Blockchain Based Offloading Strategy: Incentive, Effectiveness and Security

Weikang Liu, Bin Cao, *Senior Member, IEEE*, and Mugen Peng, *Fellow, IEEE*

***Abstract*— To securely integrate Mobile Edge Computing (MEC) into Wireless Blockchain Network (WBN), this paper proposes a framework for blockchain based offloading strategy, where blockchain nodes are categorized as blockchain users and blockchain miners from a motivation perspective. Particularly, aiming at improving the motivation ability, a block generation process is first designed for blockchain miners' short transaction processing time lower bound. Then, to further maximize the utilities of both blockchain users and blockchain miners, an optimization problem is formulated to determine an optimal strategy which involves a trade-off between the fast transaction confirmation rate required by blockchain users and the transaction fees obtained by blockchain miners. A Stackelberg game is introduced to model the interaction between the blockchain users and miners. Meanwhile, a distributed algorithm is designed to converge this strategy in an iterative manner based on the buyer-seller negotiation. Additionally, double-spending attack and selfish mining attack are analysed to examine their impact on the system performance in terms of confirmation delay and throughput while guaranteeing the high security level. Finally, extensive experiments have been conducted to show the rightness and effectiveness of the proposed equilibrium-based strategy and mathematical analysis, and some insights are discussed for the further guide as well.**

***Index Terms*— Blockchain, Internet of Things, offloading, Stackelberg game, double-spending, selfish mining.**

I. Introduction

IDENTIFIED as one of the most disruptive technologies of this century, Internet of Things (IoT) has attracted much attention from society, industry and academia as a promising technology that can enhance day to day activities, leading to diverse data flowing in its networks. Originating from human activities, most of these data is related to personal sensitive information, and thus, its security and privacy is of paramount importance [2]. Nevertheless, due to the low capability and the adopted isolated hardware solutions of IoT devices, they are the most vulnerable elements in the network to be attacked, which may result in privacy leakage and data tampering. Moreover, the IoT devices controlled by the attackers, could be the accomplices in making the centralized server inaccessible. Notably, these mentioned inevitable challenges are closely associated with IoT's centralized topology [3]. Emerging as a promising technology to build a secure and reliable ledger, blockchain presents us a decentralization solution [4]. In blockchain, participants need to solve a mathematical puzzle to generate a new block, leading to substantial and ongoing resource consumption [5]. However, constrained by the cost, size and battery life, typical IoT devices cannot provide sufficient capability to run a resource demanding blockchain protocol. Facing this dilemma, allowing computation tasks to be offload, Mobile Edge Computing (MEC) emerges as a promising solution to expand IoT devices' service capabilities [6]. Whereas, how to integrate MEC into Wireless Blockchain Network (WBN), while ensuring its security and performance, remains a challenge.

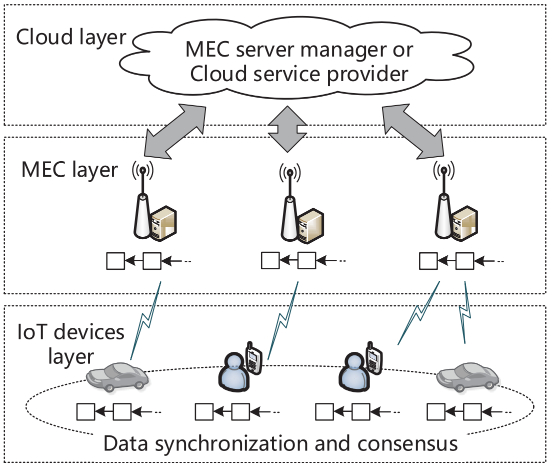

Cloud layer
MEC server manager or Cloud service provider
MEC layer
IoT devices layer
Data synchronization and consensus

Fig. 1. Typical framework for MEC enabled WBN.

Following the architectural pattern of MEC, the typical framework for MEC enabled WBN is constituted by IoT devices layer, MEC layer and cloud layer, as shown in Fig. 1 [7], [8], [9]. In this typical framework, controlled by the MEC server manager, edge servers are not the direct participants in the blockchain system, but network resources providers for IoT devices, including computational

Manuscript received 16 March 2022; revised 16 June 2022; accepted 30 June 2022. Date of publication 10 October 2022; date of current version 22 November 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB1714100 and Grant 2020YFB1806700; in part by National Natural Science Foundation of China under Grant 61925101 and Grant 61831002; and in part by the Zhejiang Laboratory under Grant 2021KF0AB03. An earlier version of this paper was presented in part at the 2020 IEEE ICC [DOI: 10.1109/ICC40277.2020.9149311]. *(Corresponding author: Bin Cao.)*

Weikang Liu and Mugen Peng are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: weikang_liu@bupt.edu.cn; pmg@bupt.edu.cn).

Bin Cao is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the Zhejiang Laboratory, Hangzhou 311121, China (e-mail: caobin@bupt.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2022.3213324.

Digital Object Identifier 10.1109/JSAC.2022.3213324

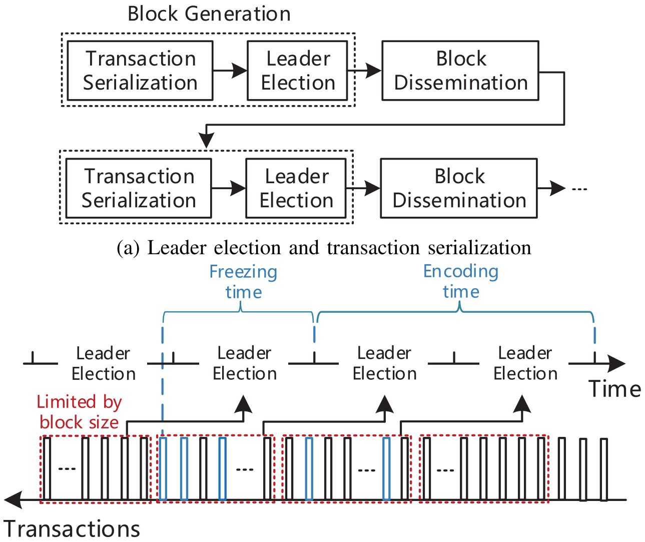
resources and storage resources [10]. However, independent of the blockchain system, the MEC server manager or the cloud service provider may become a central node, thereby undermining the blockchain system's distributed nature and further damaging its security. Therefore, to propose a blockchain based offloading framework, which can combine MEC and WBN effectively and securely, becomes the first challenge.

Nevertheless, in addition to the appropriate framework, the security of MEC enabled WBN is also inseparable from the maintenance of participants. No matter in helping motivate participants to maintain blockchain system honestly, or encouraging more participants to join in the maintenance to fight against greedy attackers, the incentive mechanism plays a key role. Based on the typical framework, most of researches treat IoT devices as blockchain miners who rent computational resources from corresponding edge servers to solve the Proof-of-work (PoW) problem on one hand, and earn mining rewards as well as transaction fees from blockchain on the other hand [11], [12], [13]. Actually, besides the mining profit, there exist other motivations to loT devices from the perspective of user experience, for example, higher transaction rates and shorter transaction processing time. Transaction processing time is defined as the duration from the blockchain user's transactions being received by blockchain miners to all the transactions being encapsulated into blocks. Thus, as one of the security elements, an incentive mechanism that considers the different motivations becomes the second challenge.

There is a relationship between the required transaction rate of blockchain users and their transaction processing time as follows,

$$
\begin{aligned}
&\text{Transaction processing time}\\
&\quad= \text{Freezing time} + \text{Encoding time},\\
&\quad= \frac{\text{Number of transactions}}{\text{Required transaction rate}}.
\end{aligned}
$$

Transaction processing time consists of freezing time and encoding time. Freezing time refers to the duration of time between a transaction being received by blockchain miners and when they are ready to encode it into a block. Encoding time is when blockchain miners encode the transactions into blocks until the blocks are successfully generated. If the blockchain user's transactions are encapsulated into multiple blocks, encoding time is the time when all the blocks are generated. Influenced by Bitcoin's coupled scheme shown in Fig. 2 (a), the leader election is only for serializing historical transactions as shown in Fig. 2 (b), thus resulting in a long freezing time for current transactions of blockchain users. However, in the perspective of blockchain users, a shorter freezing time would lead to a shorter transaction processing time and correspondingly a higher transaction rate, bringing a more extraordinary motivation ability for the whole blockchain system. To optimize Bitcoin's performance defects on freezing time, Bitcoin-NG [14] has optimized the block generation process. By introducing key blocks and micro blocks, Bitcoin-NG decouples the leader election and transaction serialization to eliminate the long system freeze between Bitcoin's leader elections. Though performance improvements have been made, the adjustments of traditional blockchain structure may lead to additional security issues, i.e., intentional key block forks [14]. Therefore, a block generation process which can reduce the lower bound on the transaction processing time without introducing risks caused by structure changing, becomes the third challenge.



(a) Leader election and transaction serialization

(b) Transaction processing time: transactions of the BU is encapsulated into two contiguous blocks

Fig. 2. Bitcoin's coupled scheme.

Aiming at tackling the above challenges, the main contributions of this paper are as follows.

- First, to avoid central nodes forming in WBN, a framework for blockchain based offloading is proposed, where edge servers are treated as blockchain miners, undertaking blockchain functionality operations and earning transaction fees, while IoT devices are treated as blockchain users, uploading transactions to blockchain with specified transaction rate requirements. In the light of different motivations of them, the entities and interactive process between them are refined. Particularly, to improve the whole blockchain system's motivation ability, a new block generation process is designed for blockchain miners' short transaction processing time lower bound.
- Second, to determine suitable required transaction rates for blockchain users and transaction price for the set of blockchain miners, an optimization problem using single-leader-multiple-followers Stackelberg game is constructed. With the proposed iterative algorithm, the optimization problem could be solved in a distributed manner to fully motivate their actions.
- Finally, two types of attack strategies, double-spending and selfish mining attacks, are analyzed and modeled to examine their impact on confirmation delay while maintaining a high-security level. In the case of double-spending attacks, the blockchain system relies on a higher confirmation threshold to reduce the impact of malicious blockchain miners. In the case of selfish mining attacks, it takes longer to reach the same confirmation threshold to make up for the reduced overall transaction rate.

The rest of this paper is organized as follows. In Section II, the framework is illustrated for blockchain based offloading

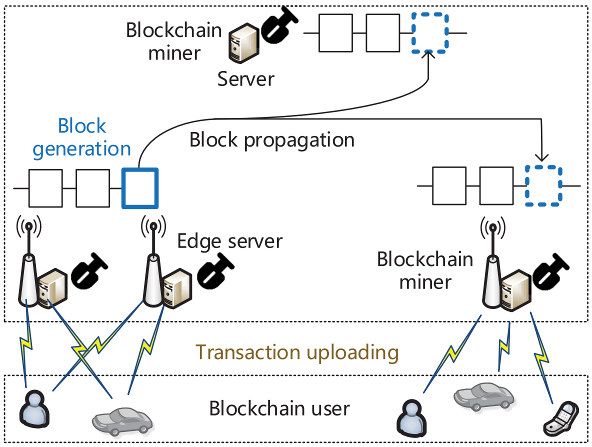


Fig. 3. Framework for blockchain based offloading strategy.

strategy. Section III presents the system model and problem formulation. Then, the optimal solution of the formulated Stackelberg game is analysed and the iterative algorithm is proposed in Section IV. Section V introduces and models the selfish mining attack and double-spending attack. Afterwards, the transaction rate, price per block and influenced confirmation delay under the two kinds of attacks are evaluated and results are discussed in Section VI. Section VII reviews the related work, and finally, the conclusion and future work are given in Section VIII.

