IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 1, 2019, accepted November 22, 2019, date of publication November 27, 2019,
date of current version December 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2956168

Mining Strategies for Completing the Longest Blockchain

SUI CHENG AND SIAN-JHENG LIN, (Member, IEEE)
CAS Key Laboratory of Electro-Magnetic Space Information, School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China

Corresponding author: Sian-Jheng Lin (sjlin@ustc.edu.cn)

This work was supported in part by the Hundred Talents Program of Chinese Academy of Sciences, and in part by the Natural Science Foundation of Anhui Province under Grant BJ2100330001.

ABSTRACT Bitcoin is an innovative decentralized cryptocurrency that records transactions in a public log, termed the blockchain. Due to delays in the Bitcoin network, the Bitcoin blockchain has the potential for forming forks. In this paper, we consider a mining strategy in which the player who created the fork can attract other miners by posting rewards and the third party (other miners) has the ability to choose the fork based on their interests. We provide a model to formalize the considered case and analyze its feasibility. Based on certain game theoretical models, the best policies for two players, termed the creator of the fork and the third party, are presented. Finally, some cases of the considered model are simulated, and the average revenue of each player is compared with the theoretical revenue to verify the correctness of our strategy. When the delay varies from 0 to 400 milliseconds, adopting our strategy can acquire 23% more revenue than adopting the default strategy. The result shows that the default strategy of a miner is not always a good choice.

INDEX TERMS Bitcoin, blockchain, fork, game theory, incentive, security.

I. INTRODUCTION

Digital cryptocurrencies have generated considerable interest in recent years, and Bitcoin comprises 90% of the market capitalization [2]. Bitcoin is a decentralized digital cryptocurrency proposed by Nakamoto [21], and it became operational in 2009. Bitcoin became a successful digital cryptocurrency through its innovative solution to double spending and its design of creating profits for participants [3].

Bitcoin records the transactions in a public log, called the blockchain. The valid transactions in the blockchain arrive at a consensus in a decentralized fashion [4]. All participants in a Bitcoin network follow a consensus protocol, known as the Nakamoto consensus [21]. With the Nakamoto consensus, it is difficult to tamper with a transaction once it is sufficiently deep in the blockchain, assuming that the attackers do not possess a large fraction of the computational power within the network [5], [7].

In the Bitcoin network, the participants, called miners, contribute their computing power to solve a computationally expensive puzzle, which is known as a proof-of-work (PoW), a hash calculation that satisfies a specific requirement. A block is able to link to the blockchain when the hashing value is less than a predetermined target threshold. The blockchain is a distributed public ledger serializing all transactions by time. A miner will broadcast its new block after discovering the hash value. If the block passes the validations, the receivers/miners append the new block to their own blockchains. Then, the miner who created this block will be rewarded, including the newly minted Bitcoins (coinbase) and transaction fees determined in the transactions [18].

To provide a smoother incentive to miners, many miners are gathered together to form a mining pool, combining their computing power [10], [15], [17]. In particular, each miner in a pool will submit a valid PoW to the administrator to demonstrate their workload [23]. When a miner identifies a new block, the miner can either submit the block to the administrator or conceal the block [24], [25]. Regardless of the strategy used by the miner, only the pool can obtain a reward via the new block.

The Nakamoto consensus does not guarantee that the blockchains of all miners are the same at all points in time. Thus, some conflicting chains may form, known as forks. When a fork occurs, these blocks are usually created by different creators, and these creators are in competition; thus, only the creator in the longest chain can win the reward.

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

In the Nakamoto consensus, miners only admit the blocks in the longest chain, and the transactions in other forks are invalid. In addition, when the longest chains are not unique, miners usually follow the highest block they received first. Figure 1 shows an example of a blockchain with the Nakamoto consensus. In this example, the longest chain is from the genesis block (A0) to the black leaf block (A8), and other blocks in shorter forks are colored white. In this example, the miners follow block (A8).

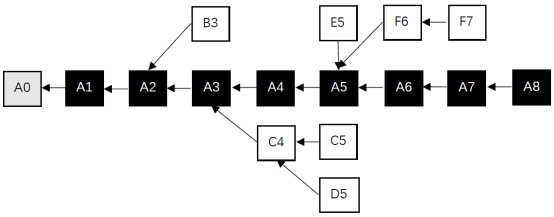

FIGURE 1. An example of a Bitcoin blockchain.

When a fork occurs, the creators may attract other miners to follow their block. In chapter 5 of [22], Narayanan et al. pointed out that the creator can ''leave big tips in blocks on the forking chain—big enough to cause miners to leave the longest chain and work on the forking chain in hopes that it will become the longest chain and they can collect the tips''. However, the article [22] does not have a thorough discussion of this topic. In this paper, we introduce a model for the topic and verify its validity via game theory. Notably, the paper [18] presents a similar model, termed whale attack. However, their motivation [18] is to attack the Bitcoin blockchain via a large transaction fee. In contrast, our motivation is not to attack Bitcoin. The proposed model allows the creators to attract other miners via announcing a bonus.

This paper consists of three parts. In the first part, we present a model in which the fork creators can publish a bonus via a smart contract to attract other miners, and these miners have the ability to change their subsequent fork at any time. In the second part, we analyze the mining strategies under the above model via a series of game theoretical models. In the final part, some simulations are given to verify the analysis results. The contributions of this paper are as follows.

1) We present a practical model in which the longest chain forks and carry out a feasibility analysis on this model.
2) With an analysis of the model, the best strategy for miners/pools is given to optimize their profits.
3) A simulation is performed to verify the analysis results.

