Time-Averaged Analysis of Selfish Mining in Bitcoin

Roozbeh Sarenche*, Ren Zhang†, Svetla Nikova*, Bart Preneel*
* COSIC, KU Leuven
† Cryptape Vanguard and Nervos
{roozbeh.sarenche,svetla.nikova,bart.preneel}@esat.kuleuven.be, ren@nervos.org

***Abstract*—A Bitcoin miner who owns a sufficient amount of mining power can perform selfish mining to increase his relative revenue. Studies have demonstrated that the time-averaged profit of a selfish miner starts to rise once the mining difficulty level gets adjusted in favor of the attacker. Selfish mining profitability lies in the fact that orphan blocks are not incorporated into the current version of Bitcoin's difficulty adjustment mechanism (DAM). Therefore, it is believed that considering the count of orphan blocks in the DAM can result in selfish mining unprofitability. In this paper, we disprove this belief by providing a formal analysis of the selfish mining time-averaged profit. We present a precise definition of the orphan blocks that can be incorporated into calculating the next epoch's target and then introduce two modified versions of DAM in which both main-chain blocks and orphan blocks are incorporated. We propose two versions of smart intermittent selfish mining, where the first one dominates the normal intermittent selfish mining and the second one results in selfish mining profitability under the modified DAMs. Moreover, we present the orphan exclusion attack with the help of which the attacker can stop honest miners from reporting the orphan blocks. Using combinatorial tools, we analyze the profitability of selfish mining accompanied by the orphan exclusion attack under the modified DAMs. Our result shows that even when considering the orphan blocks in the DAM, normal selfish mining can still be profitable; however, the level of profitability under the modified DAMs is significantly lower than that observed under the current version of Bitcoin DAM.**

***Index Terms*—Selfish mining, Bitcoin, Blockchain.**

I. Introduction

One of the major challenges in designing blockchain networks is the underlying consensus mechanism that helps the ledger get extended in a distributed manner. In fact, the consensus mechanism ensures that all the blockchain users agree upon a unified ledger without the help of any central entity [1]. The consensus mechanism used in Bitcoin is called Proof-of-Work (PoW). In PoW-based blockchains, miners try to solve a cryptographic puzzle, and the miner who manages to solve the puzzle faster than the others can add a new block to the blockchain. To incentivize miners to participate in the energy-consuming PoW-based consensus mechanisms, the miner of each block is rewarded with a specific amount of cryptocurrency. In Bitcoin, the cryptographic puzzle is to find a nonce that makes the hash of the block less than a specified difficulty target [2]. At the end of each epoch, the duration in which 2016 Bitcoin blocks are mined, there exists a *difficulty adjustment mechanism* (DAM) that aims to recalculate the difficulty target based on an estimation of the available mining power to ensure a relatively constant transaction throughput.

Since the emergence of Bitcoin, several attacks have been introduced that can threaten the progress of Bitcoin. In Bitcoin's design, it was assumed as long as more than half of the total mining power follows the protocol, the probability that a miner can obtain the next block reward is proportional to his computational power [3]. However, the selfish mining attack, presented in 2014 by Eyal and Sirer [4], has disproved the mentioned assumption. In this paper [4], the authors have shown that an attacker who owns more than 25% of the total mining power can increase his *relative revenue*, i.e., the ratio of the attacker's reward to the total reward, by deviating from mining honestly. In selfish mining, once the attacker mines a new block, he does not publish it immediately to the other mining nodes. Instead, he continues mining on his secret block. By doing so, the attacker can make honest nodes continue mining on top of the public chain whose height is less than the attacker's secret chain, and thus, some part of the honest miners' mining power gets wasted. As a result of this attack, the number of orphan blocks, the blocks that are correctly mined but not included in the main chain, increases. Due to an increase in the number of orphaned honest blocks, the attacker's profit while performing selfish mining can surpass his profit while mining honestly. It is shown that the selfish mining strategy presented in this paper [4] is not the best possible strategy for the attacker to follow. In 2016, Sapirshtein et al. presented an MDP-based algorithm to calculate the optimal selfish mining strategy based on the attacker's mining power and communication capability [5]. Since the introduction of selfish mining, several countermeasures such as [3], [4], [6]–[9] have been proposed that aim to keep the selfish miner's relative revenue as close as possible to his relative revenue while mining honestly, i.e., proportional to his mining power. Most of these countermeasures either make fundamental changes to block validity rules or propose a new fork-resolving policy [3]. For more information regarding the selfish mining defenses, one can refer to [3], [10], [11].

In 2018, Grunspan and Pérez-Marco presented the time analysis of selfish mining [12]. The authors of the paper argue that the previous selfish mining papers have used the Markov model [13] to analyze the selfish mining attack and thus ignored the time considerations in their analysis. More precisely, they mention that the relative revenue is not a proper benchmark for the attacker's profitability; and in order to analyze the attacker's profitability, one should consider the attacker's profit per unit of time, which we refer to as *time-*

averaged profit. Note that in the remainder of the paper, we always use the concept of time-averaged profit to assess profitability. The authors in this paper [12] have shown that before a difficulty adjustment mechanism, no mining strategy is more profitable than mining honestly; and to make selfish mining profitable, the difficulty level should be adjusted. Once the difficulty gets adjusted, the time-averaged profit of selfish mining starts to increase, and since then selfish mining becomes profitable. The authors [12] conclude that the selfish mining attack exploits the current difficulty adjustment mechanism of Bitcoin and suggest the idea of incorporating the count of orphan blocks in the difficulty adjustment mechanism to mitigate the attack.

Selfish mining was often considered to be impractical since it was thought that selfish mining needs to be continued for a couple of epochs to become profitable. However, the authors in [14] introduced the intermittent selfish mining strategy, where the selfish miner alternates between selfish and honest mining at every difficulty mechanism. They showed that by conducting selfish mining for just one epoch and then switching to honest mining for the next epoch, the selfish miner can gain a greater time-averaged profit in the course of two epochs compared to when he mines honestly in both of the epochs. The concept of intermittent selfish mining is similar to smart mining [15], where the attacker switches between honest mining and idling.

Knowing that selfish mining can raise the attacker's relative revenue, in this paper, we take a closer look at the time-averaged profit of selfish mining. Although the authors in [12] have suggested incorporating the count of orphan blocks in DAM, the precise definition of the orphan blocks in Bitcoin needs yet to be addressed. While it is believed that orphan inclusion in DAM can make selfish mining non-profitable, in this paper, we challenge this belief by introducing two attacks: smart intermittent selfish mining and the orphan exclusion attack. These two attacks demonstrate that selfish mining can still be profitable even when orphan blocks are included in the DAM. Our contributions include:

Formal analysis of intermittent selfish mining: Although the intermittent selfish mining attack is introduced in [14], the authors have not presented a formal analysis of the attack. This paper provides a comprehensive analysis of the time-averaged profit for different selfish mining strategies including intermittent selfish mining. Besides, we present an attack called smart intermittent selfish mining (version 1) that dominates the normal one introduced in [14].

Precise definition of orphan blocks: Authors in [12] introduced a new difficulty adjustment mechanism to incorporate orphaned blocks; however, they did not present a formal definition for the orphan blocks. In this paper, we introduce uncle blocks—orphan blocks in the same epoch—to specify the properties of the valid orphan blocks that can be incorporated in the difficulty adjustment mechanism. After defining the uncle blocks, we present two modified versions of the difficulty adjustment mechanism (the modified DAM) for Bitcoin in which, in addition to the main-chain blocks, the count of uncle blocks affects the mining difficulty of the next epoch.

Smart intermittent selfish mining (version 2): We introduce an attack called smart intermittent selfish mining (version 2) that disproves the belief that incorporating the orphan blocks in the DAM can result in the unprofitability of selfish mining.

Orphan exclusion attack: We introduce another attack, called orphan exclusion attack, with the help of which the attacker can prevent the honest miners from reporting the orphan blocks in the main chain. We show that selfish mining accompanied by the orphan exclusion attack can be profitable under the modified DAMs.

II. Preliminaries

In this section, we first present our system model. Then, we define the concepts of relative revenue and time-averaged profit. Finally, we discuss the effect of the difficulty adjustment mechanism on selfish mining profitability.

A. System model and definitions

In this paper, we use the system model introduced in [16]. We assume the system comprises a set of honest miners denoted by $\mathcal{H}$ and an adversarial miner denoted by $\mathcal{A}$. We denote by $\alpha_{\mathcal{H}}$ and $\alpha_{\mathcal{A}}$ the total honest mining power share and the adversarial mining power share, respectively, where $\alpha_{\mathcal{A}} + \alpha_{\mathcal{H}} = 1$. In our model, time is divided into rounds denoted by r. In each round, a miner can calculate multiple mining (hash) queries, the number of which is proportional to his mining power. We assume our system model is synchronous, i.e., the block published by one of the miners in round r will be delivered to all the other miners at the end of round r.

Communication capability: We denote by $\gamma_{\mathcal{A}}$ the communication capability of attacker $\mathcal{A}$. This means, in the case of a block race, where two blocks are published simultaneously by attacker $\mathcal{A}$ and an honest miner, the fraction of total honest miners that receive the block proposed by the attacker first is equal to $\gamma_{\mathcal{A}}$. The honest miners who receive the block proposed by the attacker first mine on top of the attacker's block.

The honest miners follow the honest strategy π^{H}, which is explained as follows:

Honest strategy: At the start of a new round, a miner chooses to mine on top of the longest chain available in his view. If the miner manages to mine a new block, he immediately publishes the block to all the other miners.

Attacker $\mathcal{A}$ may, however, deviate from the honest strategy and mine in a selfish way. In recent years, different selfish mining strategies have been presented such as Eyal and Sirer's selfish mining strategy $\pi^{\mathtt{SM1}}$ introduced in [4], the optimal selfish mining strategy $\pi^{\mathtt{OSM}}$ introduced in [5], and the intermittent selfish mining strategy $\pi^{\mathtt{ISM}}$ introduced in [14]. Strategies $\pi^{\mathtt{SM1}}$ and $\pi^{\mathtt{OSM}}$ are specifically designed to increase a miner's relative revenue, while strategy $\pi^{\mathtt{ISM}}$ aims to increase a miner's time-averaged profit. A summary of selfish mining strategies $\pi^{\mathtt{SM1}}$ and $\pi^{\mathtt{OSM}}$ is presented in Appendix A. Note that when referring to the selfish mining attack in a general context, we represent it using the notation $\pi^{\mathtt{SM}}$.

Definition 1 (Relative revenue)**.** *The relative revenue of attacker $\mathcal{A}$ following strategy π is defined as follows:*

$$\begin{aligned} \texttt{RelRev}^r_{\mathcal{A}}(\pi) &= \frac{N^r_{\mathcal{A}}}{\sum_{\mathcal{M}\in\{\mathcal{A},\mathcal{H}\}} N^r_{\mathcal{M}}}, \quad \textit{and} \\ \texttt{RelRev}_{\mathcal{A}}(\pi) &= \lim_{r\to\infty} \texttt{RelRev}^r_{\mathcal{A}}(\pi), \end{aligned} \tag{1}$$

where $N^r_{\mathcal{M}}$ for $\mathcal{M} \in \{\mathcal{A},\mathcal{H}\}$ denotes the number of blocks added to the main chain by miner $\mathcal{M}$ during interval $[1, r]$.

We denote by $R_{\mathcal{A}}(r)$ and $C_{\mathcal{A}}(r)$ the revenue and the mining cost of attacker $\mathcal{A}$ in round r, respectively. We denote by λ a system constant related to the block generation rate, where $\frac{1}{\lambda}$ represents the average number of rounds it takes for the whole system to mine a new block. If all the miners including attacker $\mathcal{A}$ follow the honest strategy, the average per-round revenue of attacker $\mathcal{A}$ can be obtained as follows:

$$\mathbb{E}[R_{\mathcal{A}}(r)] = \alpha_{\mathcal{A}} \cdot \lambda K \ , \tag{2}$$

where K denotes the value of the mining reward per block. If attacker $\mathcal{A}$ mines with his whole mining power, his average mining cost per round can be obtained as follows:

$$\mathbb{E}[C_{\mathcal{A}}(r)] = \alpha_{\mathcal{A}} \cdot c_{\mathcal{A}} \ , \tag{3}$$

where $c_{\mathcal{A}}$ denotes the average normalized mining cost of miner $\mathcal{A}$ per round. The profitability factor of attacker $\mathcal{A}$ is denoted by $\omega_{\mathcal{A}}$ and defined as follows:

$$\omega_{\mathcal{A}} := \frac{\mathbb{E}[R_{\mathcal{A}}(r)]}{\mathbb{E}[C_{\mathcal{A}}(r)]} = \frac{\lambda K}{c_{\mathcal{A}}} \ . \tag{4}$$

The profitability factor $\omega_{\mathcal{A}}$ represents the amount of return per each unit of money invested by attacker $\mathcal{A}$ provided that all the miners follow the honest strategy.

Definition 2 (Time-averaged profit)**.** *The time-averaged profit (per-round profit) of attacker $\mathcal{A}$ following strategy π is defined as follows:*

$$\begin{aligned} \texttt{Profit}^r_{\mathcal{A}}(\pi) &= \frac{\sum_{r'=1}^{r} \Big(R_{\mathcal{A}}(r') - C_{\mathcal{A}}(r')\Big)}{r}, \quad \textit{and} \\ \texttt{Profit}_{\mathcal{A}}(\pi) &= \lim_{r\to\infty} \texttt{Profit}^r_{\mathcal{A}}(\pi) \ . \end{aligned} \tag{5}$$

If assuming that time is divided into a set of round intervals denoted by $\texttt{cycle}$, according to the renewal reward process theorem, the time-averaged profit defined in Definition 2 can be obtained as follows:

$$\texttt{Profit}_{\mathcal{A}}(\pi) = \frac{\mathbb{E}\big[R_{\mathcal{A}}(\texttt{cycle})\big] - \mathbb{E}\big[C_{\mathcal{A}}(\texttt{cycle})\big]}{\mathbb{E}[t(\texttt{cycle})]} \ , \tag{6}$$

where $t(\texttt{cycle})$ represents the duration of $\texttt{cycle}$, and $R_{\mathcal{A}}(\texttt{cycle})$ and $C_{\mathcal{A}}(\texttt{cycle})$ denote the revenue and the mining cost of attacker $\mathcal{A}$ within the $\texttt{cycle}$, respectively.

