Greedy-Mine: Bitcoin-NG is Not Incentive Compatible

Junjie Hu and Chunxiang Xu

February 19, 2023

Greedy-Mine:Bitcoin-NG is not Incentive Compatible

Junjie Hu and Chunxiang Xu
College of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, SC, CHN
202121080707@std.uestc.edu.cn

***Abstract*— Bitcoin-NG is an extensible blockchain protocol based on the same trust model as Bitcoin. It divides each epoch into one Key-Block and multiple Micro-Blocks, effectively improving transaction processing capacity. Bitcoin-NG a special incentive mechanism (i.e., the transaction fees in each epoch are split to the current and next leader) to maintain its security. The incentive division method of Bitcoin-NG only includes some specific mining attack strategies of adversary, while ignoring the greedy attack strategy. We propose a Greedy-Mine attack strategy and prove that Bitcoin-NG mining is incentive incompatible. we summarize the computing power proportion range required for malicious adversaries to launch Greedy-Mine to obtain excess returns. we make a backward-compatibility progressive modification to Bitcoin-NG protocol that would raise the threshold of propagation factor from zero to 1. Our analytical and simulation results indicate that Bitcoin-NG mining is not incentive compatible, and Bitcoin-NG is vulnerable to Greedy-Mine attack.**

***Index Terms*—Blockchain, Mining Strategy, Incentive Mechanism, Markov Reward Model**

I. Introduction

In 2008, Nakamoto proposed the Bitcoin blockchain protocol, trying to achieve consensus under a permissionless setting [1]. Bitcoin blockchain, based on Proof of Work (PoW), effectively deters sybil attacks [2]. The blockchain can be seen as a decentralized ledger, which is composed of continuous blocks that follow certain rules and are linked through specific cryptographic methods. In Bitcoin blockchain, the first block (which does not reference any other block) is called the Genesis block. Each block is composed of block header and block body. The block header mainly includes Hash of previous, time stamp, etc. The block body includes complete transaction data. The successful applications of blockchain in the financial field [3, 4], the Internet of Things [5, 6, 7, 8], the network security field [9, 10], the public service field [11, 12], the digital copyright field [13, 14], the insurance field [15], which have made blockchain technology widely concerned by all walks of life and rapidly developed. In the process of continuous development of blockchain, its scalability problems are gradually emerging. Compared with the global payment system Visa with an average of 50000 TPS, the current blockchain system, such as Bitcoin with an average of 7 TPS, ETH with an average of 20 TPS [16], and EOS with an average of 3000 TPS, is not enough to meet the needs of modern financial transactions. In the Bitcoin blockchain, Nakamoto has chosen a fairly secure system parameter, namely, the average block output time is 10 minutes and the block size is limited to 1MB. Relevant research shows that modifying the blockchain system parameters (such as increasing the block size limit and the average block output time) can increase TPS to a certain extent, but will reduce the security level of the blockchain system [9, 17]. Therefore, redesigning the consensus protocol at the underlying blockchain has become a research hotspot in recent years.

The design of the new blockchain consensus protocol can be roughly divided into three categories: Block Classification, Parallel Chains, and Directed Acyclic Graph (DAG). In the area of block classification, FruitChain [18], Bitcoin-NG [19] divide blocks into two categories: the main blocks are responsible for choosing the longest chain of consensus protocol, and the micro blocks are responsible for packaging transactions, which can effectively improve the system throughput of blockchains. In parallel chains, OHIE [20] and Prism [21] can improve system throughput while ensuring system security. The design of Monoxide [22] is more complex. From the perspective of academic analysis, it can effectively improve throughput without certain security. And there is a trade-off between scalability and security in Monoxide. In a DAG-based design approach, Inclusive [23] only proposes basic design principles without detailed introduction to complement the protocol. In Spectre [24], transactions can be confirmed in seconds and throughput is increased by orders of magnitude over bitcoin. Phantom [25] uses a greedy algorithm to distinguish blocks mined by honest miners legally from blocks mined by malicious miners that deviate from the DAG mining protocol, and ultimately provides full order on the BlockDAG in a uniform manner by all honest nodes to meet the specific requirements for ledger timeline in smart contracts. In Conflux [26], it improves the performance of the blockchain through reasonable design and optimization of system, while ensuring the security of the blockchain. Conflux has improved the throughput of the blockchain at the consensus level and has reduced the waiting time of block confirmation. Among them, Bitcoin-NG blockchain has received extensive attention from blockchain practitioners.

Bitcoin-NG is an extensible blockchain protocol, which has the same trust model as Bitcoin. Based on the Bitcoin blockchain, it divides blocks into two categories: Key-Blocks and Micro-Blocks. Key-Blocks are responsible for participating in consensus protocol, while Micro-Blocks are responsible for packaging transactions. Bitcoin-NG improves performance by separating consensus protocols and packaging transactions. In the setting of Bitcoin-NG incentive mechanism, Eyal et al. [19] analyzed two possible malicious attacks by adversary, and obtained the division proportion of transaction fees. Jiayuan Yin [27] improved Bitcoin NG Transaction Inclusion Attack in relevant research. It modified the revenue that honest miners obtained in Transaction Inclusion Attack. The above incentive analysis based on Bitcoin-NG only includes the limited attack strategy of the adversary, while ignoring that the adversary may have more complex greedy attacks. On the basis of Bitcoin-NG incentive mechanism, we model the greedy mining strategy through Markov reward process, and comprehensively analyzes the greedy mining behavior of malicious adversaries. Specifically, we have the following three contributions:

- First of all, in Bitcoin-NG incentive mechanism, we introduce the Greedy-Mine strategy, which demonstrates that Bitcoin-NG mining is not incentive compatible.
- Secondly, on the basis of Bitcoin-NG, we summarize the computing power proportion range required for malicious adversaries to launch greedy attacks to obtain excess returns. When the greedy pool has more than 36% of the system computing power, launching a greedy attack is more advantageous than honest mining, and miners with strong computing power have a stronger incentive to adopt greedy mining.
- Finally, we make a backward-compatibility progressive modification to Bitcoin-NG protocol that would raise the threshold of propagation factor from zero to 1.