The rest of this paper is organized as follows. In Section II, we briefly introduce a number of mining strategies. In Section III, the proposed model is presented. In Section IV, we analyze the model by game theoretical models and give the best strategy. In Section V, some simulations are given to verify the analysis results. In Section VI, we discuss the effect of our mining strategy and present conclusions.

II. RELATED WORKS

There are many works concerning mining strategies. Eyal and Sirer [12] presented a mining strategy, called selfish mining, such that attackers can acquire a higher revenue than their fair share. They showed that, unless certain assumptions are made, selfish mining would be feasible in any size of mining pools. They claimed that attackers require only 25%, rather than 50%, of the whole computational power in this case.

Sapirshtein and Sompolinsky [26] provided an efficient algorithm that computes an optimal selfish mining policy. They claimed that attackers with strictly less than 25% of the computational power can still profit from selfish mining. They also demonstrated how selfish miners can execute double-spending attacks without any costs. Göbel *et al.* [13] studied the effect of communication delays on the evolution of the Bitcoin blockchain. They showed that the strategy in [12] is not profitable under a model of delays changing quickly. Möser and Böhme [20] analyzed transaction fees and their externalities. Houy [14] and Kaşkaloğlu [16] analyzed potential changes of transaction fees. Carlsten *et al.* [9] developed a new attack strategy and replayed the selfish mining attack in the context of a transaction fee.

Some papers indicated that one can attack the Bitcoin blockchain by bribing other miners. Liao and Katz [18] introduced and formalized the whale attack, which demonstrates that rationality should not be underestimated when evaluating the security of cryptocurrencies. They established informal upper bounds on the expected cost to carry out the whale attack with 100% success probability. Bonneau [6] presented various bribery attacks.

III. THE CONSIDERED MODEL

Initially, we introduce the smart contract, which is provided by Bitcoin. Then, based on the smart contract, the considered model is presented.

A. SMART CONTRACT OF BITCOIN

A smart contract is a computer protocol intended to digitally facilitate, verify, or enforce contracts [8] [19]. The Bitcoin system realizes the smart contract through its scripts. There are three types of smart contracts in Bitcoin, namely, multisignature applications, guaranteed contracts and those relying on the prophecy [11]. Figure 2 shows the flowchart of a smart contract, which consists of the following states.

1) Predefined contract: In this state, the participants (miners) of a smart contract come to an agreement. Then, they publish the contract (scripts) and set the condition.
2) Event: A state that triggers the smart contract.
3) Verify: In this state, the scripts of the contract judge whether the event satisfies the condition predefined in the smart contract.
4) Execute: In this state, the scripts of the contract execute their functions, which are predefined in smart contract.
5) Ends: In this state, the smart contract is closed.

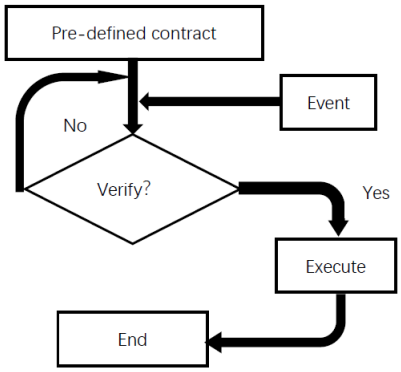

FIGURE 2. The flowchart of the smart contract in Bitcoin.

B. POSTING REWARDS VIA SMART CONTRACT

This subsection presents the model used in the subsequent analysis. As shown in Fig. 3, there are two players, termed as B and S, among which B created the block $B5$ and S created the block $S5$. The rest of the miners in the mining market are denoted as a set of players E. In particular, it is assumed that the mining strategy of each player $P_i \in E$ will follow either $B5$ or $S5$ to maximize its profit.

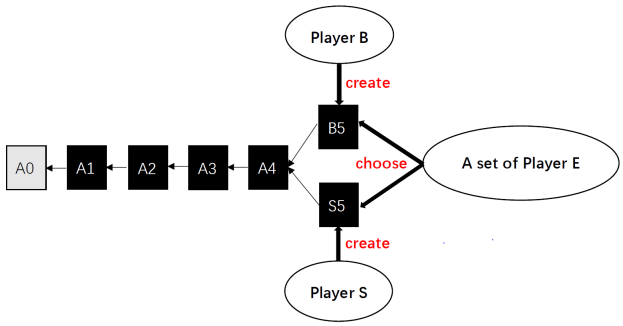

FIGURE 3. Graph representation of the considered model.

In the considered model, both B and S are able to post rewards to attract E to follow their forks via the smart contract introduced in Section III-A [27]. The smart contract in our model combines a prophecy server [21] with secure multiparty computations [1]. The prophecy server can receive an address, generate a serial number and return the hashing value of the serial number. Here, we assume that the prophecy server is secure theoretically. The sponsor of the smart contract is denoted as $Q \in \{B, S\}$. The blocks generated by Q and Q's competitor are denoted as L_q and L_c respectively. The height of L_q is denoted as t. For each node/player X, the address of X is denoted as $X.add$. The following provides the initial steps of posting a reward by Q.

1) Initialization: For $i = 1, 2, \ldots, n$, each player $P_i \in E$ holds $P_i.add$ and a pair of keys $(P_i.sk, P_i.pk)$, where $P_i.sk$ is the secret key and $P_i.pk$ is the public key. Further, Q also holds $Q.add$ and a pair of keys $(Q.sk, Q.pk)$.
2) Let $h_q = hash(L_q)$ and $h_c = hash(L_c)$. Then, Q generates an evaluation function $f_{h_q,h_c,t}$ shown in Algorithm 1, where $GetLatestBlock()$ returns the latest block in the blockchain, and $newblock.BCAdd$ denotes the output address of a basecoin in $newblock$.