III. Background

a) Unprofitability of selfish mining before a difficulty adjustment mechanism: In Bitcoin, an interval of rounds in which a set of $L = 2016$ consecutive blocks is added to the main chain is called an epoch. At the end of each epoch, there is a difficulty adjustment mechanism (DAM), which calculates the difficulty target of the upcoming epoch based on the hash power estimation of the previous epoch. Assume attacker $\mathcal{A}$ starts the selfish mining attack at the beginning of $\texttt{epoch}_1$. As it is discussed in [12] and [14], the time-averaged profit of attacker $\mathcal{A}$ under selfish mining cannot exceed his time-averaged profit under the honest strategy during $\texttt{epoch}_1$, i.e., before the next DAM. To illustrate this fact, we need to compare the attacker's profitability when following the honest strategy versus the selfish strategy. Note that the mining difficulty of $\texttt{epoch}_1$ is specified before the start of $\texttt{epoch}_1$, and thus, the attacker's strategy during $\texttt{epoch}_1$ cannot change the epoch's mining difficulty. If attacker $\mathcal{A}$ follows the honest strategy in $\texttt{epoch}_1$, he can mine a new block every $\frac{1}{\lambda\alpha_{\mathcal{A}}}$ rounds on average, and if ignoring the natural orphan occurrence, all of his blocks will be added to the main chain. If attacker $\mathcal{A}$ performs the selfish mining attack during $\texttt{epoch}_1$, since the mining difficulty is the same as the former scenario, his average mining rate is still equal to $\frac{1}{\lambda\alpha_{\mathcal{A}}}$; however, in this scenario, some of the attacker's blocks may get orphaned and remain out of the main chain due to the block races caused by the selfish mining attack. Therefore, before a DAM, the time-averaged profit of selfish mining cannot exceed the time-averaged profit of honest mining. Note that selfish mining can potentially increase the attacker's relative revenue in $\texttt{epoch}_1$. However, despite this increase in relative revenue, the attacker cannot gain a higher time-averaged profit during the first epoch of the attack.

b) The effect of DAM on selfish mining profitability: In Bitcoin, DAM is responsible for adjusting the block generation rate to ensure that, on average, it takes 10 minutes for the system to mine a new block. Therefore, the average epoch duration is equal to 2 weeks. Let $\texttt{epoch}_1$ and $\texttt{epoch}_2$ denote two consecutive epochs. If attacker $\mathcal{A}$ starts selfish mining in $\texttt{epoch}_1$, some of the both honest and adversarial blocks will get orphaned in $\texttt{epoch}_1$. Consequently, the duration in which L blocks are added to the main chain will be extended. This implies that the duration of $\texttt{epoch}_1$ will increase, exceeding the standard two-week period. At the end of $\texttt{epoch}_1$, there is a DAM that calculates the mining difficulty of $\texttt{epoch}_2$ based on the hash power estimation of $\texttt{epoch}_1$. Since the current version of Bitcoin DAM does not consider orphan blocks when estimating the active hash power of the previous epoch, the increase in the length of $\texttt{epoch}_1$ will result in a decrease in the mining difficulty of $\texttt{epoch}_2$. Therefore, during $\texttt{epoch}_2$, attacker $\mathcal{A}$ can mine a new block, on average, within a shorter period than $\frac{1}{\lambda\alpha_{\mathcal{A}}}$ rounds. This shows that starting from $\texttt{epoch}_2$, the attacker's time-averaged profit begins to increase.

c) Selfish mining profitability after the adjustment of mining difficulty: Assume attacker $\mathcal{A}$ has started selfish mining in $\texttt{epoch}_1$. Therefore, the mining difficulty of the next epoch, i.e., $\texttt{epoch}_2$, is adjusted in favor of the attacker. The time-averaged profit of attacker $\mathcal{A}$ in $\texttt{epoch}_2$ under the selfish mining strategy can be obtained as follows:

$$\texttt{Profit}_{\mathcal{A}}(\pi^{\text{SM}}) = \lambda K \cdot \texttt{RelRev}_{\mathcal{A}}(\pi^{\text{SM}}) - \alpha_{\mathcal{A}} c_{\mathcal{A}} \ . \tag{7}$$

Note that the time-averaged profit of attacker $\mathcal{A}$ under the honest strategy is equal to:

$$\texttt{Profit}_{\mathcal{A}}(\pi^{\texttt{H}}) = \alpha_{\mathcal{A}}(\lambda K - c_{\mathcal{A}}) \ . \tag{8}$$

This shows that if the attacker's selfish mining relative revenue is greater than his honest mining relative revenue, i.e., $\texttt{RelRev}_{\mathcal{A}}(\pi^{\texttt{SM}}) > \alpha_{\mathcal{A}}$, the selfish mining strategy dominates the honest strategy after the adjustment of mining difficulty. It has been shown that attacker $\mathcal{A}$ with a normal communication capability $\gamma_{\mathcal{A}} = 0.5$ needs to own more than 25% of the total mining power to achieve $\texttt{RelRev}_{\mathcal{A}}(\pi^{\texttt{SM}}) > \alpha_{\mathcal{A}}$ [4].

Knowing that selfish mining profit can surpass honest mining profit, a question arises as to why miners are unwilling to perform the selfish mining attack. One of the main reasons that answer this question is the belief that an attacker should perform selfish mining for a relatively long time to gain profit. Once the attacker starts selfish mining in $\texttt{epoch}_1$, since some of the attacker's blocks get orphaned, his gained profit in $\texttt{epoch}_1$ will be lower than his profit under honest mining. Therefore, to consider selfish mining as a profitable strategy, the gained profit by the attacker in $\texttt{epoch}_2$ (or even later epochs) should compensate for the loss the attacker has faced in $\texttt{epoch}_1$. However, if the attacker continues selfish mining for a considerable number of epochs, honest miners may decide to stop mining to avoid financial losses.

d) Intermittent selfish mining: The authors in [14] introduced the intermittent selfish mining (ISM) attack in which the attacker's time-averaged profit over two consecutive epochs surpasses the honest mining time-averaged profit. ISM disproves the belief that it is necessary to perform selfish mining for a significant number of epochs to compensate for the loss experienced during the initial selfish mining epoch. In ISM strategy, the attacker alternates between selfish mining and honest mining at every DAM. In other words, for two consecutive epochs denoted by $\texttt{epoch}_1$ and $\texttt{epoch}_2$, the attacker applies the selfish mining attack in $\texttt{epoch}_1$ and returns to honest mining in $\texttt{epoch}_2$. As already discussed, by mining in a selfish way in $\texttt{epoch}_1$, the attacker cannot increase his time-averaged profit immediately in $\texttt{epoch}_1$; however, the selfish mining attacks in $\texttt{epoch}_1$ can lead to a decrease in the mining difficulty of $\texttt{epoch}_2$, resulting in an increase in the attacker's time-averaged profit over the two consecutive epochs.

IV. Formal analysis of intermittent selfish mining

In this section, we first present a formal analysis of the intermittent selfish mining attack. Then, we introduce the smart ISM attack that can dominate the normal ISM.

A. Normal intermittent selfish mining

Despite the introduction of ISM in [14], the authors have not presented a formal analysis of ISM. We denote by $\pi^{\texttt{ISM}}$ the ISM strategy. In the ISM strategy, the attacker performs selfish mining in odd epochs, i.e., $\{\texttt{epoch}_1, \texttt{epoch}_3, \cdots\}$, and applies honest mining in even epochs, i.e., $\{\texttt{epoch}_2, \texttt{epoch}_4, \cdots\}$. Using equation 6, the attacker's time-averaged profit under the intermittent selfish mining can be obtained as follows:

$$\begin{aligned}
&\texttt{Profit}_{\mathcal{A}}(\pi^{\texttt{ISM}}) = \\
&\frac{\mathbb{E}\big[R_{\mathcal{A}}(\texttt{epoch}_{\texttt{odd}})\big] - \mathbb{E}\big[C_{\mathcal{A}}(\texttt{epoch}_{\texttt{odd}})\big]}{\mathbb{E}[t(\texttt{epoch}_{\texttt{odd}})] + \mathbb{E}[t(\texttt{epoch}_{\texttt{even}})]} + \\
&\frac{\mathbb{E}\big[R_{\mathcal{A}}(\texttt{epoch}_{\texttt{even}})\big] - \mathbb{E}\big[C_{\mathcal{A}}(\texttt{epoch}_{\texttt{even}})\big]}{\mathbb{E}[t(\texttt{epoch}_{\texttt{odd}})] + \mathbb{E}[t(\texttt{epoch}_{\texttt{even}})]} \ .
\end{aligned} \tag{9}$$

For simplicity, we refrain from using the expected value notation, denoted as $\mathbb{E}[\cdot]$, throughout the rest of the paper. The average revenue gained by the attacker in $\texttt{epoch}_{\texttt{odd}}$ and his mining cost in $\texttt{epoch}_{\texttt{odd}}$ can be obtained as follows:

$$\begin{aligned}
R_{\mathcal{A}}(\texttt{epoch}_{\texttt{odd}}) &= \texttt{RelRev}\big(\pi^{\texttt{SM}}\big) \cdot LK \ , \\
C_{\mathcal{A}}(\texttt{epoch}_{\texttt{odd}}) &= \alpha_{\mathcal{A}} c_{\mathcal{A}} \cdot t(\texttt{epoch}_{\texttt{odd}}) \ .
\end{aligned} \tag{10}$$

Similarly, the average revenue gained by the attacker in $\texttt{epoch}_{\texttt{even}}$ and his mining cost in $\texttt{epoch}_{\texttt{even}}$ can be obtained as follows:

$$\begin{aligned}
R_{\mathcal{A}}(\texttt{epoch}_{\texttt{even}}) &= \alpha_{\mathcal{A}} \cdot LK \ , \\
C_{\mathcal{A}}(\texttt{epoch}_{\texttt{even}}) &= \alpha_{\mathcal{A}} c_{\mathcal{A}} \cdot t(\texttt{epoch}_{\texttt{even}}) \ .
\end{aligned} \tag{11}$$

We denote by $M^{\texttt{total}}_{\texttt{odd}}$ and $M^{\texttt{main-chain}}_{\texttt{odd}}$ the normalized active mining power in $\texttt{epoch}_{\texttt{odd}}$ and the normalized effective mining power extending the main chain in $\texttt{epoch}_{\texttt{odd}}$, respectively. As there is no idle power in $\texttt{epoch}_{\texttt{odd}}$, we have $M^{\texttt{total}}_{\texttt{odd}} = 1$. The effective mining power $M^{\texttt{main-chain}}_{\texttt{odd}}$ represents the ratio of the number of main-chain blocks in $\texttt{epoch}_{\texttt{odd}}$ to the total number of blocks mined during $\texttt{epoch}_{\texttt{odd}}$. Due to the selfish mining attack in $\texttt{epoch}_{\texttt{odd}}$ and orphan occurrence, some part of the mining power in $\texttt{epoch}_{\texttt{odd}}$ gets wasted and does not contribute to extending the main chain. This implies that $M^{\texttt{main-chain}}_{\texttt{odd}}$ is less than 1. We define $M^{\texttt{main-chain}}_{\texttt{SM}}$ to be the normalized effective mining power under selfish mining. We have $M^{\texttt{main-chain}}_{\texttt{odd}} = M^{\texttt{main-chain}}_{\texttt{SM}}$. In Appendix A, the methods for calculating $M^{\texttt{main-chain}}_{\texttt{SM}}$ under both selfish mining strategies $\pi^{\texttt{SM1}}$ and $\pi^{\texttt{OSM}}$ are explained. Similar terms can be defined for $\texttt{epoch}_{\texttt{even}}$. Since all the miners follow the honest strategy in $\texttt{epoch}_{\texttt{even}}$, we have $M^{\texttt{main-chain}}_{\texttt{even}} = M^{\texttt{total}}_{\texttt{even}} = 1$. According to the design of the current version of Bitcoin's DAM, the duration of $\texttt{epoch}_{\texttt{odd}}$ can be calculated as follows:

$$t(\texttt{epoch}_{\texttt{odd}}) = t^{\texttt{ideal}} \cdot \frac{M^{\texttt{main-chain}}_{\texttt{even}}}{M^{\texttt{main-chain}}_{\texttt{odd}}} = \frac{L}{\lambda M^{\texttt{main-chain}}_{\texttt{SM}}} \ , \tag{12}$$

where $t^{\texttt{ideal}}$ represents the ideal epoch duration and is equal to $\frac{L}{\lambda}$. Note that under the current version of Bitcoin DAM, the epoch duration $t(\texttt{epoch}_{\texttt{odd}})$ is inversely related to the epoch's main-chain effective power $M^{\texttt{main-chain}}_{\texttt{odd}}$. The greater the amount of mining power working to extend the main chain within an epoch, the shorter the time it takes for the epoch to complete. However, the epoch duration $t(\texttt{epoch}_{\texttt{odd}})$ is directly related to the previous epoch's main-chain effective power $M^{\texttt{main-chain}}_{\texttt{even}}$. The reason is that a lower amount of main-chain effective mining power in the previous epoch results in a decrease in the mining difficulty and consequently the duration of the current epoch. Similarly, the duration of $\texttt{epoch}_{\texttt{even}}$ can be calculated as follows:

$$t(\texttt{epoch}_{\texttt{even}}) = t^{\texttt{ideal}} \cdot \frac{M_{\texttt{odd}}^{\texttt{main-chain}}}{M_{\texttt{even}}^{\texttt{main-chain}}} = \frac{LM_{\texttt{SM}}^{\texttt{main-chain}}}{\lambda} \ . \tag{13}$$

Therefore, the attacker's time-averaged profit under intermittent selfish mining can be obtained as follows:

$$\texttt{Profit}_{\mathcal{A}}(\pi^{\texttt{ISM}}) = \frac{\lambda K\Big(\texttt{RelRev}\big(\pi^{\texttt{SM}}\big) + \alpha_{\mathcal{A}}\Big)}{\frac{1}{M_{\texttt{SM}}^{\texttt{main-chain}}} + M_{\texttt{SM}}^{\texttt{main-chain}}} - \alpha_{\mathcal{A}} c_{\mathcal{A}} \ . \tag{14}$$

Mining power threshold values: We aim to address the question of how much mining power is required to make the intermittent selfish mining strategy more profitable than the honest strategy. By fixing the attacker's communication capability $\gamma_{\mathcal{A}}$, we want to calculate the minimum amount of the attacker's mining power that satisfies the following inequality:

$$\texttt{Profit}_{\mathcal{A}}(\pi^{\texttt{ISM}}) > \texttt{Profit}_{\mathcal{A}}(\pi^{\texttt{H}}) \ . \tag{15}$$

Using equations 8 and 14, we can obtain that to achieve the inequality above, the following inequality should hold:

$$\texttt{RelRev}\big(\pi^{\texttt{SM}}\big) > \big(\frac{1}{M_{\texttt{SM}}^{\texttt{main-chain}}} + M_{\texttt{SM}}^{\texttt{main-chain}} - 1\big)\alpha_{\mathcal{A}} \ . \tag{16}$$

As the coefficient of $\alpha_{\mathcal{A}}$ is greater than 1 in the inequality above, the minimum amount of mining power that makes ISM profitable is more than that in normal selfish mining. For instance, for a normal communication capability $\gamma_{\mathcal{A}} = 0.5$, the minimum amount of mining power share that makes ISM profitable is equal to 0.2773.

B. Smart intermittent selfish mining (version 1)

In intermittent selfish mining, the attacker performs the attack every other epoch. This shows that the ratio of the selfish mining period length (measured by the number of blocks added to the main chain) to the main-chain length is equal to $\frac{1}{2}$. In the smart intermittent selfish mining attack (version 1) denoted by SISM1, the attacker gains a higher amount of profit while his ratio of selfish mining period length to the main-chain length is still equal to $\frac{1}{2}$. Assume $\texttt{epoch}_{\texttt{odd}}$ and $\texttt{epoch}_{\texttt{even}}$ are two consecutive epochs in which $2L$ blocks are added to the main chain. In SISM1, the attacker performs selfish mining for $(1-\eta)L$ blocks in $\texttt{epoch}_{\texttt{odd}}$ and for ηL blocks in $\texttt{epoch}_{\texttt{even}}$, where $0 \leq \eta \leq 0.5$. For the remaining blocks in these two epochs, the attacker follows the honest mining strategy. It is clear that in SISM1, the ratio of selfish mining period length to the main-chain length is equal to $\frac{1}{2}$. Note that if $\eta = 0$, SISM1 is the same as the normal intermittent selfish mining attack. The duration of $\texttt{epoch}_{\texttt{odd}}$ and $\texttt{epoch}_{\texttt{even}}$ in SISM1 can be calculated as follows:

$$\begin{aligned} t(\texttt{epoch}_{\texttt{odd}}) &= t^{\texttt{ideal}} \frac{\eta + \frac{1-\eta}{M_{\texttt{SM}}^{\texttt{main-chain}}}}{1-\eta + \frac{\eta}{M_{\texttt{SM}}^{\texttt{main-chain}}}} \ , \\ t(\texttt{epoch}_{\texttt{even}}) &= t^{\texttt{ideal}} \frac{1-\eta + \frac{\eta}{M_{\texttt{SM}}^{\texttt{main-chain}}}}{\eta + \frac{1-\eta}{M_{\texttt{SM}}^{\texttt{main-chain}}}} \ . \end{aligned} \tag{17}$$

Therefore, the attacker's time-averaged profit under SISM1 is equal to:

$$\begin{aligned} &\texttt{Profit}_{\mathcal{A}}(\pi^{\texttt{SISM1}}) = \\ &\frac{\lambda K\Big(\texttt{RelRev}\big(\pi^{\texttt{SM}}\big) + \alpha_{\mathcal{A}}\Big)}{\frac{\eta + \frac{1-\eta}{M_{\texttt{SM}}^{\texttt{main-chain}}}}{1-\eta + \frac{\eta}{M_{\texttt{SM}}^{\texttt{main-chain}}}} + \frac{1-\eta + \frac{\eta}{M_{\texttt{SM}}^{\texttt{main-chain}}}}{\eta + \frac{1-\eta}{M_{\texttt{SM}}^{\texttt{main-chain}}}}} - \alpha_{\mathcal{A}} c_{\mathcal{A}} \ . \end{aligned} \tag{18}$$

The maximum amount of the attacker's time-averaged profit under SISM1 occurs when $\eta = 0.5$:

$$\begin{aligned} &\texttt{Profit}_{\mathcal{A}}\big(\pi^{\texttt{SISM1}} | \eta = 0.5\big) = \\ &\frac{\lambda K\Big(\texttt{RelRev}\big(\pi^{\texttt{SM}}\big) + \alpha_{\mathcal{A}}\Big)}{2} - \alpha_{\mathcal{A}} c_{\mathcal{A}} \ . \end{aligned} \tag{19}$$

Comparing equations 14 and 19, it is clear that the optimal SISM1 dominates the normal intermittent selfish mining strategy, i.e., $\texttt{Profit}_{\mathcal{A}}\big(\pi^{\texttt{SISM1}} | \eta = 0.5\big) > \texttt{Profit}_{\mathcal{A}}\big(\pi^{\texttt{ISM}}\big)$. One can consider the optimal SISM1 as the normal intermittent selfish mining with the difference that the DAM is exactly placed in the middle of the selfish mining period. Therefore, the optimal SISM1 is equivalent to performing selfish mining for half of each epoch. This shows that to increase his time-averaged profit, the attacker should keep the mining difficulty level constant and avoid inducing fluctuations. Note that the minimum amount of mining power that makes the optimal SISM1 profitable is the same as that in normal selfish mining. This implies that for a normal communication capability $\gamma_{\mathcal{A}} = 0.5$, the minimum amount of mining power share that makes SISM1 profitable is equal to 0.25.

