IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON NEW ADVANCES IN BLOCKCHAIN-BASED WIRELESS NETWORKS

Received June 28, 2020, accepted August 5, 2020, date of publication August 18, 2020, date of current version September 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3017716

A Novel Computing Power Allocation Algorithm for Blockchain System in Multiple Mining Pools Under Withholding Attack

YOURONG CHEN[1,2], HAO CHEN[2], MENG HAN[3], (Member, IEEE), BANTENG LIU[1], QIUXIA CHEN[1], AND TIAOJUAN REN[1]

[1]College of Information Science and Technology, Zhejiang Shuren University, Hangzhou 310015, China
[2]School of Information Science and Engineering, Changzhou University, Changzhou 213164, China
[3]College of Computing and Software Engineering, Kennesaw State University, Marietta, GA 30060, USA

Corresponding author: Meng Han (menghan@kennesaw.edu)

This work was supported in part by the Natural Science Foundation of Zhejiang Province of China under Grant Y21F020114 and Grant LQ18F030006, and in part by the Public Welfare Technology Application and Research Projects of Zhejiang Province of China under Grant LGF19F010005.

ABSTRACT To overcome the fast-changing block withholding attacks among multiple mining pools composed of miners in the blockchain system, this paper proposes a mining pool computing power allocation (MPPA) algorithm, which significantly improves the revenues of mining pools with block withholding attacks. MPPA first establishes the revenue optimization model of mining pools, which includes current adequate total computing power, the revenues of honest mining, and the revenues of block withholding attacks. Then MPPA calculates the revenue gain generated by block withholding attacks on other mining pools. To adjust the fixed computing power in each iteration, we have the mining pool computing power allocation algorithm with a fixed change of computing power (MPPA_F). To adjust the optimal recovery and attack computing power, we have the mining pool computing power allocation algorithm with an optimal change of computing power (MPPA_O). The simulation results demonstrate that MPPA_F and MPPA_O can find the optimized solutions of power computing allocation for each mining pool and outperform the state-of-arts such as WSFS, ALLC, and ALLD.

INDEX TERMS Computing power allocation, block withholding attacks, multiple mining pools, revenue optimization model.

I. INTRODUCTION

With the popularity of digital cryptocurrencies such as bitcoin, blockchain technology has attracted people's attention. First of all, blockchain technology is applied to digital cryptocurrencies, such as bitcoin, ETH, XRP and EOS. Considering the characteristics of blockchain technology, such as decentralization, trust mechanism and data encryption, blockchain technology is also applied to intelligent transportation, health care and other fields of the internet of things besides digital crypto-currency [1]–[4]. So we consider the blockchain in our paper. In the blockchain [5]–[8], miners follow the same accounting transaction rules and reach the consensus by the proof of work consensus mechanism (POW) [9]. Through the generation of blocks (mining) to obtain revenues, data among nodes share safely. However, a single node needs a long time to mine the block successfully. To improve the node's mining revenue, we consider that many miners form a mining pool in the blockchain. The mining pool includes a pool manager and multiple nodes. All nodes carry out mining at the same time, i.e. they carry out part work certificate. Receiving part work certificate of each node, the pool manager estimates the work of each node based on the ratio of its part work certificate. If a node generates a full work certificate, then it sends the work certificate to the pool manager. The pool manager publishes the full work certificate to the bitcoin network and distributes the revenue according to the actual workload of each node [10].

However, the security issues are still challenging the blockchain system due to selfish mining attack. In pool mining, selfish mining attack is an attack strategy based on POW. The attacker (malicious miner or malicious mining

The associate editor coordinating the review of this manuscript and approving it for publication was Yuan Gao.

pool) does not broadcast the newly mined block. It chooses to keep the block or release the block when appropriate. Pool block withholding (PBWH) attack is a recently discovered selfish mining attack. In the PBWH, the attack pool infiltrates some computing power into the attacked mining pool. Then it performs PBWH attack. That is, it retains all newly discovered blocks in the attacked pool [11], [12]. At present many scholars have studied the attack strategy between two mining pools. However, this strategy is difficult to be applied to the mutual real-time attacks among multiple mining pools. The mutual attack strategy of multiple mining pools does not consider the mining cost and the dynamic PBWH attacks among multiple mining pools. Therefore, based on the above references, we propose a mining pool computing power allocation algorithm with block withholding attacks among multiple mining pools (MPPA) [13]. Briefly, our contributions are as follows:

1. MPPA divides the computing power in the mining pool into the computing power of honest mining and computing power of block withholding attack according to their functions. Then MPPA proposes the revenue optimization model of each mining pool with the costs of honest mining and block withholding attack.

2. MPPA calculates the revenue gain generated by block withholding attacks on other mining pools. For adjusting fixed computing power in each iteration, we have the mining pool computing power allocation algorithm with fixed change of computing power (MPPA_F). For adjusting the optimal recovery and attack computing power, we have the mining pool computing power allocation algorithm with optimal change of computing power (MPPA_O).

3. MPPA_F and MPPA_O could effectively ensure the maximum revenue of single mining pool, and improve the overall revenue of all mining pools. MPPA_F and MPPA_O are suitable for the fast-changing block withholding attack environment among multiple mining pools.

The rest of the paper is organized as follows. In Section 2, we describe the related work. In Section 3, we describe our algorithm principles, which include model establishment and model solution. The simulation results are presented in Section 4. Finally, we conclude the paper and describe future work in Section 5.

II. RELATED WORK

Rosenfeld [14] firstly proposes the concept of block withholding attack in 2011. In his paper, he defines block withholding attack as a miner's act in a mining pool deliberately discarding blocks or delaying the submission of blocks to the pool manager. Then Eyal [11] considers that the mining pool can execute block withholding attacks. In his paper, in order to improve their revenues, malicious mining pools assign computing power to carry out block withholding attack on other mining pools. Afterward, Luu *et al.* [15] consider that an attacker carries out block withholding attacks on a mining pool or multiple mining pools. And they find that there is a motive for attackers to carry out block withholding attack in the long-run. Then more and more scholars study block withholding attacks. Currently, some scholars focus on block withholding attacks between two mining pools. For example, Tang *et al.* [16], [17] analyze the existing conditions of Nash equilibrium in the process of POW, and use the zero-determinant strategy to optimize the miner's strategy selection. Considering the cooperation of two mining pools, Di *et al.* [18] analyze the parameter value and corresponding mining strategy between the two mining pools according to the mining pools' computing amount. Wang *et al.* [19] propose a two-stage game model to consider whether the mining pool is open or not and whether it is attacked. It analyzes the Nash equilibrium condition of the game based on weight to obtain a strategy selection method of mining pool based on the game theory. Chatterjee *et al.* [20] establish a model of digital cryptocurrency attack. Then it proposes an ergodic average revenue game algorithm. Houlihan *et al.* [21] analyze the optimal penetration algorithm of two mining pools attacking each other under extreme symmetry, symmetry and general conditions. Then it obtains the pure Nash equilibrium condition under the anarchy state. Considering sponsored block withholding attacks, Bag *et al.* [22] analyze the computing power allocation strategy of nodes attacking one or two mining pools. Considering the computing power of mining pool and block propagation delay, Liu *et al.* [23] propose an evolutionary mining strategy with the probability, average time, revenue and other formulas of the mining pool. Kim *et al.* [24] establish a revenue model of mining pool, and analyze the impact of block withholding attack on the migration of mining pools and miners with evolutionary game theory. The works [16]–[24] mainly consider the situation of mutual attacks between two mining pools. But there are multiple mining pools attacking each other in practice.

Therefore, some scholars focus on the PBWH attack method among multiple mining pools. For example, Eyal [11] proposes the strategy selection of maximum revenue with mutual attacks among multiple mining pools. Wang *et al.* [25] regard the game behavior among multiple mining pools as an iterative prisoner's dilemma model. It uses the policy gradient algorithm of deep reinforcement learning to select the mining strategy of multiple mining pools. Considering revenues of block withholding attacks, Luu *et al.* [15] establish a revenue maximum model with computing power revenue and transaction volume revenue. Considering that a node attacks a single mining pool and multiple mining pools, Tosh *et al.* [26] propose a revenue computing method to search the Nash equilibrium point. Considering the revenue of the mining pool and miner selection, Haghighat *et al.* [27] analyze the mining pool profitability with mining pool attractiveness and miner migration rate. Considering the uncle reward, Chang *et al.* [28] propose computing power allocation strategy of mining pools to improve the revenues of mining pools. But the works [11], [15], [25]–[28] do not consider the cost of mining and the dynamic attack among multiple mining pools.

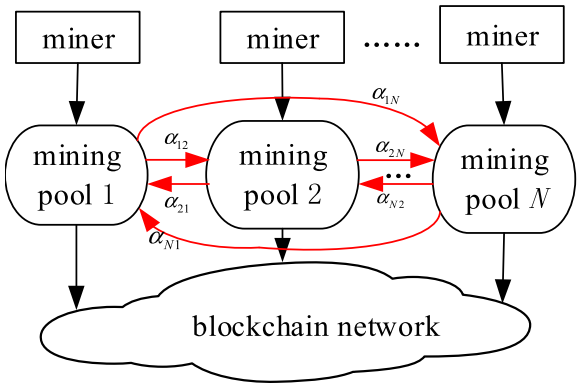

FIGURE 1. Algorithm principle diagram.

III. ALGORITHM PRINCIPLES

As shown in Fig.1, there are multiple mining pools composed of multiple miners in the network. Based on the current situation, each mining pool selects computing power for honest mining, and selects computing power to infiltrate into other mining pools for block withholding attacks. The goal of each pool is to maximize its revenue. However, the following two problems still need to be solved to obtain the optimal computing power allocation scheme of each mining pool, which could maximize their revenues. The first is how to set up the revenue mining pool' optimization model according to the current computing power of all mining pools and the computing power allocation of each mining pool. The second is how to use a heuristic algorithm to solve the optimization model according to the current situation. The details are as follows.