II. Framework for Blockchain Based Offloading Strategy

In this section, the framework for blockchain based offloading strategy is formalized. For better illustration, entities are first introduced, and then the offloading and consensus process is described.

A. Entities in Blockchain Based Offloading Strategy

As illustrated in Fig. 3, there exists two entities in the proposed framework: Blockchain Miner and Blockchain User.

Blockchain Miner, denoted by BM_i where $i \in \mathbb{M} = \{1, \ldots, M\}$, is identified by a public key BM_i^p and a private key BM_i^s. Let $\mathbb{BM} = \{BM_1, \ldots, BM_M\}$ denote blockchain miners' set. Based on the PoW consensus mechanism, each blockchain miner ought to compete with each other to get the bookkeeping right by expending CPU effort to solve the PoW problem. Once its mined block is valid according to the Longest-Chain-Rule (LCR) [4], it can acquire fees of all the transactions included in the block which is denoted by Bl. Besides the coinbase transaction created by BM_i itself, each block contains K transactions from blockchain users. Each block $Bl = (BM_i^p, preHash, nonce, \mathbb{TR}, Sig_{\mathbb{TR}}, D)$ should contain at least six types of information, where $nonce$ is the proof to verify the effort paid by BM_i, $preHash$ is the hash of the previous chained block, $\mathbb{TR} = \{TR_0, TR_1, \ldots, TR_K\}$ is the set of packed transactions, $Sig_{\mathbb{TR}}$ is the transaction signature obtained by signing hash of $\mathbb{TR}$ with BM_i^s and D is the target difficulty. Except for $\mathbb{TR}$, all other information is encapsulated in the header of Bl. In order to protect blockchain system from being flooded with a large number of meaningless transactions and ensure the service experience of users, transactions without fees are not forwarded and are therefore not included into blocks by $\mathbb{BM}$ and this is a method to counter Distributed Denial of Service attack (DDoS) and Sybil attack.

Blockchain User, denoted by BU_j where $j \in \mathbb{N} = \{1, \ldots, N\}$, is identified by a public key BU_j^p and a private key BU_j^s. Let $\mathbb{BU} = \{BU_1, \ldots, BU_N\}$ denote blockchain users' set. In order to counter against DDoS and Sybil attacks for blockchain miners, each BU is required to pay a fee for every transaction if it hopes to get the blockchain service successfully. Represented as $TR = (Data, Fee, Sig_{TR}, B_{in}^p, B_{out}^p)$, each transaction contains at least five types of information, where $Data$ is the data $\mathbb{BU}$ want to write into blockchain system, Fee is the transaction fee that $\mathbb{BU}$ need to pay, B_{in}^p is the public key of the transaction originator, B_{out}^p is public key of the transaction beneficiary and Sig_{TR} is the signature obtained by signing hash of $\{Data, Fee\}$ with transaction originator's private key B_{in}^s. Except for coinbase transactions whose beneficiary are $\mathbb{BM}$, there exists transactions between $\mathbb{BU}$. As a result, B_{out}^p can be set to BM_i^p or BU_j^p.

B. Communication Model

As shown in Fig. 3, there are two types of communication models: one for consensus among $\mathbb{BM}$, and one for interaction between $\mathbb{BU}$ and $\mathbb{BM}$. In this section, both of them are detailed.

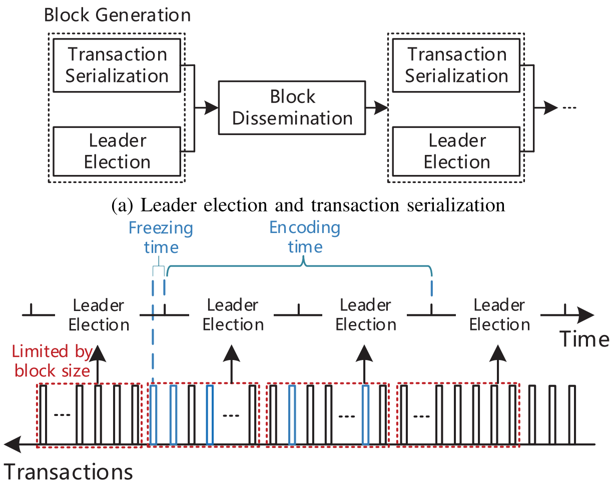
$\mathbb{BM}$ can access base stations in Next Generation Radio Access Network (NG-RAN) through the NG interface, or Evolved Universal Terrestrial Radio Access Network (E-UTRAN) through the S1 interface [15]. With the help of the base stations, they can exchange information such as blocks and transactions through the Xn and X2 interface [16]. In addition, they can also be connected to the core network through the S1 or NG interface to exchange information. The blockchain's ability to resist double-spending attacks depends on the participation of large numbers of blockchain miners. Then, to make it more flexible to join in, $\mathbb{BM}$ can also be servers in Data Network (DN) [17], and they can exchange information through the core network.

$\mathbb{BU}$ offload their workloads to $\mathbb{BM}$ through the nearby base stations. For the fear of failure or maliciousness of single access point, $\mathbb{BU}$ should broadcast their transactions to all nearby $\mathbb{BM}$ associated with base stations that can provide channel gains above a certain threshold. After that, the $\mathbb{BM}$ would forward the transactions to all the other $\mathbb{BM}$.

C. Offloading Process

The offloading and consensus process between $\mathbb{BM}$ and $\mathbb{BU}$ are presented in this section. Meanwhile, a new block generation process is proposed to be executed at the side of $\mathbb{BM}$, for frequent leader elections and high block space utilization.

- **Step1. Register.**
  In the first step, $\mathbb{BU}$ and $\mathbb{BM}$ are required to register in blockchain system. They will be assigned with a public/private key pair, which would be used for signature verification and wallet address. Without this identity, they are not allowed to participate in the service.



(a) Leader election and transaction serialization

(b) Transaction processing time: transactions of the BU is encapsulated into two contiguous blocks

Fig. 4. Proposed decoupled scheme.

- **Step2. Transaction uploading.**
  By established wireless connections to the nearby BMs with wireless access functions, $\mathbb{BU}$ can request the state information about blockchain system, i.e., the price for single transaction, transaction load capability and confirmation scores. Transaction load capability is the remaining affording transaction rate that the blockchain system can provide to blockchain users. Confirmation scores refer to the number of confirmations gained by the block to which the transaction belongs. Besides getting this information, $\mathbb{BU}$ can also upload transactions which are then stored in $\mathbb{BM}$'s transaction pool and queued for $\mathbb{BM}$'s validation, forwarding and inclusion into a candidate block. It is worth noting that the wireless link between the device and server is secure since every transaction contains a signature generated by BU_j^s [18].
- **Step3. Block generation.**
  Bitcoin's coupled scheme makes a long system freeze for transaction serialization between leader elections, and thus, there is a longer transaction processing time for $\mathbb{BU}$. As shown in Fig. 4 (a), in parallel with transaction serialization, each BM in the proposed decoupled scheme must try to construct and solve the PoW problem where the target threshold $targetHash$ is the same for $\mathbb{BM}$. Due to this design, the long system freeze can be alleviated as shown in Fig. 4 (b), leading to a shorter transaction processing time for $\mathbb{BU}$. For detailed illustration, this process is shown in **Algorithm 1**.[1]
- **Step4. Block dissemination.**

Algorithm 1 Block Generation Process

```
Input: $BM_i$, $D$ and the previous chained block $Bl_{pre}$.
Output: broadcast the Block $Bl$
 1: Calculate $preHash$
        $Hash\,(Bl_{pre}.BM_i^p + Bl_{pre}.preHash + Bl_{pre}.nonce + Bl_{pre}.Sig_{\mathbb{TR}} + Bl_{pre}.D + Bl_{pre}.other)$,
    where $other$ represents information in the block that is not relevant to transactions.
 2: $isPostBlSuc \leftarrow false$.
 3: if $BM_i$ is unregistered then
 4:     $BM_i$ has not been registered,
 5:     goto final.
 6: end if
 7: Construct the PoW problem
    $Hash(BM_i^p + preHash + nonce + D + other) < targetHash$.
 8: Try to find $nonce$ to meet the PoW problem.
 9: Select transactions from local transaction pool to form $TR_1, TR_2, \ldots, TR_K$ to fulfill the block.
10: Create the coinbase transaction
        $TR_0 \leftarrow (Data_{fee}, Null, Null, Null, BM_i^p)$,
    where $Data_{fee}$ shows that $fee = \sum_{n=1}^{K} TR_n.Fee$ is included in $Data$ and $Null$ means this special transaction needs no transaction fees, no signature and no transaction originator.
11: Get $Sig_{\mathbb{TR}}$ by signing hash of $\mathbb{TR}$ with $BM_i^s$.
12: $Bl \leftarrow (BM_i^p, preHash, nonce, \mathbb{TR}, Sig_{\mathbb{TR}}, D)$.
13: $isPostBlSuc \leftarrow postBlock(Bl)$.
14: final.
15: return $isPostBlSuc$.
```

After the block is fulfilled, it will be added to the local ledger and broadcast to peers and this operations will also be performed by other blockchain miners once they have verified it as a valid block. Including fees charged by blockchain miners, transactions in this block would not be eligible for confirmation until the block has been judged to belong to longest chain.