The normalized time-averaged profit, which is equal to $\frac{\texttt{Profit}_{\mathcal{A}}(\pi)}{\lambda K}$, is depicted in Figure 1 for multiple strategies. In this figure, the profitability factor $\omega_{\mathcal{A}}$ is set to 2, and the optimal selfish mining strategy $\pi^{\texttt{OSM}}$ is used to calculate $\texttt{RelRev}\big(\pi^{\texttt{SM}}\big)$ and $M_{\texttt{SM}}^{\texttt{main-chain}}$. An interesting observation regarding Figure 1 is that the time-averaged profit of intermittent selfish mining does not necessarily increase with the amount of mining power. The reason is that $\pi^{\texttt{OSM}}$ is specifically designed to maximize the relative revenue of selfish mining. However, $\texttt{Profit}_{\mathcal{A}}(\pi^{\texttt{ISM}})$ not only depends on $\texttt{RelRev}\big(\pi^{\texttt{SM}}\big)$, but it also depends on $M_{\texttt{SM}}^{\texttt{main-chain}}$, a point that is not considered in the design of $\pi^{\texttt{OSM}}$. This shows that the strategy optimizing the time-averaged profit of intermittent selfish mining differs from the strategy that maximizes its relative revenue. Finding the optimal time-averaged profit can be an interesting future research direction.

V. Profitability of selfish mining under the modified DAM

In this section, we introduce two modified versions of DAM, which are similar to the period-based DAM currently used in Bitcoin. In these modified DAMs, both main-chain blocks and orphan blocks are incorporated in the mining difficulty calculation of the upcoming epoch. We first provide a precise definition of the orphan blocks that can be incorporated into the modified DAM. Then, we present the modified DAMs and assess their impact on the profitability of the normal selfish mining attack.

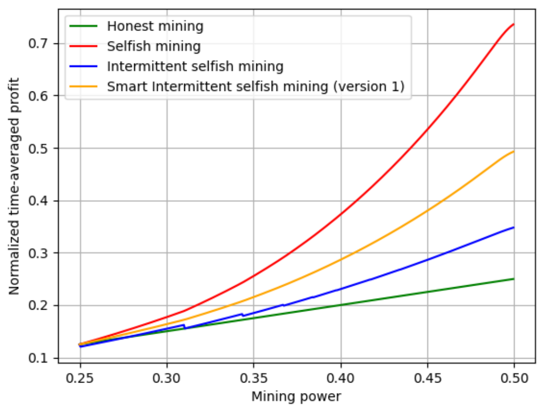

Fig. 1. Time-averaged profit under the Bitcoin's DAM

A. The uncle blocks

The current Bitcoin DAM is presented in Appendix B. To present the modified DAMs, we first need to define the uncle blocks. In the modified DAMs, we aim to consider the orphan blocks in the hash rate estimation. The uncles are the orphan blocks that are mined during the same epoch.

Definition 3 (Uncle)**.** *A block B_1 is considered an* uncle *of another block B_2 if all of the following conditions are met: (1) they are in the same epoch, sharing the same difficulty; (2) $height(B_2) > height(B_1)$; (3) B_1's parent is already embedded in B_2's chain, either as a main-chain block or as an uncle; and (4) B_2 is the first block in its chain to refer to B_1.*

Condition (1) enables uncles to contribute to a more accurate hash rate estimation, which will be exploited in our DAM. A violation of (2) contradicts the longest-chain rule. Condition (3) is to ensure that two blockchain instances with different genesis blocks do not accidentally recognize each other's blocks as uncles. Besides, B_1's parent is needed to verify the validity of B_1. Condition (4) is to prevent honest miners from reporting the same uncle multiple times.

Incorporating uncles. Miners are requested to refer to uncles—orphaned blocks in the same epoch—by embedding their hashes in the blocks. Embedding uncles contributes to a more accurate estimation of the network hash rate, thus contributing to the system's selfish mining resistance. Even a single honest block can report all previously unreported uncles in the same epoch. Note that our uncle's definition differs from that of Ethereum's deprecated PoW version [17] in that our uncle's validity does not consider how far away the uncle and the nephew's first common ancestor is. The reward distribution regarding uncles is also different from that of Ethereum. Our design issues neither uncle rewards to compensate uncle miners, nor nephew rewards to incentivize miners to embed uncles. This is because uncle and nephew rewards raise the selfish mining profit and lower the mining power threshold to perform the attack [18], [19].

In the following, two versions of the modified DAM are introduced. These two versions differ from each other in their definition of the epoch, the period at the end of which the DAM is applied. Note that our modified DAMs are nearly the same as the DAM introduced in [12], with the difference that our modified DAMs only incorporate the orphan blocks that satisfy Definition 3. In the remainder of this paper, we use the terms uncle and orphan blocks interchangeably.

B. Modified DAM with a fixed total number of blocks per epoch

Let L denote the length of each epoch in the current version of Bitcoin DAM, i.e., $L = 2016$. In this version of the modified DAM, denoted by $\mathsf{DAM}_1^{\texttt{modified}}$, an epoch is defined as an interval of rounds in which a set of L blocks (comprising both main-chain and orphan blocks) are mined and subsequently reported in the main chain. Let $\mathsf{CNT}_i^{\texttt{main-chain}}$, $\mathsf{CNT}_i^{\texttt{orphan}}$, and $\mathsf{CNT}_i^{\texttt{total}}$ denote the number of main-chain blocks, the number of orphan blocks embedded in the main chain, and the total number of both main-chain and reported orphan blocks mined during epoch i, respectively. Note that if assuming that all the orphan blocks are reported, the ratio $\mathsf{CNT}_i^{\texttt{main-chain}}/(\mathsf{CNT}_i^{\texttt{orphan}} + \mathsf{CNT}_i^{\texttt{main-chain}})$ represents the epoch's effective main-chain power, i.e., $M^{\texttt{main-chain}}$. According to the definition of epoch in $\mathsf{DAM}_1^{\texttt{modified}}$, we have $\mathsf{CNT}_i^{\texttt{total}} = L$ and $\mathsf{CNT}_i^{\texttt{main-chain}} \leq L$. This indicates that in $\mathsf{DAM}_1^{\texttt{modified}}$, the total number of blocks mined per epoch is fixed; however, the total amount of distributed reward per epoch can vary across different epochs.

Inputs and outputs. Similar to the Bitcoin DAM, our modified DAM is executed at the end of every epoch. It takes two inputs: the last epoch's target denoted by TGT_i and the last epoch's duration—the timestamp difference between epoch i and $i-1$'s last blocks—denoted by t_i. Note that TGT_i is decided by the last DAM iteration, and t_i is measured after the epoch ends. The algorithm outputs the next epoch's target denoted by TGT_{i+1}.

To adjust the target TGT based on the network hash rate, the modified $\mathsf{DAM}_1^{\texttt{modified}}$ is triggered at the end of each epoch as follows:

$$\mathsf{TGT}_{i+1} = \begin{cases} \mathsf{TGT}_i \cdot \frac{1}{\tau}, & t_i < \frac{1}{\tau} \cdot t_{\text{ideal}} \\ \mathsf{TGT}_i \cdot \tau, & t_i > \tau \cdot t_{\text{ideal}} \\ \mathsf{TGT}_i \cdot \frac{t_i}{t_{\text{ideal}}}, & \text{otherwise} \end{cases}, \tag{20}$$

where i is the epoch number, and τ is a dampening filter to prevent rapid changes of TGT. Similar to the Bitcoin DAM, we assume $t^{\texttt{ideal}}$ for $\mathsf{DAM}_1^{\texttt{modified}}$ is two weeks.

C. Modified DAM with a fixed number of main-chain blocks per epoch

We use the same notations as those introduced in the $\mathsf{DAM}_1^{\texttt{modified}}$ explanation. In this version of the modified DAM, denoted by $\mathsf{DAM}_2^{\texttt{modified}}$, an epoch is defined as an interval of rounds in which a set of L consecutive blocks is added to the main chain. According to the definition of epoch in $\mathsf{DAM}_2^{\texttt{modified}}$, we have $\mathsf{CNT}_i^{\texttt{main-chain}} = L$ and $\mathsf{CNT}_i^{\texttt{total}} \geq L$. This indicates that in $\mathsf{DAM}_2^{\texttt{modified}}$, the total amount of distributed reward per epoch is fixed; however, the total number of blocks mined per epoch may vary across different epochs.

Inputs and outputs. The modified DAM takes three inputs: the last epoch's target denoted by TGT_i, the last epoch's duration—the timestamp difference between epoch i and $i-1$'s last blocks—denoted by t_i, and the last epoch's orphaned block count—the number of uncles embedded in epoch i's main chain—denoted by $\mathsf{CNT}_i^{\texttt{orphan}}$. Among these inputs, TGT_i is decided by the last DAM iteration, while t_i and $\mathsf{CNT}_i^{\texttt{orphan}}$ are measured after the epoch ends.

To adjust the target TGT based on the network hash rate, the modified $\mathsf{DAM}_2^{\texttt{modified}}$ is triggered at the end of each epoch as follows:

$$\mathsf{TGT}_{i+1} = \begin{cases} \mathsf{TGT}_i \cdot \frac{1}{\tau}, & t_i < \frac{1}{\tau} \cdot T \\ \mathsf{TGT}_i \cdot \tau, & t_i > \tau \cdot T \\ \mathsf{TGT}_i \cdot \frac{t_i}{T}, & \text{otherwise} \end{cases} , \tag{21}$$

where $t^{\texttt{ideal}}$ is two weeks, and period T is defined as follows:

$$T = \frac{\mathsf{CNT}_i^{\texttt{orphan}} + \mathsf{CNT}_i^{\texttt{main-chain}}}{\mathsf{CNT}_i^{\texttt{main-chain}}} \cdot t^{\texttt{ideal}} \tag{22}$$

In both versions of the modified DAM, in addition to the main-chain blocks, the count of orphan blocks is considered in the calculation of the mining target. In the remainder of the paper, whenever we want to refer to a DAM that incorporates the orphan blocks, we will use the term "the modified DAM" without explicitly indicating its exact version.

D. Normal selfish mining under the modified DAM

In this section, we analyze the normal selfish mining profitability under the modified DAM. By normal selfish mining attack, we mean that the attacker follows the selfish mining strategy continuously for all the epochs.

1) Analysis under $\mathsf{DAM}_1^{\texttt{modified}}$: We first obtain the average revenue gained by the attacker in each epoch and his mining cost. Note that due to the selfish mining attack and the definition of epoch in $\mathsf{DAM}_1^{\texttt{modified}}$, the average number of main-chain blocks in each epoch is equal to $M_{\text{SM}}^{\texttt{main-chain}}L$. As a result, the total amount of distributed reward per epoch is equal to $M_{\text{SM}}^{\texttt{main-chain}}LK$.

$$\begin{aligned} &R_{\mathcal{A}}(\texttt{epoch}_i) = \texttt{RelRev}\big(\pi^{\text{SM}}\big) \cdot M_{\text{SM}}^{\texttt{main-chain}}LK \ , \\ &C_{\mathcal{A}}(\texttt{epoch}) = \alpha_{\mathcal{A}}c_{\mathcal{A}} \cdot t(\texttt{epoch}_i) \ . \end{aligned} \tag{23}$$

Then, we calculate the epoch duration. Note that $M_{i-1}^{\texttt{total}} = M_i^{\texttt{total}} = 1$. According to the design of $\mathsf{DAM}_1^{\texttt{modified}}$, the duration of each epoch can be calculated as follows.

$$t(\texttt{epoch}_i) = t^{\texttt{ideal}} \cdot \frac{M_{i-1}^{\texttt{total}}}{M_i^{\texttt{total}}} = \frac{L}{\lambda} \ . \tag{24}$$

Therefore, the time-averaged profit of normal selfish mining under $\mathsf{DAM}_1^{\texttt{modified}}$ is equal to:

$$\begin{aligned} &\texttt{Profit}_{\mathcal{A}}(\pi^{\text{SM}}, \mathsf{DAM}_1^{\texttt{modified}}) = \\ &\lambda K\Big(\texttt{RelRev}\big(\pi^{\text{SM}}\big)M_{\text{SM}}^{\texttt{main-chain}}\Big) - \alpha_{\mathcal{A}}c_{\mathcal{A}} \ . \end{aligned} \tag{25}$$

2) Analysis under $\mathsf{DAM}_2^{\texttt{modified}}$: According to the definition of epoch in $\mathsf{DAM}_2^{\texttt{modified}}$, the number of main-chain blocks and the total amount of distributed reward in each epoch are equal to L and LK, respectively. Therefore, the average revenue gained by the attacker in each epoch and his mining cost can be obtained as follows:

$$\begin{aligned} &R_{\mathcal{A}}(\texttt{epoch}_i) = \texttt{RelRev}\big(\pi^{\text{SM}}\big) \cdot LK \ , \\ &C_{\mathcal{A}}(\texttt{epoch}) = \alpha_{\mathcal{A}}c_{\mathcal{A}} \cdot t(\texttt{epoch}_i) \ . \end{aligned} \tag{26}$$

To calculate the epoch duration, one should consider that $M_{i-1}^{\texttt{total}} = 1$ and $M_i^{\texttt{main-chain}} = M_{\text{SM}}^{\texttt{main-chain}}$. According to the design of $\mathsf{DAM}_2^{\texttt{modified}}$, the duration of each epoch can be calculated as follows.

$$t(\texttt{epoch}_i) = t^{\texttt{ideal}} \cdot \frac{M_{i-1}^{\texttt{total}}}{M_i^{\texttt{main-chain}}} = \frac{L}{\lambda M_{\text{SM}}^{\texttt{main-chain}}} \ . \tag{27}$$

Therefore, the time-averaged profit of normal selfish mining under $\mathsf{DAM}_2^{\texttt{modified}}$ is equal to:

$$\begin{aligned} &\texttt{Profit}_{\mathcal{A}}(\pi^{\text{SM}}, \mathsf{DAM}_2^{\texttt{modified}}) = \\ &\lambda K\Big(\texttt{RelRev}\big(\pi^{\text{SM}}\big)M_{\text{SM}}^{\texttt{main-chain}}\Big) - \alpha_{\mathcal{A}}c_{\mathcal{A}} \ . \end{aligned} \tag{28}$$

3) Profitability of normal selfish mining under the modified DAM: As can be seen in equations 25 and 28, the time-averaged profit of normal selfish mining under both versions of the modified DAM is the same. In this section, we show that the time-averaged profit of normal selfish mining under the modified DAM is less than the honest strategy time-averaged profit, which can be calculated as $\texttt{Profit}_{\mathcal{A}}(\pi^{\text{H}}) = \lambda K\alpha_{\mathcal{A}} - \alpha_{\mathcal{A}}c_{\mathcal{A}}$. To prove this claim, we need to show that:

$$\texttt{RelRev}\big(\pi^{\text{SM}}\big)M_{\text{SM}}^{\texttt{main-chain}} \le \alpha_{\mathcal{A}} \ . \tag{29}$$

Let $M_{\text{SM},\mathcal{A}}^{\texttt{main-chain}}$ and $M_{\text{SM},\mathcal{H}}^{\texttt{main-chain}}$ denote the normalized adversarial and honest mining power share extending the main chain during the selfish mining attack, respectively. Therefore, the total power share extending the main chain during the attack is equal to $M_{\text{SM}}^{\texttt{main-chain}} = M_{\text{SM},\mathcal{A}}^{\texttt{main-chain}} + M_{\text{SM},\mathcal{H}}^{\texttt{main-chain}}$. The attacker's relative revenue under selfish mining can be obtained as follows:

$$\texttt{RelRev}\big(\pi^{\text{SM}}\big) = \frac{M_{\text{SM},\mathcal{A}}^{\texttt{main-chain}}}{M_{\text{SM},\mathcal{A}}^{\texttt{main-chain}} + M_{\text{SM},\mathcal{H}}^{\texttt{main-chain}}} \ . \tag{30}$$

Therefore,

$$\texttt{RelRev}\big(\pi^{\text{SM}}\big)M_{\text{SM}}^{\texttt{main-chain}} = M_{\text{SM},\mathcal{A}}^{\texttt{main-chain}} \ . \tag{31}$$

Note that due to the selfish mining attack, some of the adversarial blocks may get orphaned, and as a result $M_{\text{SM},\mathcal{A}}^{\texttt{main-chain}} \le \alpha_{\mathcal{A}}$. This proves the correctness of equation 29.