A. MODEL ESTABLISHMENT

We assume that there are N mining pools in the network that receive other computing power, and attack other mining pools. x_i represents the computing power of mining pool i, α_{ij} represents the computing power ratio of mining pool i in mining pool j. When $i = j$, α_{ii} represents the computing power ratio of the mining pool i for honest mining. When $i \neq j$, α_{ij} represents the computing power ratio of mining pool i in mining pool j for block withholding attack. The current effective total computing power in the blockchain is

$$A = \sum_{i=1}^{N} x_i \alpha_{ii} \tag{1}$$

where, A represents the current effective total computing power. The mining difficulty in POW is adjusted according to the current total network computing power, and each mining pool needs to consume electric energy and other resources during mining. Therefore, the revenue of honest mining for each mining pool i is

$$S_{ii} = \frac{x_i \alpha_{ii}}{A} \frac{x_i \alpha_{ii}}{x_i \alpha_{ii} + \sum_{j \neq i} x_j \alpha_{ji}} - C_w x_i \alpha_{ii}$$
$$= \frac{x_i \alpha_{ii}}{A} \frac{\alpha_{ii}}{\alpha_{ii} + \sum_{j \neq i} (x_j / x_i) \alpha_{ji}} - C_w x_i \alpha_{ii} \tag{2}$$

where, C_w represents the cost coefficient of honest mining in mining pool i, which is related to the computing power of the mining pool and the local electricity price. We consider that the cost of block withholding attack in the mining pool is different from the cost of honest mining. Then the revenue of block withholding attack in mine pool i against mine pool j is

$$S_{ij} = \frac{x_j \alpha_{jj}}{A} \frac{x_i \alpha_{ij}}{x_j \alpha_{jj} + \sum_{i \neq j} x_i \alpha_{ij}} - C_p x_i \alpha_{ij}$$
$$= \frac{\alpha_{jj}}{A} \frac{x_i \alpha_{ij}}{\alpha_{jj} + \sum_{i \neq j} (x_i / x_j) \alpha_{ij}} - C_p x_i \alpha_{ij} \tag{3}$$

where, C_p represents the cost coefficient of block withholding attack of mining pool i. $\beta_{ij} = x_i / x_j$, then the average revenue of each mining pool is

$$S_{av}^{i} = \frac{S_{ii} + \sum_{j \neq i} S_{ij}}{x_i}$$
$$= \frac{\alpha_{ii}}{A} \frac{\alpha_{ii}}{\alpha_{ii} + \sum_{j \neq i} \beta_{ji} \alpha_{ji}} - C_w \alpha_{ii}$$
$$+ \sum_{j \neq i} \left(\frac{\alpha_{jj}}{X} \frac{\alpha_{ij}}{\alpha_{jj} + \sum_{i \neq j} \beta_{ij} \alpha_{ij}} - C_p \alpha_{ij} \right) \tag{4}$$

In order to ensure its own revenue, each mining pool maximizes its revenue as much as possible. Therefore, we establish the revenue optimization model of the mining pool i as follows.

$$\max \left(\begin{array}{l} \frac{\alpha_{ii}}{\sum_{i=1}^{N} x_i \alpha_{ii}} \frac{\alpha_{ii}}{\alpha_{ii} + \sum_{j \neq i} \beta_{ji} \alpha_{ji}} - C_w \alpha_{ii} \\ + \sum_{j \neq i} \left(\frac{\alpha_{jj}}{\sum_{i=1}^{N} x_i \alpha_{ii}} \frac{\alpha_{ij}}{\alpha_{jj} + \sum_{i \neq j} \beta_{ij} \alpha_{ij}} - C_p \alpha_{ij} \right) \end{array} \right)$$
$$\text{s.t. } 0 \leq \alpha_{ii} \leq 1$$
$$0 \leq \alpha_{ij} \leq 1, \forall j$$
$$\alpha_{ii} + \sum_{j \neq i} \alpha_{ij} = 1 \tag{5}$$

B. MODEL SOLUTION

The optimization model (5) can be solved by artificial intelligence algorithms such as genetic algorithm and particle swarm optimization algorithm. But the solving process is very complex. Considering the computing power, revenue and other information of the mining pool, we use the heuristic algorithm to calculate the revenue increment of block withholding attacks on other mining pools according to computing power loss interval Δ of the mining pool. After that, according to the increase and decrease of revenue increment, the mining pool selects different strategies of computing power allocation. If the revenue increment decreases, the recovery computing power will be used for honest mining.

Otherwise, the mining computing power will be transferred to block withholding attacks on other mining pools. Therefore, the solving process of the optimization model (5) can transform into the optimal computing power allocation problem of the mining pool i and other mining pools.

According to the computing power of mining pool i and the computing power of mining pool j, the current total revenue of honest mining in mining pool i and its block withholding attack on mining pools j are calculated.

$$f(\alpha_{ii}, \alpha_{ij}) = \frac{\alpha_{ii}}{\sum_{i=1}^{N} x_i \alpha_{ii}} \frac{\alpha_{ii}}{\alpha_{ii} + \sum_{j \neq i} \beta_{ji} \alpha_{ji}} - C_w \alpha_{ii} + \frac{\alpha_{jj}}{\sum_{i=1}^{N} x_i \alpha_{ii}} \frac{\alpha_{ij}}{\alpha_{jj} + \sum_{i \neq j} \beta_{ij} \alpha_{ij}} - C_p \alpha_{ij} \quad (6)$$

When the computing power Δ of honest mining in mining pool i transforms into the computing power of block withholding attack on pool j, the revenue of mining pool i is

$$f(\alpha_{ii} - \Delta, \alpha_{ij} + \Delta) = \frac{\alpha_{ii} - \Delta}{\sum_{i=1}^{N} x_i \alpha_{ii} - x_i \Delta} \frac{\alpha_{ii} - \Delta}{\alpha_{ii} - \Delta + \sum_{j \neq i} \beta_{ji} \alpha_{ji}} - C_w \alpha_{ii} + C_w \Delta + \frac{\alpha_{jj}}{\sum_{i=1}^{N} x_i \alpha_{ii} - x_i \Delta} \frac{\alpha_{ij} + \Delta}{\alpha_{jj} + \sum_{i \neq j} \beta_{ij} \alpha_{ij} + \beta_{ij} \Delta} - C_p \alpha_{ij} - C_p \Delta \quad (7)$$

Considering the block withholding attack of each mining pool, we retain part of computing power υ for honest mining, which could attract other mining pools to be attacked, and avoid attracting the attention of the blockchain. Therefore, let $A = \sum_{i=1}^{N} x_i \alpha_{ii} \neq 0, B = \alpha_{ii} + \sum_{j \neq i} \beta_{ji} \alpha_{ji} \neq 0, C = \alpha_{jj} + \sum_{i \neq j} \beta_{ij} \alpha_{ij} \neq 0$, where A represents the total effective computing power of the whole network, B represents the total computing power in mining pool i, C represents the total computing power in mining pool j. In the mining process, honest computing power needs mine fully, and computing power of block withholding attack sometimes does not need to play its full role [29]. We use the average principle to allocate the revenue of mining pools. Therefore, the cost of block withholding attack is not higher than the cost of honest mining, namely $0 < C_P \leq C_w$. Then the revenue increment Δf_{ij} of the computing power of block withholding attack on mining pool j *is* (8), as shown at the bottom of the next page.

1) MPPA_F SOLUTION (FIXED LOSS INTERVAL OF COMPUTING POWER)

We assume that each mining pool i can obtain the information of its own, such as its total computing power B, initial computing power x_i, current revenue R_{ii}, the ratio α_{ij} of computing power used to be block withholding attack on other mining pools. It can find the total effective computing power A of the whole network and the information of mining pool j, such as its total computing power C, initial computing power x_j and current revenue R_{jj}. Because it is impossible to obtain the computing power of block withholding attack of other mining pool on itself, mining pool i can't obtain its computing power coefficient α_{ii} and the computing power coefficient α_{jj} of other mining pool j. But it can estimate the coefficient based on current revenue as follows.

$$\alpha_{ii} = R_{ii} A / x_i, \alpha_{jj} = R_{jj} A / x_j \quad (9)$$

After each mining pool has known the computing power coefficient α_{ii} of its own and other mining pool j, the computing power allocation strategy of each mining pool i according to fixed revenue increment Δf_{ij} is as follows.

Step 1: After a certain time ΔT, according to formula (8), the mining pool i calculates revenue increment Δf_{ij} between itself and other mining pools. Then it establishes a revenue increment matrix $\Delta\gamma_1$ by the calculation result.

$$\Delta\gamma_1 = [\ \Delta f_{i1} \quad \Delta f_{i2} \quad \ldots \quad \Delta f_{ij} \quad \ldots \quad \Delta f_{iN}\ ] \quad (10)$$

where, $\Delta f_{ij} = 0, if\ i = j$. It represents that the mining pool i does not carry out block withholding attacks on its own mining pools.

Step 2: Mining pool i analyzes the revenue increment matrix $\Delta\gamma_1$. If $\Delta f_{ij} < 0$, it records Δf_{ij} and the corresponding mining pool number. Then it obtains the set of $\Delta f_{ij} < 0$. Mining pool i selects the mining pool, which has the maximum absolute value of revenue increment as the object of computing power recovery. If multiple mining pools have the same maximum absolute value of revenue increment, mining pool i randomly selects one mining pool as the object of computing power recovery.

Step 3: If the recoverable computing power of the object is higher than Δ, mining pool i recovers the computing power Δ for honest mining. Otherwise, it recovers all the computing power.

Step 4: Mining pool i calculates $\Delta f'_{ij}$ between mining pool i and other mining pool j respectively with the formula (8). Then it obtains the revenue increment matrix $\Delta\gamma_2$ by the calculation result.

Step 5: Mining pool i analyzes revenue increment matrix $\Delta\gamma_2$. If $\Delta f'_{ij} > 0$, it records $\Delta f'_{ij}$ and the corresponding mining pool number. Then it obtains the set of $\Delta f'_{ij} > 0$. Mining pool i actively selects the mining pool, which has the maximum value of revenue increment as the object of computing power assignment. If multiple mining pools have the same maximum value of revenue increment, mining pool i randomly selects one mining pool as the computing power assignment object.

Step 6: Mining pool i needs to reserve the computing power υ for honest mining. If $x_i \alpha_{ii} - \upsilon$ is higher than Δ, it assigns computing power Δ to block withholding attack on the computing power assignment object. Otherwise, it assigns $x_i \alpha_{ii} - \upsilon$. Skip back to step 1.

The pseudo code of MPPA_F is as follows:

Each mining pool repeats steps 1-6 above to adjust the its computing power allocation from time to time. Finally, they can obtain their computing power allocation strategy to improve their revenue.

