$ is its unit cost. Assume that the payment received by miner i is proportional to h_i. Then the utility of miner i can be defined as the difference between payment and cost:

$$u_i = \frac{h_i}{\sum_{j \in M} h_j} P - h_i k_i \tag{2.1}$$

For the reason that the mining process is subject to Poisson process [4], we can get the utility of the platform in expectation:

$$u_0 = \frac{\sum_{j\in M} h_j}{A + \sum_{j\in M} h_j} R - P \tag{2.2}$$

where $A = \frac{D\times 2^{32}}{10min}$ is the total hash power in Bitcoin network. We can estimate it easily from the difficulty of finding a valid block, D, which is adjusted periodically by the Bitcoin network. We suppose that A is a constant because it is almost stable for two weeks (the approximate period when Bitcoin network adjusts D). The probability of finding a valid block by the platform is proportional to its total computing power in the whole network. R is the reward the platform can obtain if it finds a valid block.

The objective of the platform is to decide the optimal value of P such as to maximize (2.2), while each miner $i \in M$ decides its hash quantity h_i to maximize (2.1) for the given value of P. Since no rational user is willing to mine with negative utility, user i will set $h_i = 0$ when $P \leq k_i \sum_{j\neq i \cap j\in M} h_j$.

Our objective is to design an incentive mechanism for mining pool satisfying the following four desirable properties:

- Computational Efficiency: A mechanism is computationally efficient if the outcome can be computed in polynomial time.
- Individual Rationality: Each participating miner will have a non-negative utility.
- Profitability: The value brought by the miners should be at least as large as the total payment paid to the miners. Note that profitability here is profitability in expectation because of the randomness of Bitcoin mining.
- Uniqueness: The combination of strategies is called *Nash Equilibrium*, where each player's equilibrium strategy is to maximize his/her expected utility, while all other players follow the equilibrium strategy. Uniqueness requires that there exists only one *Nash Equilibrium*. Being uniqueness, we can predict and compute the equilibrium strategies of all players exactly.

3 Incentive Mechanism

We model the mining process as *Stackelberg* game, which can be called *Mining* game. There are two phases in this mechanism: In the first phase (called payment determination), the platform announces its payment P; in the second phase (called hash determination), each miner strategizes its mining plan to maximize its own utility. Therefore, the platform is the leader and the miners are the followers in our *Mining* game. The strategy of the platform is its payment P. The strategy of any miner i is its hash amount h_i. Let $\boldsymbol{h} = (h_1, h_2, \ldots, h_n)$ denote the strategy profile of all miners' strategies. Let h_{-i} denote the strategy profile excluding h_i. As a notational convention, we write $\boldsymbol{h} = (h_i, h_{-i})$.

Note that the second process of the *Mining* game itself can be considered as a non-cooperative game, which we call the *Hash Determination* (*HD*) game. Given *Stackelberg* game formulation, we introduce the following two definitions:

Definition 1 (Nash Equilibrium, NE). *A set of strategies* $(h_1^{ne}, h_2^{ne}, \ldots, h_n^{ne})$ *is a Nash Equilibrium of the HD game if for any user i,*

$$u_i(h_i^{ne}, h_{-i}^{ne}) \geq \overline{u}_i\left(h_i, h_{-i}^{ne}\right)$$

for any $h_i \geq 0$*, where* u_i *is defined in (2.1).*

Definition 2 (Subgame Perfect Nash equilibrium). *The Stackelberg game can be solved by finding the Subgame Perfect Nash Equilibrium (SPNE), i.e. the strategy profile serves best for each player, given the strategies of the other player, and entails every player playing in a Nash Equilibrium in every subgame.*

3.1 Hash Determination

We first introduce the concept of best response strategy.

Definition 3 (Best Response Strategy). *Given* h_{-i}*, the strategy is miner i's best response strategy, denoted by* $\beta_i(h_{-i})$*, if it maximizes* $u_i(h_i, h_{-i})$ *over all* $h_i \geq 0$.