VI. Smart intermittent selfish mining (version 2)

In the previous section, we showed that the normal selfish mining attack is unprofitable under the modified DAM. In this section, we present a new version of the selfish mining attack called smart intermittent selfish mining (version 2), which can be profitable even under the modified version of DAM. This attack can be considered as the combination of smart mining [15] and ISM. In the smart intermittent selfish mining (version 2) denoted by SISM2, during $\texttt{epoch}_{\texttt{even}} \in$

$\{\texttt{epoch}_0, \texttt{epoch}_2, \cdots\}$, attacker $\mathcal{A}$ divides his mining power share $\alpha_\mathcal{A}$ into two parts: the idle mining power and the honest mining power. We assume the attacker's idle mining power share and honest mining power share are equal to $e\alpha_\mathcal{A}$ and $(1-e)\alpha_\mathcal{A}$, where $0 \le e \le 1$. However, during $\texttt{epoch}_{\text{odd}} \in \{\texttt{epoch}_1, \texttt{epoch}_3, \cdots\}$, attacker $\mathcal{A}$ uses all his mining power share to perform selfish mining. Our goal is to show that SISM2 can be more profitable than honest mining under the modified DAM. Note that for the analysis of this section, we assume that all the orphan blocks are reported, and thus, incorporated in the modified DAM.

A. Analysis under $\mathsf{DAM}_1^{\texttt{modified}}$

The average revenue gained by the attacker in $\texttt{epoch}_{\text{even}}$ and his mining cost in $\texttt{epoch}_{\text{even}}$ can be obtained as follows:

$$
\begin{aligned}
R_\mathcal{A}(\texttt{epoch}_{\text{even}}) &= \frac{(1-e)\alpha_\mathcal{A}}{1-e\alpha_\mathcal{A}} \cdot LK \ , \\
C_\mathcal{A}(\texttt{epoch}_{\text{even}}) &= (1-e)\alpha_\mathcal{A} c_\mathcal{A} \cdot t(\texttt{epoch}_{\text{even}}) \ .
\end{aligned} \tag{32}
$$

Note that due to the selfish mining in $\texttt{epoch}_{\text{odd}}$ and the definition of epoch in $\mathsf{DAM}_1^{\texttt{modified}}$, the number of main-chain blocks in $\texttt{epoch}_{\text{odd}}$ is equal to $M_{\text{SM}}^{\texttt{main-chain}} L$. As a result, the total amount of distributed reward in $\texttt{epoch}_{\text{odd}}$ is equal to $M_{\text{SM}}^{\texttt{main-chain}} LK$. The average revenue gained by the attacker in $\texttt{epoch}_{\text{odd}}$ and his mining cost in $\texttt{epoch}_{\text{odd}}$ can be obtained as follows.

$$
\begin{aligned}
R_\mathcal{A}(\texttt{epoch}_{\text{odd}}) &= \texttt{RelRev}\big(\pi^{\text{SM}}\big) \cdot M_{\text{SM}}^{\texttt{main-chain}} LK \ , \\
C_\mathcal{A}(\texttt{epoch}_{\text{odd}}) &= \alpha_\mathcal{A} c_\mathcal{A} \cdot t(\texttt{epoch}_{\text{odd}}) \ .
\end{aligned} \tag{33}
$$

Note that $M_{\text{odd}}^{\texttt{total}} = 1$ and $M_{\text{even}}^{\texttt{total}} = 1 - e\alpha_\mathcal{A}$. According to the design of $\mathsf{DAM}_1^{\texttt{modified}}$, the duration of $\texttt{epoch}_{\text{even}}$ and $\texttt{epoch}_{\text{odd}}$ can be calculated as follows:

$$
t(\texttt{epoch}_{\text{even}}) = t^{\texttt{ideal}} \cdot \frac{M_{\text{odd}}^{\texttt{total}}}{M_{\text{even}}^{\texttt{total}}} = \frac{L}{\lambda(1-e\alpha_\mathcal{A})} \ , \tag{34}
$$

and

$$
t(\texttt{epoch}_{\text{odd}}) = t^{\texttt{ideal}} \cdot \frac{M_{\text{even}}^{\texttt{total}}}{M_{\text{odd}}^{\texttt{total}}} = \frac{L(1-e\alpha_\mathcal{A})}{\lambda} \ . \tag{35}
$$

Therefore, the time-averaged profit of SISM2 under $\mathsf{DAM}_1^{\texttt{modified}}$ is equal to:

$$
\begin{aligned}
&\texttt{Profit}_\mathcal{A}(\pi^{\texttt{SISM2}}, \mathsf{DAM}_1^{\texttt{modified}}) = \\
&\frac{\lambda K\Big(\texttt{RelRev}\big(\pi^{\text{SM}}\big) M_{\text{SM}}^{\texttt{main-chain}} + \frac{(1-e)\alpha_\mathcal{A}}{1-e\alpha_\mathcal{A}}\Big)}{1-e\alpha_\mathcal{A} + \frac{1}{1-e\alpha_\mathcal{A}}} \\
&+ \frac{e\alpha_\mathcal{A} c_\mathcal{A} \cdot \frac{1}{1-e\alpha_\mathcal{A}}}{1-e\alpha_\mathcal{A} + \frac{1}{1-e\alpha_\mathcal{A}}} - \alpha_\mathcal{A} c_\mathcal{A} \ .
\end{aligned} \tag{36}
$$

B. Analysis under $\mathsf{DAM}_2^{\texttt{modified}}$

The average revenue gained by the attacker in $\texttt{epoch}_{\text{even}}$ and his mining cost in $\texttt{epoch}_{\text{even}}$ can be obtained as follows:

$$
\begin{aligned}
R_\mathcal{A}(\texttt{epoch}_{\text{even}}) &= \frac{(1-e)\alpha_\mathcal{A}}{1-e\alpha_\mathcal{A}} \cdot LK \ , \\
C_\mathcal{A}(\texttt{epoch}_{\text{even}}) &= (1-e)\alpha_\mathcal{A} c_\mathcal{A} \cdot t(\texttt{epoch}_{\text{even}}) \ .
\end{aligned} \tag{37}
$$

According to the definition of epoch in $\mathsf{DAM}_2^{\texttt{modified}}$, the number of main-chain blocks and the total amount of distributed reward in $\texttt{epoch}_{\text{odd}}$ are equal to L and LK, respectively. The average revenue gained by the attacker in $\texttt{epoch}_{\text{odd}}$ and his mining cost in $\texttt{epoch}_{\text{odd}}$ can be obtained as follows.

$$
\begin{aligned}
R_\mathcal{A}(\texttt{epoch}_{\text{odd}}) &= \texttt{RelRev}\big(\pi^{\text{SM}}\big) \cdot LK \ , \\
C_\mathcal{A}(\texttt{epoch}_{\text{odd}}) &= \alpha_\mathcal{A} c_\mathcal{A} \cdot t(\texttt{epoch}_{\text{odd}}) \ .
\end{aligned} \tag{38}
$$

Note that $M_{\text{odd}}^{\texttt{total}} = 1$ and $M_{\text{even}}^{\texttt{total}} = 1 - e\alpha_\mathcal{A}$. According to the design of $\mathsf{DAM}_2^{\texttt{modified}}$, the duration of $\texttt{epoch}_{\text{even}}$ and $\texttt{epoch}_{\text{odd}}$ can be calculated as follows:

$$
t(\texttt{epoch}_{\text{even}}) = t^{\texttt{ideal}} \cdot \frac{M_{\text{odd}}^{\texttt{total}}}{M_{\text{even}}^{\texttt{main-chain}}} = \frac{L}{\lambda(1-e\alpha_\mathcal{A})} \ , \tag{39}
$$

and

$$
t(\texttt{epoch}_{\text{odd}}) = t^{\texttt{ideal}} \cdot \frac{M_{\text{even}}^{\texttt{total}}}{M_{\text{odd}}^{\texttt{main-chain}}} = \frac{L(1-e\alpha_\mathcal{A})}{\lambda M_{\text{SM}}^{\texttt{main-chain}}} \ . \tag{40}
$$

Therefore, the time-averaged profit of SISM2 under $\mathsf{DAM}_2^{\texttt{modified}}$ is equal to:

$$
\begin{aligned}
&\texttt{Profit}_\mathcal{A}(\pi^{\texttt{SISM2}}, \mathsf{DAM}_2^{\texttt{modified}}) = \\
&\frac{\lambda K\Big(\texttt{RelRev}\big(\pi^{\text{SM}}\big) + \frac{(1-e)\alpha_\mathcal{A}}{1-e\alpha_\mathcal{A}}\Big) + e\alpha_\mathcal{A} c_\mathcal{A} \cdot \frac{1}{1-e\alpha_\mathcal{A}}}{\frac{1-e\alpha_\mathcal{A}}{M_{\text{SM}}^{\texttt{main-chain}}} + \frac{1}{1-e\alpha_\mathcal{A}}} \\
&- \alpha_\mathcal{A} c_\mathcal{A} \ .
\end{aligned} \tag{41}
$$

C. Profitability of SISM2 under the modified DAM

In this section, we show that SISM2 can be more profitable than honest mining even under the modified DAM (when the orphan blocks are reported). We first define the profitability advantage of strategy π, which is denoted by $\texttt{P}^{\texttt{adv}}(\pi)$, as follows:

$$
\texttt{P}^{\texttt{adv}}(\pi) := \frac{\texttt{Profit}_\mathcal{A}(\pi) - \texttt{Profit}_\mathcal{A}(\pi^{\texttt{H}})}{\lambda K} \ . \tag{42}
$$

$\texttt{P}^{\texttt{adv}}(\pi) > 0$ indicates that strategy π is more profitable than honest mining. Depending on the profitability factor $\omega_\mathcal{A}$, the mining power share $\alpha_\mathcal{A}$, the communication capability $\gamma_\mathcal{A}$, and the version of the modified DAM, the amount of attacker's idle mining power share in $\texttt{epoch}_{\text{even}}$ that maximizes the SISM2 time-averaged profit may vary. The maximum amount

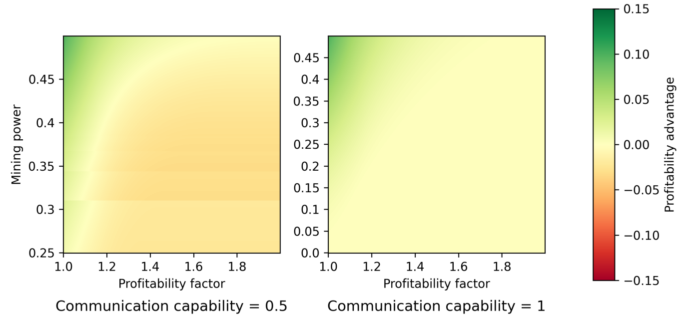

Fig. 2. Profitability advantage of SISM2 under $\mathsf{DAM}_1^{\texttt{modified}}$

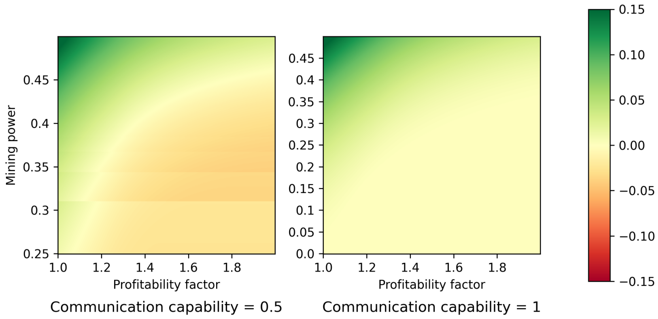

Fig. 3. Profitability advantage of SISM2 under $\mathsf{DAM}_2^{\texttt{modified}}$

of profitability advantage of SISM2 under $\mathsf{DAM}_1^{\texttt{modified}}$ and under $\mathsf{DAM}_2^{\texttt{modified}}$ are depicted in Figure 2 and Figure 3, respectively. These figures represent the profitability advantage as a function of $\alpha_{\mathcal{A}}$ and $\omega_{\mathcal{A}}$ for two distinct values of communication capability. Note that to draw these figures, we have used the optimal selfish mining strategy [5] to calculate the attacker's relative revenue and main-chain effective power in $\texttt{epoch}_{\texttt{odd}}$. As can be seen, at some points in the maps depicted in Figure 2 and Figure 3, $\mathtt{P}^{\mathrm{adv}}(\pi^{\mathtt{SISM2}}) > 0$, showing that selfish mining can be profitable even when the orphan blocks are incorporated into the DAM.

Intuition behind SISM2 profitability SISM2 profitability lies in the fact that the idle mining power in $\texttt{epoch}_{\texttt{even}}$ results in a decrease in the mining difficulty of $\texttt{epoch}_{\texttt{odd}}$. As a result, the attacker can mine a new block in a shorter period of time during $\texttt{epoch}_{\texttt{odd}}$, which leads to collecting a greater reward. If the extra collected reward in $\texttt{epoch}_{\texttt{odd}}$ can compensate for the loss of being idle in $\texttt{epoch}_{\texttt{even}}$, the attack becomes profitable. For the attackers whose profitability factor $\omega_{\mathcal{A}}$ is relatively low, being idle does not cause a huge profit loss, and consequently, SISM2 can be more profitable than honest mining. At the time of writing on February 26, 2024, the average Bitcoin profitability factor is equal to 1.071 [20]. In Appendix C, we show that under $\mathsf{DAM}_1^{\texttt{modified}}$, SISM2 cannot be more profitable than smart honest mining [15], in which the miner switches between honest mining and being idle. However, under $\mathsf{DAM}_2^{\texttt{modified}}$, SISM2 can even be more profitable than smart honest mining.

VII. Orphan Exclusion Attack

As already shown in the previous section, an intelligently executed selfish mining attack can be more profitable than honest mining even if orphan blocks are considered in the DAM. In the analysis of the previous section, we have assumed that all the orphan blocks get reported by the honest miners and subsequently incorporated in the modified DAM. However, the attacker can impose an attack, we refer to as the orphan exclusion attack (OEA), that stops honest miners from reporting some of the orphan blocks. By performing the orphan exclusion attack, a selfish miner can increase his time-averaged profit under the modified DAM. In this section, we explain the orphan exclusion attack and try to analyze for how long the attacker can prevent the honest miners from reporting the orphan blocks.

A. The attack explanation

In this attack, the attacker tries to orphan a set of consecutive honest blocks at the end of each epoch including the last honest block of the epoch. Whenever a few blocks are left to the end of each epoch, the attacker starts orphaning the public chain. To do so, the attacker separates his private chain from the public chain, i.e., forks the public chain, and tries to extend his private chain. The attack is considered to be successful if the following two conditions are satisfied:

1) Starting from the fork point, the attacker's private chain manages to orphan the public chain.
2) The last block of the epoch is included in the attacker's private chain.