2) MPPA_O SOLUTION (OPTIMAL CHANGE OF LOSS INTERVAL OF COMPUTING POWER)

Because each mining pool's situation is different, mining pool i calculates optimal attack computing power and optimal recovery computing power according to the computing power information of mining pool i and other mining pool j. That is, the mining pool calculates the revenue increment Δf_{ij}^{at} with the formula (8) when the optimal value's range of attack computing power is $0 \le \Delta \le \alpha_{ii}$. If $\Delta f_{ij}^{at} > 0$, the computing power Δ is beneficial for block withholding attack on the mining pool j. Otherwise, it is not beneficial. The mining pool selects the maximum Δf_{ij}^{bat} and selects the corresponding Δ_{ij}^{bat} as optimal attack computing power. Similarly the mining pool calculates the revenue increment Δf_{ij}^{re} when the optimal value's range of recovery computing power is $-\alpha_{ij} \le \Delta \le 0$. If $\Delta f_{ij}^{re} > 0$, the recovery computing power Δ is beneficial. Otherwise, it is not beneficial. The pool selects the maximum Δf_{ij}^{bre} and selects the corresponding $\left|\Delta_{ij}^{bre}\right|$ as optimal recovery computing power. The specific calculation method of optimal attack computing power and optimal recovery computing power is as follows.

The following formula by substituting the variable Δ into formula (8) is obtained. Formula (11), as shown at the bottom of the next page, transforms into the following form.

$$\Delta f_{ij}^{at} = \frac{\phi_1\Delta^4 + \phi_2\Delta^3 + \phi_3\Delta^2 + \phi_4\Delta}{\phi_5\Delta^3 + \phi_6\Delta^2 + \phi_7\Delta + \phi_8} \tag{12}$$

where, the definitions of $\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7$ and ϕ_8 are as follows.

$$\begin{aligned}
\phi_1 &= ABC\beta_{ij}x_iC_w - ABC\beta_{ij}x_iC_p,\\
\phi_2 &= ABC\beta_{ij} - \alpha_{ii}^2Cx_i\beta_{ij} - \alpha_{jj}\alpha_{ij}x_i\beta_{ij}B + ABC^2x_iC_w\\
&\quad - A^2BC\beta_{ij}C_w - AB^2Cx_i\beta_{ij}C_w - ABC^2x_iC_p\\
&\quad + A^2BC\beta_{ij}C_p + AB^2Cx_i\beta_{ij}C_p,\\
\phi_3 &= ABC^2 - \alpha_{ii}^2C^2x_i + \alpha_{ii}^2AC\beta_{ij} + \alpha_{ii}^2BC\beta_{ij}x_i - 2\\
&\quad \times \alpha_{ii}ABC\beta_{ij} + \alpha_{jj}\alpha_{ij}x_i\beta_{ij}B^2 - AC\alpha_{jj}B - \alpha_{jj}\alpha_{ij}Cx_iB\\
&\quad + \alpha_{jj}\alpha_{ij}A\beta_{ij}B + A^2B^2C\beta_{ij}C_w - A^2BC^2C_w\\
&\quad - AB^2C^2x_iC_w - A^2B^2C\beta_{ij}C_P + A^2BC^2C_P\\
&\quad + AB^2C^2x_iC_p,\\
\phi_4 &= C^2\alpha_{ii}^2A + C^2\alpha_{ii}^2Bx_i - 2\times\alpha_{ii}ABC^2 + AC\alpha_{jj}B^2\\
&\quad + \alpha_{jj}\alpha_{ij}Cx_iB^2 - \alpha_{jj}\alpha_{ij}A\beta_{ij}B^2 + A^2B^2C^2C_w\\
&\quad - A^2B^2C^2C_p,\\
\phi_5 &= ABCx_i\beta_{ij},\\
\phi_6 &= ABC^2x_i - A^2BC\beta_{ij} - AB^2Cx_i\beta_{ij},\\
\phi_7 &= -A^2BC^2 - AB^2C^2x_i + A^2B^2C\beta_{ij}\\
\phi_8 &= A^2B^2C^2,
\end{aligned} \tag{13}$$

Considering that each mining pool retains the computing power υ for honest mining, the total computing power of honest mining in each mining pool is not equal to 0 in any case. Moreover, the computing power B in mining pool i and computing power C in mining pool j are not equal to 0. That is, $A > 0, B > 0, C > 0$. Therefore, when $0 \le \Delta \le \alpha_{ii}$, $(A - x_i\Delta)(B - \Delta) > 0$, $(A - x_i\Delta)(C + \beta_{ij}\Delta) > 0$, the denominator in formula (12) is not equal to 0. Then we use the derivative method to get the maximum value with the definition field $0 \le \Delta \le \alpha_{ii}$ of formula (8)-(12) [30].

Calculate the derivative of formula (12). The following formula is obtained in (14), as shown at the bottom of the next page. where, the definitions of $\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6$ and φ_7 are as follows.

$$\begin{aligned}
&\varphi_1 = \phi_1\phi_5, \varphi_2 = 2\times\phi_1\phi_6, \varphi_3 = 3\times\phi_1\phi_7 + \phi_2\phi_6 - \phi_3\phi_5,\\
&\varphi_4 = 4\times\phi_1\phi_8 + 2\times\phi_2\phi_7 - 2\times\phi_4\phi_5,\\
&\varphi_5 = 3\times\phi_2\phi_8 + \phi_3\phi_7 - \phi_4\phi_6, \varphi_6 = 2\times\phi_3\phi_8, \varphi_7 = \phi_4\phi_8
\end{aligned} \tag{15}$$

The formula (14) transforms into the following form.

$$g_1(\Delta) = \varphi_1\Delta^6 + \varphi_2\Delta^5 + \varphi_3\Delta^4 + \varphi_4\Delta^3 + \varphi_5\Delta^2 + \varphi_6\Delta + \varphi_7 \tag{16}$$

$$\begin{aligned}
\Delta f_{ij} &= f(\alpha_{ii} - \Delta, \alpha_{ij} + \Delta) - f(\alpha_{ii}, \alpha_{ij})\\
&= \frac{\alpha_{ii} - \Delta}{A - x_i\Delta}\frac{\alpha_{ii} - \Delta}{B - \Delta} - \frac{\alpha_{ii}}{A}\frac{\alpha_{ii}}{B} + \frac{\alpha_{jj}}{A - x_i\Delta}\frac{\alpha_{ij} + \Delta}{C + \beta_{ij}\Delta} - \frac{\alpha_{jj}}{A}\frac{\alpha_{ij}}{C} + (C_w - C_p)\Delta\\
&= \frac{AB(\Delta^2 - 2\alpha_{ii}\Delta + \alpha_{ii}^2) - \alpha_{ii}^2(AB - A\Delta - Bx_i\Delta + x_i\Delta^2)}{AB(A - x_i\Delta)(B - \Delta)} + (C_w - C_p)\Delta\\
&\quad + \frac{AC(\alpha_{jj}\alpha_{ij} + \alpha_{jj}\Delta) - \alpha_{jj}\alpha_{ij}(AC + A\beta_{ij}\Delta - Cx_i\Delta - x_i\beta_{ij}\Delta^2)}{AC(A - x_i\Delta)(C + \beta_{ij}\Delta)}\\
&= \left(\begin{array}{l}\dfrac{(AB - \alpha_{ii}^2x_i)\Delta + (\alpha_{ii}^2A + \alpha_{ii}^2Bx_i - 2\alpha_{ii}AB)}{AB(A - x_i\Delta)(B - \Delta)}\\ + \dfrac{\alpha_{jj}\alpha_{ij}x_i\beta_{ij}\Delta + (AC\alpha_{jj} + \alpha_{jj}\alpha_{ij}Cx_i - \alpha_{jj}\alpha_{ij}A\beta_{ij})}{AC(A - x_i\Delta)(C + \beta_{ij}\Delta)} + (C_w - C_p)\end{array}\right)\Delta
\end{aligned} \tag{8}$$

According to formula (14), the monotonicity of Δf_{ij}^{at} relates to the positive and negative of g(Δ). The function $g_1(\Delta)$ is a sixth-order equation. The equation can be solved to calculate the roots by genetic algorithm, artificial bee colony algorithm, particle swarm optimization algorithm and other artificial intelligence. The root-finding method in MATLAB, MathWorks and other software can be used to calculate the root Δ_i^{ZS} of $g_1(\Delta) = 0$. Considering the number of real roots and the image property of higher-order polynomial function, the pool calculates the maximum value of Δf_{ij}^{at} with the monotonicity of $g_1(\Delta)$. The specific steps of each mining pool i are as follows.

Step 1: If there is no real root Δ_i^{ZS} of $g_1(\Delta)$ in the range of $0 \le \Delta \le \alpha_{ii}$, it means that $0 \le \Delta \le \alpha_{ii}$ does not intersect with horizontal ordinate. Mining pool i calculates $g_1(0)$ and $g_1(\alpha_{ii})$. Otherwise, skip to step 3. If $g_1(0) > 0$ and $g_1(\alpha_{ii}) > 0$, $\Delta f_{ij}^{at\prime} > 0$ which represents that formula (12) is monotonically increasing. Then $\Delta_{ij}^{bat} = \alpha_{ii}$ and return. Otherwise, skip to step 2.

Step 2: If $g_1(0) < 0$ and $g_1(\alpha_{ii}) < 0$, $\Delta f_{ij}^{at\prime} < 0$ which represents that formula (12) is monotonic decreasing. Then $\Delta_{ij}^{bat} = 0$ and return. Otherwise, skip to step 3.

Step 3: Mining pool i substitutes several real roots of $g_1(\Delta)$ in the range of $0 \le \Delta \le \alpha_{ii}$ into formula (11) to calculate the values of $F_1(0)$, $F_1(\alpha_{ii})$ and $F_1(\Delta_i^{ZS})$. It selects the maximum value as optimal attack increment revenue Δf_{ij}^{bat}. Then it takes the corresponding Δ as Δ_{ij}^{bat}, which is current optimal attack computing power.

Each mining pool performs steps 1-3 above to obtain optimal attack computing power Δ_{ij}^{bat} and optimal attack increment revenue Δf_{ij}^{bat}. Therefore, it is convenient to adjust the allocation of computing power which improves the revenue.