II. Bitcoin-NG and it's protocol

A. Overview

Bitcoin-NG is an extensible blockchain protocol based on the same trust model as Bitcoin. On the basis of Bitcoin blockchain, it improves the blockchain performance under the Nakamoto consensus by separating consensus protocols and packaging transactions. The time is divided into multiple epochs, and each epoch contains a leader (i.e. blocks in the main chain). The tenure of each leader is about 10 minutes, during which the transactions in the transaction pool are packaged.

B. Key-Block and Micro-Block

Bitcoin-NG divides blocks into two categories: Key-Blocks and Micro- Blocks. Key-Blocks are responsible for consensus agreements, and Micro-Blocks are responsible for packaging transactions.

1) ***Key-Blocks: Consensus Protocol.*** Similar to Bitcoin blockchain, the Key-Block contains reference to the previous block, current GMT time, coin-base transactions for block awards, target value, nonce, and public keys for subsequent micro blocks. Miners must traverse nonces while mining until the PoW Puzzle is successfully solved, making the hash of their Key-Block header smaller than the target, and setting the coin-base transaction for the block reward to output to their account address (it is closely related to the hash of the public key). Each time a miner tries nonce, it can be seen as a Bernonlli trail, which forms a Bernonlli process. Therefore, the process of miners mining Key-Blocks is memoryless. Bitcoin-NG adjusts the difficulty of mining puzzle through changing the target value through GMT time stored in the block header to maintain the average block generation rate so as to ensure the security of the system.

2) ***Micro-Blocks: Packaging Transaction.*** When a miner generates a legitimate Key-Block, it becomes the leader within the current epoch. Leader can package transactions to generate Micro-Blocks at a rate below a predefined maximum rate. The predefined maximum rate for the Micro-Blocks is much higher than the average generation rate for the Key-Blocks. In this setting, the leader packages transactions in the transaction pool to generate micro-blocks within each epoch. The Micro-Block header contains a reference to the previous block, the current GMT time, the hash of its account, and the signature of the Micro-Block header. Micro-blocks have no contribution to consensus protocol and are responsible only for packaging transactions, which is critical to ensuring system consensus through incentives.

C. Protocol

In Bitcoin-NG, Micro-blocks are generated frequently, the current local state of each node, therefore, may be inconsistent, leading to forking. As shown in Figure 1. At this time when the Key-Block 1 is generated, the Micro-Blocks 1' and 2 may not have been received yet. Ultimately, Micro-Blocks 1' and 2' become orphan blocks, where transactions are not executed. Therefore, users who see the Micro-Blocks in the blockchain should wait for a period of network propagation until other Key-Blocks are generated after the Micro-Blocks before the transaction is confirmed.

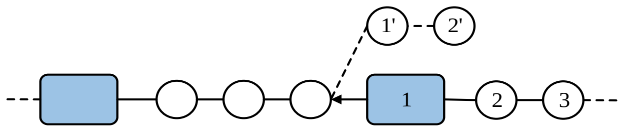

Fig. 1. Forking in Bitcoin-NG

To motivate miners to mine honestly and ensure the security of the system, leaders in each epoch receive two rewards: the one is to get a coin-base reward for each Key-block that is generated. The second is the transaction fees for the Micro-Blocks generated by the leader. The packaging transaction fees should be shared by two adjacent leaders before and after the current epoch. To ensure security and promote honest mining by miners in accordance with the Bitcoin-NG protocol, 40% of these transaction fees are earned by the leader of current epoch and 60% by subsequent leaders. See Figure 2 for details.

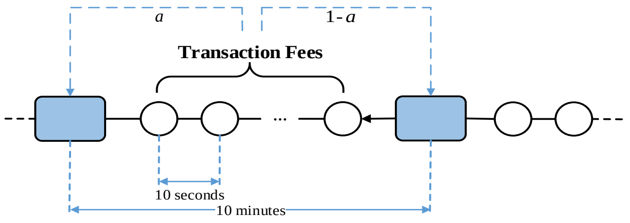

Fig. 2. Transaction Incentives in Bitcoin-NG

III. Incentive Analysis of Bitcoin-NG

A. *Original Incentive Analysis*

Original incentive analysis of Bitcoin-NG contains two types of malicious attack strategies: Transaction Inclusion Attack and Longest Chain Extension Attack.

*1) **Transaction Inclusion Attack:*** We assume that the transaction fees obtained by the leader in the current period accounts for r_leader of the total transaction fees, the remain (1-r_leader) belong to the next leader, and the computing power of adversary accounts for a of the total system computing power. When the adversary generates a Key-Block and packages transactions to generate a series of Micro-Blocks, they may increase their revenue by trying to earn 100% of the transaction fees through selfish mining. If next Key-Block is generated by honest miners, the adversary would wait for the transactions contained in the previously packaged Micro-Block to be packaged by the current leader and package into the Micro-Block. Then adversary try to mine Key-Blocks after the Micro-Block. In this case, adversary can only earn (1-r_leader) transaction fees. See Figure 3 for specific representation. In order to urge all miners to mine honestly according to Bitcoin-NG protocol, the revenue obtained through Transaction Inclusion Attach should be less than the revenue obtained from honest mining. We can get equation 1.