By performing this attack, the attacker manages to orphan some of the honest blocks that will never be reported inside the other honest blocks of the main chain. The scenario of the orphan exclusion attack is depicted in Figure 4. Assume the attacker starts the attack when L_1^{OEA} blocks are left to the end of epoch i and manages to orphan the public chain after L^{OEA} blocks, where $L_1^{\mathsf{OEA}} \leq L^{\mathsf{OEA}}$. We use L^{OEA} to denote the length of the orphan exclusion attack. As a result of the attack, L_1^{OEA} blocks get orphaned at the end of epoch i. Because there is no honest block included in the main chain from the start of the attack till the end of epoch i, these L_1^{OEA} orphaned honest blocks cannot be reported and consequently cannot be incorporated in the DAM deciding the difficulty of epoch $i+1$. In addition, if assuming $L^{\mathsf{OEA}} = L_1^{\mathsf{OEA}} + L_2^{\mathsf{OEA}}$, the attacker manages to orphan L_2^{OEA} honest blocks at the start of epoch $i+1$. According to condition (3) in the uncle definition, prior to reporting the L_2^{OEA} honest blocks orphaned at the start of epoch $i+1$, the honest miners should report their ancestors, i.e., the L_1^{OEA} honest blocks orphaned at the end of epoch i. However, according to condition (1) in the uncle definition, the honest blocks in epoch $i+1$ cannot report the orphaned blocks of the previous epochs. As a result, these L_2^{OEA} orphaned honest blocks cannot be reported by the honest blocks in epoch $i+1$ and consequently cannot be incorporated in the DAM deciding the difficulty of epoch $i+2$. Therefore, by imposing a successful orphan exclusion attack, in total, L^{OEA} orphaned honest blocks cannot be reported in the current and next epochs. The orphan exclusion reduces the mining difficulty calculated by the modified DAM in favor of the attacker, which can result in an increase in the selfish mining time-averaged profit.

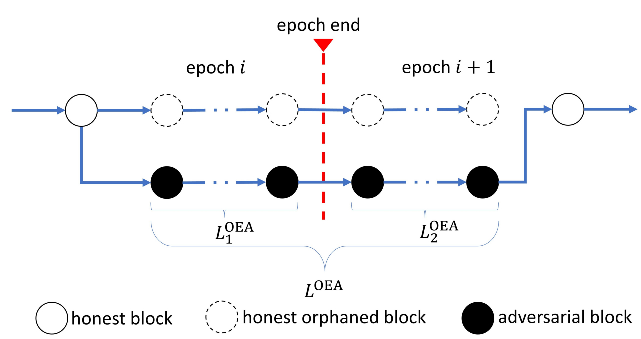

Fig. 4. Orphan exclusion attack

The attacker should decide on the starting and ending time of the orphan exclusion attack. The greater the length of the orphan exclusion attack, the more profitable the selfish mining. However, increasing the length of the orphan exclusion attack reduces the success probability of the attack. Assume the attacker starts the attack when a few blocks are left to the end of an epoch. If during the attack until the end of the epoch, the attacker's private chain always has a lead over the public chain, the attacker can easily orphan the public chain and finish the attack successfully. However, there is a possibility that in the middle of the attack and before reaching the end of the epoch, the public chain gets a lead over the attacker's private chain. In this situation, the attacker should decide whether he wants to continue mining on top of his private chain or stop the orphan exclusion attack and join the public chain. On the one hand, if the attacker decides to stop the attack before reaching the end of the epoch, regardless of whether he managed to orphan the honest blocks or not, the attack cannot cause a reduction in the difficulty level specified by the upcoming modified DAM because the remaining honest blocks added to the main chain before the end of the epoch can report the orphaned blocks. On the other hand, if the attacker decides to continue mining on top of his private chain, he will risk losing more blocks. Therefore, the attacker should devise a strategy for the orphan exclusion attack that can maximize the length of the orphan exclusion attack.

In the remainder of the paper, we aim to calculate the attacker's time-averaged profit under the modified DAMs while performing both selfish mining and orphan exclusion attacks. The first step towards calculating the attacker's time-averaged profit is to calculate the length of the orphan exclusion attack. This length represents the period during which the attacker can prevent the honest miners from adding an honest block to the blockchain and reporting the orphaned blocks. It is obvious that if the attacker's mining power is less than the honest miners', the attacker cannot continue orphaning all the honest blocks forever, and there will be an honest block that gets added to the blockchain and terminates the orphan exclusion attack [21]. Note that the concept of honest block exclusion is similar to the suppression concept introduced in [22], where the attacker tries to suppress the honest blocks and put them out of the main chain. While the authors in [22] have focused on calculating the number of suppressed honest blocks, in this paper, we try to find the length of **consecutive** suppressed honest blocks.

B. The length of the orphan exclusion attack under $DAM_1^{modified}$

In this section, we aim to calculate the average length of the orphan exclusion attack performed by attacker $\mathcal{A}$ at the end of each epoch under the modified DAMs. To calculate the length of the longest possible orphan exclusion attack, we assume attacker $\mathcal{A}$ enjoys the highest communication capability and can predict future block miners. We first explain the impact of communication capability and predictability on the orphan exclusion attack.

Definition 4 (Mining sequence)**.** *A mining sequence, which is denoted by S, is an ordered list of blocks that specifies the miner of each block. B_i^H ($B_i^{\mathcal{A}}$) is used to represent the i^{th} honest (adversarial) block in the mining sequence S. The blocks in S are ordered by the time at which they are mined.*

Note that not all the blocks of S are included in the main chain since some of them may get orphaned. We assume all the orphans in S are caused by the attack, namely, there is no naturally orphaned block in S. We first explain the predictive capability.

Predictive capability: The attacker can predict the elements of mining sequence S in advance. In other words, the attacker knows which of the upcoming blocks are mined either by himself or by the honest miners.

Note that, in Bitcoin, no miner knows who is the next block proposer until they solve the puzzle or receive a new block from the blockchain network, i.e., no miners enjoy the predictive capability. In Appendix D, we discuss how an attacker without the predictive capability can perform the OEA attack. In this section, to calculate the maximum length of the orphan exclusion attack, we assume that $\mathcal{A}$ enjoys the predictive capability and his communication capability ($\gamma_{\mathcal{A}}$) is equal to 1. We argue that in such a scenario, $\mathcal{A}$ can impose the longest possible orphan exclusion attack at the end of each epoch. Possessing the predictive capability helps $\mathcal{A}$ not only to successfully finish all the orphan exclusion attacks but to impose the longest possible attack at the end of each epoch. Consider the mining sequence $S = \{B_1^{\mathrm{H}}, B_1^{\mathcal{A}}, B_2^{\mathrm{H}}, B_3^{\mathrm{H}}, B_2^{\mathcal{A}}, B_3^{\mathcal{A}}, B_4^{\mathcal{A}}, B_4^{\mathrm{H}}, B_5^{\mathrm{H}}, \cdots\}$ depicted in Figure 5. Assume a real-world attacker, who has mined block $B_1^{\mathcal{A}}$, decides to keep the block secret and start the orphan exclusion attack. Since the next block, i.e., B_2^{H}, is mined by the honest miners, the block race situation occurs. In this case, the real-world attacker should decide whether he wants to publish block $B_1^{\mathcal{A}}$ or continue the orphan exclusion attack. Continuing the attack can increase the risk of losing mined blocks for the real-world attacker. However, since $\mathcal{A}$ is aware of the mining sequence, he knows he will eventually win the chain race and successfully finish the attack. Therefore, at the end of each epoch, $\mathcal{A}$ either does not start the orphan exclusion attack or imposes a successful one. Moreover, at the end of each epoch, there is a possibility that an attacker can impose successful orphan exclusion attacks of different lengths. For instance,

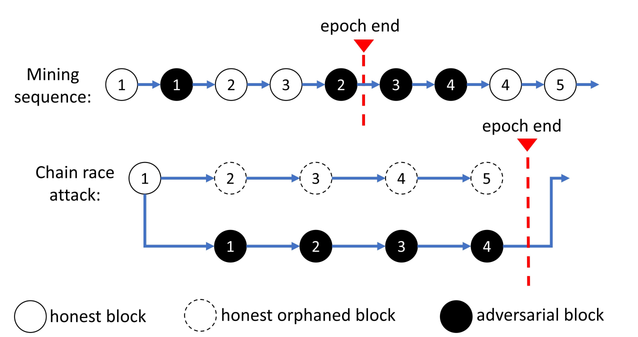

Fig. 5. Comparison of the orphan exclusion attack imposed by a real-world attacker and $\mathcal{A}$

TABLE I
THE LENGTH OF THE ORPHAN EXCLUSION ATTACK

mining power share	0.25	0.3	0.35	0.4	0.45
L^{OEA} ($\mathsf{DAM}_1^{\texttt{modified}}$)	1.45	2.99	6.83	18.36	87.02
L^{OEA} ($\mathsf{DAM}_2^{\texttt{modified}}$)	15.48	23.87	40.02	78.67	225.79

assume after block B_5^{H}, there exist a long set of consecutive blocks mined by honest miners. In this scenario, the attacker can start a successful orphan exclusion attack both at block $B_1^{\mathcal{A}}$ and block $B_2^{\mathcal{A}}$. However, the length of the former attack is equal to 4 and the length of the latter one is equal to 3. This shows that $\mathcal{A}$ can specify the start and end points of the attack in a way that maximizes the length of the attack. Having $\gamma_{\mathcal{A}} = 1$ helps $\mathcal{A}$ to orphan the maximum possible number of honest blocks (the same number as the orphan exclusion attack length).

A comprehensive analysis for calculating the length of the orphan exclusion attack under $\mathsf{DAM}_1^{\texttt{modified}}$ and $\mathsf{DAM}_2^{\texttt{modified}}$ are presented in Appendix E and Appendix F, respectively. Table I presents the average length of the orphan exclusion attack under $\mathsf{DAM}_1^{\texttt{modified}}$ and $\mathsf{DAM}_2^{\texttt{modified}}$ performed by attacker $\mathcal{A}$, who enjoys the predictive capability and the highest possible communication capability. The results presented in Table I show that the length of orphan exclusion attack under $\mathsf{DAM}_2^{\texttt{modified}}$ is longer than that under $\mathsf{DAM}_1^{\texttt{modified}}$. This indicates that selfish mining accompanied by the orphan exclusion attack can be more profitable under $\mathsf{DAM}_2^{\texttt{modified}}$ compared to that under $\mathsf{DAM}_1^{\texttt{modified}}$.

VIII. SELFISH MINING ACCOMPANIED BY THE ORPHAN EXCLUSION ATTACK

In this section, we aim to assess the effect of the orphan exclusion attack on selfish mining profitability under the modified DAM. As shown in Section V-D, the normal selfish mining attack in which the attacker follows selfish mining continuously for all the epochs is not profitable under the modified DAM. Here, we show that applying the orphan exclusion attack can make the normal selfish mining attack profitable under the modified DAM. For our analysis in this section, we assume that the attacker manages to perform a successful orphan exclusion attack at the end of each epoch, where the attack length is denoted by L^{OEA}. This indicates that the last L^{OEA} blocks of the main chain in each epoch are adversarial, and L^{OEA} honest blocks get orphaned without being reported in each epoch.

A. Analysis under $\mathsf{DAM}_1^{\texttt{modified}}$

We first calculate the average revenue gained by the attacker in each epoch and his mining cost. Note that out of the first $L_{\text{epoch}} - L^{\mathsf{OEA}}$ blocks of the epoch, only $M_{\mathsf{SM}}^{\texttt{main-chain}}(L_{\text{epoch}} - L^{\mathsf{OEA}})$ blocks are added to the main chain, out of which $\texttt{RelRev}(\pi^{\mathsf{SM}}) M_{\mathsf{SM}}^{\texttt{main-chain}}(L_{\text{epoch}} - L^{\mathsf{OEA}})$ blocks are adversarial. The last L^{OEA} blocks of the epoch that are added to the main chain are adversarial blocks.

$$\begin{aligned}&R_{\mathcal{A}}(\texttt{epoch}_i) = \\ &\texttt{RelRev}(\pi^{\mathsf{SM}}) M_{\mathsf{SM}}^{\texttt{main-chain}}(L_{\text{epoch}} - L^{\mathsf{OEA}})K + L^{\mathsf{OEA}}K \ , \\ &C_{\mathcal{A}}(\texttt{epoch}_i) = \alpha_{\mathcal{A}} c_{\mathcal{A}} \cdot t(\texttt{epoch}_i) \ .\end{aligned} \tag{43}$$

According to the design of the modified DAM introduced in section V-B, the target of $\texttt{epoch}_i$ can be calculated as follows:

$$\mathsf{TGT}_i = \mathsf{TGT}_{i-1} \frac{t(\texttt{epoch}_{i-1})}{t^{\texttt{ideal}}} \ . \tag{44}$$

Since the total amount of mining power and the miners' strategy are consistent throughout the whole epochs, the duration and the mining target of all the epochs would be the same, i.e., $\mathsf{TGT}_i = \mathsf{TGT}_{i-1}$ and $t(\texttt{epoch}_i) = t(\texttt{epoch}_{i-1})$. Therefore, by using equation 44, the duration of each epoch can be calculated as follows:

$$t(\texttt{epoch}_i) = t^{\texttt{ideal}} = \frac{L_{\text{epoch}}}{\lambda} \ . \tag{45}$$

Therefore, the time-averaged profit of normal selfish mining accompanied by the orphan exclusion attack under $\mathsf{DAM}_1^{\texttt{modified}}$ is equal to:

$$\begin{aligned}&\texttt{Profit}_{\mathcal{A}}(\pi^{\mathsf{SM\text{-}OEA}}, \mathsf{DAM}_1^{\texttt{modified}}) = \\ &\lambda K \Big(\texttt{RelRev}(\pi^{\mathsf{SM}}) M_{\mathsf{SM}}^{\texttt{main-chain}} \big(\frac{L_{\text{epoch}} - L^{\mathsf{OEA}}}{L_{\text{epoch}}}\big) + \frac{L^{\mathsf{OEA}}}{L_{\text{epoch}}} \Big) \\ &- \alpha_{\mathcal{A}} c_{\mathcal{A}} \ .\end{aligned} \tag{46}$$

B. Analysis under $\mathsf{DAM}_2^{\texttt{modified}}$

We first calculate the average revenue gained by the attacker in each epoch and his mining cost. Note that out of the first $L_{\text{epoch}} - L^{\mathsf{OEA}}$ main-chain blocks, only $\texttt{RelRev}(\pi^{\mathsf{SM}})(L_{\text{epoch}} - L^{\mathsf{OEA}})$ blocks are adversarial. The last L^{OEA} blocks of the main chain are adversarial blocks.

$$\begin{aligned}&R_{\mathcal{A}}(\texttt{epoch}_i) = \texttt{RelRev}(\pi^{\mathsf{SM}})(L_{\text{epoch}} - L^{\mathsf{OEA}})K + L^{\mathsf{OEA}}K \ , \\ &C_{\mathcal{A}}(\texttt{epoch}_i) = \alpha_{\mathcal{A}} c_{\mathcal{A}} \cdot t(\texttt{epoch}_i) \ .\end{aligned} \tag{47}$$

According to the design of the modified DAM introduced in section V-C, the target of $\texttt{epoch}_i$ can be calculated as follows:

$$\mathsf{TGT}_i = \mathsf{TGT}_{i-1} \frac{t(\texttt{epoch}_{i-1})}{\frac{\mathsf{CNT}_i^{\texttt{orphan-total}} + L_{\text{epoch}} - L^{\mathsf{OEA}}}{L_{\text{epoch}}} t^{\texttt{ideal}}} \ , \tag{48}$$

where $\mathsf{CNT}_i^{\texttt{orphan-total}}$ represents the total number of both reported and non-reported orphan blocks in $\texttt{epoch}_i$. Since the total amount of mining power and the miners' strategy are consistent throughout the whole epochs, the duration and the mining target of all the epochs would be the same, i.e., $\mathsf{TGT}_i = \mathsf{TGT}_{i-1}$ and $t(\texttt{epoch}_i) = t(\texttt{epoch}_{i-1})$. Therefore, by using equation 48, the duration of each epoch can be calculated as follows:

$$t(\texttt{epoch}_i) = \frac{\mathsf{CNT}_i^{\texttt{orphan-total}} + L_{\text{epoch}} - L^{\mathsf{OEA}}}{L_{\text{epoch}}} t^{\texttt{ideal}} \ . \tag{49}$$