Similarly, the calculation method of optimal recovery computing power is as follows. The pool substitutes variable Δ into formula (8) for calculation. Then it gets formula (17), as shown at the bottom of the next page. As the subsequent processing method is similar to the specific calculation method of optimal attack computing power Δ_{ij}^{bat}. Referring above theory, the pool obtains the optimal recovery computing power $\left|\Delta_{ij}^{btr}\right|$ and the corresponding optimal recovery increment revenue Δf_{ij}^{bre}.

The computing power allocation strategy of mining pool i is obtained by successfully calculating the optimal attack computing power and optimal recovery computing power. The specific implementation steps of each mining pool i are as follows.

Step 1: After a certain period of time ΔT, mining pool i uses the formula (17) and the derivative method to obtain the derivative $\Delta f_{ij}^{re\prime}$. Then it converts the derivative $\Delta f_{ij}^{re\prime}$ into $g_2(\Delta)$.

Step 2: Mining pool i solves the real root $\Delta_{i,re}^{ZS}$ by the root program. Calculating the $F_2(\Delta_{i,re}^{ZS})$ of each real root, $F_2(0)$ and $F_2(-\alpha_{ij})$, mining pool i selects the maximum value in $F_2(0)$, $F_2(-\alpha_{ij})$, and $F_2(\Delta_{i,re}^{ZS})$ of each real root. Then the maximum value is recovery increment revenue Δf_{ij}^{bre}. The optimal recovery computing power is an independent variable $\left|\Delta_{ij}^{btr}\right|$ which corresponds to the recovery increment revenue Δf_{ij}^{bre}. If there are the multiple same $\left|\Delta_{ij}^{btr}\right|$, then the pool i randomly selects one as current optimal recovery computing power.

Step 3: The maximum value matrix $\Delta\lambda$ of recovery computing power consists of optimal recovery increment revenue Δf_{ij}^{bre}. The optimal value matrix $\Delta\sigma$ of recovery computing

$$
\begin{aligned}
&\Delta f_{ij}^{at} \\
&= F_1(\Delta) = \frac{\left(ABC\beta_{ij}x_iC_w - ABC\beta_{ij}x_iC_p\right)\Delta^4}{ABCx_i\beta_{ij}\Delta^3 + \left(ABC^2x_i - A^2BC\beta_{ij} - AB^2Cx_i\beta_{ij}\right)\Delta^2 + \left(-A^2BC^2 - AB^2C^2x_i + A^2B^2C\beta_{ij}\right)\Delta + A^2B^2C^2} \\
&\quad + \frac{\left(ABC\beta_{ij} - \alpha_{ii}^2Cx_i\beta_{ij} - \alpha_{jj}\alpha_{ij}x_i\beta_{ij}B + ABC^2x_iC_w - A^2BC\beta_{ij}C_w - AB^2Cx_i\beta_{ij}C_w - ABC^2x_iC_p + A^2BC\beta_{ij}C_p + AB^2Cx_i\beta_{ij}C_p\right)\Delta^3}{ABCx_i\beta_{ij}\Delta^3 + \left(ABC^2x_i - A^2BC\beta_{ij} - AB^2Cx_i\beta_{ij}\right)\Delta^2 + \left(-A^2BC^2 - AB^2C^2x_i + A^2B^2C\beta_{ij}\right)\Delta + A^2B^2C^2} \\
&\quad + \frac{\left(ABC^2 - \alpha_{ii}^2C^2x_i + \alpha_{ii}^2AC\beta_{ij} + \alpha_{ii}^2BC\beta_{ij}x_i - 2\times\alpha_{ii}ABC\beta_{ij} + \alpha_{jj}\alpha_{ij}x_i\beta_{ij}B^2 - AC\alpha_{ij}B - \alpha_{ij}\alpha_{ij}Cx_iB + \alpha_{jj}\alpha_{ij}A\beta_{ij}B\right)\Delta^2}{ABCx_i\beta_{ij}\Delta^3 + \left(ABC^2x_i - A^2BC\beta_{ij} - AB^2Cx_i\beta_{ij}\right)\Delta^2 + \left(-A^2BC^2 - AB^2C^2x_i + A^2B^2C\beta_{ij}\right)\Delta + A^2B^2C^2} \\
&\quad + \frac{\left(A^2B^2C\beta_{ij}C_w - A^2BC^2C_w - AB^2C^2x_iC_w - A^2B^2C\beta_{ij}C_P + A^2BC^2C_P + AB^2C^2x_iC_p\right)\Delta^2}{ABCx_i\beta_{ij}\Delta^3 + \left(ABC^2x_i - A^2BC\beta_{ij} - AB^2Cx_i\beta_{ij}\right)\Delta^2 + \left(-A^2BC^2 - AB^2C^2x_i + A^2B^2C\beta_{ij}\right)\Delta + A^2B^2C^2} \\
&\quad + \frac{\left(C^2\alpha_{ii}^2A + C^2\alpha_{ii}^2Bx_i - 2\times\alpha_{ii}ABC^2 + AC\alpha_jB^2 + \alpha_{jj}\alpha_{ij}Cx_iB^2 - \alpha_{jj}\alpha_{ij}A\beta_{ij}B^2 + A^2B^2C^2C_w - A^2B^2C^2C_p\right)\Delta}{ABCx_i\beta_{ij}\Delta^3 + \left(ABC^2x_i - A^2BC\beta_{ij} - AB^2Cx_i\beta_{ij}\right)\Delta^2 + \left(-A^2BC^2 - AB^2C^2x_i + A^2B^2C\beta_{ij}\right)\Delta + A^2B^2C^2}, \\
&\quad 0 \le \Delta \le \alpha_{ii}
\end{aligned}
\tag{11}
$$

$$
\begin{aligned}
\Delta f_{ij}^{at\prime} = &\frac{\left(\begin{array}{l}(\phi_1\phi_5)\Delta^6 + (2\times\phi_1\phi_6)\Delta^5 + (3\times\phi_1\phi_7 + \phi_2\phi_6 - \phi_3\phi_5)\Delta^4 + \\ (4\times\phi_1\phi_8 + 2\times\phi_2\phi_7 - 2\times\phi_4\phi_5)\Delta^3 + (3\times\phi_2\phi_8 + \phi_3\phi_7 - \phi_4\phi_6)\Delta^2\end{array}\right)}{(\phi_5\Delta^3 + \phi_6\Delta^2 + \phi_7\Delta + \phi_8)^2} \\
&+ \frac{(2\times\phi_3\phi_8)\Delta + \phi_4\phi_8}{(\phi_5\Delta^3 + \phi_6\Delta^2 + \phi_7\Delta + \phi_8)^2}
\end{aligned}
\tag{14}
$$

power consists of the corresponding optimal value $\left|\Delta_{ij}^{btr}\right|$.

$$\begin{aligned}\Delta\lambda &= [\ \Delta f_{i1}^{bre} \quad \Delta f_{i2}^{bre} \quad \ldots \quad \Delta f_{ij}^{bre} \quad \ldots \quad \Delta f_{iN}^{bre}\ ],\\ \Delta\sigma &= [\ \left|\Delta_{i1}^{btr}\right| \quad \left|\Delta_{i2}^{btr}\right| \quad \ldots \quad \left|\Delta_{ij}^{btr}\right| \quad \ldots \quad \left|\Delta_{iN}^{btr}\right|\ ]\end{aligned} \tag{18}$$

where, $\Delta f_{ij}^{bre} = 0$ and $\Delta_{ij}^{btr} = 0, ifi = j$. It represents that mining pool i does not carry out the block withholding attack power on its own mining pool and recover the computing power from its own mining pool.

Step 4: Mining pool i analyzes the maximum value matrix $\Delta\lambda$ of recovery computing power. If $\Delta f_{ij}^{bre} > 0$, it records Δf_{ij}^{bre}, corresponding $\left|\Delta_{ij}^{btr}\right|$ and mining pool number. Then it obtains the set of all $\Delta f_{ij}^{bre} > 0$. If there are multiple same maximum values, the mining pool i randomly selects a maximum value to obtain its corresponding mining pool and optimal recovery computing power. Otherwise, the mining pool i selects the mining pool and optimal recovery computing power corresponding to the maximum value. Then it actively recovers the optimal recovery computing power of selected mining pool for honest mining.

Step 5: Mining pool i uses the derivative method to obtain derivative $\Delta f_{ij}^{at\prime}$. Then it converts the derivative $\Delta f_{ij}^{at\prime}$ into $g_1(\Delta)$.

Step 6: Mining pool i solves real roots $\Delta_{i,btr}^{ZS}$ by root program to obtain the $F_1(\Delta_{i,btr}^{ZS})$ of each real root, $F_1(0)$ and $F_1(\alpha_{ii})$, mining pool i selects the maximum value in $F_1(0)$, $F_1(\alpha_{ii})$ and $F_1(\Delta_{i,btr}^{ZS})$ of each real root. The maximum value is optimal attack increment revenue Δf_{ij}^{bat}. The corresponding independent variable Δ_{ij}^{bat} is an optimal attack computing power. If there are multiple same Δ_{ij}^{bat}, then the pool i randomly selects one as current optimal attack computing power.

Step 7: The maximum attack computing power matrix $\Delta\eta$ consists of calculated Δf_{ij}^{bat}. The optimal attack computing power matrix $\Delta\rho$ consists of corresponding optimal attack computing power Δ_{ij}^{bat}.

$$\begin{aligned}\Delta\eta &= [\ \Delta f_{i1}^{bat} \quad \Delta f_{i2}^{bat} \quad \ldots \quad \Delta f_{ij}^{bat} \quad \ldots \quad \Delta f_{iN}^{bat}\ ],\\ \Delta\rho &= [\ \Delta_{i1}^{bat} \quad \Delta_{i2}^{bat} \quad \ldots \quad \Delta_{ij}^{bat} \quad \ldots \quad \Delta_{iN}^{bat}\ ]\end{aligned} \tag{19}$$

where, $\Delta f_{ij}^{bat} = 0$ and $\Delta_{ij}^{bat} = 0, ifi = j$. It represents that mining pool i does not assign its computing power to carry out block withholding attack power on its own mining pool.

Step 8: Mining pool i analyzes the maximum attack computing power matrix $\Delta\eta$. If $\Delta f_{ij}^{bat} > 0$, it records $\Delta f_{ij}^{bat} > 0$, corresponding Δ_{ij}^{bat} and mining pool number. Then it obtains the set of $\Delta f_{ij}^{bat} > 0$. If there are multiple same maximum values, the mining pool i randomly selects a maximum value to obtain its corresponding mining pool and optimal attack computing power. Otherwise, the mining pool i selects the mining pool and optimal attack computing power corresponding to the maximum value. Then it assigns the optimal attack computing power to carry out block withholding attack on the selected mining pool.