III. System Model and Problem Formulation

A. System Model

In the PoW consensus mechanism, the effective growth rate of the main chain contributed by $\mathbb{BM}$ remains stable, and the larger the set $\mathbb{BM}$, the lower the block completion probability for each BM's hashing operations, which are independent of each other. Therefore, to find a block whose hash value is less than $targetHash$, each BM needs to perform hashing operations at a tremendous rate with an extremely low success probability for each operation. Then, the block completions at each BM can be modelled as a Poisson process [19]. Let $\{\lambda_1, \ldots, \lambda_i\}, i \in \mathbb{M}$ denote the block completion rate of $\mathbb{BM}$ measured by the number of blocks per hour. Therefore, the aggregate block completion of this blockchain system can also

[1] Leader election corresponds to steps 7 and 8, while transaction serialization corresponds to steps 9-12. $postBlock(Bl)$ represents broadcasting the block to blockchain system. $isPostBlSuc$ indicates whether $postBlock(Bl)$ executes successfully, with success as true and failure as false. From **Algorithm 1**, though transaction serialization is observed to be executed after leader election, they can actually run in parallel as shown in Fig. 4 (a). Different from Bitcoin whose PoW problem changes with its encapsulated $\mathbb{TR}$, $\mathbb{TR}$ in this paper would not affect the constructed PoW problem. Therefore, the long system freeze for transaction serialization between leader elections can be alleviated.

be modelled as a Poisson process with rate

$$\Lambda = \sum_{i=1}^{M} \lambda_i. \tag{1}$$

Due to the fact that a mined block is not immediately synchronized to the entire system because of transmission delay, it will routinely happen that simultaneous blocks may spread in the system and each miner individually chooses which block to accept. Thus, the effective growth rate (denoted by Γ) will be less than Λ. With K transactions accepted in each block, the effective transaction rate of blockchain system can be modeled as a Poisson process with rate $K\Gamma$. With price β per block paid to $\mathbb{BM}$, it is reasonable for BU_j to require an appropriate transaction rate $\gamma_j, j \in \mathbb{N}$. Thus, we can get the overall effective transaction rate $K\Gamma$ as follows,

$$K\Gamma = \sum_{j=1}^{N} \gamma_j. \tag{2}$$

To simplify our analysis, but can be extended into a generic case, the block completion rate of different BMs can be set to the same value with $\lambda_i \equiv \lambda$, for all $i \in \mathbb{M}$. Therefore, the aggregate block completion rate $\Lambda = M\lambda$. According to [19], the block propagation delays between blockchain miners BM_m and BM_l where $\{m, l\} \in \mathbb{M}, m \neq l$ are sampled from an exponential distribution with rate $\mu_{m,l}$. For all $m \neq l, \{m, l\} \in \mathbb{M}$, communication intervals can be also set to the same value with $\mu_{m,l} \equiv \mu$. Based on the conclusion of [19], the effective growth rate Γ behaves as

$$\Gamma = M\lambda\left(1 - \frac{\lambda A_M}{\mu}\right) = \Lambda\left(1 - \frac{\Lambda A_M}{M\mu}\right), \tag{3}$$

where $A_M = \sum_{m=1}^{M-1} \frac{1}{m}$ is the M^{th} harmonic number and $\mu \gg \lambda$.

For a given Γ, there are two feasible solutions to Λ. Based on (2) and (3), they can be shown as follows,

$$\begin{aligned} \Lambda^1 &= \frac{M\mu(K-\sigma)}{2AK} < \frac{M\mu}{2A}, \\ \Lambda^2 &= \frac{M\mu(K+\sigma)}{2AK} > \frac{M\mu}{2A}. \end{aligned} \tag{4}$$

where

$$0 < \sigma = \sqrt{\frac{K\left(MK\mu - 4A\sum_{j=1}^{N}\gamma_j\right)}{M\mu}} < K. \tag{5}$$

However, when $\Lambda = \frac{M\mu}{2A}$, Γ gets its extreme value. As a result, when $\Lambda = \Lambda^2$, there would be more computational resources consumption. Thus, for a better state, Λ^2 is discarded, and then $\Lambda = \Lambda^1$.

Let c denote the cost of a BM to produce a unit of computational power, and then the cost for each block is cr where r is the miner's computing power or hash power [20]. Learning from [21], there exists a relationship $\lambda = {}^{r}/_{D}$ where D is the difficulty of mining a block. Thus, the cost of generating a block is

$$cr = cD\frac{r}{D} = cD\lambda = cD\frac{\Lambda}{M}. \tag{6}$$

Considering a practical constraint that D must be an appropriate value which can avoid generating block too fast [4] and each BM has finite computation capacity. Therefore, there exists a constraint λ_{max} that makes $\lambda \leq \lambda_{max}$. As a result, we have

$$\Lambda = M\lambda \leq M\lambda_{max}. \tag{7}$$

Then, based on (3) and (7), there exists a fixed upper bound Γ_{max} for Γ with the corresponding value M,

$$\Gamma_{\max} = \begin{cases} M\lambda_{\max}\left(1 - \dfrac{\lambda_{\max}A_M}{\mu}\right) & \lambda_{\max} < \dfrac{\mu}{2A}, \\ \dfrac{M\mu}{4A} & \lambda_{\max} \geq \dfrac{\mu}{2A}. \end{cases} \tag{8}$$

Hence, according to (2), the constraint on $\mathbb{BU}$'s requirements for their transaction rates can be shown as follows,

$$\sum_{j=1}^{N} \gamma_j = K\Gamma \leq K\Gamma_{max}. \tag{9}$$

B. Stackelberg Game Formulation

In order to encourage $\mathbb{BM}$ to share their computational resources, it is reasonable for them to claim for an amount of fees for each transaction. Meanwhile, $\mathbb{BU}$ can pay a fee to require different transaction rates based on their service levels [22]. Thus, a Stackelberg game can be formulated to optimize the interaction between the $\mathbb{BU}$ and $\mathbb{BM}$ to determine the optimal strategy. However, there is no a long lasting relationship between the specific BU and specific BM, because fees attached to the transaction uploaded by specific BU are earned by $\mathbb{BM}$ in a competitive manner. Moreover, due to the fact that every BM shares the same information about state of blockchain system, they can have the same reaction, and then further verify whether other BMs' reactions are correct or not. Hence, this Stackelberg game is a single-leader-multiple-followers game where $\mathbb{BM}$ can be treated as the leader and BUs are the followers.

At BU_j's side, the utility of BU_j includes the satisfaction degree and incentive cost, i.e., the transaction fee. The satisfaction degree is evaluated by the logarithmic function which is widely used in mobile computing and wireless communication domains [23], [24], [25], and the incentive cost is equal to its required transaction rate times the price paid for each transaction. Therefore, in order to maximize the utility by requiring considerable transaction rate γ_j measured by the number of transactions per hour, the optimization problem for BU_j can be expressed as follows,

$$\begin{aligned} \max_{\gamma_j} \ & U_{BU_j} = \alpha_j \log(1+\gamma_j) - \frac{\beta}{K}\gamma_j, \\ s.t. \ & \sum_{j=1}^{N} \gamma_j \leq K\Gamma_{max}, \\ & \gamma_j \geq 0, \quad \forall j \in \mathbb{N}, \end{aligned} \tag{10}$$

where α_j is the weight factor for BU_j, ${}^{\beta}/_{K}$ is the transaction price. Thus, $\beta\gamma_j/K$ denotes the average fees paid by BU_j

per hour. $K\Gamma_{max}$ is the maximum transaction rate blockchain system can afford.

At $\mathbb{BM}$'s side, the utility of them is defined as charged transaction fees minus computational resources consumption. Thus, to maximize their revenue, the optimization problem can be expressed as

$$\max_{\beta}\ U_{\mathbb{BM}} = \beta\Gamma - cD\frac{\Lambda}{M}\times\Lambda, \tag{11}$$

where $\beta\Gamma$ is the reward earned by $\mathbb{BM}$ in mining Γ confirmed blocks per hour, and $(cD\Lambda/M)\,\Lambda$ is the cost of the computational resources expended by $\mathbb{BM}$ in mining Λ blocks per hour.

IV. Optimal Solution Analysis

In this section, based on the Karush-Kuhn-Tucker (KKT) conditions [26] and backward induction method [27], the optimization problem of both the leader and followers are firstly analysed. Then, in the distributed environment, using the iterative update function proposed in this paper, the optimal solution is obtained. Finally, the optimal solution is proved to be a Stackelberg Equilibrium (SE) in the proposed game.

A. Analysis of Follower's Optimization Problem

For BU_j, the first order derivation of U_{BU_j} with respect to γ_j can be expressed as follows,

$$\frac{\partial U_{BU_j}}{\partial\gamma_j} = \frac{\alpha_j}{\gamma_j+1} - \frac{\beta}{K}, \tag{12}$$

and then the second order derivation is,

$$\frac{\partial^2 U_{BU_j}}{\left(\partial\gamma_j\right)^2} = -\frac{\alpha_j}{\left(\gamma_j+1\right)^2} < 0. \tag{13}$$

Obviously, we can see that U_{BU_j} is a concave function of γ_j. Since the constraint (9) is affine, the Lagrangian Relaxation with the multiplier u_j can be applied to solve this optimization problem (10) at BU_j as follows

$$\begin{aligned} L_{BU_j}(\gamma_j,u_j) &= \alpha_j\log\left(1+\gamma_j\right) - \frac{\beta}{K}\gamma_j \\ &\quad - u_j\left(\gamma_j + \sum_{i\neq j}\gamma_i - K\Gamma_{max}\right), \end{aligned} \tag{14}$$

where $L_{BU_j}(\gamma_j,u_j)$ is the Lagrangian of the utility function (10) BU_j and $u_j, j\in\mathbb{N}$ are the Lagrangian multipliers for the inequality constraint (10). The KKT conditions can be obtained as follows where $*$ represent the optimal solution.

$$\begin{aligned} u_j^*\left(\gamma_j^* + \sum_{i\neq j}\gamma_i - K\Gamma_{max}\right) &= 0, \\ \gamma_j^* + \sum_{i\neq j}\gamma_i - K\Gamma_{max} &\le 0, \\ \gamma_j^* > 0,\quad u_j^* &\ge 0. \end{aligned} \tag{15}$$

Let $\frac{\partial L_{BU_j}(\gamma_j,u_j)}{\partial\gamma_j} = 0$, the optimal γ_j^* can be obtained

$$\gamma_j^* = \frac{K\alpha_j}{\beta + Ku_j^*} - 1. \tag{16}$$

It is easy to see that γ_j^* is a function of β, which means that to obtain γ_j^*, the corresponding β is a necessary information. Besides, the information about the maximum transaction rate $K\Gamma_{max}$ and the current transaction flow $\sum_{i\neq j}\gamma_i$ are also necessary.

B. Analysis of Leader's Optimization Problem

For $\mathbb{BM}$, the first order derivation of $U_{\mathbb{BM}}$ with respect to β can be expressed as follows,

$$\frac{\partial U_{\mathbb{BM}}}{\partial\beta} = \Gamma + \beta\sum_{j=1}^{N}\frac{\partial\Gamma}{\partial\gamma_j}\frac{\partial\gamma_j}{\partial\beta} - \frac{2cD}{M}\Lambda\sum_{j=1}^{N}\frac{\partial\Lambda}{\partial\gamma_j}\frac{\partial\gamma_j}{\partial\beta}, \tag{18}$$

and then the second order derivation is,

$$\begin{aligned} \frac{\partial^2 U_{\mathbb{BM}}}{\left(\partial\beta\right)^2} &= 2\sum_{j=1}^{N}\frac{\partial\Gamma}{\partial\gamma_j}\frac{\partial\gamma_j}{\partial\beta} - \frac{2cD}{M}\left\{\left(\sum_{j=1}^{N}\frac{\partial\Lambda}{\partial\gamma_j}\frac{\partial\gamma_j}{\partial\beta}\right)^2\right. \\ &\quad \left. + \Lambda\sum_{j=1}^{N}\left[\frac{\partial^2\Lambda}{\left(\partial\gamma_j\right)^2}\left(\frac{\partial\gamma_j}{\partial\beta}\right)^2 + \frac{\partial\Lambda}{\partial\gamma_j}\frac{\partial^2\gamma_j}{\left(\partial\beta\right)^2}\right]\right\} \\ &\quad + \beta\sum_{j=1}^{N}\left[\frac{\partial^2\Gamma}{\left(\partial\gamma_j\right)^2}\left(\frac{\partial\gamma_j}{\partial\beta}\right)^2 + \frac{\partial\Gamma}{\partial\gamma_j}\frac{\partial^2\gamma_j}{\left(\partial\beta\right)^2}\right]. \end{aligned} \tag{19}$$