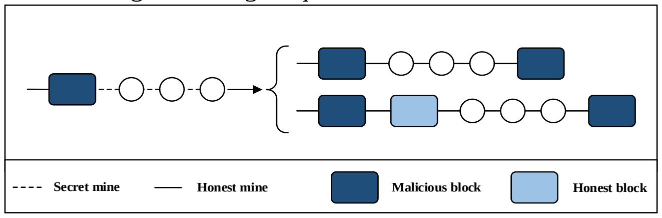

Fig. 3. Transaction Incentives in Bitcoin-NG

$$\overbrace{\boldsymbol{\alpha\cdot 100\%}}^{\substack{\textit{sefish-mine succeed}\\ (\textit{win } 100\%)}} + \overbrace{\boldsymbol{(1-\alpha)\cdot\alpha\cdot(100\% - r_{leader})}}^{\substack{\textit{mine succeed after tx}\\ (\textit{win}(100\%-r_{leader}))}} < \overbrace{\boldsymbol{r_{leader}}}^{\textit{honest mine}} \qquad (1)$$

According to equation 1, we can get $r_{leader} > 1 - \frac{1-\alpha}{1+\alpha-\alpha^2}$. We assume that adversary owned the mining power $\alpha < \frac{1}{4}$, we can get $r_{leader} > 37\%$.

*2) **Longest Chain Extension Attack:*** In order to improve revenue, the adversary can avoid Micro-Blocks, and directly mine to generate a new Key-Block after the Key-Block, and then package these transactions to generate Micro-Blocks. See Figure 4 for details. The revenue that the adversary obtained by Longest Chain Extension Attack must be less than the revenue obtained by honest mining according to Bitcoin-NG protocol. We can get equation 2.

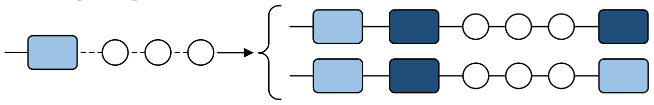

Fig. 4. Longest Chain Extension Attack in Bitcoin-NG

$$\overbrace{\boldsymbol{\alpha^2\cdot 100\%}}^{\textit{Win } 100\%} + \overbrace{\boldsymbol{\alpha\cdot(1-\alpha)\cdot r_{leader}}}^{\textit{Win } r_{leader}} < \overbrace{\boldsymbol{\alpha\cdot(1-r_{leader})}}^{\textit{Honest mine}} \qquad (2)$$

According to equation 2, we can get $r_{leader} < \frac{1-\alpha}{2-\alpha}$. Assume that adversary owned the mining power $\alpha < \frac{1}{4}$, we can get $r_{leader} < 43\%$.

B. *Modified Incentive Analysis*

Jiayuan Yin [27] improved Bitcoin-NG Transaction Inclusion Attack in relevant research. It modified the revenue that honest miners obtained in Transaction Inclusion Attack. In this case, honest miners can not only obtain transaction fees r_{leader}, but also have the probability α of obtaining transaction fees $(1 - r_{leader})$ following the successful generation of new Key-Blocks. Specific description is shown in Figure 5. And we can get equation 3.

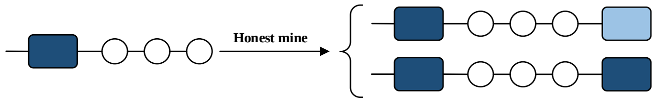

Fig. 5. Modified Transaction Inclusion Attack in Bitcoin-NG

$$\overbrace{\boldsymbol{\alpha\cdot 100\%}}^{\substack{\textit{sefish-mine succeed}\\ (\textit{win } 100\%)}} + \overbrace{\boldsymbol{(1-\alpha)\cdot\alpha\cdot(100\% - r_{leader})}}^{\substack{\textit{mine succeed after tx}\\ (\textit{win}(100\%-r_{leader}))}} < \overbrace{\boldsymbol{r_{leader} + \alpha\cdot(1-r_{leader})}}^{\textit{honest mine}} \qquad (3)$$

According to equation 3, we can get $r_{leader} > \frac{\alpha}{1+\alpha}$. Assume that adversary owned the mining power $\alpha < \frac{1}{4}$, we can get $r_{leader} > 25\%$.

To sum up, we can get $\frac{\alpha}{1+\alpha} < r_{leader} < \frac{1-\alpha}{2-\alpha}$. Assume that adversary owned the mining power $\alpha < \frac{1}{4}$, we can get $20\% < r_{leader} < 43\%$. The incentive parameter $r_{leader} = 40\%$ selected in Bitcoin-NG meets the security requirements.

C. *Defects of The Traditional Incentive Analysis of Bitcoin-NG*

The above incentive analysis based on Bitcoin-NG only includes the limited attack strategies of adversary, while ignoring that the adversary may have more complex greedy attack strategies. For example, in the first stage, the adversary fails to package a whale transaction, but it may reverse the longest chain by generating a new block and attack successfully to obtain whale transaction fees. On the basis of Bitcoin-NG's original incentive analysis above, we defined the greedy attack of adversary, modeled the greedy adversary through the Markov reward process, and analyzed the mining strategies of the greedy adversary comprehensively.

IV. Markov Model of Greedy-Mine Strategy

*1) **Hypothesis:*** We first make some hypotheses about the blockchain system and the adversary before building the model. We define that the proportion of computing power owned by the adversary, called greedy pool, accounts for α of the total computing power of the whole network system. When disagreements arise on the longest legal chain of the blockchain system, the proportion of honest miners mining in malicious branches is γ (γ is also referred to as transmission factor), and the proportion of mining in honest branch is $1 - \gamma$. In the incentive analysis, only the transaction fees are considered, ignoring coin-base reward (this assumption follows Bitcoin-NG incentive analysis).