Step 9: If $x_i\alpha_{ii} - \upsilon$ is higher than Δ_{ij}^{bat}, mining pool i assigns computing power Δ_{ij}^{bat} to carry out block withholding attack on the assignment object. Otherwise, it assigns computing power $x_i\alpha_{ii} - \upsilon$. Skip back to step 1.

Each mining pool repeats steps 1-9 to adjust the allocation of its computing power from time to time. Finally, it can obtain the computing power allocation strategy to improve its revenue.

The pseudo code of MPPA_O is as follows:

3) TIME COMPLEXITY ANALYSIS

According to the characteristics of mining pool allocation, we analyze the time complexity of MPPA_F and MPPA_O.

MPPA_F in each mining pool mainly includes three parts: revenue increment matrix calculation, computing power recovery and assignment, and revenue calculation of the mining pool. The first part is to calculate the computing power revenue increment of its own mining pool and other mining pools by formula (8), that is, the time complexity is $\Theta(N)$, where N represents the mining pool quantity. The second part is the computing power recovery and assignment of each mining pool, that is, the time complexity is $\Theta(N)$. The third part is to calculate the current revenue of the mining pool by formula (4), that is, the time complexity is $\Theta(N)$. Therefore, the time complexity of MPPA_F is $\Theta(N)$.

MPPA_O in each mining pool mainly includes three parts: optimal computing power calculation, computing power recovery and assignment, and revenue calculation of the mining pool. The first part is to calculate the root of function (16) of its own mining pool and other mining pools by root program. According to its root, the pool calculates the computing power revenue increment, that is, the time complexity is $\Theta(\chi N)$, where χ represents the time complexity of the selected root method. The second part and the third part are the same as MPPA_F. Therefore, the time complexity

$$\begin{aligned}\Delta f_{ij}^{bre} &= F_2(\Delta)\\ &= \left(\begin{array}{c}\dfrac{(AB-\alpha_{ii}^2x_i)\Delta + (\alpha_{ii}^2A + \alpha_{ii}^2Bx_i - 2\alpha_{ii}AB)}{AB(A-x_i\Delta)(B-\Delta)} + (C_w - C_p) +\\ \dfrac{\alpha_{jj}\alpha_{ij}x_i\beta_{ij}\Delta + (AC\alpha_{jj} + \alpha_{jj}\alpha_{ij}Cx_i - \alpha_{jj}\alpha_{ij}A\beta_{ij})}{AC(A-x_i\Delta)(C+\beta_{ij}\Delta)}\end{array}\right)\Delta, -\alpha_{ij} \le \Delta \le 0\end{aligned} \tag{17}$$

Algorithm 1 MPPA_F Solution

Input: the computing power coefficient of each mining pool $\alpha_{ij}, \forall i, \forall j$

Output: the solutions of power computing allocation for each mining pool

for $i = 1; i <= N; i++$ do
  for $j = 1; j <= N; j++$ do
    if $i = j$ then $\Delta\gamma_1[j-1] \leftarrow 0$
    else $\Delta\gamma_1[j-1] \leftarrow \Delta f_{ij}$
    end if
  end for
  for $j = 1; j < N; j++$ do
    if $|\Delta\gamma_1[j-1]| < |\Delta\gamma_1[j]|$ and $\Delta\gamma_1[j] < 0$
  the object of computing power recovery $\leftarrow j$
    end if
  end for
  if recoverable computing power $> \Delta$
    then
      recover computing power Δ for honest mining
    else
      recover all computing power for honest mining
    end if
    for $j = 1; j < N; j++$ do
      if $i = j$ then $\Delta\gamma_2[j-1] \leftarrow 0$
      else $\Delta\gamma_2[j-1] \leftarrow \Delta f'_{ij}$
      end if
    end for
      for $j = 1; j < N; j++$ do
    if $|\Delta\gamma_2[j-1]| < |\Delta\gamma_2[j]|$ and $\Delta\gamma_2[j] > 0$
    the object of computing power assignment $\leftarrow j$
      end if
    end for
    if $x_i\alpha_{ii} - \upsilon > \Delta$ then
      assign computing power Δ to carry out block withholding attack
    else
      assign computing power $x_i\alpha_{ii} - \upsilon$ to carry out block withholding attack
    end if
end for

Algorithm 2 MPPA_O Solution

Input: the computing power coefficient of each mining pool $\alpha_{ij}, \forall i, \forall j$

Output: the solutions of power computing allocation for each mining pool

for $i = 1; i <= N; i++$ do
  for $j = 1; j <= N; j++$ do
    if $i = j$ then $\Delta\lambda[j-1] \leftarrow 0$, $\Delta\sigma[j-1] \leftarrow 0$
    else $\Delta\lambda[j-1] \leftarrow \Delta f_{ij}^{bre}$, $\Delta\sigma[j-1] \leftarrow \left|\Delta_{ij}^{btr}\right|$
    end if
  end for
  for $j = 1; j < N; j++$ do
    if $\Delta\lambda[j-1] < \Delta\lambda[j]$ and $\Delta\lambda[j] > 0$
  the object of computing power recovery $\leftarrow j$
    end if
  end for
  recover the optimal recovery computing power
  for $j = 1; j < N; j++$ do
    if $i = j$ then $\Delta\eta[j-1] \leftarrow 0$, $\Delta\rho[j-1] \leftarrow 0$
    else $\Delta\eta[j-1] \leftarrow \Delta f_{ij}^{bat}$, $\Delta\rho[j-1] \leftarrow \Delta_{ij}^{bat}$
    end if
  end for
  for $j = 1; j < N; j++$ do
    if $\Delta\eta[j-1] < \Delta\eta[j]$ and $\Delta\eta[j] > 0$
  the object of computing power assignment $\leftarrow j$
    end if
  end for
  if $x_i\alpha_{ii} - \upsilon > \Delta_{ij}^{bat}$ then
    assign computing power Δ_{ij}^{bat} to carry out block withholding attack
  else
    assign computing power $x_i\alpha_{ii} - \upsilon$ to carry out block withholding attack
  end if
end for

of MPPA_O is $\Theta(\chi N)$. MPPA_O is more complex than MPPA_F.

IV. ALGORITHM SIMULATION

A. SIMULATION PARAMETERS AND PERFORMANCE PARAMETERS

In order to verify the performance of MPPA, we use MATLAB software to perform algorithm simulation and comparison. In the algorithm simulation, we first numerically simulate the computing power allocation of each mining pool, and calculate its honest mining revenue and block withholding attack revenue by formula (4). In the MPPD_F, mining pool calculates the revenue increment by formula (8), and performs computing power recovery and computing power assignment through the algorithm steps. In the MFFD_A, mining pool uses the derivative method to calculate optimal attack computing power and optimal recovery computing power by formula (17), and performs computing power recovery and computing power assignment through the algorithm steps. Then they obtain the revenue of each mining pool through multiple iterations. The simulation parameters are in Table 1. Then we study the influence of cost of honest mining, cost of block withholding attack, mining pool quantity, computing power allocation among mining pools and attack ratio among mining pools. We use Win Stay Fail Shift(WSFS) [9], ALL Cooperate (ALLC) [10], All Defect(ALLD) [10], MPPA_F and MPPA_O to calculate the average revenue of mining pool under different mining pool quantity. Among them, ALLC and ALLD are fixed strategy for the computing power allocation of mining pools. All the

TABLE 1. Simulation parameter.

Parameter Name	Value
mining pool quantity N	10
initial computing power of each mining pool	1
iteration times	200
proportion of mutual attacks	100%
cost C_w of honest mining	0.05
cost C_p of block withholding attack	0.01
reserve computing power υ	0.01
computing power loss interval Δ in MPPA_F	0.01
network revenue value in each iteration	1000

mining pools in ALLC strategy are always fully honest mining. In WSFS strategy, the mining pool initially adopts the full honest mining strategy. If the revenue is lower than threshold, it will switch between full honest mining strategy and full block withholding attack strategy. Otherwise, the strategy will remain unchanged. ALLD strategy chooses full honest mining strategy for some mining pools, and the other mining pools always adopt a full block withholding attack.

We define the average revenue of the mining pool as

$$R_{av}^{t} = \frac{\sum_{i=1}^{N} S_t^i}{N} \tag{20}$$

where, S_{av}^{i} represents the revenue of mining pool i in the tth iteration, N represents the mining pool quantity, R_{av}^{t} represents the average revenue of mining pool after the tth iteration.

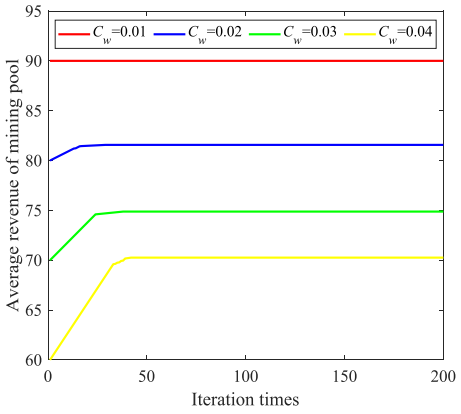

FIGURE 2. Influence of cost of honest mining on the average revenue in MPPA_F.

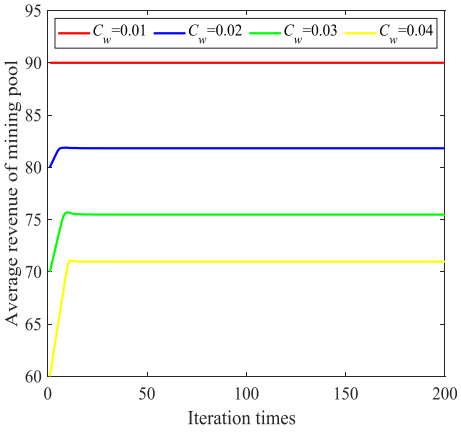

FIGURE 3. Influence of cost of honest mining on the average revenue in MPPA_O.