According to **Appendix A**, we can know that $\frac{\partial^2 U_{\mathbb{BM}}}{(\partial\beta)^2} < 0$, which indicates that $U_{\mathbb{BM}}$ is a concave function of β. Therefore, the optimal price β^* set by $\mathbb{BM}$ can be gotten when $U_{\mathbb{BM}}(\beta^*)$ is the extreme value. Let $\frac{\partial U_{\mathbb{BM}}}{\partial\beta} = 0$, then

$$\Gamma + \beta\sum_{j=1}^{N}\frac{\partial\Gamma}{\partial\gamma_j}\frac{\partial\gamma_j}{\partial\beta} - \frac{2cD}{M}\Lambda\sum_{j=1}^{N}\frac{\partial\Lambda}{\partial\gamma_j}\frac{\partial\gamma_j}{\partial\beta} = 0. \tag{20}$$

In a distributed environment, it is hard to get the closed form of β^* from (16) and (20), because blockchain miners do not know the utility function of blockchain users. However, with the classic iterative method [23], according to (21) where

$$\frac{\partial^2 U_{\mathbb{BM}}}{\left(\partial\beta\right)^2} = -\sum_{j=1}^{N}\frac{2\,\alpha_j K u_j}{\left(\beta+Ku_j\right)^3} - \frac{2cD}{M}\left\{\left(\sum_{j=1}^{N}\frac{1}{\sigma}\frac{K\alpha_j}{\left(\beta+Ku_j\right)^2}\right)^2 + \Lambda\sum_{j=1}^{N}\left[\frac{2\,A_M\,K}{M\,\mu\sigma^3}\frac{K^2\alpha_j^2}{\left(\beta+Ku_j\right)^4} + \frac{1}{\sigma}\frac{2\,K\alpha_j}{\left(\beta+Ku_j\right)^3}\right]\right\} < 0. \tag{17}$$

Algorithm 2 Iterative Updating Function

```
Input: $c$, $D$, $\mu$, $\lambda_{max}$, M, N, K, $\alpha_j, j \in \mathbb{N}$ ($\mathbb{N} = \{1, \ldots, N\}$
Output: the converged $\beta$ and $\gamma_j, j \in \mathbb{N}$ ($\mathbb{N} = \{1, \ldots, N\}$
Get $\Gamma_{max}$ using (8)
Let the price $\beta^t = c + \Delta$ where $\Delta$ is an extremely small positive number
Let $U_{\mathbb{BM}}^{distance} = abs(U_{\mathbb{BM}}^{t+1} - U_{\mathbb{BM}}^{t})$ and $\Delta_{U_{\mathbb{BM}}}$ is the convergence precision of $U_{\mathbb{BM}}$
Set iteration number index $t = 0$
Initialize $converged \leftarrow false$, $U_{\mathbb{BM}}^{distance} > \Delta_{U_{\mathbb{BM}}}$
while $not\ converged$ do
    for each blockchain user $BU_j, j \in \mathbb{N}$ ($\mathbb{N} = \{1, \ldots, N\}$ do
        Set the Lagrangian multiplier $u_j^t = 0$
        Get $\beta^t$ and $\sum_{i \neq j} \gamma_i$ according to blockchain state, update the corresponding transaction rate $\gamma_j^t$ using (16)
        if $\gamma_j^t > K\Gamma_{max} - \sum_{i \neq j} \gamma_i$ then
            Get the right $u_j^t$ according to (22)
            $\gamma_j^t = K\Gamma_{max} - \sum_{i \neq j} \gamma_i$
        end if
    end for
    Get $U_{\mathbb{BM}}^t$ based on equation (11)
    For blockchain miners:
      if $U_{\mathbb{BM}}^{distance} < \Delta_{U_{\mathbb{BM}}}$ then
        $converged \leftarrow true$
        $\beta^* = \beta^t$ and $\gamma_j^* = \gamma_j^t, j \in \mathbb{N}$ ($\mathbb{N} = \{1, \ldots, N\}$
      end if
      Get $\gamma_j^t$ and $u_j^t$ from blockchain user $BU_j, j \in \mathbb{N}$ ($\mathbb{N} = \{1, \ldots, N\}$, update the block price $\beta^{t+1}$ according to (23)
    $t = t + 1$
end while
return $\beta^*$ and $\gamma_j^*, j \in \mathbb{N}$ ($\mathbb{N} = \{1, \ldots, N\}$)
```

$h(\cdot)$ means that β^* is a function of $\frac{\partial \gamma_j^*}{\partial \beta}$ and γ_j^*, the optimal β^* can be worked out iteratively, and concrete progress will be represented in the following subsection.

$$\beta^* = h\left(\frac{\partial \gamma_j^*}{\partial \beta}, \gamma_j^*\right) = \frac{\frac{2cD}{M}\Lambda \sum\limits_{j=1}^{N} \frac{\partial \Lambda}{\partial \gamma_j}\Big|_{\gamma_j = {\gamma_j}^*} \cdot \frac{\partial {\gamma_j}^*}{\partial \beta} - \Gamma}{\sum\limits_{j=1}^{N} \frac{\partial \Gamma}{\partial \gamma_j}\Big|_{\gamma_j = {\gamma_j}^*} \cdot \frac{\partial {\gamma_j}^*}{\partial \beta}}. \tag{21}$$

C. Distributed Iterative Updating Algorithm

Combined with the above analysis, an updating scheme in the distributed environment could be designed to achieve the optimal solution. In the proposed Stackelberg game, based on the computing cost c, β firstly is assigned an appropriate value which satisfies $\frac{\partial U_{\mathbb{BM}}(\beta)}{\partial \beta} > 0$. Then according to (16), after receiving the price β^t (t denote the iteration index) set by blockchain miners, the corresponding γ_j^t can be obtained at blockchain user BU_j. Before sending it as game information to blockchain miners for price updating, it is necessary to check whether $u_j^t = 0$ and $\gamma_j = \gamma_j^t$ meet the KKT conditions. When $\gamma_j^t > K\Gamma_{max} - \sum_{i \neq j} \gamma_i$ which tells that the constraint $\gamma_j + \sum_{i \neq j} \gamma_i \leq K\Gamma_{max}$ is active. Then, in this condition, $\gamma_j^t = K\Gamma_{max} - \sum_{i \neq j} \gamma_i$, and u^t can be gotten according to (22),

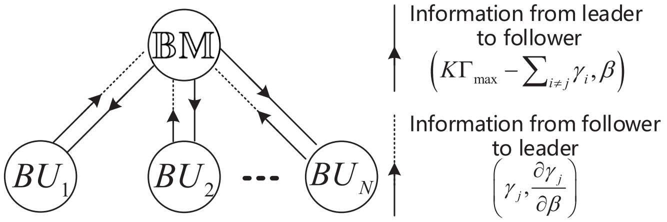

BM
BU1
BU2
BUN
Information from leader to follower
$\left(K\Gamma_{\max} - \sum_{i \neq j} \gamma_i, \beta\right)$
Information from follower to leader
$\left(\gamma_j, \frac{\partial \gamma_j}{\partial \beta}\right)$

Fig. 5. Information exchange between $\mathbb{BM}$ and $BU_j, j \in \mathbb{N}$.

$$u_j^t = \frac{\alpha_j}{K\Gamma_{\max} + 1 - \sum_{i \neq j} \gamma_i} - \frac{\beta^t}{K}. \tag{22}$$

Next, from information γ_j^t and $\frac{\partial \gamma_j}{\partial \beta}\Big|_{\left(\beta^t, u_j^t\right)}$ sent by the blockchain user BU_j, based on (20), (21) and **Appendix A**, blockchain miners can update their price by using the updating function (23) as follows,

$$\beta^{t+1} = \frac{\sum_{j=1}^{N} \gamma_j^t}{K \sum_{j=1}^{N} \frac{\alpha_j}{\left(\beta^t + K u_j^t\right)^2}} + \frac{\mu c D \left(K - \sigma^t\right)}{A_M \sigma^t}, \tag{23}$$

where $\sigma^t = \sqrt{\frac{\left(MK\mu - 4A\sum_{j=1}^{N} \gamma_j^t\right)K}{M\mu}}$.

After that, β^{t+1} would be sent to blockchain users for the next round updating. Repeating the process until both the β and γ_j converge to a unique fixed value finally, the utilities of both the users and miners cannot increase any more. Finally, the optimal solution is achieved. To better elucidate the above, the detail of the process is illustrated in **Algorithm 2** and information exchange phases is shown in Fig. 5.

D. Convergence of the Iterative Updating Function

Combined with (16) and (23), the updating function of β can be re-arranged as below,

$$\beta^{t+1} = \frac{K \sum_{j=1}^{N} \frac{\alpha_j}{\beta^t + K u_j^t} - N}{K \sum_{j=1}^{N} \frac{\alpha_j}{\left(\beta^t + K u_j^t\right)^2}} + \frac{\mu c D K}{A_M \sigma\left(\beta^t\right)} - \frac{\mu c D}{A_M}, \tag{24}$$

where $\sigma\left(\beta^t\right) = \sqrt{\frac{\left(MK\mu + 4AN - 4AT\sum_{j=1}^{N} \frac{\alpha_j}{\beta^t + K u_j^t}\right)K}{M\mu}}$ and

$$u_j^t = \begin{cases} 0, \quad \gamma_j^t + \sum_{i \neq j} \gamma_i \leq K\Gamma_{\max}, \\ \dfrac{\alpha_j}{K\Gamma_{\max} + 1 - \sum_{i \neq j} \gamma_i} - \dfrac{\beta^t}{K}, \\ \quad \gamma_j^t + \sum_{i \neq j} \gamma_i > K\Gamma_{\max}. \end{cases} \tag{25}$$

Obviously, the price updating function with constraints is also convergent if that with no constraint is convergent. Let $u_j^t = 0$ for all $t \in \mathbb{R}$ and $j \in \mathbb{N}, (\mathbb{N} = \{1, \ldots, N\})$, which means that the constraint is inactive, then there exists,

$$I(\beta) = \frac{\beta K \alpha_{all} - N\beta^2}{K\alpha_{all}} + \frac{\mu c D K}{A_M \sigma(\beta)} - \frac{\mu c D}{A_M}, \tag{26}$$

where $\sigma(\beta) = \sqrt{\frac{(MK\mu+4AN)K}{M\mu} - \frac{4AK^2\alpha_{all}}{M\mu\beta}}$ and $\alpha_{all} = \sum_{j=1}^{N}\alpha_j$. Therefore, this problem can be solved by demonstrating the convergence of the iteration in (26).

From [29], the standard function defined in Definition 1 will converge to a unique fixed point.