2) ***Greedy-Mine Strategy:*** In our model, adversary will exploit greedy mining strategy to try to obtain excess returns. For the sake of simplicity and generality, we assume that miners are divided into two categories: one is the minority mining pool following the greedy mining strategy, and the other is the majority pool following the honest mining strategy. It is not significant whether honest miners are a single pool, a series of pools or individual miners.

The key idea of the greedy mining strategy is that the greedy pool tries to compete with honest pool for the longest legal chain, forcing honest pool to waste their computing power on the non-longest legal chain, and finally the greedy pool gets all the transaction fees in an epoch.

Greedy pool generates a new block selectively in the current blockchain, eventually making the greedy branch the longest legal chain, so as to achieve the purpose of obtaining all transaction fees in a certain epoch. Generally speaking, in some epoch, there is a whale transaction (whale transaction refers to the transaction involving very high transaction fees, and block rewards can be neglected compared with whale transaction). Greedy pool will try to generate Key-Blocks before the Micro-Blocks containing whale transactions and generate the Micro-Blocks containing the whale transactions, even if the whale transactions have been packaged into Micro-Blocks by other honest pool and generated. Greedy pool will attempt to make their branch the longest legal chain, wasting the computing power of honest pool, so as to gain disproportionate reward.

With the above intuition, we can define the greedy mining strategy. The strategy is driven by the mining events of greedy pool or honest pool. The decision of greedy pool is determined by the specific state of whale transactions in the current blockchain. We divide $state_{tx}$ into three categories: $state_{tx} = 0$ refers that whale transaction has not been packaged. $state_{tx} = 1$ refers that whale transaction has been packaged by greedy pool. And $state_{tx} = 2$ is that whale transaction has been packaged by honest pool. The initialization of Greedy-Mine is described in the following algorithm 1.

```
Algorithm 1. Initialization of Greedy-Mine
1:  On Init
2:      $state_{tx} = 0$
3:      $length(honest\ branch) = 0$
4:      $length(adversarial\ branch) = 0$
5:      Mine at the head of longest branch.
```

Fig. 6. Initialization of Greedy-Mine

When greedy pool finds a Key-Block, if whale transaction has not been packaged at this time, it releases the Key-Block and the Micro-Block containing whale transactions, sets $state_{tx} = 1$, adds one to the length of adversarial branch. If whale transaction has been packaged by greedy pool and the length of adversarial branch is not shorter than the length of honest branch at this time, it will generate the Key-Block and add one to the length of the branch where the greedy miner is located. The competition ends and adversarial pool gets all whale transaction fees. If whale transaction has been packaged by honest pool and the length of honest branch is longer than adversarial branch, greedy pool will generate the Key-Block and add one to the length of the branch. If the competition is not over, greedy pool will try to mine at the head of greedy branch. The specific strategies of greedy pool are described in the following algorithm 2.

```
Algorithm 2. Greedy-Mine of Greedy pool finds a Key-block
1:  On Greedy pool finds a Key-block
2:      $\Delta = length(adversarial\ branch) - length(honest\ branch)$
3:      if $state_{tx} = 0$ then
4:          $state_{tx} = 1$
5:          $length(adversarial\ branch)$
6:          $= length(adversarial\ branch) + 1$
7:      else if $state_{tx}\ ! = 0$ then
8:          if $\Delta < 0$ then
9:              $length(adversarial\ branch)$
10:             $= length(adversarial\ branch) + 1$
11:         else if $\Delta\ ! < 0$ then
12:             Competition ends
13:     Mine at the head of greedy branch.
```

Fig. 7. Greedy-Mine of Greedy pool finds a Key-block

When honest pool finds a Key-Block, they still mine with the honest strategy according to different whale transaction states. If whale transaction has not been packaged this moment, it will generate the Key-Block and the Micro-Block containing whale transactions, and set $state_{tx} = 1$. If whale transaction has been packaged at this time, it can be divided into three states according to the difference between the branch length of greedy miners and the branch length of honest miners. If the length of honest branch is longer than greedy branch, honest pool generates Key-Block at the head of honest branch, and adds one to the length of the honest branch. If the length of honest branch is balanced with the length of the adversarial branch, different mining strategies will affect the results of competition. Specifically, if honest pool appends honest branch, add one to the length of honest branch. Otherwise, add one to the length of adversarial branch. If the length of adversarial is longer than the length of honest branch, competition ends and adversarial pool gets all whale transaction fees. If the competition is not over, honest pool will mine at the head of the longest branch. The specific strategies of honest pool are described in the following algorithm 3.

```
Algorithm 3. Greedy-Mine of Honest pool finds a Key-block
1:  On Honest pool finds a Key-block
2:      $\Delta = length(adversarial\ branch) - length(honest\ branch)$
3:      if $state_{tx} = 0$ then
4:          $state_{tx} = 2$
5:          $length(honest\ branch) = length(honest\ branch) + 1$
6:      else if $state_{tx}\ ! = 0$ then
7:          if $\Delta < 0$ then
8:              $length(honest\ branch) = length(honest\ branch) + 1$
9:          else if $\Delta > 0$ then
10:             Competition ends
11:         else if $\Delta = 0$ then
12:             if honest pool appends honest branch then
13:                 $length(honest\ branch) = length(honest\ branch) + 1$
14:             else if honest pool appends adversarial branch then
15:                 $length(adversarial\ branch)$
16:                 $= length(adversarial\ branch) + 1$
17:     Mine at the head of the longest branch
```