B. ANALYSIS OF SIMULATION RESULTS

1) INFLUENCE OF COST OF HONEST MINING

To analyze the influence of cost of honest mining on the average revenue in MPPA_F and MPPA_O, we select the cost 0.01, 0.02, 0.03 and 0.04 of honest mining, cost 0.01 of block withholding attack and other parameters in Table 1. As shown in Fig.2 and Fig.3, with the increase of cost of honest mining, the convergence rate and convergence value of average revenue of the mining pool in MPPA_F and MPPA_O gradually decrease, but the decline degrees of convergence value slow down. This is because: when the cost of honest mining and cost of block withholding attack are both 0.01, the computing power of block withholding attack among mining pools is relatively small. So the average revenue of mining pools of the two algorithms converge to the optimal values quickly (both algorithms are 90). With the increase of cost of honest mining, MPPA_F and MPPA_O need more time to allocate computing power for block withholding attacks. So their rates of convergence decrease. At the same time, the convergence value of the average revenue of the mining pool decreases due to the decrease of mining revenue. However, the two algorithms can adjust computing power. Then the computing power of block withholding attack among mining pools increases gradually and the computing power of honest mining decreases accordingly. It not only improves the block withholding attack revenue of mining pools, but also reduces the influence of cost of honest mining on the average revenue of mining pools, resulting in the convergence value of the average revenue of mining pools becoming 81.6, 74.9, 70.3 in MPPA_F and 81.8, 75.5, 71 in MPPA_O. Therefore, the decline of the convergence value of the average revenue of mining pools slows down.

2) INFLUENCE OF COST OF BLOCK WITHHOLDING ATTACK

To analyze the influence of cost of block withholding attack on the average revenue of mining pool in MPPA_F and MPPA_O, we select the cost 0.05 of honest mining, cost 0.02, 0.03, 0.04, 0.05 of block withholding attack and other parameters in Table 1. As shown in Fig.4 and Fig.5, with the increase of cost of block withholding attack, the convergence rate of average revenue of mining pools in MPPA_F and MPPA_O gradually increase, but their convergence values gradually decrease, and the degrees of decline slow down. This is because: when the cost of block withholding attack is 0.01 and the cost of honest mining is 0.05, the computing

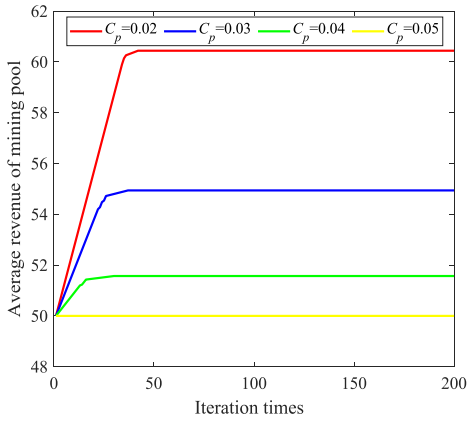

FIGURE 4. Influence of cost of block withholding attack on MPPA_F.

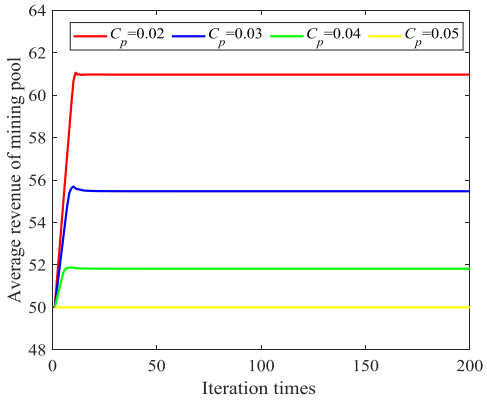

FIGURE 5. Influence of cost of block withholding attack on MPPA_O.

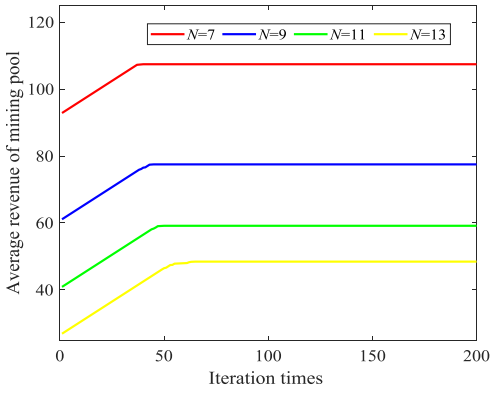

FIGURE 6. Influence of mining pool quantity in MPPA_F.

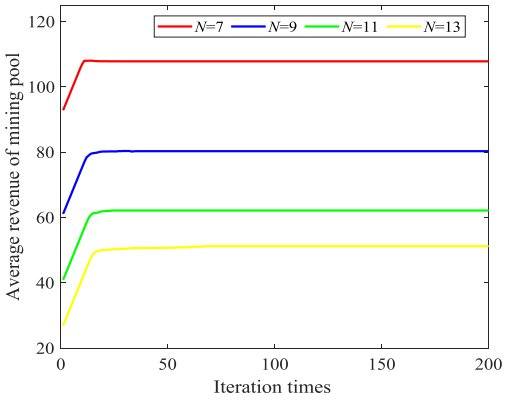

FIGURE 7. Influence of mining pool quantity in MPPA_O.

power of block withholding attack among mining pools in the two algorithms is relatively large. So it takes a long time to allocate the computing power of mining pools, and their average revenues converge to the optimal values (60.5 in MPPA_F and 61 in MPPA_O). With the increase of the cost of block withholding attack, MPPA_F and MPPA_O can complete the computing power allocation of block withholding attack in a shorter time, resulting in a faster rate of convergence. At the same time, the mining pool needs to consume more cost to carry out block withholding attacks, resulting in the decline of convergence value of the mining pool's average revenue. However, due to the algorithm's automatic adjustment of computing power allocation, the computing power of honest mining among mining pools gradually increases, and the computing power of block withholding attack begins to decline. It not only improves the mining revenue of mining pools, but also reduces the influence of cost of block withholding attacks on the average revenue of algorithms, resulting in the convergence value of average revenue of mining pool becoming 54.9, 51.5, 50 in MPPA_F and 55.5, 51.8, 50 in MPPA_O. Therefore, it slows down the decline of the convergence value of the average revenue of the mining pool. At the same time, before the average revenue of mining pools reaches the equilibrium state, some mining pools assign more computing power to carry out block withholding attack and obtain more revenue. Then some mining pools do not obtain reasonable revenue. They adjust the computing power by MPPA_O, which improves their revenue. Finally, the average revenue of the mining pools fluctuates a little before reaching the equilibrium state.

3) INFLUENCE OF MINING POOL QUANTITY

To analyze the influence of mining pool quantity on the average revenues of mining pool in MPPA_F and MPPA_O, we select the cost 0.05 of honest mining, cost 0.01 of block withholding attack, mining pool quantity 7, 9, 11, 13 and other parameters in Table 1. As shown in Fig.6 and Fig.7, with the increase of mining pool quantity in MPPA_F and MPPA_O, the initial average revenues of the mining pool gradually decrease. Then the convergence rate and convergence value of average revenues of the mining pool gradually decrease, but the decrease of convergence values slow down. This is because: according to formula (8), the initial computing power of each mining pool is the same and the mining pool does not use block withholding attack in the beginning. With the increase of mining pool quantity, the effective computing power in the whole network also rises correspondingly and the network revenue value of each iteration is same. So the initial average revenue and convergence value of the

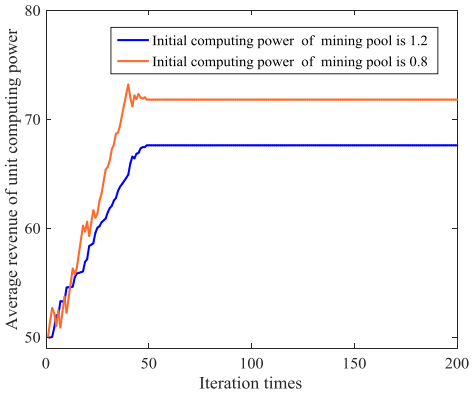

FIGURE 8. Comparison chart of different initial mining computing power in MPPA_F.

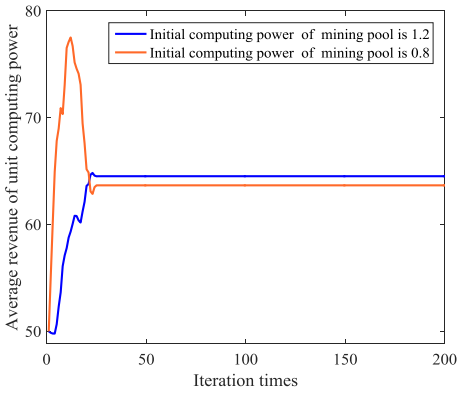

FIGURE 9. Comparison chart of different initial mining computing power in MPPA_O.

mining pool obtained through mining decreases gradually. Due to the limited revenue of computing power of honest mining in each mining pool, the mining pool automatically adjusts the computing power [31]. It lets more computing power carry out block withholding attacks, resulting in a decrease in its convergence rate. At the same time, the two algorithms automatically adjust the computing power allocation, and the block withholding attack revenues of mining pool increase, so the convergence value in MPPA_F changes from 107.4 to 77.5, 59.2, 48.5, and the convergence value in MPPA_O changes from 107.8 to 80 3, 62.2, 51.2. The decline of the convergence value of the average revenue of mining pools slows down.

4) INFLUENCE OF COMPUTING POWER ALLOCATION AMONG MINING POOLS

To analyze the influence of uneven allocation of computing power among mining pools on the average revenue of unit computing power of mining pool in MPPA_F and MPPA_O, we select the initial computing power 1.2 of 50% mining pools, the initial computing power 0.8 of other 50% mining pools, and other parameters in Table 1. The average revenues of the unit computing power of the mining pool are defined as the ratio of total revenue and total computing power of all mining pools. As shown in Fig.8 and fig.9, at the beginning of the algorithm, mining pools play games with each other and adjust their computing power to carry out honest mining and mutual block withholding attacks. Therefore, before the equilibrium state is reached, the average revenues of mining pools have obvious jitter, but the curves converge to equilibrium. As shown in fig.8, in MPPA_F, the convergence value of average revenue of unit computing power in the pools with small initial computing power is slightly greater than that in the pools with large initial computing power. This is because: MPPA_F uses a fixed loss interval of computing power to adjust computing power. The mining pool with small initial computing power can quickly adjust its computing power to occupy the advantage of revenue allocation, which improves the average revenue of unit computing power. As shown in Fig.9, in MPPA_O, the convergence value of unit computing power in the mining pool with large initial computing power is slightly greater than that in the mining pool with small initial computing power. This is because: MPPA_O uses the optimization method to calculate the optimal recovery computing power and optimal attack computing power. The mining pool with small initial computing power can adjust its own computing power and occupy advantage in the early iteration. But the mining pool with large initial computing power not only can quickly adjust its own computing power according to the current computing power situation, but also use the advantage of larger computing power to carry out block withholding attack. It occupies an advantage in revenue distribution to improve the average revenue of its unit computing power. In conclusion, MPPA_F is slightly beneficial to the mining pool's revenue acquisition with small initial computing power. MPPA_O is slightly beneficial to the revenue acquisition of the mining pool with large initial computing power.