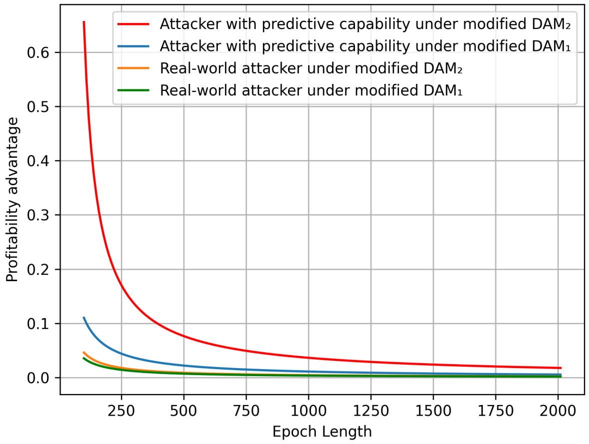

Fig. 6. The profitability advantage of strategy $\pi^{\text{SM-OEA}}$ under the modified DAM

As $M_{\text{SM}}^{\texttt{main-chain}} = \frac{L_{\text{epoch}}}{\mathsf{CNT}_i^{\texttt{orphan-total}} + L_{\text{epoch}}}$, we have:

$$t(\text{epoch}_i) = \frac{L_{\text{epoch}} - M_{\text{SM}}^{\texttt{main-chain}} L^{\mathsf{OEA}}}{M_{\text{SM}}^{\texttt{main-chain}} L_{\text{epoch}}} \cdot \frac{L_{\text{epoch}}}{\lambda} \quad . \tag{50}$$

Therefore, the time-averaged profit of normal selfish mining accompanied by the orphan exclusion attack under $\mathsf{DAM}_2^{\texttt{modified}}$ is equal to:

$$\begin{aligned} &\texttt{Profit}_{\mathcal{A}}(\pi^{\text{SM-OEA}}, \mathsf{DAM}_2^{\texttt{modified}}) = \\ &\lambda K \Big(\frac{M_{\text{SM}}^{\texttt{main-chain}} L_{\text{epoch}}}{L_{\text{epoch}} - M_{\text{SM}}^{\texttt{main-chain}} L^{\mathsf{OEA}}} \Big) \times \\ &\Big(\texttt{RelRev}(\pi^{\text{SM}}) \big(\frac{L_{\text{epoch}} - L^{\mathsf{OEA}}}{L_{\text{epoch}}} \big) + \frac{L^{\mathsf{OEA}}}{L_{\text{epoch}}} \Big) - \alpha_{\mathcal{A}} c_{\mathcal{A}} \quad . \end{aligned} \tag{51}$$

C. *Profitability of normal selfish mining accompanied by the orphan exclusion attack under the modified DAM*

Due to the orphan exclusion attack, the mining difficulty of the subsequent epoch decreases, which results in an increase in selfish mining profitability. In Figure 6, the profitability advantage of normal selfish mining accompanied by the orphan exclusion attack is depicted as a function of epoch length L_{epoch} for the attacker with $\alpha_{\mathcal{A}} = 0.4$ and $\gamma_{\mathcal{A}} = 1$ under both versions of the modified DAM. As can be seen in Figure 6, $\texttt{P}^{\texttt{adv}}(\pi^{\text{SM-OEA}}) > 0$, indicating that the normal selfish mining attack can be profitable even if the orphans are incorporated in the DAM. Additionally, it can be observed that a reduction in the epoch length can result in a more destructive selfish mining attack. This highlights a trade-off in the design of the DAM between resistance to selfish mining and sensitivity to hash rate fluctuations.

IX. Discussion

There has been a significant debate surrounding selfish mining, one of the most influential attacks against Bitcoin, regarding its lack of occurrence. Although it is proved that selfish mining in Bitcoin under the current DAM is profitable, Bitcoin has been running smoothly without strong evidence of this attack for almost a decade. In fact, we have high confidence that the attack has not happened because the observable orphan rate is relatively low. This raises the question of why miners have refrained from deploying the selfish mining attack despite its potential profitability. The most influential argument is miners' goodwill to keep the network secure [23]. Although a miner can gain a greater amount of cryptocurrency (reward) on average by performing the selfish mining attack, due to the decrease in the cryptocurrency price caused by the detection of the attack, the miner cannot obtain a financial profit. However, if the attacker manages to perform selfish mining with an acceptable forking rate, the attack cannot be detected easily [24]. Additionally, it may happen that with the passage of time and updating the underlying protocols, the cryptocurrency can recover its price, making the reward gathered by selfish mining in the past valuable again.

Although we argued that selfish mining can still be profitable even when the orphans are incorporated in the DAM, as can be seen in Figure 6, the modified versions of the DAM introduced in this paper can significantly limit the increase in the real-world attacker's selfish mining time-averaged profit, especially when the epoch length is relatively long. The absence of the selfish mining attack in Bitcoin so far is not sufficient evidence to dismiss the possibility of its occurrence in the future. Therefore, implementing a solution to defend against selfish mining can be a reasonable decision provided that the overhead caused by the proposed solution does not exceed its benefits. Compared to the Bitcoin DAM, the modified DAM imposes higher communication and storage costs to the Bitcoin network because referring to orphans in the honest blocks would increase the block size. However, firstly since only the hash of orphan blocks is reported, and secondly, due to the very low forking rate in the normal situation, the added network and storage costs by the modified DAM would be negligible. Therefore, by applying the modified DAM, at the cost of a very small increase in communication and storage costs, we can significantly decrease selfish mining profitability.

As a comparison between two versions of the modified DAM introduced in this paper, it is worth mentioning that under $\mathsf{DAM}_1^{\texttt{modified}}$, the smart intermittent selfish mining (version 2) profitability cannot surpass smart honest mining profitability. However, under $\mathsf{DAM}_2^{\texttt{modified}}$, SISM2 profitability can surpass smart honest mining profitability. Moreover, the average length of the orphan exclusion attack under $\mathsf{DAM}_1^{\texttt{modified}}$ is less than that under $\mathsf{DAM}_2^{\texttt{modified}}$. Therefore, between these two versions of the modified DAM, $\mathsf{DAM}_1^{\texttt{modified}}$ seems to be the superior choice to implement.

Although it is shown in this paper that the modified DAM is relatively secure against selfish miners, it is advisable to formally analyze the behavior of the modified DAM in the presence of Byzantine adversaries to strongly justify the modified DAM adoption beyond its help with mitigating the selfish mining attack. In this paper, we tried to carefully define the uncle block properties in a way that mitigates Byzantine behaviors. As a future work, one can use the Bitcoin backbone model introduced in [25] to demonstrate that under the introduced modified DAM and uncle block definition, Bitcoin can satisfy common-prefix and chain-quality properties.

Note that in this paper, we analyzed selfish mining profitability under period-based difficulty adjustment mechanisms. By period-based, we mean that the DAM is applied at the end of a fixed period [14]. However, there exist other types of difficulty adjustment mechanisms such as DAMs with a sliding window that can be modified to incorporate the orphan blocks. Note that the orphan exclusion attack introduced in this paper is not specific to period-based DAMs and can also be applied to DAMs with a sliding window. As a future work, one can analyze selfish mining profitability under a DAM with a sliding window that incorporates the orphan blocks.

X. Conclusion

While it was widely believed that incorporating the count of the orphan blocks in the difficulty adjustment mechanism of Bitcoin can make selfish mining unprofitable, in this paper, we disproved the mentioned belief by proposing two attacks: the smart intermittent selfish mining attack and the orphan exclusion attack. These two attacks make selfish mining more profitable compared the the honest strategy even when the orphan blocks are incorporated in the DAM. In this paper, we calculated the profitability of different selfish mining strategies under different DAMs. Besides, to analyze the effect of the orphan exclusion attack on the profitability of selfish mining, we used probability analysis and combinatorial tools to calculate bounds for the length of the orphan exclusion attack. Knowing the orphan exclusion attack length, we obtained the timed-averaged profit of selfish mining accompanied by the orphan exclusion attack under the modified DAMs.

Acknowledgments

This work was supported by CyberSecurity Research Flanders with reference number VR20192203. In addition, this work was supported in part by the Flemish Government through FWO SBO project SNIPPET S007619N.

References

[1] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *Ieee Access*, vol. 7, pp. 22 328–22 370, 2019.

[2] D. Fullmer and A. S. Morse, "Analysis of difficulty control in bitcoin and proof-of-work blockchains," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 5988–5992.

[3] R. Zhang and B. Preneel, "Publish or perish: A backward-compatible defense against selfish mining in bitcoin," in *Cryptographers' Track at the RSA Conference*. Springer, 2017, pp. 277–292.

[4] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *International conference on financial cryptography and data security*. Springer, 2014, pp. 436–454.

[5] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 515–532.

[6] E. Heilman, "One weird trick to stop selfish miners: Fresh bitcoins, a solution for the honest miner," in *International Conference on Financial Cryptography and Data Security*. Springer, 2014, pp. 161–162.

[7] L. Bahack, "Theoretical bitcoin attacks with less than half of the computational power (draft)," *arXiv preprint arXiv:1312.7013*, 2013.

[8] M. Saad, L. Njilla, C. Kamhoua, and A. Mohaisen, "Countering selfish mining in blockchains," in *2019 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2019, pp. 360–364.

[9] S. Solat and M. Potop-Butucaru, "Brief announcement: Zeroblock: Timestamp-free prevention of block-withholding attack in bitcoin," in *International Symposium on Stabilization, Safety, and Security of Distributed Systems*. Springer, 2017, pp. 356–360.

[10] K. Nicolas, Y. Wang, and G. C. Giakos, "Comprehensive overview of selfish mining and double spending attack countermeasures," in *2019 IEEE 40th Sarnoff Symposium*. IEEE, 2019, pp. 1–6.

[11] R. Zhang and B. Preneel, "Lay down the common metrics: Evaluating proof-of-work consensus protocols' security," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 175–192.

[12] C. Grunspan and R. Pérez-Marco, "On profitability of selfish mining," *arXiv preprint arXiv:1805.08281*, 2018.

[13] P. A. Gagniuc, *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons, 2017.

[14] K. A. Negy, P. R. Rizun, and E. G. Sirer, "Selfish mining re-examined," in *International Conference on Financial Cryptography and Data Security*. Springer, 2020, pp. 61–78.

[15] G. Goren and A. Spiegelman, "Mind the mining," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, 2019, pp. 475–487.

[16] M. Mirkin, Y. Ji, J. Pang, A. Klages-Mundt, I. Eyal, and A. Juels, "Bdos: Blockchain denial-of-service," in *Proceedings of the 2020 ACM SIGSAC conference on Computer and Communications Security*, 2020, pp. 601–619.

[17] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, 2014.

[18] F. Ritz and A. Zugenmaier, "The impact of uncle rewards on selfish mining in ethereum," in *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2018, pp. 50–57.

[19] C. Feng and J. Niu, "Selfish mining in ethereum," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 1306–1316.

[20] "Bitcoin - average mining costs," https://en.macromicro.me/charts/29435/bitcoin-production-total-cost.

[21] A. Dembo, S. Kannan, E. N. Tas, D. Tse, P. Viswanath, X. Wang, and O. Zeitouni, "Everything is a race and nakamoto always wins," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 859–878.

[22] A. Kiayias, N. Leonardos, and D. Zindros, "Mining in logarithmic space," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 3487–3501.

[23] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "Sok: Research perspectives and challenges for bitcoin and cryptocurrencies," in *2015 IEEE symposium on security and privacy*. IEEE, 2015, pp. 104–121.

[24] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden markov decision process," *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.

[25] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol with chains of variable difficulty," in *Annual International Cryptology Conference*. Springer, 2017, pp. 291–323.

[26] "Implementation of optimal selfish mining strategies in bitcoin," https://github.com/nirenzang/Optimal-Selfish-Mining-Strategies-in-Bitcoin/blob/master/README.md.

[27] T. Davis, "Catalan numbers," *Unpublished notes, November*, vol. 26, 2006.

[28] H. Chen, "Interesting series associated with central binomial coefficients, catalan numbers and harmonic numbers." *J. Integer Seq.*, vol. 19, no. 1, pp. 16–1, 2016.

[29] D. H. Lehmer, "Interesting series involving the central binomial coefficient," *The American Mathematical Monthly*, vol. 92, no. 7, pp. 449–457, 1985.

[30] J. Dutka, "The incomplete beta function—a historical profile," *Archive for history of exact sciences*, pp. 11–29, 1981.

Appendix A
Selfish mining strategies

A. Eyal and Sirer's selfish mining strategy

The authors in [4] presented the first selfish mining strategy $\pi^{\texttt{SM1}}$ that increases the attacker's relative revenue. In this paper, the Markov chain is used to analyze the strategy $\pi^{\texttt{SM1}}$. Let $l_{\mathcal{A}}$ and $l_{\mathcal{H}}$ denote the length of the attacker's chain and the length of the honest chain, respectively. The set of actions is composed of four different actions, which are `adopt`, `overwrite`, `match`, and `wait`. `adopt` means the selfish miner leaves his secret chain and continues mining on top of the honest chain. `overwrite` represents that the attacker publishes his secret chain that is longer than the honest chain. `match` means once the honest miners mine a new block, the attacker publishes a conflicting block with the same height. And finally, `wait` means that the attacker continues mining on top of his secret chain. The attacker's strategy is as follows:

- $l_{\mathcal{H}} > l_{\mathcal{A}}$: `adopt`
- $l_{\mathcal{H}} = l_{\mathcal{A}} = 1$: `match`
- $l_{\mathcal{H}} = l_{\mathcal{A}} - 1 \geq 1$: `overwrite`
- Otherwise: `wait`

Formulas for calculating the attacker's relative revenue $\texttt{RelRev}_{\mathcal{A}}(\pi^{\texttt{SM1}})$ and the effective active mining power $M^{\texttt{main-chain}}_{\texttt{SM1}}$ are presented in [4].

B. Optimal selfish mining

The optimal selfish mining strategy $\pi^{\texttt{OSM}}$ introduced in [5] aims to maximize the attacker's relative revenue. The authors have used Markov Decision Process (MDP) to find the optimal strategy. Each state of selfish mining can be represented using a tuple $(l_{\mathcal{A}}, l_{\mathcal{H}}, \texttt{fork})$, where $l_{\mathcal{A}}$ denotes the length of the attacker's chain, $l_{\mathcal{H}}$ is the length of the honest chain, and `fork` gives information regarding the miner of the latest block. The set of actions is similar to $\pi^{\texttt{SM1}}$. The implementation presented in [26] can be used to calculate the attacker's relative revenue $\texttt{RelRev}_{\mathcal{A}}(\pi^{\texttt{OSM}})$ and the effective active mining power $M^{\texttt{main-chain}}_{\texttt{OSM}}$.

Appendix B
Bitcoin DAM

To mine a new block, miners try to find a nonce for which the block hash is smaller than a target TGT, which is computed by the last iteration of DAM at the end of the previous epoch. DAM aims to maintain the block production rate constant, which results in relatively stable transaction throughput regardless of the total mining power available in the network.

To adjust the target TGT based on the network hash rate, a DAM is triggered after every epoch of L_{epoch} main-chain blocks [25]:

$$\mathsf{TGT}_{i+1} = \begin{cases} \mathsf{TGT}_i \cdot \frac{1}{\tau}, & t_i < \frac{1}{\tau} \cdot t_{\text{ideal}} \\ \mathsf{TGT}_i \cdot \tau, & t_i > \tau \cdot t_{\text{ideal}} \\ \mathsf{TGT}_i \cdot \frac{t_i}{t_{\text{ideal}}}, & \text{otherwise} \end{cases}, \tag{52}$$

where i is the epoch number, t_{ideal} is the ideal time duration of an epoch, τ is a dampening filter to prevent rapid changes of TGT, and t_i is the actual time duration of the last epoch, as reported in the blocks. In Bitcoin, $L_{\text{epoch}} = 2016$, $\tau = 4$, and t_{ideal} is two weeks.