Fig. 8. Greedy-Mine of Greedy pool finds a Key-block

TABLE I. THE STATE DESCRIPTION

$state$	Description
s	Whale transaction has not been packaged.
h_0	Whale transaction is packaged by greedy pool and has not been linked to any new Key-Blocks since.
h_1	Whale transaction is packaged by the greedy pool and a Key-Block linked later is also generated by the greedy pool (a termination state).
$h_{0k}(k \geq 1)$	Whale transaction is packaged by greedy pool and then only honest Key-Blocks are linked, and the branch length of honest Key-Blocks is $k+1$.
h_{10}	Whale transaction is packaged by greedy pool, and the length of the honest branch is equal to the greedy branch.
$h_{1k}(k \geq 1)$	Whale transaction is packaged by greedy pool, and the length of the branch is k longer than the greedy branch.
a_0	Whale transaction is packaged by honest pool and no new Key-Blocks are linked later.
a_1	Whale transaction is packed by honest pool and only one honest Key-Block is linked afterwards.
$a_{0k}(k \geq 0)$	Whale transactions are packaged and generated by honest pool and then link to honest Key-blocks.
a_2	Whale transaction on the honest branch is packaged and generated by the honest pool, and then link to an honest block.
$a_{1k}(k \geq 0)$	On the basis of $state\ a_2$, new honest Key-Blocks are generated after honest branch, and the length of honest branch is $k+1$ longer than that of greedy branch.
$a_{2k}(k \geq 0)$	On the basis of $state\ a_2$, new Key-Blocks are generated after honest and greedy branches, and the difference between the length of honest branch and the length of greedy branch is k.

Under the greedy mining strategy, the greedy pool can obtain all whale transactions if they attack successfully. On the contrary, nothing is gained.

*3) **State Probability Transition:*** We construct state probability transition model based on greedy mining strategy and honest mining strategy. The following Table 1 shows the state description. The following figure 9 shows the state

*4) **probability transition process:*** We define the proportion of computing power owned by the adversary pool accounts for a of the total computing power of the whole network system. When all miners of the system disagree on the longest legal chain, the proportion of honest miners mining in malicious branches is γ (γ is also referred to as transmission factor), and the proportion of mining in honest branch is $1-\gamma$.

$state\ s$ indicates that the whale transaction has not been packaged. $state\ h_0$ indicates that the whale transaction is packaged by greedy pool and has not been linked to any new Key-Blocks since. The $state\ h_1$ is a termination state, which means that the whale transaction is packaged by the greedy pool and a Key-Block linked later is also generated by the greedy pool. All the whale transaction fees are obtained by the greedy pool. The $state\ h_{0k} (k \geq 1)$ indicate that the whale transaction is packaged by greedy pool and then only honest Key-Blocks are linked. The branch length of honest Key-Blocks is $k+1$. $state\ h_{10}$ indicates that the whale transaction is packaged by greedy pool, and the length of the honest branch is equal to the greedy branch. $state\ h_{1k} (k \geq 1)$ indicates that the whale transaction is packaged by greedy pool, and the length of the branch is k longer than the greedy branch. $state\ a_0$ indicates that the whale transaction is packaged by honest pool and no new Key-Blocks are linked later. $state\ a_1$ indicates that the whale transaction is packed by honest pool and only one honest Key-Block is linked afterwards. $state\ a_{0k} (k \geq 0)$ indicates that whale transactions are packaged and generated by honest pool and then link to honest Key-blocks. The length of these honest branch is $k+2$. $state\ a_2$ indicates that the whale transaction on the honest branch is packaged and generated by the honest pool, and then link to an honest block. The greedy pool tries to launch a greedy attack. After the previous Key-Block of the whale transaction, the Key-Block is released. The adversary pool packages the whale transaction to generate a Micro-Block, forming a competitive fork. $state\ a_{1k} (k \geq 0)$ indicates that on the basis of $state\ a_2$, new honest Key-Blocks are generated after honest branch, and the length of honest branch is $k+1$ longer than that of greedy branch. $state\ a_{2k} (k \geq 0)$ indicates that on the basis of $state\ a_2$, new Key-Blocks are generated after honest and greedy branches, and the difference between the length of honest branch and the length of greedy branch is k.

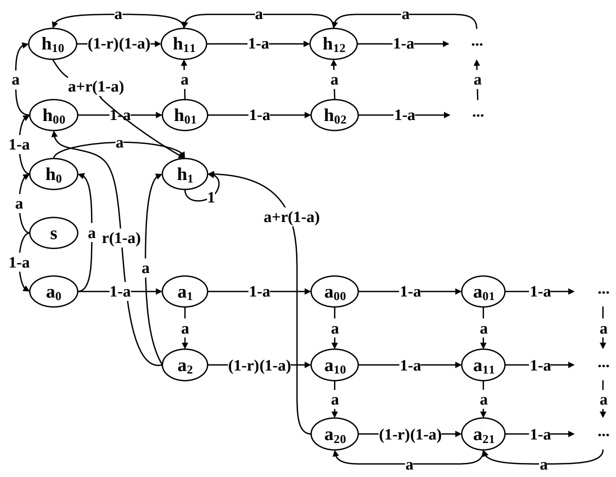

Fig. 9. State Probability Transition Process

*5) **Equation of State Probability:*** According to the state probability transition process in Section 4.3, we analyze the state probability distribution and get the following equation (4).