Definition 1: A function $I(\beta)$ is standard if for all $\beta \geq 0$, the following properties are satisfied [29]:

- *Positivity: $I(\beta) > 0$,*
- *Monotonicity: If $\beta \geq \beta'$, then $I(\beta) \geq I(\beta')$,*
- *Scalability: For all $\alpha > 1$, $\alpha I(\beta) > I(\alpha\beta)$.*

Then, the convergence of the iteration in (26) can be demonstrated by proving $I(\beta)$ is a standard function [29] according to **Appendix B**. After the price updating is proved to be converged, according to (16), required transaction rates of BUs are also converged.

E. Stackelberg Equilibrium Solution

To prove the optimal solution analysed above is also the SE solution in which any participant has no motivation to deviate, the SE in this model can be stated in Definition 2.

Definition 2: When β is fixed, if γ_j^* satisfies $L_{BU_j}(\gamma_j^*, u_j^*) \geq L_{BU_j}(\gamma_j, u_j)$ for all $j \in \mathbb{N}$ where $\mathbb{N} = \{1, \ldots, N\}$, and when $\{\gamma_j, \forall j \in \mathbb{N}\}$ is fixed, if β^* satisfied $U_{\mathbb{BM}}(\beta^*) \geq U_{\mathbb{BM}}(\beta)$, the Strategy Profile $(\beta^*, \{\gamma_j^*, \forall j \in \mathbb{N}\})$ is a SE in the proposed game.

Then the conclusion that $(\beta^*, \{\gamma_j^*, \forall j \in \mathbb{N}\})$ is equal to the SE solution $(\beta^{SE}, \{\gamma_j^{SE}, \forall j \in \mathbb{N}\})$ can be drawn according to **Appendix C**.

V. Security Analysis

In this section, double-spending attack and selfish mining attack are introduced and modeled. As two of the most typical attacks in blockchain, they would contribute to different damages to this system. In double-spending attack, an attacker, who aims at taking back money spent at the merchant recently, would try to generate a parasite chain to roll back its transactions after the handover of goods [31], thus leading to a financial loss in this attacked merchant. In selfish mining attack, with the purpose of obtaining more revenue, an attacker would keep its discovered blocks private according to certain strategies. Consequently, honest blockchain miners would continue to mine on the shorter public branch of the blockchain [32], thus the aggregate block completion rate would be reduced.

Let Λ_m denotes block completion rate in the malicious blockchain miner, no matter which attack it attempts. Based on the analysis in Section III, the effective block completion rate of honest blockchain miners is $\Gamma = \frac{\sum_{j=1}^{N}\gamma_j}{K}$. With n confirmations, which means that the transaction is deemed to be valid after it has been buried under n blocks, confirmation delay is $T_{delay} = \frac{n}{\Gamma} = \frac{Kn}{\sum_{j=1}^{N}\gamma_j}$ hours.

Due to the design that each block's PoW problem is different, based on hash of its previous block [4], mining events for different blocks are independent.

$$P\{\text{the newest is mined} \mid \text{previous block is mined}\} = P\{\text{the newest is mined}\}. \quad (27)$$

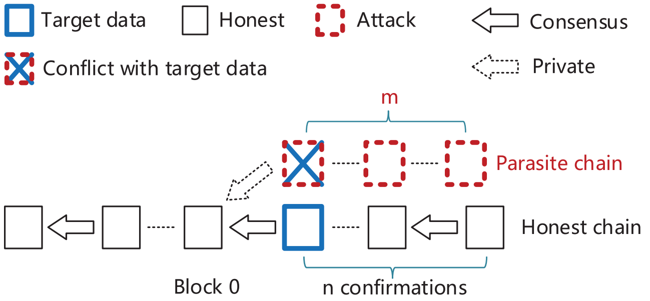


Fig. 6. Double-spending attack.

Hence, according to [33], probabilities that the newest block is mined by honest blockchain miners and the malicious can be obtained as follows,

$$P\{\text{the newest is mined by the honest}\} = \frac{\Gamma}{\Gamma + \Lambda_m} = p, \quad (28)$$

$$P\{\text{the newest is mined by the attacker}\} = \frac{\Lambda_m}{\Gamma + \Lambda_m} = q. \quad (29)$$

A. Double-Spending Attack Model

According to LCR adopted in this paper, to launch a successful double-spending attack, the attacker must issue a parasite chain which is longer than the honest chain after n confirmations. As illustrated in Fig. 6, including conflicting data, this branch would be mined privately on **block 0** which is closely prior to block containing the target data.

Considering an attack strategy where the attacker would not try to publish an double-spending attack until it has mined the newest block before the honest, learned from [30] and [31], the probability of a successful attack is

$$P\{\text{attack succeeds}\} = \begin{cases} 1 + \sum_{m=1}^{n} \binom{m+n-2}{m-1} \left(p^{m-1}q^n - p^n q^{m-1}\right), \\ \quad p > q \wedge n \geq 1, \\ 1, \quad (p \leq q \wedge n \geq 1) \vee n = 0. \end{cases} \quad (30)$$

To make $P\{\text{attack succeeds}\}$ smaller than a given threshold $\epsilon \in (0, 1]$, there exists a requirement on confirmations n as follows,

$$n^*(\varepsilon) = \min\{n : P\{\text{attack succeeds}\} \leq \varepsilon\}. \quad (31)$$

Thus, the necessary confirmation delay is $T_{delay} = \frac{Kn^*(\varepsilon)}{\sum_{j=1}^{N}\gamma_j}$.

B. Selfish Mining Attack Model

Same with double-spending attack, selfish mining attack is also launched by intentional forking. However, different purposes lead to different strategies. To earn more transaction fees, the attacker might adopt a strategy that changes dynamically with the state of blockchain. [32] has detailed a selfish mining algorithm, Selfish-Mine, whose insight is to have the honest blockchain miners waste their computational resources

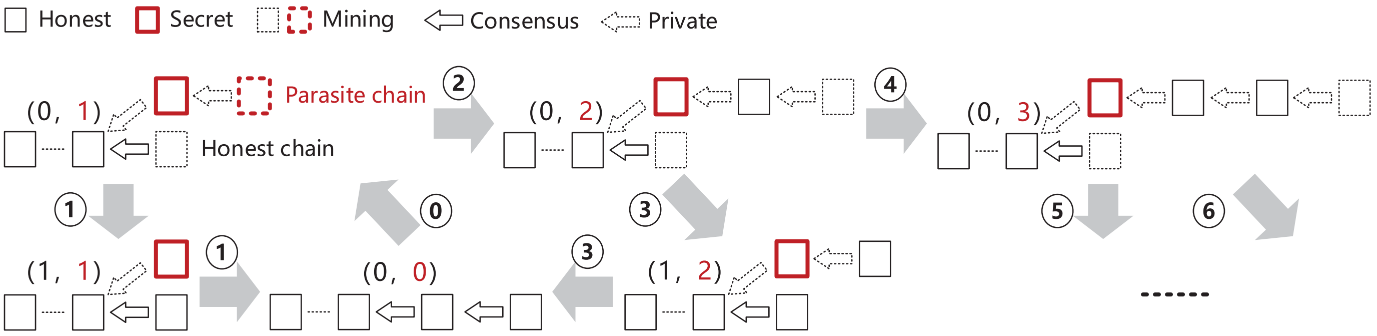


Fig. 7. Influenced consensus process under Selfish-Mine.

on attempting to extend blocks that will not be included in the main branch of the blockchain. However, in addition to helping the attacker obtain an extra revenue, Selfish-Mine would also reduce effective growth rate Γ by influencing the consensus process of blockchain system. Therefore, different from [32] whose focus is on the analysis of expected rewards from Selfish-Mine, the influenced consensus process would be analysed in this paper.

Denoting the consensus length of blockchain at time t as $L(t)$, branches of honest blockchain miners and attacker can be described by $X_h(t) = L(t) + Y_h(t)$ and $X_m(t) = L(t) + Y_m(t)$ respectively. $Y_h(t)$ represents the length of the honest branch beyond the consensus length, and $Y_m(t)$ represents the length of the attacker's kept secret branch beyond the consensus length. (Y_h, Y_m) denotes the separate branches' states in the honest and the attacker, and $(0,0)$ refers to the state where the honest and the attacker reach an agreement on the head block to be extended. As shown in Fig. 7, the influenced consensus process can be illustrated as below,

⓪ $(0,0) \to (0,1)$: Once the attacker mines a block before the honest, attack begins.

① $(0,1) \to (1,1) \to (0,0)$: This case means that the honest mines a block before attacker mines its second block, the attacker would choose to release its branch, resulting in equiheight forks in the consensus chain. After a new block is mined either by the honest or the attacker, an agreement on the head block to be extended would be reached, finally leading to the new consensus which is waited to be broken as ⓪.

② $(0,1) \to (0,2)$: In the case that the attacker mines its second block before the honest mines their first one, the attacker would choose to keep mining along its parasite chain.

③ $(0,2) \to (1,2) \to (0,0)$: The attacker would not issue its parasite chain until the difference has been narrowed to 1, and so would it when the state is $(Y_m - 2, Y_m) \to (Y_m - 1, Y_m) \to (0,0)$. At that time, what attacker has here is a waiting game, just as ⓪.

④ $(0,2) \to (0,3)$: When the difference is greater than 1, the attacker would still mine along its parasite chain, and so would it when the state is (Y_h, Y_m) where $Y_m - 2 > Y_h$.

Thus, a simple Markovian model can be formulated as Fig. 8 to analyse this influenced consensus process described above,

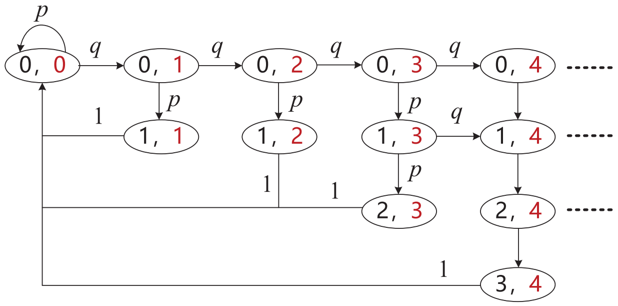


Fig. 8. Simple Markovian model under selfish mining attack.