Algorithm 1 $f_{h_q,h_c,t}()$: Evaluation Procedure

Output: An address or NULL.

```
newblock ← GetLatestBlock()
if newblock.height ≠ t + 1 then
    return NULL
end if
if newblock.parenthash = h_c then
    return Q.add
end if
if newblock.parenthash = h_q then
    return newblock.BCAdd
end if
return NULL
```

3) Let $hash_0 = hash(f_{h_q,h_c,t})$ denote the hashing value of $f_{h_q,h_c,t}$. Q creates a prophecy server PS that stores $hash_0$. Q sends PS $Q.add$. Then, PS generates a serial number s_0 and calculates $h_0 = hash(s_0)$. PS stores $(Q.add, s_0)$ and sends Q the hashing value h_0.
4) Each P_i sends $P_i.pk$ to Q. Then, Q sends P_i the address $PS.add$ and the evaluation function $f_{h_q,h_c,t}$.
5) P_i sends PS its address $P_i.add$. Then, PS generates a serial number s_i and calculates $h_i = hash(s_i)$. PS stores $(P_i.add, s_i)$ and sends P_i the hashing value h_i. Then, each P_i sends Q the value h_i.
6) Q creates a transaction $TX1$, whose input(s) $TX1.in$ are given by Q and the output $TX1.out$ is an address randomly chosen by Q. In addition, $TX1$ also includes the scriptSig and the scriptPubKey. The scriptSig is the signature of the unredeemed transactions corresponding to $TX1.in$ signed by $Q.sk$. The scriptPubKey is described as

$$\begin{aligned}&((hash(s) = h_0 \bigwedge ver(TX1, Q.pk)) \bigvee \ldots\\ &\bigvee (hash(s) = h_i \bigwedge ver(TX1, P_i.pk)) \bigvee \ldots\\ &\bigvee (hash(s) = h_n \bigwedge ver(TX1, P_n.pk))),\end{aligned} \tag{1}$$

which uses the notations introduced in [1], for $i = 1, 2, \ldots, n$. In (1), s denotes the serial number, and $ver(TX1, \bullet)$ is a signature of $TX1$ by $\bullet$. Then, Q broadcasts $TX1$.
7) When a $P_i \in E$ received $TX1$, P_i verifies $TX1$ by the scriptSig and the scriptPubKey of $TX1$. If $TX1$ passes the verification, P_i will trust the bounty. Otherwise, P_i will quit the smart contract.

After the initial steps, the prophecy server PS can verify the new blocks generated by Q or each $P_i \in E$ via the following steps.

1) When a player $p \in \{Q\} \cup E$ obtains a new block, p broadcasts the block.

2) p sends PS the evaluation function $f_{h_q,h_c,t}()$. Then, PS performs the evaluation steps shown in Algorithm 2. If the evaluation passed, PS will send p a serial number s.

Algorithm 2 Evaluation Steps in a Prophecy Server

Input: The evaluation function $f_{h_q,h_c,t}()$.
Output: The result of the verification or a string.

```
if hash(f_{h_q,h_c,t}()) ≠ hash_0 or f_{h_q,h_c,t}() = NULL then
    return false
end if
if Q.add = f_{h_q,h_c,t}() then
    return s_0
end if
for each i ∈ [1, n] do
    if P_i.add = f_{h_q,h_c,t}() then
        return s_i
    end if
end for
return false
```

3) p creates a transaction $TX2$, whose input(s) are $TX1.out$ and the output is the address given by p. In addition, $TX2$ also has the scriptSig and the scriptPubKey, where the scriptSig includes a signature of $TX1$ signed by $p.sk$, and the serial number s; the scriptPubKey includes the signature of $TX2$ by $p.sk$.
4) p broadcasts $TX2$ and s. If p is the first one generating the new block, the s given by PS corresponds to $p.add$. Other players can verify $TX2$ by scriptSig in $TX2$, and $TX2$ will be included in a block. Then, p can obtains the reward.

C. ANALYSIS

1) FEASIBILITY

The proposed model uses the prophecy servers [21] and the secure multiparty computations [1]. To apply this model, the system shall possess the following conditions.

1) The player B (or S) can judge whether the highest blocks of the blockchain have forks.
2) The player in E can accept the smart contract for the bounty.
3) The player B (or S) can increase the bounty amount after perceiving other smart contracts posted by the competitors.

For the first condition, as the proposed model is applied on the Bitcoin system, each player maintains a local copy of the blockchain. Thus, each player can detect the existence of forks when there are more than two blocks with the same height. For the second condition, when a player wants to post a reward, the player will create a transaction $TX1$ (see Step 6 of Section III-B) and broadcast it. Then, other players will receive the transaction $TX1$, and the player can identify $TX1$ as a posted reward by checking the content of scriptPubKey (1). For the final condition, the player can post another reward to increase the bounty amount. In summary, the proposed model is feasible.

2) STABILITY

The proposed model is applied on the blockchain system. With the proposed model, E can choose a fork to follow, depending on the computing power and the network delay. However, in the proposed model, E cannot change their strategies before increasing the height of the longest chain. Thus, the proposed model is stable.