$$\begin{cases} p_s = 1 \\ p_{a_0} = (1-\alpha)p_s \\ p_{h_0} = \alpha + \alpha \cdot p_{a_0} \\ p_{h_{00}} = (1-\alpha)p_{h_0} + \gamma(1-\alpha)p_{a_2} \\ p_{h_{10}} = \alpha(p_{h_{00}} + p_{h_{11}}) \\ p_{a_1} = (1-\alpha)p_{a_0} \\ p_{a_2} = \alpha \cdot p_{\alpha_1} \\ p_{a_{00}} = (1-\alpha)p_{a_1} \\ p_{a_{10}} = (1-\gamma)(1-\alpha)p_{a_2} + \alpha \cdot p_{a_{00}} \\ p_{a_{20}} = \alpha \cdot p_{\alpha_{10}} + \alpha \cdot p_{\alpha_{21}} \end{cases} \tag{4}$$

We divide the state probability transition process into two parts. Since the termination state probability p_{h_1} is affected by the probabilities of state probability $p_{a_{20}}$ and state probability $p_{h_{10}}$, we separately analyze state probability $p_{a_{20}}$ and state probability $p_{h_{10}}$ according to the state transition process. According to the state transition process, we can reduce the state probability $p_{a_{20}}$ to the state probability $p_{a_{10}}$ (see equation (5)). Similarly, the state probability $p_{h_{10}}$ can be simplified to be expressed only by the state probability $p_{h_{00}}$ (see equation (6)).

$$p_{\alpha_{20}} = \alpha \cdot p_{\alpha_{10}} \cdot \sum_{k=0}^{\infty} \alpha^k (1-\alpha)^k \tag{5}$$

$$p_{h_{10}} = \alpha \cdot p_{h_{00}} \cdot \sum_{k=0}^{\infty} \alpha^k (1-\alpha)^k \tag{6}$$

According to the above equation, we can get the following results (refer to Appendix 1 for specific solution details):

$$\begin{cases} p_{h_0} = \alpha(2-\alpha) \\ p_{a_2} = \alpha(1-\alpha)^2 \\ p_{h_{10}} = \dfrac{\alpha^2(1-\alpha)(2-\alpha) + \gamma\alpha^2(1-\alpha)^3}{1-\alpha(1-\alpha)} \\ p_{a_{20}} = \dfrac{(2-\gamma)\alpha^2(1-\alpha)^3}{1-\alpha(1-\alpha)} \end{cases} \tag{7}$$

V. Revenue Analysis

1) *Revenue Analysis of Honest Mine:*

(1) Adversary pool generates two Key-Blocks in succession, and generates Micro-Block packaging whale transaction. Adversary pool obtains revenue of $\alpha^2 \cdot 100\%$.

(2) Adversary pool first generates a Key-Block, and generates Micro-Block packaging whale transaction. Honest pool generates a new Key-Blocks after the Micro-block. Adversary pool obtains revenue of $\alpha(1-\alpha) \cdot r_{leader}$.

(3) Honest pool first generates a Key-Block, and generates Micro-Block packaging whale transaction. Adversary pool generates a new Key-Blocks after the Micro-Block. Adversary pool obtains revenue of $(1-\alpha)\alpha(100\% - r_{leader})$.

According to the above three honest mining strategies, we calculate the revenue expectation that the honest pool with computing power of a can obtain:

2) *Revenue Analysis of Greedy Mine:*

(4) $state\ h_0$ is transferred to termination $state\ h_1$. Greedy pool can obtain revenue of $\alpha \cdot p_{h_0} \cdot 100\%$.

(5) $state\ h_{10}$ is transferred to termination $state\ h_1$. Greedy pool can obtain revenue of $(\alpha + \gamma(1-\alpha)) \cdot p_{h_{10}} \cdot 100\%$.

(6) $state\ a_2$ is transferred to termination $state\ h_1$. Greedy pool can obtain revenue of $\alpha \cdot p_{a_2} \cdot 100\%$.

(7) $state\ a_{20}$ is transferred to termination $state\ h_1$. Greedy pool can obtain revenue of $(\alpha + \gamma(1-\alpha)) \cdot p_{a_{20}} \cdot 100\%$.

According to the above analysis on the termination $state\ h_1$ of greedy mining, we calculate that the revenue expectation that greedy pool with computing power a can obtain is:

$$\begin{aligned} r_{Honest-Mine} &= \overbrace{\alpha^2 \cdot 100\%}^{Case(1)} + \overbrace{\alpha(1-\alpha) \cdot r_{leader}}^{Case(2)} + \overbrace{(1-\alpha)\alpha(100\% - r_{leader})}^{Case(3)} \\ &= a \end{aligned} \tag{8}$$

$$\begin{aligned} r_{Greedy-Mine} &= \overbrace{a \cdot p_{h_0} \cdot 100\%}^{Case(4)} + \overbrace{(\alpha + \gamma(1-\alpha))p_{h_{10}} \cdot 100\%}^{Case(5)} + \overbrace{(\alpha \cdot p_{a_2} \cdot 100\%}^{Case(6)} + \overbrace{(\alpha + \gamma(1-\alpha))p_{a_{20}} \cdot 100\%}^{Case(7)} \\ &= \frac{(2a^6 - 9a^5 + 16a^4 - 13a^3 + 4a^2)\gamma - a^6 + 3a^5 - 2a^4 - 2a^3 + 3a^2}{a^2 - a + 1} \end{aligned} \tag{9}$$

VI. Simulation and Experimental Result

We next present a systematic evaluation of the benefits of the adversary exploiting greedy mining strategy. We also evaluated the minimum computing power threshold that greedy pool may exploit greedy mining strategy to obtain disproportionate revenue. Figure 10 shows the revenue that adversary with different computing power can obtain by launching greedy mining strategies under different propagation factor parameters, and the revenue that can be obtained with honest Bitcoin-NG protocol.