and its stationary distribution $\pi(y_h, y_m)$ satisfies equations are shown as follows,

$$\pi(0,0) = p\pi(0,0) + \pi(1,1) + \sum\nolimits_{y_m=2}^{\infty} \pi(y_m - 1, y_m), \tag{32}$$

and

$$\pi(1,1) = p\pi(0,1), \tag{33}$$

for $y_m \geq 1$ and $y_h = 0$,

$$\pi(y_h, y_m) = q\pi(y_h, y_m - 1), \tag{34}$$

for $1 \leq y_h \leq y_m - 3$,

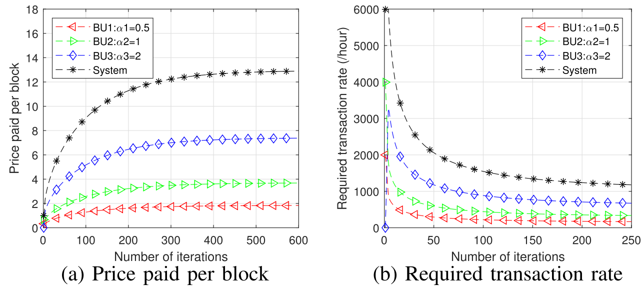
$$\pi(y_h, y_m) = p\pi(y_h - 1, y_m) + q\pi(y_h, y_m - 1), \tag{35}$$

for $y_m - 2 \leq y_h \leq y_m - 1$ and $y_m \geq 3$,

$$\pi(y_h, y_m) = p\pi(y_h - 1, y_m). \tag{36}$$

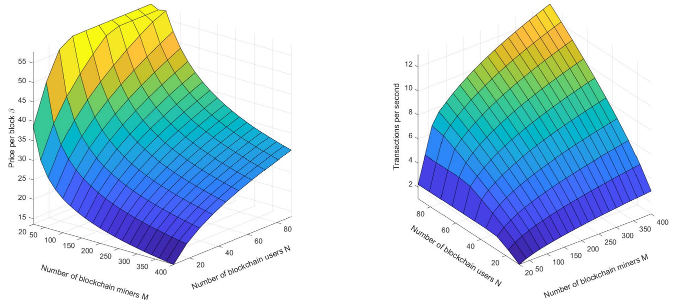
When $(Y_h, Y_m) = (0,0)$, the attack is not launched since the attacker has not mined a block ahead of the honest, hence the practical effective growth rate of blockchain is supported by honest blockchain miners. When $(Y_h, Y_m) \neq (0,0)$, selfish mining attack is launched and computational power of the honest makes no effort. In that case, the practical effective growth rate of blockchain is provided by attacker. Then, calculated from the stationary distribution $\pi(y_h, y_m)$, which can be obtained by setting truncation level T in **Appendix D**, there exists a conclusion,

$$\Gamma' = \pi(0,0)\Gamma + [1 - \pi(0,0)]\Lambda_m. \tag{37}$$



(a) Price paid per block (b) Required transaction rate

Fig. 9. Price paid per block vs. required transaction rate.



(a) Price per block (b) Overall transaction rate

Fig. 10. Performance in different system states.

VI. Numerical Results and Discussions

In this section, the effectiveness and security of the proposed offloading strategy are illustrated. Firstly, convergence of the iterative updating process is showed. Secondly, in terms of price per block β and overall effective transaction rate $K\Gamma$, extensive experiments are numerically conducted. Then, as to double spending attack, the necessary confirmations n and corresponding confirmation delay T_{delay} are examined. Finally, the reduced transaction rate $K\Gamma'$ and corresponding T_{delay} under selfish mining attack are also evaluated.

A. Simulation Settings

According to [28], it can be learned that the average communication delay in Bitcoin is 12.6s and its effective block completion rate is $6/h$. Hence, the overall communication interval parameter can be set $\mu = 285/h$. Combined with Bitcoin's throughput which is generally limited to 7 transactions per second [3], we can get that 1 block is generated and a maximum of 4000 transactions can be approximately written into blockchain every 10 minutes. Thus, the number of transactions in each block K can be set to 4000. Let $cD = 1$ which can also be set to other appropriate positive values. The max block completion rate of single device is fixed $\lambda_{max} = 0.1/h$, showing that each blockchain miner can generate a maximum of one block every ten hours. Let the malicious blockchain miner's block completion rate $\Lambda_m = 1/h$. All the numerical results are obtained using MATLAB.

B. Convergence of the Proposed Game

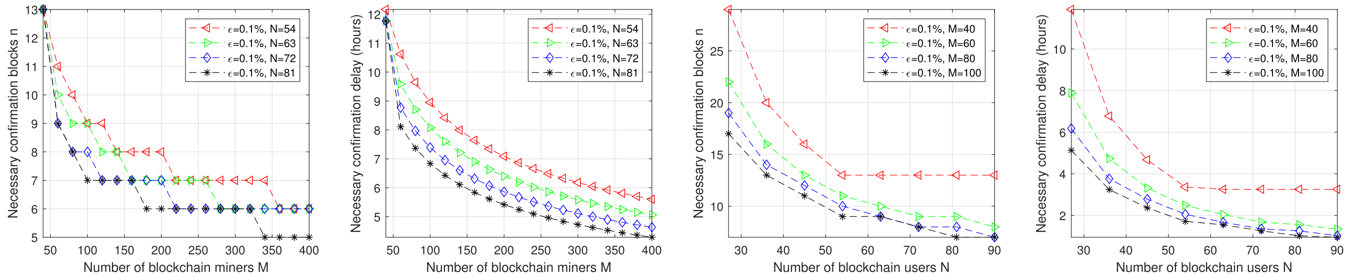
In this experiment, let the number of blockchain miners be $M = 15$ and the number of blockchain users be $N = 3$ with $[\alpha_1, \alpha_2, \alpha_3] = [0.5, 1, 2]$. For better illustration, Fig. 9 also shows the iterations of price paid per block by blockchain users and the overall required transaction rate. From Fig. 9 (a), it is easy to see that as the number of iteration goes up, the price increases and finally converges to a stable value, which is the optimal price paid per block. In Fig. 9 (b), opposite to the price's increasing trend, the required transaction rate decreases with each iteration and finally converges to a stable value. This correlation is consistent with the monotone decreasing property of $\gamma_j^* = \frac{K\alpha_j}{\beta + Ku_j^*} - 1$ with respect to β. Seen from Fig. 9 (b), with higher weight factor α_j, the blockchain user's required transaction rate would also be higher. Meanwhile, higher required transaction rate also brings higher price paid per block and that conclusion can be obtained by comparing the three curves of blockchain users in Fig. 9 (a).

C. Price and Effective Transaction Rate

With $\alpha_j = 4$ where $j \in \mathbb{N}$, Fig. 10 shows the performance varying the number of blockchain users and blockchain miners. The variation tendency of price per block β is showed in Fig. 10 (a) and the overall transaction rate is showed in Fig. 10(b). In Fig. 10 (a), with increasing numbers of blockchain users, it is not difficult to see that β increases more and more slowly. For example, when $M = 20$, β would eventually approach to a constant value. This is because the affordable overall transaction rate of $\mathbb{BM}$ would gradually meet its upper bound with increasing N, as is showed by the curve with $M = 20$ in Fig. 10 (b). Stable transaction rates with N lead to stable β with N. When more and more blockchain miners participate in this system, it is easy to see that β would decrease more and more slowly in Fig. 10 (a). However, expressed as the equation (8), when $\lambda_{max} < \frac{\mu}{2A}$, Γ_{max} will increase with M, and when $\lambda_{max} \geq \frac{\mu}{2A}$, $\Gamma_{max} = \frac{M\mu}{4A}$ is also an increasing function of M. Γ_{max} is not going to reach a stable value with increasing M. Economically, high supply would bring about low prices. Therefore, in Fig. 10 (a), β would not converge to a fixed value with M. Correspondingly, as the curves with constant N in Fig. 10 (b), the overall required transaction rate would get higher would not converge to a fixed value with M as well.

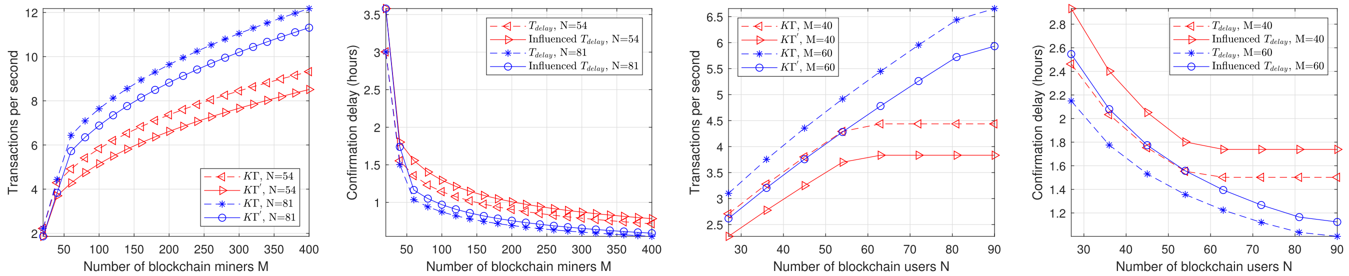
D. Double-Spending Attack

In this experiment, let $P\{\text{attack succeeds}\} \leq \epsilon = 0.001$. Under the same security level $1 - \epsilon = 99.9\%$, Fig. 11 shows the impact of double-spending attack varying the number of blockchain users and blockchain miners. In Fig. 11 (a), with increasing number of blockchain miners, the necessary confirmations n goes down step by step. Though there are cases where the same n exists for different N and M, for example, on the curve $N = 81$ of Fig. 11 (a), n at $M = 200$ is same with that at $M = 300$, meaning higher overall transaction rate, more blockchain miners would still bring about lower T_{delay}, as is showed in Fig. 11 (b). Similar properties can also be found in Fig. 11 (c) and (d). However, when the overall transaction rate meets its upper bound, more blockchain users would not contribute to less confirmations n, as is showed by the curve with $M = 40$ in Fig. 11 (c) As a result, T_{dealy} would



(a) n vs. M (b) T_{delay} vs. M (c) n vs. N (d) T_{delay} vs. N

Fig. 11. Impact of double-spending attack.



(a) Overall transaction rate vs. M (b) T_{delay} vs. M (c) Overall transaction rate vs. N (d) T_{delay} vs. N

Fig. 12. Impact of selfish mining attack.

no longer go down after N reaches beyond 60 in $M = 40$ of Fig. 11 (d). In summary, while maintaining the high security level, more blockchain miners or more blockchain users within the transaction load capacity, is conducive to lower T_{dealy}.

E. Selfish Mining Attack

In this experiment, let truncation level $T = 10$ and the confirmations $n = 6$. Selfish mining would not result in data tampering, however, when maintaining the high security level determined by n, it would affect the overall transaction rate, and then affect $T_{dealy} = n/\Gamma$, which is closely associated with blockchain users' security experience. Fig. 7 shows the impact of selfish mining attack varying the number of blockchain users and blockchain miners. As explained in Fig. 10 (b), the overall transaction rate would not reach a stable value with increasing M. Correspondingly, $K\Gamma'$ and the influenced transaction rate $K\Gamma'$ exhibit the same property in Fig. 7 (a). It is easy to see that $K\Gamma' < K\Gamma$ under the same system state, leading to the result that influenced T_{delay} is greater than the uninfluenced in Fig. 7 (b). Meanwhile, in Fig. 7 (c) and (d), $K\Gamma'$ and influenced T_{delay} experience a similar correspondence about $K\Gamma$ and T_{delay} to that of Fig. 7(a) and (b).

VII. Related Work

As an appealing decentralized technology, blockchain has been used in wired networks like Bitcoin, Nxt, Ethereum and so on [3]. To extend blockchain technology to wireless networks, many researches have been done, which can be divided into two categories. One is the lightweight blockchain mechanism for resource-constrained IoT devices, represented by the Lightweight Scalable blockchain proposed in [34], and the other is to pass the obstacle of IoT devices' constrained resources with the help of MEC. [35] proposes a MEC-enabled framework for blockchain-based video streaming, and to incentivize more users to participate, where resource allocation, offloading scheduling and adaptive block size are jointly considered. To encourage mobile devices to share their own resource, [23] designs a game theoretic approach based incentive mechanism. Although many studies have been done on the combination of MEC and blockchain, they have ignored the mismatch problem between the centralized architecture of MEC and the decentralized architecture of blockchain system.