IV. GAME THEORETICAL ANALYSIS

As shown in Figure 3, a mining network is associated with a set of parameters

$$\Gamma = (\mathcal{M}, \mathcal{P}, \sigma, D, \lambda, T), \tag{2}$$

where $\mathcal{M} = \{B, S, E\}$ denotes the set of mining pools; $\mathcal{P} = \{p_B, p_S, p_E\}$ denotes the ratios of computing power of B, S and E; and $1 = p_B + p_S + p_E$. Without loss of generality, assume that $p_B > p_S$. Furthermore, $\sigma = \{\sigma_B, \sigma_S\}$ denotes the rewards posted by B and S, respectively. D is the delay between any two mining pools. λ denotes the expected number of new blocks per second, and this value is $\lambda = 1/600$ according to the Bitcoin specification. The coinbase is T (miners receive T Bitcoins for generating a new block through mining). In addition, we assume that changing the mining strategy is cost-free for B, S and E.

In this section, we analyze the profits of S, B and E. The block $B5$ is generated at the time t_1, and $S5$ is generated at the time t_2. If $t_1 \geq t_2$, S will always follow $B5$ due to $p_B > p_S$, and thus, we only discuss $t_1 < t_2$. In this section, we analyze the profits of S, B and E in terms of two assumptions. First, we consider that the communication between any two pools does not have delay $D = 0$. Second, we consider $D > 0$.

A. BASELINE MODEL WITHOUT DELAY

We consider $D = 0$ in (2), which means there is no delay within the mining network. As $D = 0$, we have $t_1 \approx t_2$, or else S will follow $B5$ to continue the mining work. The following shows the profits of B and S via the strategies they adopted.

1) If S adopts the default mining strategy, B can also adopt the default mining strategy. In this case, the profit of B is given by

$$\frac{p_B + p_E}{p_B + p_S + p_E} \times T, \tag{3}$$

and the profit of S is given by

$$\frac{p_S}{p_B + p_S + p_E} \times T. \tag{4}$$

2) If S posts a reward via the method in Section III-B, while B adopts the default mining strategy, then the profit of B is given by

$$\frac{p_B}{p_B + p_S + p_E} \times T, \tag{5}$$

TABLE 1. Profits of S and B without network delay.

		B	
		Default strategy	Posting a reward
S	Default strategy	$\frac{p_S}{p_B+p_S+p_E} \times T, \frac{p_B+p_E}{p_B+p_S+p_E} \times T$	
	Posting a reward	$\frac{p_S+p_E}{p_B+p_S+p_E} \times (T-\sigma_S), \frac{p_B}{p_B+p_S+p_E} \times T$	$\frac{p_S+p_E}{p_B+p_S+p_E} \times (T-\sigma_S), \frac{p_B+p_E}{p_B+p_S+p_E} \times (T-\sigma_B)$

and the profit of S is given by

$$\frac{p_S+p_E}{p_B+p_S+p_E} \times (T-\sigma_S). \quad (6)$$

3) If both S and B post rewards via the method in Section III-B, the profit of B is given by

$$\frac{p_B+p_E}{p_B+p_S+p_E} \times (T-\sigma_B), \quad (7)$$

and the profit of S is given by

$$\frac{p_S+p_E}{p_B+p_S+p_E} \times (T-\sigma_S). \quad (8)$$

4) If S adopts the default mining strategy, the players in E will follow B, which has higher computing power than S. Thus, B does not need to post a reward.

The profit of each player is summarized in Table 1. The following provides the Nash equilibrium of this case.

Theorem 1: The best strategy for S is to post a reward between $\frac{p_E \times T}{p_E+p_B}$ *and* $\frac{T \times p_E}{p_E+p_S}$. *In this case, B can follow the default strategy, and E will follow the block that provides the best reward.*

Proof: If B adopts the default strategy, posting a reward is the best strategy for S, if

$$\frac{p_S+p_E}{p_B+p_S+p_E} \times (T-\sigma_S) > \frac{p_S}{p_B+p_S+p_E} \times T. \quad (9)$$

Then, we have

$$\sigma_S < \frac{T \times p_E}{p_E+p_S}. \quad (10)$$

If S posts a reward, the best strategy for B is also posting a reward, if

$$\frac{p_B+p_E}{p_B+p_S+p_E} \times (T-\sigma_B) > \frac{p_B}{p_B+p_S+p_E} \times T. \quad (11)$$

Then, we have

$$\sigma_B < \frac{p_E \times T}{p_E+p_B}. \quad (12)$$

From (10) and (12), if $\frac{p_E \times T}{p_E+p_B} \leq \sigma_S < \frac{T \times p_E}{p_E+p_S}$, the strict Nash equilibrium provides that S can post a reward while B can adopt the default strategy. This completes the proof. □

B. BASELINE MODEL WITH NETWORK DELAY

We consider $D > 0$ in (2). In this case, we have $t_2 - t_1 < D$, or else S will follow $B5$ to continue the mining work. At the time $t_1 + D$, both S and E receive $B5$, and then S immediately posts a reward via the method in Section III-B. Given a miner network Γ, $\gamma_i = \gamma(\Gamma) \in [0, 1]$ denotes the probability that a block belonging to the longest chain was mined by the miner i (see [17] for more details), and $\beta = \beta(\Gamma)$ denotes the rate of the block added to the longest chain per second (see [26] for more details). From [26], we have that B has a greater advantage than S and that, as D increases, the advantage of B extends. The following provides the profit of E by following $B5$ and $S5$, respectively.