Based on the definition of *NE*, every player is playing its best response strategy in a *NE*. From (2.1), we know that $h_i \leq \frac{P}{k_i}$ because u_i will be negative otherwise. To study the best response strategy of miner i, we compute the derivatives of u_i with respect to h_i:

$$\frac{\partial u_i}{\partial h_i} = \frac{1}{\sum_{j \in M} h_j} P - \frac{h_i}{\left(\sum_{j \in M} h_j\right)^2} P - k_i \tag{3.1}$$

$$\frac{\partial^2 u_i}{\partial h_i^2} = -\frac{2p \sum_{j \in M \setminus \{i\}} h_j}{\sum_{j \in M} h_j} < 0 \tag{3.2}$$

Since the second-order derivative of u_i is negative, the utility u_i is a strictly concave function with h_i. Therefore, given any $P > 0$ and any strategy profile h_{-i} of the other miners, the best response strategy $\beta_i(h_{-i})$ of user i is unique, if it exists. If the strategy of all other miners $j \neq i$ is $h_j = 0$, then miner i does not have a best response strategy, as it can have a utility arbitrarily close to P, by setting h_i to a sufficiently small positive number. Therefore, we are only interested in the best response for miner i when $\sum_{j \in M \setminus \{i\}} h_j > 0$. Setting the first derivative of u_i to 0, we have

$$\frac{1}{\sum_{j \in M} h_j} P - \frac{h_i}{\left(\sum_{j \in M} h_j\right)^2} P - k_i = 0 \tag{3.3}$$

Solving for h_i in (3.3), we obtain

$$h_i = \sqrt{\frac{P\sum_{j\in M\setminus\{i\}} h_j}{k_i}} - \sum_{j\in M\setminus\{i\}} h_j \tag{3.4}$$

Remark: h_i is the total hash that can make i achieve maximum utility in the current mining pool. Of course, i can put the remaining hash power to any other pools.

If the right-hand side of (3.4) is positive, it is also the best response strategy of miner i, due to the concavity of u_i. If the right-hand side of (3.4) is less than or equal to 0, then miner i does not participate in the mining task by setting $h_i = 0$ (to avoid a deficit). Hence we have

$$\beta(h_i) = \begin{cases} 0 & if\ P \le k_i \sum\limits_{j\neq i \cap j\in M} h_j \\ \sqrt{\frac{P\sum_{j\in M\setminus\{i\}} h_j}{K_i}} - \sum\limits_{j\in M\setminus\{i\}} h_j & otherwise \end{cases} \tag{3.5}$$

These analyses lead to Algorithm 1 for computing an *NE* of the *HD* game.

Algorithm 1: Computation of the NE

Sort miners according to their unit costs, $k_1 \le k_2 \le \ldots \le k_n$;
$S \leftarrow \{1,2\},\ i \leftarrow 3$;
while $i \le n$ **and** $k_i < \frac{k_i + \sum_{j\in S} k_j}{|S|}$ **do**
  $S \leftarrow S \cup \{i\}, i \leftarrow i+1$;
end
for each $i \in M$ **do**
  if $i \in S$ **then** $h_i^{ne} = \frac{(|S|-1)P}{\sum_{j\in S} k_j}\left(1 - \frac{(|S|-1)k_i}{\sum_{j\in S} k_j}\right)$;
  else $h_i^{ne} = 0$;
end
return $h^{\mathrm{ne}} = (h_1^{\mathrm{ne}}, h_2^{\mathrm{ne}}, \ldots, h_n^{\mathrm{ne}})$;

Theorem 1. *The strategy profile* $h^{\mathrm{ne}} = \left(h_1^{ne}, h_2^{ne}, \ldots, h_n^{ne}\right)$ *computed by Algorithm 1 is a NE of the HD game.*

PROOF 1: From Algorithm 1, we get:

(1) for $i \notin S$, $k_i \ge \frac{\sum_{j\in S} k_j}{n_0 - 1}$

(2) $\sum_{j\in S} h_j^{ne} = \frac{(|S|-1)P}{\sum_{j\in s} k_j}$

(3) for $i \in S$, $\sum_{j\in S\setminus\{i\}} h_j^{ne} = \frac{(|S|-1)^2 P k_i}{\left(\sum_{j\in S} k_j\right)^2}$

There are two cases:

① For $i \notin S$: It is obvious that $k_i \sum_{j\in S\setminus\{i\}} h_j^{ne} = k_i \sum_{j\in S} h_j^{ne}$. Using (1) and (2), we get $k_i \sum_{j\in S\setminus\{i\}} h_j^{ne} \geq P$. According to (3.5), we have $\beta\left(h_{-i}^{ne}\right) = 0$. So, it is the best response strategy given h_{-i}^{ne} for $\notin S$.

② For $i \in S$: From the Line 3 of Algorithm 1, we get $(i-1)\, k_i < \sum_{j=1}^{i} k_j$. Then

$$\begin{aligned}(n_0-1)k_i &= (i-1)k_i + (n_0-i)k_i \\ &< \sum_{j=1}^{i} k_j + \sum_{j=i}^{n} k_j = \sum_{j=1}^{n} k_j\end{aligned}$$

Thus, $k_i < \frac{\sum_{j=1}^{n} k_j}{n_0-1}$. Furthermore, using (3) we have

$$\begin{aligned}k_i \sum_{j\in M\setminus\{i\}} h_j^{ne} = k_i \sum_{j\in S\setminus\{i\}} h_j^{ne} &= k_i \frac{(n_0-1)^2 P k_i}{\left(\sum_{j\in S} k_j\right)^2} = P\frac{(n_0-1)^2 k_i^2}{\left(\sum_{j\in S} k_j\right)^2} \\ &< P\frac{(n_0-1)^2\left(\frac{\sum_{j\in S} k_j}{n_0-1}\right)^2}{\left(\sum_{j\in S} k_j\right)^2} = P\end{aligned}$$

Thus, $k_i < \frac{P}{\sum_{j\in M\setminus\{i\}} h_j^{ne}}$. According to (3.5), we have

$$\beta\left(h_{-i}^{ne}\right) = \sqrt{\frac{P\sum_{j\in M\setminus\{i\}} h_j}{k_i}} - \sum_{j\in M\setminus\{i\}} h_j = \frac{(n_0-1)P}{\sum_{j\in S} h_j} - \frac{(n_0-1)^2 P h_i}{\left(\sum_{j\in S} h_j\right)^2} = h_i^{ne}$$

In summary of ① and ②, h^{ne} is an *NE* of *HD* game. ■

Theorem 2. *The NE in Theorem 1 is unique.*

PROOF 2: First, we assume that there exists one miner $i \in M$ whose $h_i' \neq h_i^{ne}$, but it also satisfies $u_i\left(h_i', h_{-i}^{ne}\right) \geq u_i\left(h_i, h_{-i}^{ne}\right)$ for any $h_i > 0$.

① If $i \notin S$, There must have $h_i' > 0$ for the reason that $h_i' \neq h_i^{ne}$ and $h_i^{ne} = 0$. However, it cannot change the truth that $k_i < \frac{k_i + \sum_{j\in S} k_j}{|S|}$, which means that $k_i \sum_{j\in S\setminus\{i\}} h_j^{ne} \geq P$ (In proof 1). So, its h_i' have to be 0 in order to avoid a deficit. $h_i' = 0$ is contradict with $h_i' > 0$.

② If $i \in S$, reminding that (2.1) is a concave function and it reaches the maximum when $h_i = h_i^{ne}$. So, $u_i\left(h_i', h_{-i}^{ne}\right) < u_i\left(h_i^{ne}, h_{-i}^{ne}\right)$. Which is contradict with $u_i\left(h_i', h_{-i}^{ne}\right) \geq u_i\left(h_i, h_{-i}^{ne}\right)$ for any $h_i > 0$.