Contrary to increasing numbers of transactions, existing blockchain technologies' low throughput and high delay has significantly hindered it from being extensively put into use. From the aspects of block generation process, data storage and data transmission, [36] has summarized the existing blockchain scalability technologies. As the first blockchain system, Bitcoin has clarified the corresponding operation logic, which has been used for reference by subsequent blockchain systems. In view of the long freezing time brought by Bitcoin's block generation process, [14] presents an idea that decoupling leader election from transaction serialization by introducing key blocks and micro blocks in Bitcoin-NG. Based on this idea, ByzCoin is proposed in [37] to solve the problem in Bitcoin-NG which is retained from Bitcoin, i.e, temporary forks and leader's maliciousness. Combining PoW with Practical Byzantine Fault Tolerance (PBFT) algorithm, leader's maliciousness can be restrained in time. Further, to reduce per-round communication complexity in the consensus group, collective signing is employed to scale PBFT protocols to large groups. In this paper, without changing the block structure and compromising the decentralized nature, we propose a different solution from the perspective of the way blocks are associated with transactions.

Largely known for its security and privacy, more and more importance has been attached to blockchain. However, there still exist many vulnerabilities which would lead to various security threats, among which double-spending and selfish mining attacks are the two most typical threats. [31] uses stochastic processes to model the double-spending attack on PoW-based blockchains and analyzes the probabilities of success. [30] extends the analysis of the double-spending attack in [31] to DAG-based blockchains. Because the model developed in [30] and [31] is also applicable to this paper, we use the relevant conclusions directly in a short space rather than deducing them again in this paper. As to selfish mining, [32] describes a strategy that the malicious collusion can use to obtain more mining profit. In the context of this strategy, [28] studies the evolution of Bitcoin blockchain in the presence of communication delay. However, the effect of this strategy on the effective growth rate of blockchain systems is overlooked. This neglect point is closely related to the user experience of blockchain users, and with this different analysis objective, the model in this paper is different.

VIII. Conclusion and Future Work

In this paper, as to the situation where resource constrained IoT devices do not have effective ability to become a full node in WBN, we firstly propose a framework where IoT devices can upload transactions into blockchain system through edge servers. Further, at the side of blockchain miners, a block generation process is designed to satisfy frequent leader election and high block space utilization simultaneously. Then, to coordinate both the needs of the blockchain users and blockchain miners, a single-leader-multiple-followers Stackelberg game is formulated to model the interaction between them. Meanwhile, in order to match the distributed characteristics of blockchain system, an iterative updating function is proposed to achieve the SE in a distributed manner. Next, the impacts of two typical attacks in blockchain are analysed where stochastic model is used to examine the probability of a successful double-spending attack and markovian chain is used to model the consensus process under selfish mining attack. Finally, besides validating the convergence and effectiveness of the iterative updating function, numerical results show the influence trends of the attacks on blockchain system.

In future work, focusing on blockchain scalability technologies, the consensus mechanism, sharding scheme and so on would be studied. Furthermore, a scalable blockchain system would try to be proposed. To analyse its performance, a simple software prototype would be implemented.

Appendix A

To prove that $U_{\mathbb{BM}}$ has an extreme value, its convexity should be analysed. Firstly, from equation (16), with the corresponding β, γ_j^* can be calculated. Then, its first order derivation and second order derivation to β can be gotten

$$\frac{\partial\gamma_j}{\partial\beta} = -\frac{K\alpha_j}{\left(\beta+Ku_j\right)^2}, \qquad \frac{\partial^2\gamma_j}{\left(\partial\beta\right)^2} = \frac{2K\alpha_j}{\left(\beta+Ku_j\right)^3}. \tag{38}$$

According to equation (2) and (4), the first and second derivation of Γ and Λ are shown as follows,

$$\frac{\partial\Gamma}{\partial\gamma_j} = \frac{1}{K}, \qquad \frac{\partial^2\Gamma}{\left(\partial\gamma_j\right)^2} = 0. \tag{39}$$

$$\frac{\partial\Lambda}{\partial\gamma_j} = \frac{1}{\sigma}, \qquad \frac{\partial^2\Lambda}{\left(\partial\gamma_j\right)^2} = \frac{2AK}{M\mu\sigma^3}. \tag{40}$$

where $\sigma = \sqrt{\frac{K\left(MK\mu-4A\sum_{j=1}^{N}\gamma_j\right)}{M\mu}}$. As a result, we can know that $\frac{\partial^2 U_{\mathbb{BM}}}{\left(\partial\beta\right)^2} < 0$ according to (17), as shown at the bottom page 6 which indicates that $U_{\mathbb{BM}}$ is a concave function of β.

Appendix B

Based on the following proofs, $I\left(\beta\right)$ can be proved to be the standard function.

Proof of Positivity: According to equation (23), $I\left(\beta\right)$ can also be shown as follows where $u_j^t = 0$ for all $t \in \mathbb{R}$ and $j \in \mathbb{N}, \left(\mathbb{N} = \{1, \ldots, N\}\right)$,

$$I\left(\beta\right) = \frac{\sum_{j=1}^{N}\gamma_j^t}{K\sum_{j=1}^{N}\frac{\alpha_j}{\left(\beta+Ku_j^t\right)^2}} + \frac{\mu cD\left(K-\sigma^t\right)}{A\sigma^t}. \tag{41}$$

As $\sigma^t < K$ from equation (5), it's easy to get that $I\left(\beta\right) > 0$. □

Proof of Monotonicity: Comparing $I(\beta)$ and $I(\beta')$ in a basic way with $\beta \geq \beta'$, there exists

$$I\left(\beta\right) - I\left(\beta'\right) = \frac{\left(\beta-\beta'\right)}{K\alpha_{all}}\left[K\alpha_{all} - N\left(\beta+\beta'\right)\right] + \frac{\mu cDK}{A}\left[\frac{1}{\sigma\left(\beta\right)} - \frac{1}{\sigma\left(\beta'\right)}\right]. \tag{42}$$

Learned from [3], Bitcoin's throughput is about 25200 transactions per hour which is far from meeting users' needs. Then without contradicting reality, it is reasonable to assume that the optimal transaction rate γ_j^* measured the number of transactions by per hour is much greater than 0. Therefore, according to equation (16), there exists an inequality that $\gamma_j^* = \frac{K\alpha_j}{\beta+Ku_j^*} - 1 \gg 0$. Hence, under the condition that $u_j^t = 0$ for all $t \in \mathbb{R}$ and $j \in \mathbb{N}, \left(\mathbb{N} = \{1, \ldots, N\}\right)$, it is easy to get that $K\sum_{j=1}^{N}\alpha_j = K\alpha_{all} \gg N\beta$. Then, $K\alpha_{all} > N\left(\beta+\beta'\right)$.

Since $\beta - \beta' > 0$ and $\sigma(\beta)$ is an increasing function of β, $I(\beta) \geq I(\beta')$ can be claimed. □

Proof of Scalability: With $\alpha > 1$, comparing $\alpha I(\beta)$ and $I(\alpha\beta)$ in an element-wise manner, there exists

$$\alpha I\left(\beta\right) - I\left(\alpha\beta\right) = \frac{N\alpha\beta^2}{K\alpha_{all}}\left(\alpha-1\right) + \alpha\left[\frac{\mu cDK}{A\sigma\left(\beta\right)} - \frac{\mu cD}{A}\right] - \left[\frac{\mu cDK}{A\sigma\left(\alpha\beta\right)} - \frac{\mu cD}{A}\right]. \tag{43}$$

Because of the increasing property of function $\sigma(\beta)$, equation (26) can be scaled as follows,

$$\begin{aligned}\alpha I\left(\beta\right)-I\left(\alpha\beta\right)>{}&\frac{N\alpha\beta^{2}}{K\alpha_{all}}\left(\alpha-1\right)+\alpha\left[\frac{\mu cDK}{A\sigma\left(\beta\right)}-\frac{\mu cD}{A}\right]\\&-\left[\frac{\mu cDK}{A\sigma\left(\beta\right)}-\frac{\mu cD}{A}\right]\\={}&\left(\alpha-1\right)\left[\frac{\mu cDK}{A\sigma\left(\beta\right)}-\frac{\mu cD}{A}\right]+\frac{N\alpha\beta^{2}}{K\alpha_{all}}\\&\times\left(\alpha-1\right).\end{aligned}\tag{44}$$

As $\sigma^{t} < K$ from equation (5), it is not difficult to get that $\alpha I\left(\beta\right) > I\left(\alpha\beta\right)$. □

Appendix C

Based on the following properties, the optimal solution can be proved to be equal to the SE solution.

Property 1: For BU_j, when β is fixed, γ_j^* is the global optimum which can maximize $L_{BU_j}(\gamma_j, u_j)$.

According to equation (45), where $\frac{\partial^2 U_{BU_j}}{(\partial\gamma_j)^2} < 0$ as analysed in equation (13) and $g(\gamma_j) = -u_j(\sum_{j=1}^{N}\gamma_j - K\Gamma_{max})$, $L_{BU_j}(\gamma_j, u_j)$ is a concave function of γ_j. Satisfying the condition of definition 1, $L_{BU_j}(\gamma_j, u_j)$ can get its maximum when $\gamma_j = \gamma_j^*$. Thus, it is also the SE solution γ_j^{SE}

$$\frac{\partial^{2}L_{BU_j}\left(\gamma_j,u_j\right)}{\left(\partial\gamma_j\right)^{2}}=\frac{\partial^{2}U_{BU_j}}{\left(\partial\gamma_j\right)^{2}}+\frac{\partial^{2}g\left(\gamma_j\right)}{\left(\partial\gamma_j\right)^{2}}<0.\tag{45}$$

Property 2: For blockchain miners, the required transaction rate γ_j of BU_j decreases as the price β increases according to equation (38). This makes sense because when blockchain miners increase their price, blockchain users will have lower requirements for the transaction rate.

Property 3: For blockchain miners, when blockchain users get the desired transaction rate γ_j, β^* is the optimal price which can maximize $U_{\mathbb{BM}}(\beta)$ under the condition that $\frac{\partial^2 U_{\mathbb{BM}}}{(\partial\beta)^2} < 0$ according to equation (17).