Lemma 1: If E follows B5, the profit of E, denoted as E_1, *is given by*

$$E_1 = \frac{\beta(\Gamma_{B,E})}{\beta(\Gamma_{B,E})+\beta(\Gamma_S)} \times T \times \gamma(\Gamma_{B,E})_E.$$

Proof: In this case, the mining network Γ can be divided into two networks $\Gamma_S = (S, p_S, \sigma_S, D, \lambda_S, T)$ and $\Gamma_{B,E} = (\{B, E\}, \{p_B, p_E\}, \sigma_B, D, \lambda_{B,E}, T)$, where $\lambda_S + \lambda_{B,E} = \lambda$. From Eq. (2) of [17], the probability that a block in $\Gamma_{B,E}$ belonging to the longest chain was mined by E is given by

$$\gamma(\Gamma_{B,E})_E = \frac{p_E^2 e^{2D\lambda p_E} - p_E p_B (2\frac{e^{2D\lambda(p_E+p_B)}-1}{e^{2D\lambda p_E}+e^{2D\lambda p_B}-2} - 1)}{p_E^2 e^{2D\lambda p_E} - p_B^2 e^{2D\lambda p_B}}. \quad (13)$$

From Theorem 9 of [26], the rate of a block added to the longest chain in Γ_S per second is given by

$$\beta(\Gamma_S) = p_S \times \lambda, \quad (14)$$

and the rate of a block added to the longest chain in $\Gamma_{B,E}$ per second is given by

$$\beta(\Gamma_{B,E}) = \frac{(\lambda p_E)^2 e^{2D\lambda p_E} - (\lambda p_B)^2 e^{2D\lambda p_B}}{\lambda p_E e^{2D\lambda p_E} - \lambda p_B e^{2D\lambda p_B}}. \quad (15)$$

Then, the probability that a block is added to the longest chain belonging to $\Gamma_{B,E}$ is given by $\frac{\beta(\Gamma_{B,E})}{\beta(\Gamma_{B,E})+\beta(\Gamma_S)}$. From (13), (14) and (15), we have

$$E_1 = \frac{\beta(\Gamma_{B,E})}{\beta(\Gamma_{B,E})+\beta(\Gamma_S)} \times T \times \gamma(\Gamma_{B,E})_E. \quad (16)$$

This completes the proof. □

Lemma 2: If E follows S5, the profit of E, denoted as E_2, *is given by*

$$E_2 = \frac{\beta(\Gamma_{S,E})}{\beta(\Gamma_{S,E})+\beta(\Gamma_B)} \times \gamma(\Gamma_{S,E})_E \times T.$$

Proof: In this case, the mining network Γ can be divided into two networks $\Gamma_B = (B, p_B, \sigma_B, D, \lambda_B, T)$ and $\Gamma_{S,E} = (\{S, E\}, \{p_S, p_E\}, \sigma_S, D, \lambda_{S,E}, T)$, where $\lambda_B + \lambda_{S,E} = \lambda$. From Eq. (2) of [17], the probability that a block in $\Gamma_{S,E}$ belonging to the longest chain was mined by E is given by

$$\gamma(\Gamma_{S,E})_E = \frac{p_E^2 e^{2D\lambda p_E} - p_E p_S (2\frac{e^{2D\lambda(p_E+p_S)}-1}{e^{2D\lambda p_E}+e^{2D\lambda p_S}-2} - 1)}{p_E^2 e^{2D\lambda p_E} - p_S^2 e^{2D\lambda p_S}}. \quad (17)$$

TABLE 2. Profits of *S* and *B* with network delay.

		B	
		Default strategy	Posting a reward
S	Default strategy	$\frac{\beta(\Gamma_S)}{\beta(\Gamma_{B,E})+\beta(\Gamma_S)} \times T, \frac{\beta(\Gamma_{B,E})}{\beta(\Gamma_{B,E})+\beta(\Gamma_S)} \times T$	
	Posting a reward	$\frac{\beta(\Gamma_{S,E})}{\beta(\Gamma_{S,E})+\beta(\Gamma_B)} \times (T-\sigma_S), \frac{\beta(\Gamma_B)}{\beta(\Gamma_{S,E})+\beta(\Gamma_B)} \times T$	$\frac{\beta(\Gamma_{S,E})}{\beta(\Gamma_{S,E})+\beta(\Gamma_B)} \times (T-\sigma_S), \frac{\beta(\Gamma_{B,E})}{\beta(\Gamma_{B,E})+\beta(\Gamma_S)} \times (T-\sigma_B)$

From Theorem 9 of [26], the rate of a block added to the longest chain in Γ_B per second is given by

$$\beta(\Gamma_B) = p_B \times \lambda, \tag{18}$$

and the rate of a block added to the longest chain in $\Gamma_{S,E}$ per second is given by

$$\beta(\Gamma_{S,E}) = \frac{(\lambda p_E)^2 e^{2D\lambda p_E} - (\lambda p_S)^2 e^{2D\lambda p_S}}{\lambda p_E e^{2D\lambda p_E} - \lambda p_S e^{2D\lambda p_S}}. \tag{19}$$

Then, the probability that a block is added to the longest chain belonging to $\Gamma_{B,E}$ is given by $\frac{\beta(\Gamma_{S,E})}{\beta(\Gamma_{S,E})+\beta(\Gamma_B)}$. From (17), (18) and (19), we have

$$E_2 = \frac{\beta(\Gamma_{S,E})}{\beta(\Gamma_{S,E})+\beta(\Gamma_B)} \times \gamma(\Gamma_{S,E})_E \times T. \tag{20}$$

This completes the proof. □

To attract other miners, from (16) and (20), the σ_S posted by S must satisfy

$$\sigma_S \geq E_1 - E_2. \tag{21}$$

Figure 4 shows the curve E_1/E_2 with various values of (p_B, p_S, p_E) for a network delay of less than 35 msec. As shown in Figure 4, each curve gives a threshold D', which is a point at $E_1/E_2 = 1$. We can see that $E_1/E_2 < 1$ when the delay is less than D'. In this case, we have the bounty $\sigma_S = 0$; otherwise, S shall post a reward larger than $E_1 - E_2$ to attract miners. In addition, we have $D' = 0$ when $p_E > 0.5$. For $p_E \leq 0.5$ and smaller $(p_B - p_S)$, D' becomes larger.