In summary of ① and ②, there is no any miner $i \in M$ whose $h_i^{'} \neq h_i^{ne}$, and it still satisfies $u_i\left(h_i^{'}, h_{-i}^{ne}\right) \geq u_i\left(h_i, h_{-i}^{ne}\right)$ for any $h_i > 0$. ■

3.2 Platform Utility Maximization

According to the above analysis, the platform, which is the leader in the *Stackelberg* game, knows that there exists a unique NE for the miner for any given value of P. Hence the platform can maximize its utility by setting the optimal value of P. Substituting h^{ne} into (2.2), we have

$$u_0 = \frac{X}{A+X} - P \tag{3.6}$$

where $X = \sum_{j\in S} \frac{(|S|-1)P}{\sum_{j\in S} k_j}\left(1 - \frac{(|S|-1)k_i}{\sum_{j\in S} k_j}\right)$, and $X' = \frac{\partial X}{\partial P} = \sum_{j\in S} \frac{(|S|-1)}{\sum_{j\in S} k_j}\left(1 - \frac{(|S|-1)k_i}{\sum_{j\in S} k_j}\right)$. Obviously, X' is a constant. We use Y to represent X'.

Theorem 3. There exists a unique *Stackelberg* Equilibrium (P^*, h^{ne}) in the Mining game, where P^* is the unique value of P to maximize the platform utility in (3.6) over $P \in [0, \infty)$.

PROOF 3: We have

$$\frac{\partial u_0}{\partial P} = \frac{AY}{(A+X)^2} - 1 \tag{3.7}$$

$$\frac{\partial^2 u_0}{\partial P^2} = -\frac{2AY^2}{(A+X)^3} < 0 \tag{3.8}$$

Therefore the utility u_0 defined in (3.6) is a strictly concave function of P, for any $P \in [0, \infty)$. Since the value of u_0 in (3.6) is 0 if $P = 0$, and goes to $-\infty$ when P goes to ∞, it has a unique maximum value P^* that can be effectively computed using either bisection or Newton's method. ■

In the following, we present the analysis, demonstrating that *Mining* game can achieve the desired properties.

Theorem 4. *Mining game is computationally efficient, individually rational, profitable, and has unique Equilibrium.*

PROOF 4: The Sorting in Line 1 can be done in $O(nlogn)$ time. The while-loop (Lines 3–5) requires a total time of $O(n)$. The for-loop (Lines 6–9) requires a total time of $O(n)$. Hence the time complexity of Algorithm 1 is $O(nlogn)$.

The property of individually rational is obvious from (3.3). The property of profitability is also obvious because the pool can always set $P = 0$ in (2.2) to get $u_0 = 0$ (In this case, all miners should set $h_i = 0$ according to (3.5)). This means that u_0 can be at least 0 because the u_0 defined in (3.6) is a strictly concave function of P. The uniqueness of Equilibrium has been proved in Theorem 3. ■

4 Performance Evaluation

We consider that the block reward R is 100. The default number of miners in the pool is 100. We assume the cost of each miner subjects to normal distribution or uniform distribution with $\mu = 4.0788 \times 10^{-12}$, which can be estimated from the miners in [7].