Appendix C
Comparison between SISM2 and SHM

In the smart honest mining [15] denoted by SHM, during $\texttt{epoch}_{\texttt{even}}$, attacker $\mathcal{A}$ divides his mining power into two parts: the idle mining power and the honest mining power. We assume the attacker's idle mining power share and honest mining power share are equal to $e\alpha_{\mathcal{A}}$ and $(1-e)\alpha_{\mathcal{A}}$, where $0 \leq e \leq 1$. However, attacker $\mathcal{A}$ mines honestly in $\texttt{epoch}_{\texttt{odd}}$. Therefore, the SHM time-averaged profit is equal to:

$$\begin{aligned} \texttt{Profit}_{\mathcal{A}}(\pi^{\texttt{SHM}}) = & \frac{\lambda K\left(\alpha_{\mathcal{A}} + \frac{(1-e)\alpha_{\mathcal{A}}}{1-e\alpha_{\mathcal{A}}}\right) + e\alpha_{\mathcal{A}} c_{\mathcal{A}} \cdot \frac{1}{1-e\alpha_{\mathcal{A}}}}{1 - e\alpha_{\mathcal{A}} + \frac{1}{1-e\alpha_{\mathcal{A}}}} \\ & - \alpha_{\mathcal{A}} c_{\mathcal{A}} \ . \end{aligned} \tag{53}$$

If the attacker enjoys the highest possible communication capability $\gamma_{\mathcal{A}} = 1$, SISM2 time-averaged profit under $\mathsf{DAM}_1^{\texttt{modified}}$ is the same as smart honest mining time-averaged profit obtained in equation 53. However, if attacker enjoys the highest possible communication capability $\gamma_{\mathcal{A}} = 1$, SISM2 time-averaged profit under $\mathsf{DAM}_2^{\texttt{modified}}$ is equal to:

$$\begin{aligned} & \texttt{Profit}_{\mathcal{A}}(\pi^{\texttt{SISM2}}, \mathsf{DAM}_2^{\texttt{modified}}) = \\ & \frac{\lambda K\left(\frac{\alpha_{\mathcal{A}}}{1-\alpha_{\mathcal{A}}} + \frac{(1-e)\alpha_{\mathcal{A}}}{1-e\alpha_{\mathcal{A}}}\right) + e\alpha_{\mathcal{A}} c_{\mathcal{A}} \cdot \frac{1}{1-e\alpha_{\mathcal{A}}}}{\frac{1-e\alpha_{\mathcal{A}}}{1-\alpha_{\mathcal{A}}} + \frac{1}{1-e\alpha_{\mathcal{A}}}} - \alpha_{\mathcal{A}} c_{\mathcal{A}} \ . \end{aligned} \tag{54}$$

For all the values of e greater than zero, we have: $\texttt{Profit}_{\mathcal{A}}(\pi^{\texttt{SISM2}}, \mathsf{DAM}_2^{\texttt{modified}}) > \texttt{Profit}_{\mathcal{A}}(\pi^{\texttt{SHM}})$. The intuitive reason is that the relative revenue gained in $\texttt{epoch}_{\texttt{even}}$ and the duration of $\texttt{epoch}_{\texttt{even}}$ are the same in both strategies. Therefore, the key difference lies in $\texttt{epoch}_{\texttt{odd}}$. The time-averaged profit during $\texttt{epoch}_{\texttt{odd}}$ is the same for both strategies. However, the duration of $\texttt{epoch}_{\texttt{odd}}$ in SISM2 under $\mathsf{DAM}_2^{\texttt{modified}}$ is longer than that in SHM. Since the time-averaged profit during $\texttt{epoch}_{\texttt{odd}}$ is greater than that during $\texttt{epoch}_{\texttt{even}}$, the longer duration of $\texttt{epoch}_{\texttt{odd}}$ in SISM2 under $\mathsf{DAM}_2^{\texttt{modified}}$ makes SISM2 more profitable than smart honest mining.

Appendix D
Orphan exclusion attack performed by a real-world attacker

Let $\mathcal{A}$ be an attacker who does not possess the predictive capability. To perform the orphan exclusion attack, attacker $\mathcal{A}$ follows the subsequent strategy:

- Keep the adversarial chain secret whenever the length of the adversarial chain is greater than the length of the honest chain.
- Publish the adversarial chain once the length of the honest chain becomes equal to the length of the adversarial chain.

If the attacker's communication capability $\gamma_{\mathcal{A}}$ is equal to 1, the attacker does not risk losing any blocks while performing the orphan exclusion attack. If the epoch end is placed in the middle of one of the chain race iterations, the attack is considered to be successful.

Appendix E
The orphan exclusion attack length under $\mathsf{DAM}_1^{\text{MODIFIED}}$

As the first step toward obtaining the orphan exclusion attack length, we define the terms "chain race" and "longest dominant chain".

Definition 5 (Chain race)**.** *For two adversarial blocks $B_i^{\mathcal{A}}, B_j^{\mathcal{A}} \in S$, where $i \leq j$, the chain race started at $B_i^{\mathcal{A}}$ and ended at $B_j^{\mathcal{A}}$ is the race between $\mathcal{A}$'s private chain and the public chain that satisfies the following properties:*

- *Before $B_i^{\mathcal{A}}$, both the private chain and the public chain share the same chain denoted as $\mathsf{C}^{\lceil B_i^{\mathcal{A}}}$.*
- *$\mathcal{A}$'s private chain is $\mathsf{C}^{\lceil B_i^{\mathcal{A}}} || \{B_i^{\mathcal{A}}, \cdots, B_j^{\mathcal{A}}\}$, where $\{B_i^{\mathcal{A}}, \cdots, B_j^{\mathcal{A}}\}$ is the set of consecutive adversarial blocks in mining sequence S starting at $B_i^{\mathcal{A}}$ and ending at $B_j^{\mathcal{A}}$.*
- *The public chain is $\mathsf{C}^{\lceil B_i^{\mathcal{A}}} || \{B_{i'}^{H}, \cdots, B_{j'}^{H}\}$, where $\{B_{i'}^{H}, \cdots, B_{j'}^{H}\}$ is the set of consecutive honest blocks in mining sequence S starting at $B_{i'}^{H}$ and ending at $B_{j'}^{H}$, where $B_{i'}^{H}$ as well as $B_{j'}^{H}$ are respectively the first honest block after $B_i^{\mathcal{A}}$ and the last honest block before $B_j^{\mathcal{A}}$ in S.*

We say $\mathcal{A}$ wins the chain race starting at $B_i^{\mathcal{A}}$ and ending at $B_j^{\mathcal{A}}$ if the length of the set $\{B_i^{\mathcal{A}}, \cdots, B_j^{\mathcal{A}}\}$ is grater than the length of the set $\{B_{i'}^{H}, \cdots, B_{j'}^{H}\}$. The length of a chain race is defined as the length of the adversarial fork.

The expression "$\mathcal{A}$ wins the chain race starting at $B_i^{\mathcal{A}}$ and ending at $B_j^{\mathcal{A}}$" indicates that if $\mathcal{A}$ forks the main chain at $B_i^{\mathcal{A}}$, he can orphan the honest miners' consecutive blocks mined after $B_i^{\mathcal{A}}$ and before $B_j^{\mathcal{A}}$.

Definition 6 (Longest dominant chain)**.** *The longest dominant chain of an adversarial block $B_i^{\mathcal{A}}$, which is represented by $\mathsf{LDC}(B_i^{\mathcal{A}})$, is a set of consecutive adversarial blocks sampled from the mining sequence S that satisfies the following properties:*

- *$\mathsf{LDC}(B_i^{\mathcal{A}})$ starts at $B_i^{\mathcal{A}}$ and ends at an adversarial block, e.g., $B_j^{\mathcal{A}}$, where $i \leq j$. We have $\mathsf{LDC}(B_i^{\mathcal{A}}) = \{B_i^{\mathcal{A}}, \cdots, B_j^{\mathcal{A}}\}$.*
- *$\mathcal{A}$ wins the chain race starting at $B_i^{\mathcal{A}}$ and ending at $B_j^{\mathcal{A}}$.*
- *There is no $k > j$ such that $\mathcal{A}$ wins the chain race starting at $B_i^{\mathcal{A}}$ and ending at $B_k^{\mathcal{A}}$.*

Let $L^{\mathsf{LDC}}(B_i^{\mathcal{A}})$ denote the length of the longest dominant chain starting at $B_i^{\mathcal{A}}$, i.e., $L^{\mathsf{LDC}}(B_i^{\mathcal{A}}) = |\mathsf{LDC}(B_i^{\mathcal{A}})|$. Note that $\min(L^{\mathsf{LDC}}(B_i^{\mathcal{A}})) = 1$, and this occurs when $\mathsf{LDC}(B_i^{\mathcal{A}})$ comprises only $B_i^{\mathcal{A}}$.

Theorem 1. *The average length of the longest dominant chain starting at an adversarial block, e.g., $B_i^{\mathcal{A}}$, can be calculated as follows:*

$$\mathbb{E}[L^{LDC}(B_i^{\mathcal{A}})] = \frac{2\alpha(1-\alpha)}{(1-2\alpha)^2} + 1 \ . \tag{55}$$

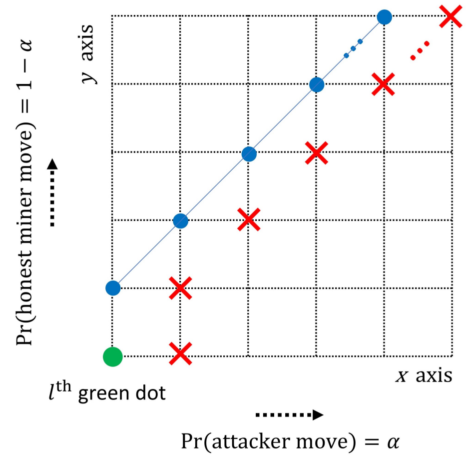

Fig. 7. The mining path starting at $(0,0)$ and never reaching the line $y = x$ at $x \geq 1$

Note that through this paper, we use a two-dimensional (x, y)-grid to depict the chain race as shown in Figure 8. The mining sequence is represented by a path on this grid. Whenever the honest miners mine a new block, the mining path moves one step up, and whenever $\mathcal{A}$ mines a new block, the mining path moves one step to the right. The grid-based chain race representation provides us with a strong tool to analyze different event probabilities within blockchains. To prove Theorem 2, we first need to present Lemma 1.

Lemma 1. *The probability of the event that the mining path starting at $(0,0)$ never reaches the line $y = x$ for $x \geq 1$, which is denoted by P_{NR}, is equal to $1 - 2\alpha$.*

Proof of Lemma 1. This lemma can be proved using straightforward methods such as a normal random walk. However, as a warm-up, we use the grid-based chain race representation to prove this theorem. Later in this paper, we will use the grid-based approach to prove other complicated theorems where the straightforward methods are insufficient. Consider the mining path represented in Figure 7. We first calculate the probability of the complementary event. The complementary event occurs when the mining path for at least one time reaches one of the cross marks depicted in Figure 7. At the start, if the mining path moves one to the right, it reaches the cross mark in point $(1,0)$. To reach the other cross marks, the path needs to move one up and reach the first blue dot in point $(0,1)$. The number of paths that start from the blue dot in $(0,1)$ and reach one of the cross marks for the first time at (i,i), for $i \geq 1$, is equal to the number of paths from $(0,1)$ to the blue dot in $(i-1,i)$ without passing below the line $y = x+1$. The latter one is equal to the $i-1^{\text{th}}$ Catalan number [27]. The i^{th} Catalan number, denoted by C_i, can be calculated as $C_i = \frac{1}{i+1}\binom{2i}{i}$ [28]. Therefore, we have:

$$\overline{P_{\mathsf{NR}}} = \alpha + \alpha(1-\alpha)\sum_{i=0}^{\infty} C_i(\alpha(1-\alpha))^i = 2\alpha \ . \tag{56}$$

The result of the series above is presented in [29]. Finally, we have:

$$P_{\mathsf{NR}} = 1 - \overline{P_{\mathsf{NR}}} = 1 - 2\alpha \ . \tag{57}$$

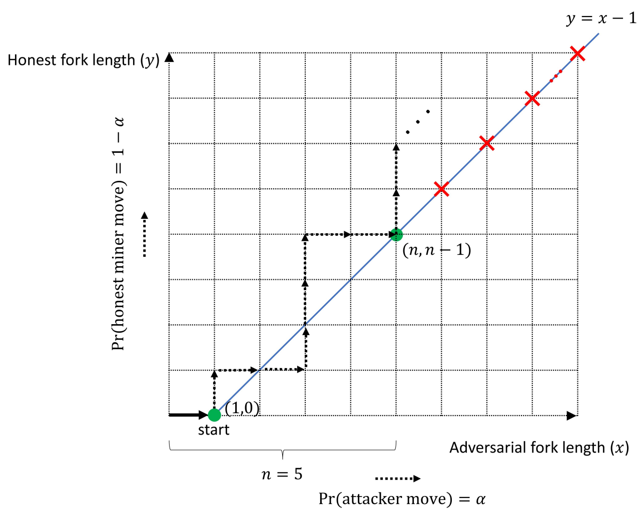

Fig. 8. Chain race representation

□

Proof of Theorem 1. The chain race starting at $B_i^{\mathcal{A}}$ is depicted in Figure 8. Since the chain race starts with an adversarial block, i.e., $B_i^{\mathcal{A}}$, the mining path always moves to the right, i.e., point $(1,0)$, as the first step. Assume $L^{\mathsf{LDC}}(B_i^{\mathcal{A}}) = n$, where $n \geq 1$. This indicates that the mining path reaches the line $y = x - 1$ for the last time at point $(n, n-1)$. The number of paths from point $(1,0)$ to point $(n, n-1)$ is equal to $\binom{2(n-1)}{n-1}$. Therefore, the probability that the mining path starting at point $(0,1)$ reaches the point $(n, n-1)$ is equal to $\binom{2(n-1)}{n-1}\big(\alpha(1-\alpha)\big)^{n-1}$. According to Lemma 1, the probability that the mining path starting at $(n, n-1)$ never reaches the line $y = x - 1$ again is equal to $1 - 2\alpha$. As a result, the probability that $L^{\mathsf{LDC}}(B_i^{\mathcal{A}}) = n$ is equal to $\binom{2(n-1)}{n-1}\big(\alpha(1-\alpha)\big)^{n-1}(1-2\alpha)$. Finally, the expected value of $L^{\mathsf{LDC}}(B_i^{\mathcal{A}})$ can be obtained as follows:

$$\begin{aligned}
\mathbb{E}\big[L^{\mathsf{LDC}}(B_i^{\mathcal{A}})\big] &= \sum_{n=1}^{\infty} n\binom{2(n-1)}{n-1}(\alpha(1-\alpha))^{n-1}(1-2\alpha) \\
&= \sum_{n=1}^{\infty}(n-1)\binom{2(n-1)}{n-1}(\alpha(1-\alpha))^{n-1}(1-2\alpha) \\
&+ \sum_{n=1}^{\infty}\binom{2(n-1)}{n-1}(\alpha(1-\alpha))^{n-1}(1-2\alpha) \\
&= \sum_{n=0}^{\infty} n\binom{2n}{n}(\alpha(1-\alpha))^{n}(1-2\alpha) \\
&+ \sum_{n=1}^{\infty}\binom{2n}{n}(\alpha(1-\alpha))^{n}(1-2\alpha) = \frac{2\alpha(1-\alpha)}{(1-2\alpha)^3}(1-2\alpha) \\
&+ \frac{1}{1-2\alpha}(1-2\alpha) = \frac{2\alpha(1-\alpha)}{(1-2\alpha)^2} + 1 \ .
\end{aligned} \tag{58}$$

Formulas to solve the series above, which involve central binomial coefficients, are presented in [29]. □

In order to have a successful orphan exclusion attack at the end of $\mathtt{epoch}_i$, there should exist an adversarial block $B^{\mathcal{A}}$ in $\mathtt{epoch}_i$ whose longest dominant chain includes the epoch end. The existence of such a longest dominant chain $\mathsf{LDC}(B^{\mathcal{A}})$ indicates that a subset of adversarial blocks within $\mathsf{LDC}(B^{\mathcal{A}})$ forms the last main-chain blocks of $\mathtt{epoch}_i$. In other words, there is no honest block that can get added to the main chain after $\mathsf{LDC}(B^{\mathcal{A}})$ within $\mathtt{epoch}_i$. Consequently, the honest blocks that get orphaned by the adversarial fork $\mathsf{LDC}(B^{\mathcal{A}})$ cannot be reported and included in the modified DAM. Let $BS = \{B_1^{\mathcal{A}}, \cdots, B_N^{\mathcal{A}}\}$ be the set of adversarial blocks within $\mathtt{epoch}_i$ whose longest dominant chains can result in a successful orphan exclusion attack. Note that not all the adversarial blocks in $\mathtt{epoch}_i$ can be the starting block of a successful orphan exclusion attack. For instance, the orphan exclusion attack starting at one of the adversarial blocks that is far from the epoch end has almost no chance of being successful since the epoch end would not be included in that longest dominant chain. To maximize his profit, the attacker should perform the longest possible orphan exclusion attack. Therefore, the length of the orphan exclusion attack is equal to the length of the longest LDC among the set $\{\mathsf{LDC}(B_i^{\mathcal{A}}) | B_i^{\mathcal{A}} \in BS\}$.