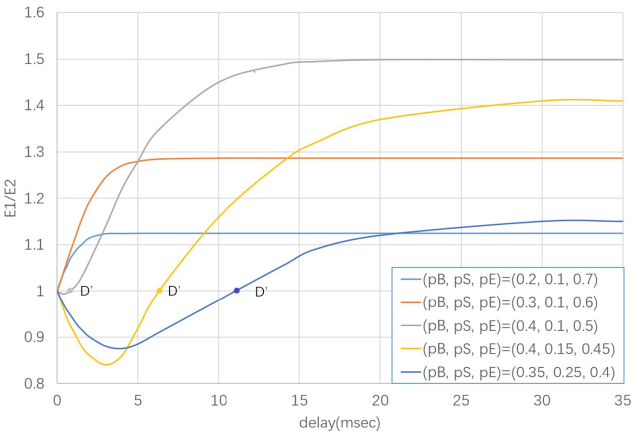

FIGURE 4. The ratio E_1/E_2, where $\lambda = 5/3$ (per msec).

Then, we discuss the profits of S and B via the strategies they adopted.

1) If S adopts the default strategy, B can also adopt the default strategy. In this case, from (14) and (15), the profit of B is given by

$$\frac{\beta(\Gamma_{B,E})}{\beta(\Gamma_{B,E})+\beta(\Gamma_S)} \times T, \tag{22}$$

and the profit of S is given by

$$\frac{\beta(\Gamma_S)}{\beta(\Gamma_{B,E})+\beta(\Gamma_S)} \times T. \tag{23}$$

2) If S posts a reward while B adopts the default strategy, from (18) and (19), the profit of B is given by

$$\frac{\beta(\Gamma_B)}{\beta(\Gamma_{S,E})+\beta(\Gamma_B)} \times T, \tag{24}$$

and the profit of S is given by

$$\frac{\beta(\Gamma_{S,E})}{\beta(\Gamma_{S,E})+\beta(\Gamma_B)} \times (T-\sigma_S). \tag{25}$$

3) If both S and B post rewards, from (15) and (19), the profit of B is given by

$$\frac{\beta(\Gamma_{B,E})}{\beta(\Gamma_{B,E})+\beta(\Gamma_S)} \times (T-\sigma_B), \tag{26}$$

and the profit of S is the same as (25).

4) If S adopts the default mining strategy, the players in E will follow B, which has higher computing power than S. Thus, B does not need to post a reward.

The profits are summarized in Table 2. We can obtain the Nash equilibrium of this case as in Section IV-A. From Table 2, we have a corollary as follows.

Corollary 1: We have the following observations.

1) *The profits of B, S and E are determined by their computing powers.*
2) *The increasing/decreasing rate of the profits are determined by $p_B - p_S$.*

Proof: For the first point, when there is no delay within the mining network, (7) and (8) show that the profits of B, S and E are determined by p_B, p_S and p_E. When E chooses S and $D \to \infty$, the profit of B is

$$\frac{p_E p_B + p_B p_S}{p_E^2 + p_E p_S + p_S^2 + p_E p_B + p_B p_S} \times T,$$

and the profit of S is

$$\frac{p_E^2 + p_E p_S + p_S^2}{p_E^2 + p_E p_S + p_S^2 + p_E p_B + p_B p_S} \times T.$$

Hence, the profits of B, S and E are determined by p_B, p_S and p_E.

For the second point, when p_S decreased and p_B increased, the profit of S decreased, and the profit of B increased. From Figure 4, the threshold D' decreases when $p_B - p_S$ increases. These findings show that the increasing/decreasing rate of the profits becomes faster when $p_B - p_S$ increases. When E chooses B, we obtain similar results. Hence, the increasing/decreasing rates of the profits are determined by $p_B - p_S$. □

V. SIMULATION

This section presents a number of simulations to verify our results. In following simulations, the computing power of S, B and E are $p_S = 0.25$, $p_B = 0.35$, and $p_E = 0.40$, respectively. Figure 5 shows the locations of 1000 nodes in a network, and each node has a unit of computing power. Thus, these nodes are classified as three pools S, B and E, and the size of each pool is determined by its computing power. In each pool, a node is chosen as the administrator. In the simulations, the administrators of S, B and E are at $(971, 494)$, $(477, 963)$ and $(488, 912)$ in Figure 5, respectively. The simulations follow the following conditions.

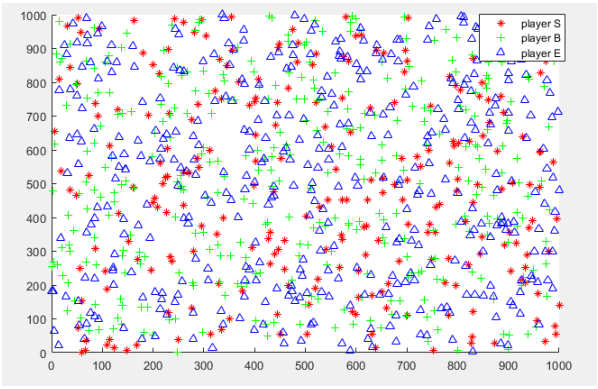

FIGURE 5. The distribution of nodes.