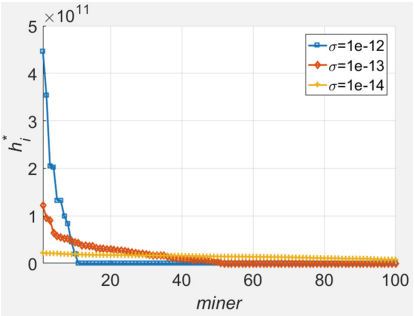

Fig. 2. Hash amount provided by each miner under normal distribution

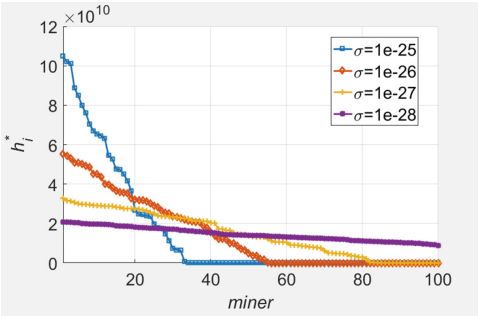

Fig. 3. Hash amount provided by each miner under uniform distribution

To explore the influence of σ further, we pick some meaningful value of σ and plot Figs. 2 and 3. We find that when σ is not large enough, there is only part of the miners provide hash to the pool, which means that other miners set $h_i = 0$. Second, we find that the larger σ is, the steeper the curve. Extremely, the curve can be a horizontal line when $\sigma = 0$. Third, notice that the unit cost of miners is sorted by $k_1 \leq k_2 \leq \ldots \leq k_n$. We find that miners with lower cost are willing to provide more hash to the pool since the NE computed by Algorithm 1 is a decreasing function with the unit cost. Accordingly, from Fig. 2 we find the miners, who contribute more hash, will be paid more.

5 Related Work

Since launched in 2009, Bitcoin has received lots of attention in the research community. Eyal *et al.* [8], and Kiayias *et al.* [9] all focus on the problem called selfish mining in Bitcoin network. Rosenfeld *et al.* [4], Schrijvers *et al.* [5], and Lewenberg *et al.* [6] focus on profit distribution in a mining pool. Eyal *et al.* [10] focus on improving the protocol of the Bitcoin network. However, there isn't much work taking the cost into consideration.

6 Conclusion

We have proposed a novel pool mining mechanism based on *Stackelberg* game to incentive the rational miners in Bitcoin mining pool. Through both theoretical analysis and simulations, we demonstrate that the proposed mechanism achieves computational efficiency, individual rationality, and profitability. Moreover, we show that the *Stackelberg* game has a unique Equilibrium.

Acknowledgements. This work has been supported in part by the NSFC (No. 61872193, 61872191, 61872197).

References

1. Liu, Y., et al.: Incentive mechanism for computation offloading using edge computing: a Stackelberg game approach. Comput. Netw. **129**, 399–409 (2017)
2. Xu, J., Xiang, J., Yang, D.: Incentive mechanisms for time window dependent tasks in mobile crowdsensing. IEEE Trans. Wireless Commun. **14**(11), 6353–6364 (2015)
3. Xu, J., Rao, Z., Xu, L., et al.: Incentive mechanism for multiple cooperative tasks with compatible users in mobile crowd sensing via online communities. In: IEEE Transactions on Mobile Computing, (2019)
4. Rosenfeld, M.: Analysis of bitcoin pooled mining reward systems. arXiv preprint arXiv: 1112.4980 (2011)
5. Schrijvers, O., Bonneau, J., Boneh, D., Roughgarden, T.: Incentive compatibility of bitcoin mining pool reward functions. In: Grossklags, J., Preneel, B. (eds.) FC 2016. LNCS, vol. 9603, pp. 477–498. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54970-4_28
6. Lewenberg, Y., et al.: Bitcoin mining pools: a cooperative game theoretic analysis. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (2015)
7. Btcfans Homepage. http://mining.btcfans.com/. Last accessed 21 Apr 2019
8. Eyal, I., Sirer, E.G.: Majority is not enough: bitcoin mining is vulnerable. Commun. ACM **61**(7), 95–102 (2018)
9. Kiayias, A., Koutsoupias, E., Kyropoulou, M., et al.: Blockchain mining games. In: Proceedings of the 2016 ACM Conference on Economics and Computation, pp. 365–382. ACM (2016)
10. Eyal, I., Gencer, A.E., Sirer, E.G., et al.: Bitcoin-ng: a scalable blockchain protocol. In: 13th Symposium on Networked Systems Design and Implementation, pp. 45–59 (2016)