5) INFLUENCE OF ATTACK RATIO AMONG MINING POOLS

To analyze the influence of computing power allocation among mining pools on the average revenues of MPPA_F and MPPA_O, we select the attack ratio 100%, 90%, 80%, 70% among mining pools, and other parameters in Table 1. As shown in Fig.10 and Fig.11, with the decrease of attack ratio among mining pools, the convergence value of block withholding attack revenue in MPPA_F and MPPA_O decreases gradually. But the convergence value of honest mining revenue increase gradually. Moreover, the convergence value of block withholding attack revenue decreases more than that of honest mining revenue. The dominant position of revenue changes from block withholding attack revenue to honest mining revenue. This is because: with the decline of attack ratio among mining pools, the computing power of honest mining among mining pools gradually rises, while computing power of block withholding attack gradually decreases. So the mining revenue also gradually rises, and the block withholding attack revenue gradually decreases. The factor also directly leads to the dominant position of honest mining revenue. At the same time, due to the decline of attack ratio, the mining pools reduce the number of attack objects. Therefore, in MPPA_F, the convergence value of honest mining

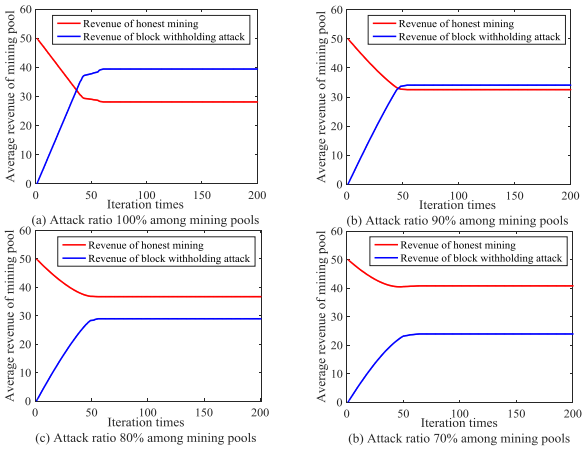

FIGURE 10. Revenue comparison chart in MPPA_F when attack ratio among mining pools changes.

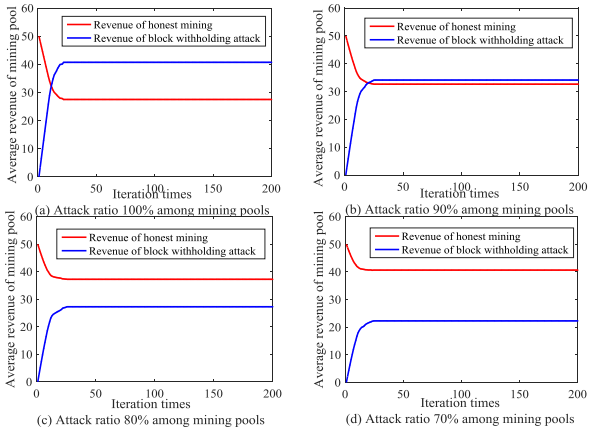

FIGURE 11. Revenue comparison chart in MPPA_O when attack ratio among mining pools changes.

revenue increases by 12%, 11%, 9%, and the convergence value of block withholding attack revenue decreases by 13%, 14%, 17%. In MPPA_O, the numbers are 15%, 13%, 7% and 16%, 20%, 18% respectively. In the two algorithms, the decrease range of convergence value of block withholding attack revenue is larger than an increased range of honest mining revenue.

6) INFLUENCE OF LOSS INTERVAL OF COMPUTING POWER

To analyze the influence of loss interval of computing power on the average revenue of the mining pool in MPPA_F, we select the cost 0.05 of honest mining, cost 0.01 of block withholding attack, loss interval 0.01, 0.02, 0.03, 0.04 of computing power and other parameters in Table 1. Because MPPA_O can calculate the optimal loss interval of computing power, it is unnecessary to consider the influence of loss interval of computing power. As shown in Fig.12, with the increase of the loss interval of computing power, the convergence value of the average revenue of the mining pool in MPPA_F gradually decreases and the decrease range becomes larger, but its convergence rate gradually increases.

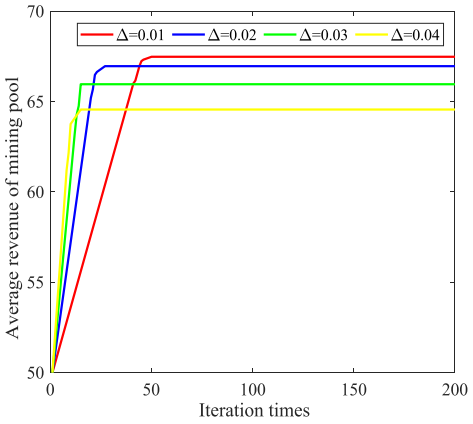

FIGURE 12. Influence of loss interval of computing power in MPPA_F.

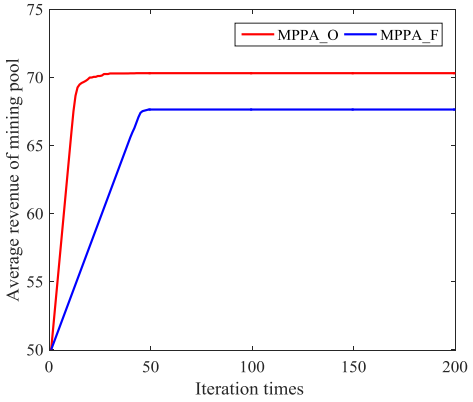

FIGURE 13. Revenue comparison of MPPA_F and MPPA_O.

This is because: MPPA_F realizes computing power allocation of the mining pool by setting a loss interval of computing power. If the loss interval of computing power sets too large, the convergence can be achieved at a faster rate, but it only converges to different local optimal solutions. There is no good adjustment in computing power. Then the convergence value of the average revenue of the mining pool decreases. The larger the loss interval of computing power, the farther the optimal local solution from the optimal global solution is, and the larger the decrease range are. Then a small loss interval of computing power can effectively improve the convergence value of the average revenue of the mining pool. However, if the interval of each computing power adjustment is less, the algorithm will spend more time to find the optimal value and reduce the convergence rate.

7) ALGORITHM PERFORMANCE COMPARISON

We select parameters in Table 1 to calculate the average revenues of the mining pool in MPPA_F and MPPA_O. As shown in Fig.13 and Fig.14, although the convergence rate and convergence value of the average revenue of the mining pool in MPPA_O are better than that in MPPA_F, the computing time in MPPA_O is significantly longer than that in MPPA_F. This is because: MPPA_O calculates the optimal recovery computing power and attack computing power

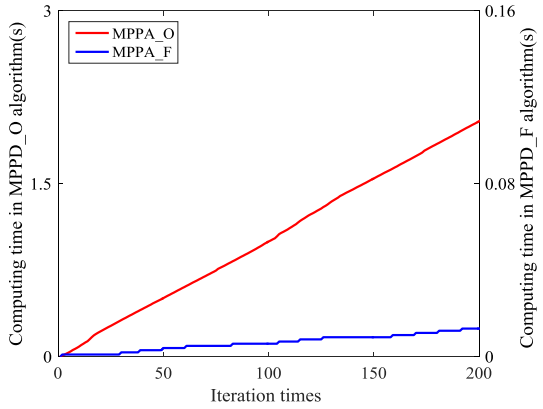

FIGURE 14. Computing time comparison of MPPA_F and MPPA_O.

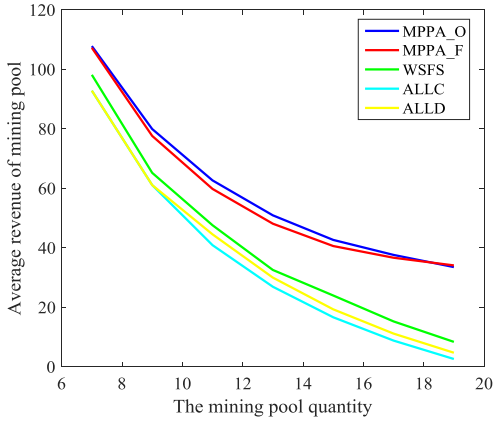

FIGURE 15. Comparison of various strategies.

to adjust computing power with the optimization method. So mining pools in MPPA_O converge at the faster convergence rate, and they can obtain the optimal convergence value of the average revenue of the mining pool. MPPA_F adjusts the interval loss of computing power by fixed value to distribute computing power of the mining pool. The allocation strategy converges to an local optimal solution, so the convergence rate and the convergence value of the average revenue of the mining pool in MPPA_F are lower than that in MPPA_O. Because of the optimization method, MPPA_O spends more time to slove the maximum value. The computing time in MPPA_O is much longer than that in MPPA_F. At the same time, MPPA_F is easy to fall into the optimal local solution during the iterative process, so the computing time of MPPA_F presents a ladder shape.

We select the mining pool quantity 7, 9, 11, 13, 15, 17, 19 and other parameters in Table 1 to calculate the convergence values of average revenue of mining pool in WSFS, ALLC, ALLD, MPPA_F and MPPA_O. As shown in Fig.15, with the increase of mining pool quantity, the average revenues of the mining pool in WSFS, ALLC, ALLD, MPPA_F and MPPA_O gradually decrease. The average revenues of the mining pool in ALLC, ALLD and WSFS are relatively close, while that in MPPA_O and MPPA_O are relatively close and obviously larger than the first three algorithms. This is because: with the increase of mining pool quantity and the same network revenue value for each iteration, WSFS, ALLC, ALLD, MPPA_F and MPPA_O reduce the honest mining revenue of each mining pool, resulting in the gradual decline of average revenue of mining pool. All the mining pools in ALLC insist on honest mining all the time and do not maintain revenue through block withholding attack, so the average revenue of mining pool is the lowest. WSFS adjusts computing power allocation of each mining pool through threshold value. ALLD only selects some mining pools to carry out block withholding attack. Therefore, the average revenue of the mining pool in ALLD is poor, and the average revenue of the mining pool in WSFS is larger than that in ALLD. However, due to single adjustment of computing power in WSFS, MPPA_O and MPPA_F dynamically adjust the computing power of mining pool to improve the average revenue of the mining pool by calculating the revenue increment. So the average revenues of the mining pool in MPPA_O and MPPA_F are larger than that in WSFS, ALLC and ALLD.