- At the instants of a Poisson process, blocks were mined by the randomly selected nodes. On average, a block was added to the longest chain every ten minutes.
- Every node maintains a local copy of the blockchain.
- The simulation terminates when S or B fails the competition.
- The delay is defined as $Delay = Coef\_Delay \times Distance$, where $Coef\_Delay \in [0.002, 0.04]$ is a real number and $Distance$ is determined by the Euclidean distance between two nodes. In particular, for the delay between any two pools, $Distance$ is the distance between the administrators of two pools. For the delay of a node in a pool, $Distance$ is the distance between the node and its administrator.
- Given each $Coef\_Delay \in [0.002, 0.04]$, we run the simulation 1000 times and record the revenue of S, B and E.

The following discusses the simulation results.

A. SIMULATION WITHOUT DELAY

In this simulation, we simulated 10 rounds without delay between any two pools. In each round, we simulated 1000 times to compute the average profit of S. Figure 6 shows the average profits of S without any delay when S adopts posting a reward and the default strategy. Notably, each point in Figure 6 (and Figures 7 and 8) represents the average of 1000 simulation results. We can see that S can always obtain a higher profit by posting a reward.

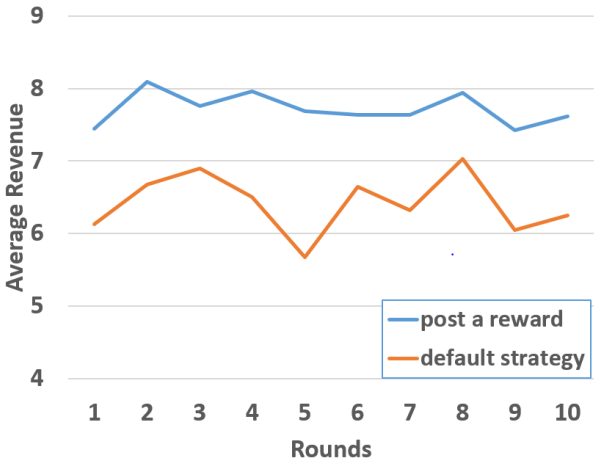

FIGURE 6. The average profit of S without delay.

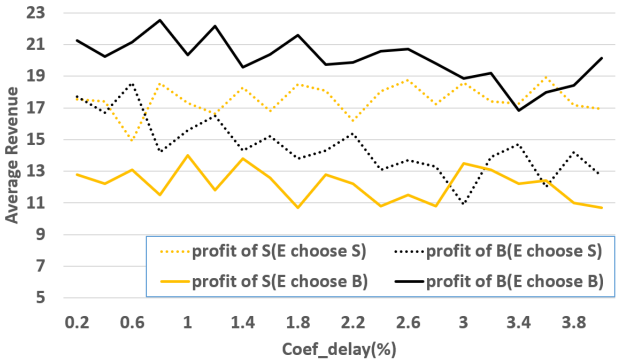

FIGURE 7. The average profit of S and B with delay.

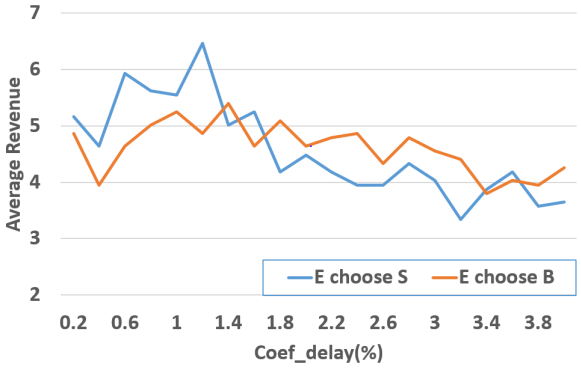

FIGURE 8. The average profit of E with delay.

B. SIMULATION WITH DELAY

In this simulation, we simulated 20 rounds and $Coef\_Delay \in [0.002, 0.04]$. Figure 7 shows the profits of B and S with different strategies. In Figure 7, the solid and dotted lines are used to represent the events describing "E choosing B" and "E choosing S", respectively. The black and yellow lines represent the profits of B and S, respectively. From Figure 7, one can see when E has a higher computing power than S and B, and the increasing amount of the profit of S is almost equal to that of B. This implication can be explained by the following analysis.

When $D \to \infty$, from (23), the profit of S adopting the default strategy is

$$\frac{p_B p_S + p_E p_S}{p_E^2 + p_B^2 + p_E p_B + p_B p_S + p_E p_S} \times T. \tag{27}$$

From (25), the profit of S posting a reward is

$$\frac{p_E^2 + p_S^2 + p_E p_S}{p_E^2 + p_S^2 + p_E p_B + p_B p_S + p_E p_S} \times T. \tag{28}$$

From (24), the profit of B adopting the default strategy is

$$\frac{p_B p_E + p_E p_S}{p_E^2 + p_S^2 + p_E p_B + p_B p_S + p_E p_S} \times T. \tag{29}$$

From (26), the profit of B posting a reward is

$$\frac{p_E^2 + p_B^2 + p_E p_B}{p_E^2 + p_B^2 + p_E p_B + p_B p_S + p_E p_S} \times T. \tag{30}$$

The difference of (27) and (28) is equal to the difference of (29) and (30). However, the denominator in (27) is smaller, and thus, S has a higher profit increasing ratio than B. This answers the implication stated above.

In Figure 8, we use $p_E = 0.10$. We can see that when the computing power of E is less than S and B, it is better for E to choose S when the delay is short; otherwise, it is better to choose B.