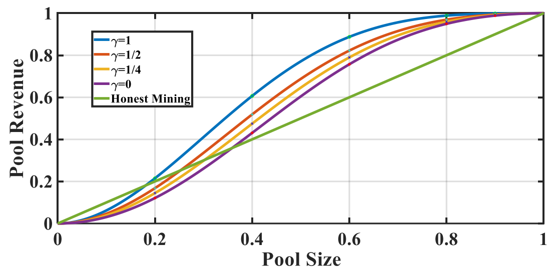

Fig. 10. Greedy revenue using the Greedy-Mine strategy for different propagation factors γ, compared to the honest mining protocol.

We find that the simulation results are consistent with the theoretical analysis, both of which show that when greedy miners have higher computing power, they will obtain higher revenue by adopting greedy mining strategies. It demonstrates that the Bitcoin-NG mining system is not incentive compatible even in the presence of an honest majority. More specifically, we set the propagation factor to four cases ($\gamma = 0$、0.25、0.5、1). The experimental results show that when $\gamma = 0$, the minimum computing power owned by greedy pool to launch a greedy attack is $\alpha = 0.36$. When $\gamma = 0.25$, $\alpha = 0.30$. When $\gamma = 0.5$, $\alpha = 0.25$. When $\gamma = 1$, $\alpha = 0.18$. Figure 11 shows the minimum computing power owned by the adversary for greedy mining to obtain disproportionate revenue under different propagation factor parameters γ.

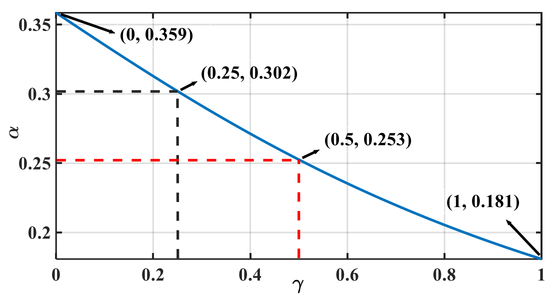

Fig. 11. For a given γ, the threshold α shows the minimum power greedy mining that will trump the honest protocol.

When the propagation factor γ is larger, honest miners are more likely to contribute to the greedy branch. The greater the probability that the greedy branch will eventually become the longest legal chain, the less computing power adversary need to exploit greedy mining strategy to obtain disproportionate revenue. We neglect the influence of the propagation parameter γ. The minimum computing power ratio owned by the adversary to exploit the greedy mining strategy to obtain excess revenue is 0.36.

VII. Conclusion

In our work, we researched the greedy mining strategy and incentive mechanism of Bitcoin-NG blockchain protocol. Bitcoin-NG is scalable, however, its incentive mechanism is not set with comprehensive considerations, ignoring possible high-level malicious attacks. Greedy adversary with a proportion of computing power resources may launch greedy attack to obtain excess revenue. We find that the higher greedy pool has the computing power, the higher revenue expectation they can obtain. Moreover, the propagation factor γ will also affect the revenue of greedy mining. The larger propagation factor, the more profitable to adversary to launch greedy mining. No matter how the propagation factor changes, when the greedy pool has computing power no less than 36% of the total computing power of the system, launching greedy mining is more profitable than honest mining. Miners with strong computing power have stronger motivation to adopt greedy mining. The Bitcoin-NG mining system is not incentive compatible.

APPENDIX

PROBABILITY CALCULATION

We analyze this state Probability Transition to learn the probabilities of it being in its different states. We obtain the following equations with the state probability transition process shown in Fig. 9.

$$\boldsymbol{p_s = 1} \tag{10}$$

$$\boldsymbol{p_{a_0} = (1-\alpha)p_s = 1-\alpha} \tag{11}$$

$$\boldsymbol{p_{h_0} = \alpha + \alpha \cdot p_{a_0} = \alpha + \alpha(1-\alpha) = \alpha(2-\alpha)} \tag{12}$$

$$\boldsymbol{p_{a_1} = (1-\alpha)p_{a_0} = (1-\alpha)^2} \tag{13}$$

$$\boldsymbol{p_{a_2} = \alpha \cdot p_{\alpha_1} = \alpha(1-\alpha)^2} \tag{14}$$

$$\begin{aligned}\boldsymbol{p_{h_{00}}} &= \boldsymbol{(1-\alpha)p_{h_0} + \gamma(1-a)p_{a_2}} \\ &= \boldsymbol{\alpha(1-\alpha)(2-\alpha) + \gamma(1-\alpha)^3\alpha}\end{aligned} \tag{15}$$

$$\boldsymbol{p_{a_{00}} = (1-\alpha)p_{a_1} = (1-\alpha)^3} \tag{16}$$

$$\begin{aligned}\boldsymbol{p_{a_{10}}} &= \boldsymbol{(1-\gamma)(1-\alpha)p_{a_2} + \alpha \cdot p_{a_{00}}} \\ &= \boldsymbol{(1-\gamma)\alpha(1-\alpha)^3 + \alpha(1-\alpha)^3} \\ &= \boldsymbol{(2-\gamma)\alpha(1-\alpha)^3}\end{aligned} \tag{17}$$

$$\begin{aligned}\boldsymbol{p_{a_{20}}} &= \boldsymbol{a \cdot p_{a_{10}} \cdot \sum_{k=0}^{\infty} a^k(1-a)^k} \\ &= \boldsymbol{a \cdot p_{a_{10}} \cdot \lim_{n\to\infty} \frac{1\cdot\left(1-\left(a(1-a)\right)^n\right)}{1-a(1-a)}} \\ &= \boldsymbol{a \cdot p_{a_{10}} \cdot \frac{1}{1-a(1-a)}} \\ &= \boldsymbol{\frac{(2-\gamma)a^2(1-a)^3}{1-a(1-a)}}\end{aligned} \tag{18}$$