V. CONCLUSION

The paper proposes a mining pool computing power allocation algorithm (MPPA) with block withholding attacks among multiple mining pools. First, the algorithm considers that there are multiple mining pools composed of miners and pool manager in the network. The mining pools carry out dynamic block withholding attacks on other mining pools. Considering honest mining and block withholding attack, we establish the mining pool optimization model, which includes the current effective total computing power, the honest mining revenue of each mining pool, the revenue of block withholding attack and the average revenue of each mining pool, etc. Secondly, according to the optimization model, MPPA calculates the revenue gain generated by block withholding attacks on other mining pools. For adjusting fixed computing power in each iteration, we have the mining pool computing power allocation algorithm with a fixed change of computing power (MPPA_F). For adjusting the optimal recovery and attack computing power, we have the mining pool computing power allocation algorithm with an optimal change of computing power (MPPA_O). Finally, we analyze the influence of cost of honest mining, cost of block withholding attack, mining pool quantity, computing power allocation among mining pools, attack ratio among mining pools and loss interval of computing power on the average revenue of the mining pool. Then we compare the average revenues of mining pool in WSFS, ALLC, ALLD, MPPA_F and MPPA_O.

The simulation results show that MPPA can find the optimal computing power allocation strategy for each mining pool to reasonably allocate computing power. Therefore, MPPA not only improves the average revenue of the mining pool, but also reduces the influence of block withholding attack of other mining pools on mining pool revenue of its own. MPPA outperforms the state-of-arts such as WSFS,

ALLC and ALLD. Since MPPA does not consider the selection of miners in the mining pool yet, the next stage's goal is to consider the miner individual and propose the mining revenue maximization algorithm of miners.

REFERENCES

[1] Y. Yong, X. Ni, S. Zeng, and F. Wang, "Blockchain consensus algorithms: The state of the art and future trends," *Acta Automatica Sinica*, vol. 44, no. 11, pp. 2011–2022, Nov. 2018.

[2] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, Jan. 2019.

[3] A. Singh, R. M. Parizi, M. Han, A. Dehghantanha, H. Karimipour, and K. K. R. Choo, "Public blockchains scalability: An examination of sharding and segregated witness," in *Proc. Blockchain Cybersecurit, Trust Privacy.*, vol. 79, Mar. 2020, pp. 203–232.

[4] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 577–590, Aug. 2018.

[5] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Jun. 2019.

[6] Q. Ren, K. L. Man, M. Li, and B. Gao, "Using blockchain to enhance and optimize IoT-based intelligent traffic system," in *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, Jan. 2019, pp. 1–4.

[7] Y.-T. Yang, L.-D. Chou, C.-W. Tseng, F.-H. Tseng, and C.-C. Liu, "Blockchain-based traffic event validation and trust verification for VANETs," *IEEE Access*, vol. 7, pp. 30868–30877, 2019.

[8] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 4196–4205, Dec. 2019.

[9] S. Zhu, Z. Cai, H. Hu, Y. Li, and W. Li, "ZkCrowd: A hybrid blockchain-based crowdsourcing platform," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4196–4205, Jun. 2020.

[10] Y. Li, X. Zhang, H. Wu, J. Liu, D. Tang, and X. Tao, "DQN for multi-layer game based mining competition in VEC network," in *Proc. Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2019, pp. 1–6.

[11] I. Eyal, "The miner's dilemma," in *Proc. IEEE SSP*, San Jose, CA, USA, May 2015, pp. 85–103.

[12] J. Han, J. Zou, H. Jiang, and Q. Xu, "Research on mining attacks in bitcoin," *J. Cryptolog. Res.*, vol. 5, no. 5, pp. 470–483, Dec. 2018.

[13] T. Zhu, J. Li, Z. Cai, Y. Li, and H. Gao, "Computation scheduling for wireless powered mobile edge computing networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Jul. 2020, pp. 596–605.

[14] M. Rosenfeld, "Analysis of bitcoin pooled mining reward systems," 2011, *arXiv:1112.4980*. [Online]. Available: http://arxiv.org/abs/1112.4980

[15] 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 *Proc. IEEE 28th Comput. Secur. Found. Symp.*, Jul. 2015, pp. 397–411.

[16] C. B. Tang, Z. Yang, Z. L. Zheng, Z. Y. Chen, and X. Li, "Game dilemma analysis and optimization of pow consensus algorithm," *Acta Automatica Sinica.*, vol. 43, no. 9, pp. 1520–1531, Sep. 2017.

[17] C. Tang, C. Li, X. Yu, Z. Zheng, and Z. Chen, "Cooperative mining in blockchain networks with zero-determinant strategies," *IEEE Trans. Cybern.*, early access, May 22, 2019, doi: 10.1109/TCYB.2019.2915253.

[18] W. Di, X. D. Liu, X. B. Yan, R. Peng, and G. Li, "Equilibrium analysis of bitcoin block withholding attack: A generalized model," *Rel. Eng. Syst. Saf.*, vol. 185, no. 1, pp. 318–328, May 2019.

[19] Y. Wang, C. Tang, F. Lin, Z. Zheng, and Z. Chen, "Pool strategies selection in PoW-based blockchain networks: Game-theoretic analysis," *IEEE Access*, vol. 7, pp. 8427–8436, 2019.

[20] K. Chatterjee, A. K. Goharshady, R. Ibsen-Jensen, and Y. Velner, "Ergodic mean-payoff games for the analysis of attacks in crypto-currencies," in *Proc. ICCT*, Beijing, China, Aug. 2018, pp. 1–22.

[21] C. A. Houlihan and N. Shah, "The pure price of anarchy of pool block withholding attacks in bitcoin mining," in *Proc. AAAI*, Honolulu, HI, USA, Jul. 2019, pp. 1–10.

[22] S. Bag, S. Ruj, and K. Sakurai, "Bitcoin block withholding attack: Analysis and mitigation," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1967–1978, Aug. 2017.

[23] X. Liu, W. Wang, D. Niyato, N. Zhao, and P. Wang, "Evolutionary game for mining pool selection in blockchain networks," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 760–763, Oct. 2018.

[24] S. Kim and S.-G. Hahn, "Mining pool manipulation in blockchain network over evolutionary block withholding attack," *IEEE Access*, vol. 7, pp. 144230–144244, 2019.

[25] T. Wang, S. Y. Yu, and B. M. Xu, "Research on proof of work mining dilemma based on policy gradient algorithm," *J. Comput. Appl.*, vol. 39, no. 5, pp. 1336–1342, May 2019.

[26] D. K. Tosh, S. Shetty, X. Liang, C. A. Kamhoua, K. A. Kwiat, and L. Njilla, "Security implications of blockchain cloud with analysis of block withholding attack," in *Proc. 17th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGRID)*, May 2017, pp. 458–467.

[27] A. Toroghi Haghighat and M. Shajari, "Block withholding game among bitcoin mining pools," *Future Gener. Comput. Syst.*, vol. 97, pp. 482–491, Aug. 2019.

[28] S. Y. Chang, Y. Park, S. Wuthier, and C. W. Chen, "Uncle-block attack: Blockchain mining threat beyond block withholding for rational and uncooperative miners," in *Proc. ACNS*, Bogota, Colombia, Jun. 2019, pp. 241–258.

[29] M. Alharby and A. van Moorsel, "The impact of profit uncertainty on miner decisions in blockchain systems," *Electron. Notes Theor. Comput. Sci.*, vol. 340, pp. 151–167, Oct. 2018.

[30] *Department of Mathematics, Tong Ji University, Advanced Mathematics*, 7th ed, Higher Education, Press, Beijing, China, 2014, pp. 66–69.

[31] A. P. Joshi, M. Han, and Y. Wang, "A survey on security and privacy issues of blockchain technology," *Math. Found. Comput.*, vol. 1, no. 2, pp. 121–147, 2018.

YOURONG CHEN received the B.S. degree in communication engineering from Zhejiang University of Technology, Hangzhou, China, in 2004, the M.S. degree in communication engineering communication and information system from Zhejiang University of Technology, Hangzhou, China, in 2007, and the Ph.D. degree in control theory and control engineering from Zhejiang University of Technology, Hangzhou, China, in 2012. Since 2007, he is currently a full professor in Zhejiang Shuren University. His research interests include network security and internet of things engineering.

HAO CHEN received the B.S. degree in communication engineering from Zhejiang Shuren University, Hangzhou, China, in 2019. He is currently pursuing the master's degree with Changzhou University. His research interests include network security and the Internet of Things engineering.

MENG HAN (Member, IEEE) is currently an Assistant Professor with the College of Computing and Software Engineering, Kennesaw State University–Marietta. His research interests include data-driven intelligence, data-driven AI security and privacy, and blockchain technologies. He is also an ACM member and an IEEE COMSOC member.

BANTENG LIU received the B.S. and M.S. degrees in communication engineering from the Zhejiang University of Technology, Hangzhou, China, in 2009, and the Ph.D. degree in control theory and control engineering from Zhejiang University, Hangzhou, in 2017. He is currently an Associate Professor with Zhejiang Shuren University. His research interests include wireless mobile communication and nondestructive flaw detection.

QIUXIA CHEN received the B.Eng. degree in automation and the Ph.D. degree in control theory and control engineering from the Zhejiang University of Technology, Hangzhou, China, in 2005 and 2011, respectively. She is currently a Senior Engineer with Zhejiang Shuren University. Her research interests include network security, model predictive control, and deep learning.

TIAOJUAN REN received the B.S. degree in dynamic testing and automation from Zhejiang University, Hangzhou, China, in 1987, and the M.S. degree in information management from Korea Giant Buddha University, South Korea, in 2005. She is currently a Professor with Zhejiang Shuren University. Her research interests include network security and wireless and mobile communications.