VI. CONCLUSION

In this paper, a new mining strategy for miners/pools that considers the occurrence of forks in the longest chain is discussed. In the considered model, there are three players: two of them are the creators of the forks, and the other player will choose one of the forks to follow. The practicality of the model in the Bitcoin network is discussed. In the analysis, two models are considered, where the first model does not have a network delay and the second one does. Then, some results are provided. First, if the delay in the Bitcoin network is less than a threshold, it is better for other miners or pools to follow the fork whose creator has a smaller computing power; otherwise, the larger pool's fork is a better choice. Second, for the two pools that created the forks, the pool with a smaller computing power may post a larger reward to attract other miners. There are some extensions of our work in the future. To enhance the reality of the considered model, we can consider models with more miners/pools. In addition, we can consider the best mining policy for an unstable network.

REFERENCES

[1] M. Andrychowicz, S. Dziembowski, D. Malinowski, and L. Mazurek, "Secure multiparty computations on bitcoin," in *Proc. IEEE Symp. Secur. Privacy*, May 2014, pp. 443–458.
[2] A. M. Antonopoulos, *Mastering Bitcoin*. Newton, MA, USA: O'Reilly, 2014.
[3] 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.
[4] G. Bigi, A. Bracciali, G. Meacci, and E. Tuosto, "Validation of decentralised smart contracts through game theory and formal methods," in *Programming Languages with Applications to Biology and Security*. Cham, Switzerland: Springer, 2015, pp. 142–161.
[5] R. Böhme, N. Christin, B. Edelman, and T. Moore, "Bitcoin: Economics, technology, and governance," *J. Econ. Perspect.*, vol. 29, no. 2, pp. 38–213, 2015.
[6] J. Bonneau, "Why buy when you can rent?" in *Proc. Int. Conf. Financial Cryptography Data Secur.* Berlin, Germany: Springer, 2016, pp. 19–26.
[7] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "SoK: Research perspectives and challenges for bitcoin and cryptocurrencies," in *Proc. IEEE Symp. Secur. Privacy*, May 2015, pp. 104–121.
[8] V. Buterin, "A next-generation smart contract and decentralized application platform," White Paper 3, 2014. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper
[9] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 154–167.
[10] N. T. Courtois and L. Bahack, "On subversive miner strategies and block withholding attack in bitcoin digital currency," 2014, *arXiv:1402.1718*. [Online]. Available: https://arxiv.org/abs/1402.1718
[11] K. Delmolino, M. Arnett, A. Kosba, A. Miller, and E. Shi, "Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2016, pp. 79–94.
[12] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Commun. ACM*, vol. 61, no. 7, pp. 95–102, 2018.
[13] J. Göbel, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor, "Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay," *Perform. Eval.*, vol. 104, pp. 23–41, Oct. 2016.
[14] N. Houy, "The economics of bitcoin transaction fees," *GATE WP*, vol. 1407, pp. 1–13, Feb. 2014.
[15] B. Johnson, A. Laszka, J. Grossklags, M. Vasek, and T. Moore, "Game-theoretic analysis of DDoS attacks against bitcoin mining pools," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2014, pp. 72–86.
[16] K. Kaskaloglu, "Near zero Bitcoin transaction fees cannot last forever," in *Proc. Int. Conf. Digit. Secur. Forensics*, 2014, pp. 91–99.
[17] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein, "Bitcoin mining pools: A cooperative game theoretic analysis," in *Proc. Int. Conf. Auto. Agents Multiagent Syst.*, 2015, pp. 919–927.
[18] K. Liao and J. Katz, "Incentivizing blockchain forks via whale transactions," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Cham, Switzerland: Springer, 2017, pp. 264–279.
[19] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Cham, Switzerland: Springer, 2017, pp. 357–375.
[20] M. Möser and R. Böhme, "Trends, tips, tolls: A longitudinal study of Bitcoin transaction fees," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2015, pp. 19–33.
[21] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Tech. Rep., 2008. [Online]. Available: http://bitcoin.org/bitcoin.pdf
[22] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton, NY, USA: Princeton Univ. Press, 2016.
[23] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS P)*, Mar. 2016, pp. 305–320.

[24] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2016, pp. 515–532.

[25] O. Schrijvers, J. Bonneau, D. Boneh, and T. Roughgarden, "Incentive compatibility of bitcoin mining pool reward functions," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2016, pp. 477–498.

[26] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in bitcoin," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2015, pp. 507–527.

[27] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, and J. J. Kishigami, "Blockchain contract: A complete consensus using blockchain," in *Proc. IEEE 4th Global Conf. Consum. Electron. (GCCE)*, Oct. 2015, pp. 577–578.

SUI CHENG received the B.E. degree in information security from the University of Science and Technology of China (USTC), Hefei, China, in 2017, where he is currently pursuing the M.Sc. degree. His research interest includes mining strategies in Bitcoin.

SIAN-JHENG LIN (M'16) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from National Chiao Tung University, Hsinchu, Taiwan, in 2004, 2006, and 2010, respectively. From 2010 to 2014, he held a Postdoctoral position at the Research Center for Information Technology Innovation, Academia Sinica. From 2014 to 2016, he held a postdoctoral position at the Electrial Engineering Department, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia. He was a part-time Lecturer with Yuanpei University, from 2007 to 2008, and Hsuan Chuang University, from 2008 to 2010. He is currently a Project Researcher with the School of Information Science and Technology, University of Science and Technology of China (USTC), Hefei, China. In recent years, his research interest includes algorithms for MDS codes and its applications to storage systems.