Appendix D

Combined with $\sum_{(y_h,y_m)}\pi\left(y_h,y_m\right) = 1$, $\pi\left(0,0\right)$ can be calculated by translating $\pi\left(y_h,y_m\right)$ into an expression about $\pi\left(0,0\right)$. From equations (32)-(36), the following regularities can be obtained,

$$\pi\left(1,1\right)=pq\pi\left(0,0\right),\tag{46}$$

for $y_h = 0$,

$$\pi\left(y_h,y_m\right)=q^{y_m}\pi\left(0,0\right),\tag{47}$$

for $1 \le y_h \le y_m - 2$,

$$\pi\left(y_h,y_m\right)=p\sum_{y_m'=y_h+2}^{y_m}q^{y_m-y_m'}\pi\left(y_h-1,y_m'\right),\tag{48}$$

for $y_h = y_m - 1$ and $y_h \ge 1$,

$$\pi\left(y_h,y_m\right)=p\pi\left(y_h-1,y_m\right).\tag{49}$$

However, according to equation (48) and (49), the general formula of $\pi\left(y_h,y_m\right)$ where $1 \le y_h \le y_m - 2$ is so complicated that it is hard to show the closed form of $\pi\left(0,0\right)$ when $y_m \to \infty$. Based on the analysis above, $\pi\left(y_h,y_m\right)$ will become smaller with larger pair $\left(y_h,y_m\right)$. Then without loss of generality, truncating the process at $y_m = T$ $(T \ge 4)$ showed as Fig. 13, will not make too much of an impact on $\pi\left(y_h,y_m\right)$ when $y_m < T$.

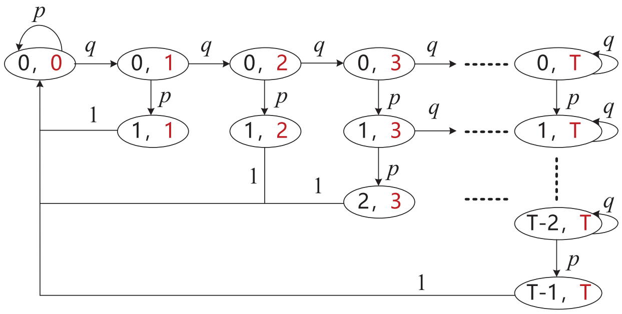


Fig. 13. Simple Markovian model with truncation level $y_m = T$.

With truncation level $y_m = T$, same as equations (46)-(49) for $y_m < T$, when $y_m = T$, there exist different regularities For $y_h = 0$,

$$\pi\left(y_h,T\right)=\frac{q^{T}\pi\left(0,0\right)}{p},\tag{50}$$

for $1 \le y_h \le T - 2$,

$$\pi\left(y_h,T\right)=\frac{q\pi\left(y_h,T-1\right)+p\pi\left(y_h-1,T\right)}{p},\tag{51}$$

for $y_h = y_m - 1$ and $y_h \ge 1$, it is same as $y_m < T$. Hence, based on the analysis above, it can be easily worked out by computing tools such as MATLAB.

References

[1] W. Liu, B. Cao, L. Zhang, M. Peng, and M. Daneshmand, “A distributed game theoretic approach for blockchain-based offloading strategy,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[2] M. Villari, M. Fazio, S. Dustdar, O. Rana, L. Chen, and R. Ranjan, “Software defined membrane: Policy-driven edge and Internet of Things security,” *IEEE Cloud Comput.*, vol. 4, no. 4, pp. 92–99, Jul. 2017.

[3] B. Cao et al., “When Internet of Things meets blockchain: Challenges in distributed consensus,” *IEEE Netw.*, vol. 33, no. 6, pp. 133–139, Nov./Dec. 2019.

[4] S. Nakamoto. (2009). *Bitcoin: A Peer-to-Peer Electronic Cash System.* [Online]. Available: https://bitcoin.org/bitcoin.pdf

[5] H.-N. Dai, Z. Zheng, and Y. Zhang, “Blockchain for Internet of Things: A survey,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8076–8094, Oct. 2019.

[6] H. He, H. Shan, A. Huang, Q. Ye, and W. Zhuang, “Edge-aided computing and transmission scheduling for LTE-U-enabled IoT,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 7881–7896, Dec. 2020.

[7] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, “When mobile blockchain meets edge computing,” *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 33–39, Aug. 2018.

[8] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, “Mobile edge computing: A survey,” *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[9] B. Cao, L. Zhang, Y. Li, D. Feng, and W. Cao, “Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework,” *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 56–62, Mar. 2019.

[10] M. Liu, F. R. Yu, Y. Teng, V. C. Leung, and M. Song, “Computation offloading and content caching in wireless blockchain networks with mobile edge computing,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11008–11021, Jun. 2018.

[11] N. Zhao, H. Wu, and Y. Chen, “Coalition game-based computation resource allocation for wireless blockchain networks,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8507–8518, Oct. 2019.

[12] Y. Jiao, P. Wang, D. Niyato, and K. Suankaewmanee, "Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 9, pp. 1975–1989, Sep. 2019.

[13] S. Guo, Y. Dai, S. Guo, X. Qiu, and F. Qi, "Blockchain meets edge computing: Stackelberg game and double auction based task offloading for mobile blockchain," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5549–5561, May 2020.

[14] I. Eyal, A. E. Gencer, E. G. Sirer, and R. V. Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *Proc. NSDI*, 2016, pp. 45–59.

[15] *Next Generation Radio Access Network (NG-RAN); NG General Aspects and Principles*, document 3GPP TS 38.410 V17.0.0, Release 17, Apr. 2022.

[16] *Next Generation Radio Access Network (NG-RAN); Xn General Aspects and Principles*, document 3GPP TS 38.420 V17.0.0, Release 17, Apr. 2022.

[17] *System Architecture for the 5G System (5GS); Stage 2*, document 3GPP TS 23.501 v17.4.0, Release 17, Mar. 2022.

[18] Y. Sun, L. Zhang, G. Feng, B. Yang, B. Cao, and M. A. Imran, "Blockchain-enabled wireless Internet of Things: Performance analysis and optimal communication node deployment," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5791–5802, Jun. 2019.

[19] N. Papadis, S. Borst, A. Walid, M. Grissa, and L. Tassiulas, "Stochastic models and wide-area network measurements for blockchain design and analysis," in *Proc. IEEE Conf. Comput. Commun.*, Honolulu, HI, USA, Apr. 2018, pp. 2546–2554.

[20] N. Dimitri, "Bitcoin mining as a contest," *Ledger*, vol. 2, pp. 31–37, Sep. 2017. http://doi.org/10.5195/ledger.2017.96

[21] B. Biais, C. Bisière, M. Bouvard, and C. Casamatta, "The blockchain folk theorem," *Rev. Financial Stud.*, vol. 32, no. 5, pp. 1662–1715, May 2019.

[22] BSN Development Association. (2020). *Blockchain-Based Service Network Technical White Paper*. [Online]. Available: https://www.bsnbase.com/sys/file/downLoadTechnicalWhite?type=EN

[23] B. Cao, S. Xia, J. Han, and Y. Li, "A distributed game methodology for crowdsensing in uncertain wireless scenario," *IEEE Trans. Mobile Comput.*, vol. 19, no. 1, pp. 15–28, Jan. 2020.

[24] C. Xu, M. Sheng, V. S. Varma, T. Q. S. Quek, and J. Li, "Wireless service provider selection and bandwidth resource allocation in multi-tier HCNs," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 5108–5124, Dec. 2016.

[25] L. Zhang, B. Cao, Y. Li, M. Peng, and G. Feng, "A multi-stage stochastic programming-based offloading policy for fog enabled IoT-eHealth," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 411–425, Feb. 2021.

[26] H. Hindi, "A tutorial on convex optimization," in *Proc. Amer. Control Conf.*, 2004, pp. 3252–3262.

[27] M. Liu and Y. Liu, "Price-based distributed offloading for mobile-edge computing with computation capacity constraints," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 420–423, Jun. 2018.

[28] 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.

[29] R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341–1348, Sep. 1995.

[30] Y. Li et al., "Direct acyclic graph-based ledger for Internet of Things: Performance and security analysis," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1643–1656, Aug. 2020.

[31] M. Rosenfeld, "Analysis of hashrate-based double spending," 2014, *arXiv:1402.2009*.

[32] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Commun. ACM*, vol. 61, no. 7, pp. 95–102, Jul. 2018.

[33] S. M. Ross, *Introduction to Probability Models*, 11th ed. New York, NY, USA: Academic, 2014.

[34] A. Dorri, "LSB: A lightweight scalable blockchain for IoT security and anonymity," *J. Parallel Distrib. Comput.*, vol. 134, pp. 180–197, Dec. 2019.

[35] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 695–708, Jan. 2019.

[36] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu and Y. Liu, "A survey on the scalability of blockchain systems," *IEEE Netw.*, vol. 33, no. 5, pp. 166–173, Sep. 2019.

[37] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in *Proc. USENIX Secur.*, 2016, pp. 96–279.

Weikang Liu received the B.E. degree in information and communication engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree with the State Key Laboratory of Networking and Switching Technology. His research interests include blockchain and the Internet of Things.

Bin Cao (Senior Member, IEEE) received the Ph.D. degree (Hons.) in communication and information systems from the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China (UESTC), in 2014. From April 2012 to December 2012, he was an International Visitor at the Institute for Infocomm Research (I2R), Singapore. He was a Research Fellow at the National University of Singapore from July 2015 to July 2016. He is currently an Associate Professor with the State Key Laboratory of Network and Switching Technology, Beijing University of Posts and Telecommunications (BUPT). His research interests include blockchain systems, the Internet of Things, and mobile edge computing, and he has extensive publications in IEEE/ACM Transactions on Networking, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Mobile Computing, IEEE Transactions on Multimedia, IEEE Transactions on Vehicular Technology, IEEE Transactions on Industrial Informatics, IEEE Transactions on Could Computing, IEEE Internet of Things Journal, IEEE Sensors Journal, *IEEE Communications Magazine*, IEEE Wireless Communications, and *IEEE Network*, and three of them are ESI Hot/Highly Cited Papers. He served as a TPC member for numerous conferences. He is the Founding Vice Chair of Special Interest Group on Wireless Blockchain Networks in IEEE Cognitive Networks Technical Committee. He received the IEEE Outstanding Leadership Award 2020 and the IEEE Broadcast Technology Society 2021 Best Paper Award. He served as the Symposium Co-Chair for IEEE ICNC 2018 and the Blockchain Workshop Co-Chair for CyberC 2019 and IEEE Blockchain 2020. He is an Associate Editor of IEEE Transactions on Mobile Computing and a Lead Guest Editor of IEEE Internet of Things Journal for Special Issue on "Blockchain-Enabled Internet of Things." He is the Co-Chair of big data track of IEEE GLOBECOM 2022. He also served as a Guest Editor for IEEE Sensors Journal and IEEE Transactions on Industrial Informatics.

Mugen Peng (Fellow, IEEE) received the Ph.D. degree in communication and information systems from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2005. Afterward, he joined BUPT, where he has been a Full Professor with the School of Information and Communication Engineering since 2012. In 2014, he was an Academic Visiting Fellow with Princeton University, Princeton, NJ, USA. He leads a Research Group focusing on wireless transmission and networking technologies with the State Key Laboratory of Networking and Switching Technology, BUPT. He has authored/coauthored over 100 refereed IEEE journal articles and over 300 conference proceeding articles. He was a recipient of the 2018 Heinrich Hertz Prize Paper Award, the 2014 IEEE ComSoc AP Outstanding Young Researcher Award, and the Best Paper Award in the JCN 2016 and IEEE WCNC 2015. He is on the Editorial/Associate Editorial Board of the *IEEE Communications Magazine*, the IEEE Internet of Things Journal, and IEEE Access.