$$\begin{aligned}\boldsymbol{p_{h_{10}}} &= \boldsymbol{a \cdot p_{h_{00}} \cdot \sum_{k=0}^{\infty} a^k(1-a)^k} \\ &= \boldsymbol{a \cdot p_{h_{00}} \cdot \lim_{n\to\infty} \frac{1\cdot\left(1-\left(a(1-a)\right)^n\right)}{1-a(1-a)}} \\ &= \boldsymbol{a \cdot p_{h_{00}} \cdot \frac{1}{1-a(1-a)}} \\ &= \boldsymbol{\frac{a^2(1-a)(2-a) + \gamma\cdot a^2(1-a)^3}{1-a(1-a)}}\end{aligned} \tag{19}$$

REFERENCES

[1] Nakamoto S . Bitcoin: A Peer-to-Peer Electronic Cash System[J]. consulted, 2008.

[2] Douceur, John R.. "The Sybil Attack", International Workshop on Peer-to-Peer Systems, pp. 251-260, 2002.

[3] Fanning, Kurt, and David P. Centers. "Blockchain And Its Coming Impact On Financial Services", JOURNAL of CORPORATE ACCOUNTING and FINANCE, pp. 53-57, 2016.

[4] Hyvärinen, Hissu et al. "A Blockchain-Based Approach Towards Overcoming Financial Fraud in Public Sector Services.", Web Intelligence, pp. 441-456, 2017.

[5] Khan, Minhaj Ahmad, and Khaled Salah. "IoT security: Review, blockchain solutions, and open challenges.", FUTURE GENERATION COMPUTER SYSTEMS-THE INTERNATIONAL JOURNAL of ESCIENCE, pp. 395-411, 2018.

[6] Ferrag, Mohamed Amine et al. "Blockchain Technologies for the Internet of Things: Research Issues and Challenges.", Computing Research Repository, pp. 2188-2204, 2019.

[7] Dai, Hong-Ning et al. "Blockchain for Internet of Things: A Survey.", arXiv: Networking and Internet Architecture, pp. 8076-8094, 2019.

[8] Sharma, Pradip Kumar et al. "Blockchain-based Distributed Framework for Automotive Industry in a Smart City", IEEE Transactions on Industrial Informatics, pp. 4197-4205, 2019.

[9] Pass, Rafael et al. "Analysis of the Blockchain Protocol in Asynchronous Networks.", Theory and Application of Cryptographic Techniques, pp. 643-673, 2017.

[10] Lind, Joshua et al. "Teechain: a secure payment network with asynchronous blockchain access", Symposium on Operating Systems Principles, pp. 63-79, 2019.

[11] Mettler, Matthias. "Blockchain Technology In Healthcare The Revolution Starts Here", International Conference on e-Health Networking, Applications and Services, pp. 520-522, 2016.

[12] Xie, Junfeng et al. "A Survey of Blockchain Technology Applied to Smart Cities: Research Issues and Challenges", IEEE Communications Surveys and Tutorials, pp. 2794-2830, 2019.

[13] Savelyev, Alexander. "Copyright in the Blockchain Era: Promises and Challenges", Computer Law & Security Review, pp. 550-561, 2018.

[14] Meng, Zhaoxiong et al. "Design Scheme of Copyright Management System Based on Digital Watermarking and Blockchain", Computer Software and Applications Conference, pp. 359-364, 2018.

[15] Meskini, F. Z., and R. Aboulaich Islamic. "Multi-agent based simulation of a smart insurance using Blockchain technology", International Conference on Intelligent Computing, pp. 1-6, 2019.

[16] Wood G. Ethereum: A secure decentralised generalised transaction ledger[J]. Ethereum project yellow paper, pp. 1-32, 2014.

[17] Croman, Kyle et al. "On Scaling Decentralized Blockchains - (A Position Paper).", Financial Cryptography, pp. 106-125, 2016.

[18] Pass, Rafael, and Elaine Shi. "FruitChains: A Fair Blockchain.", Principles of Distributed Computing, pp. 315-324, 2017.

[19] Eyal, Ittay et al. "Bitcoin-NG: A Scalable Blockchain Protocol", Networked Systems Design and Implementation, pp. 45-59, 2016.

[20] Yu, Haifeng et al. "OHIE: Blockchain Scaling Made Simple", Security and Privacy, pp. 90-105, 2020.

[21] Lei, Yang et al. "Prism: Scaling Bitcoin by 10,000x", arXiv preprint arXiv, 2019.

[22] Jiaping Wang and Hao Wang. "Monoxide: Scale out blockchains with asynchronous consensus zones", In 16th USENIX Symposium on Networked Systems Design and Implementation, pp. 95-112, 2019.

[23] Lewenberg, Yoad et al. "Inclusive Block Chain Protocols.", Financial Cryptography, pp. 528-547, 2015.

[24] Sompolinsky, Yonatan et al. "SPECTRE : Serialization of Proof-of-work Events : Confirming Transactions via Recursive Elections", semanticscholar, 2017.

[25] Sompolinsky, Yonatan, and Aviv Zohar. "PHANTOM: A Scalable BlockDAG Protocol.", IACR Cryptology ePrint Archive, 2018.

[26] Li, Chenxing et al. "Scaling Nakamoto Consensus to Thousands of Transactions per Second.", Computing Research Repository, 2018.

[27] Yin, Jiayuan et al. "Revisiting The Incentive Mechanism Of Bitcoin-Ng", Australasian Conference on Information Security and Privacy, pp. 706-71, 2018.