Here, we first define a promising block within $\mathtt{epoch}_i$, characterized by having one of the longest LDCs that can result in a successful orphan exclusion attack. The longest dominant chain of this block can help us calculate a lower bound and upper bound for the length of the orphan exclusion attack.

Definition 7 (Block height). *Let $S = \{B_0, B_1, B_2, \cdots\}$ be the mining sequence, where the block miners can be honest or adversarial. Let h_i denote the height of block B_i. We define the height of genesis block B_0 to be equal to 0. The height of the block B_i for $i \geq 1$ can be obtained as follows:*

$$h_i = \begin{cases} h_{i-1} - 1 \ , & \textit{if block } B_i \textit{ is adversarial.} \\ h_{i-1} + 1 \ , & \textit{if block } B_i \textit{ is honest.} \end{cases} \tag{59}$$

The height of the block B_i with respect to the block B_j is defined to be equal to $h_i - h_j$.

Definition 8 (Epoch promising block). *Let $B_{end} \in S$ represent the last block of $\mathtt{epoch}_i$ under the condition that all the blocks within the mining sequence get added to the main chain. The promising block of $\mathtt{epoch}_i$ is defined to be an adversarial block that has the highest height among all the adversarial blocks before and including block B_{end}. If multiple of these blocks exist, the one that is farthest from B_{end} is considered the promising block.*

As an example, assume $S_i = \{\cdots, B_{n-9}^{\mathcal{H}}, B_{n-8}^{\mathcal{H}}, B_{n-7}^{\mathcal{H}}, B_{n-6}^{\mathcal{H}}, B_{n-5}^{\mathcal{A}}, B_{n-4}^{\mathcal{A}}, B_{n-3}^{\mathcal{H}}, B_{n-2}^{\mathcal{A}}, B_{n-1}^{\mathcal{A}}, B_n^{\mathcal{H}}\}$ represent the mining sequence of $\mathtt{epoch}_i$. The height of blocks within mining sequence S_i with respect to the last block of $\mathtt{epoch}_i$, i.e., $B_n^{\mathcal{H}}$, is as follows: $\{\cdots, -1, 0, 1, 2, 1, 0, 1, 0, -1, 0\}$. As can be seen, the highest height among adversarial blocks belongs to block $B_{n-5}^{\mathcal{A}}$ with $h_{n-5} = 1$. Note that since the adversarial mining power share is less than half, the height of blocks has an increasing pattern in a long-term perspective. This indicates that the height of blocks before $B_n^{\mathcal{H}}$ and after $B_n^{\mathcal{H}}$ with respect to $B_n^{\mathcal{H}}$ converge to $-\infty$ and ∞, respectively.

Therefore, if assuming that there is no other adversarial block before $B^{\mathcal{A}}_{n-5}$ whose height with respect to $B^{\mathcal{H}}_{n}$ is greater than or equal to $h_{n-5} = 1$, $B^{\mathcal{A}}_{n-5}$ is the promising block of $\mathtt{epoch}_i$.

Lemma 2. *Let $B^{\mathcal{A}}_i$ and $B^{\mathcal{A}}_j$ denote two adversarial blocks, where $j > i$. The attacker can win the chain race starting at $B^{\mathcal{A}}_i$ and ending at $B^{\mathcal{A}}_j$ if and only if $h_i \geq h_j$.*

Proof. Let n denote the number of adversarial blocks in the adversarial fork starting at $B^{\mathcal{A}}_i$ and ending at $B^{\mathcal{A}}_j$. $h_i \geq h_j$ indicates that the number of honest blocks in the mining sequence between two adversarial blocks $B^{\mathcal{A}}_i$ and $B^{\mathcal{A}}_j$ is less than or equal to $n-1$. Therefore, the attacker can win the chain race starting at $B^{\mathcal{A}}_i$ and ending at $B^{\mathcal{A}}_j$. The reverse direction can be proved in an analogous way. □

We first explain why the longest dominant chain starting at the epoch promising block is among the longest LDCs that can result in a successful orphan exclusion attack. Since the promising block of an epoch is located near the epoch end, its longest dominant chain has a relatively high probability of ending after and including the epoch end. Therefore, the LDC starting at the epoch promising block has a high chance of leading to a successful orphan exclusion attack. Moreover, the LDC starting at the epoch promising block is among the longest LDCs that start prior to the epoch end and end afterward. According to Lemma 2, the LDC starting at the promising block continues as long as the height of subsequent adversarial blocks remains less than or equal to that of the promising block. Since the promising block has the highest height among the adversarial blocks before the epoch end, a relatively long list of blocks after the epoch end is required to exceed the epoch's promising height.

To find a lower bound and an upper bound for the length of the orphan exclusion attack, we use the grid-based representation of the mining path. In the grid-based representation, we assume that point $(0, 0)$ represents the end of $\mathtt{epoch}_i$. This implies that the mining path in the upper right quadrant and in the lower left quadrant belong to the mining sequence within $\mathtt{epoch}_{i+1}$ and $\mathtt{epoch}_i$, respectively. The mining path in $\mathtt{epoch}_{i+1}$ starts at point $(0, 0)$ and moves forward within the upper right quadrant. For $\mathtt{epoch}_i$ which is located in the lower left quadrant, we define the term "reversed mining path". The reversed path originates at point $(0, 0)$, which corresponds to the last block of $\mathtt{epoch}_i$, and moves backward within the lower left quadrant, towards the previous blocks in $\mathtt{epoch}_i$. Let moving to point P on the mining path represent a block B. To find the height of block B with respect to the last block of $\mathtt{epoch}_i$ in the grid representation, one should draw a line with slop 1 at point P. The y-intercept of the line is equal to the height of block B with respect to the last block of $\mathtt{epoch}_i$.

Lemma 3. *Let $r \geq 1$ and $s \geq 0$. The probability that the **reversed** mining path (within the lower left quadrant) starting at $(0, 0)$ reaches the line $y = x + r$ for the last time at point $(-s - r, -s)$ without never passing the line $y = x + r$ is denoted by $P^1(r, s)$ and can be obtained as follows:*

$$P^1(r, s) = \binom{r + 2s}{s} \frac{r + 1}{r + s + 1} \alpha^{r+s}(1-\alpha)^s(1-2\alpha) \ . \quad (60)$$

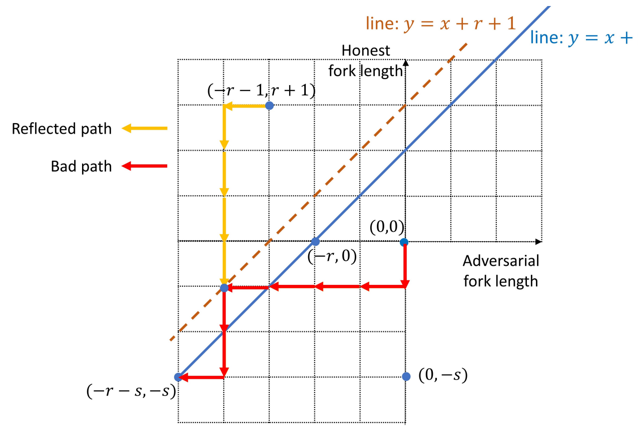

Fig. 9. Mining path representation

Proof. We first find the number of paths from point $(0, 0)$ to point $(-s-r, -s)$ without passing the line $y = x+r$. Consider the mining path depicted in Figure 9. The total number of paths from point $(0, 0)$ to point $(-s-r, -s)$ is equal to $\binom{r+2s}{s}$. Some of these $\binom{r+2s}{s}$ paths, however, pass the line $y = x+r$, which we refer to as bad paths. Being a bad path implies that the path reaches the line $y = x + r + 1$ before reaching the point $(-s-r, -s)$. For each bad path P, we define the initial part of the path to be equal to the part of the path P before reaching the line $y = x + r + 1$. For each bad path P, we define a new path P' by reflecting the initial part of the path across the line $y = x + r + 1$ as depicted in Figure 9. By doing so, we can generate a one-to-one mapping between the bad paths and the reflected paths that start at point $(-r-1, r+1)$ and end at point $(-s - r, -s)$. Therefore, the number of bad paths is equal to $\binom{r+2s}{s-1}$. Consequently, the number of paths from point $(0, 0)$ to point $(-s - r, -s)$ without passing the line $y = x + r$ can be obtained as follows:

$$\binom{r + 2s}{s} - \binom{r + 2s}{s - 1} = \binom{r + 2s}{s} \frac{r + 1}{r + s + 1} \ . \quad (61)$$

The probability that the reversed mining path starting at $(0, 0)$ reaches point $(-s-r, -s)$ without passing the line $y = x+r$ is equal to $\binom{r+2s}{s}\frac{r+1}{r+s+1}\alpha^{r+s}(1-\alpha)^s$. According to Lemma 1, the probability that the reversed mining path starting at $(-s-r, -s)$ never reaches the line $y = x + r$ again in the future is equal to $1 - 2\alpha$. Therefore, the probability that the reversed mining path starting at $(0, 0)$ reaches the line $y = x+r$ for the last time at point $(-s - r, -s)$ without never passing the line $y = x+r$ is equal to $\binom{r+2s}{s}\frac{r+1}{r+s+1}\alpha^{r+s}(1-\alpha)^s(1-2\alpha)$. □

Lemma 4. *Let $r \geq 0$ and $s \geq 1$. The probability that the reversed mining path (within the lower left quadrant) starting at $(0, 0)$ reaches the point $(0, -r)$ and afterward reaches the line $y = x - r$ for the last time at point $(-s - r, -s)$ without never passing the line $y = x - r$ is denoted by $P^2(r, s)$ and can be obtained as follows:*

$$P^2(r, s) = \binom{2s}{s} \frac{1}{s + 1} \alpha^s(1-\alpha)^{r+s}(1-2\alpha) \ . \quad (62)$$

Proof. The probability that the reversed mining path starting at $(0, 0)$ reaches the point $(0, -r)$ is equal to $(1 - \alpha)^r$. The number of paths from point $(0,-r)$ to point $(-s,-r-s)$ without passing the line $y=x-r$ is equal to the s^{th} Catalan number, which can be obtained as $\binom{2s}{s}\frac{1}{s+1}$. The probability that the reversed mining path starting at $(0,0)$ reaches the point $(0,-r)$ and afterwards reaches the point $(-s-r,-s)$ without never passing the line $y = x-r$ is equal to $\binom{2s}{s}\frac{1}{s+1}\alpha^s(1-\alpha)^{s+r}$. According to Lemma 1, the probability that the reversed mining path starting at $(-s-r,-s)$ never reaches the line $y = x-r$ again in the future is equal to $1-2\alpha$. Therefore, the probability that the reversed mining path starting at $(0,0)$ reaches the point $(0,-r)$ and afterward reaches the line $y = x-r$ for the last time at point $(-s-r,-s)$ without never passing the line $y = x-r$ is equal to $\binom{2s}{s}\frac{1}{s+1}\alpha^s(1-\alpha)^{r+s}(1-2\alpha)$. □

Lemma 5. *Let $r \geq 1$ and $k \geq 0$. The probability that the mining path (within the upper right quadrant) starting at $(0,0)$ passes the line $y = x+r$ for the last time at point $(k,k+r)$ is denoted by $P^3(r,k)$ and can be obtained as follows:*

$$P^3(r,k) = \binom{r+2k-1}{k}\alpha^k(1-\alpha)^{r+k-1}(1-2\alpha) \ . \quad (63)$$

Proof. The number of paths from point $(0,0)$ to point $(k,k+r-1)$ is equal to $\binom{r+2k-1}{k}$. Therefore, the probability that the mining path starting at $(0,0)$ reaches the point $(k,k+r-1)$ is equal to $\binom{r+2k-1}{k}\alpha^k(1-\alpha)^{r+k-1}$. The event that the mining path starting at $(0,0)$ passes the line $y = x+r$ for the last time at point $(k,k+r)$ is equivalent to the event that the mining path starting at $(0,0)$ reaches the point $(k,k+r-1)$ and afterward never reaches the line $y = x+r-1$ again. According to Lemma 1, the probability that the mining path starting at $(k,k+r-1)$ never reaches the line $y = x+r-1$ again in the future is equal to $1-2\alpha$. Therefore, the probability that the mining path starting at $(0,0)$ passes the line $y = x+r$ for the last time at point $(k,k+r)$ is equal to $\binom{r+2k-1}{k}\alpha^k(1-\alpha)^{r+k-1}(1-2\alpha)$. □

Lemma 6. *Let $r \geq 0$ and $k \geq 1$. The probability that the mining path (within the upper right quadrant) starting at $(0,0)$ gets below the line $y = x-r$ (reaches the line $y = x-r-1$) and then passes the line $y = x-r$ for the last time at point $(k+r,k)$ is denoted by $P^4(r,k)$ and can be obtained as follows:*

$$P^4(r,k) = \binom{r+2k-1}{r+k}\alpha^{r+k}(1-\alpha)^{k-1}(1-2\alpha) \ . \quad (64)$$

The proof of Lemma 6 is similar to the proof of Lemma 5.

1) A lower bound for the length of the orphan exclusion attack under $\mathsf{DAM}_1^{modified}$: Assume $\mathsf{LDC}(B_i^*)$ contains $N_{i+1}^{\mathcal{A}}$ adversarial blocks belonging to $\texttt{epoch}_{i+1}$ and orphans $N_i^{\mathcal{H}}$ honest blocks belonging to $\texttt{epoch}_i$. If $N_{i+1}^{\mathcal{A}} \geq N_i^{\mathcal{H}}$, the epoch end is included in $\mathsf{LDC}(B_i^*)$, and therefore, $\mathsf{LDC}(B_i^*)$ results in a successful orphan exclusion attack. Let $L^{\textsf{OEA-min}}$ denote a lower bound for the length of the orphan exclusion attack. We consider as follows:

- If $\mathsf{LDC}(B_i^*)$ does not result in a successful orphan exclusion attack, $L^{\textsf{OEA-min}} = 0$.
- If $\mathsf{LDC}(B_i^*)$ results in a successful orphan exclusion attack, $L^{\textsf{OEA-min}} = L^{\mathsf{LDC}}(B_i^*)$.

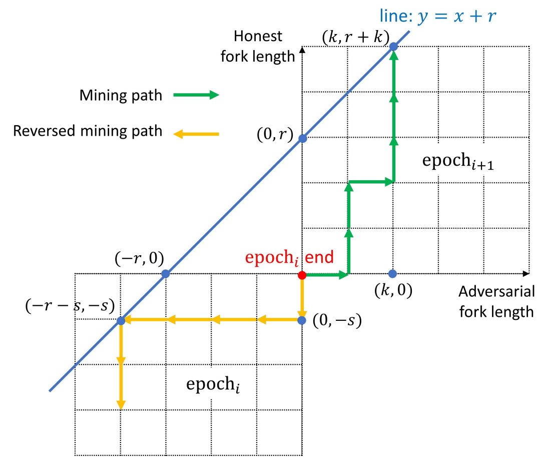

Fig. 10. Mining path representation

Let B_i^* and h_i^* denote the $\texttt{epoch}_i$'s promising block and its height with respect to the last block of $\texttt{epoch}_i$, respectively. We analyze the orphan exclusion attack in both scenarios where h_i^* is non-negative and negative.

Assume the scenario in which $h_i^* = r-1$, where $r \geq 1$. In this case, h_i^* is non-negative. The illustration of this scenario is depicted in Figure 10. Assume further that the promising block is located where the mining path moves from point $(-r-s,-s)$ to point $(-r-s+1,-s)$. This implies that $\mathsf{LDC}(B_i^*)$ starts at point $(-r-s,-s)$. The event that $\mathsf{LDC}(B_i^*)$ starts at point $(-r-s,-s)$ is equivalent to the event that the reversed mining path starting at point $(0,0)$ reaches the line $y = x+r$ at $x < 0$ for the last time at point $(-r-s,-s)$ without never passing the line $y = x+r$ at $x < 0$. $\mathsf{LDC}(B_i^*)$ ends at an adversarial block that is the last adversarial block located below the line $y = x+r$. Let the mining path in the upper right quadrant pass the line $y = x+r$ for the last time at point $(k,k+r)$. This implies that the length of $\mathsf{LDC}(B_i^*)$ is equal to $r+s+k$. Note that the longest dominant chain of the promising block B_i^* does not necessarily lead to a successful orphan exclusion attack. To have a successful attack, the final block of $\texttt{epoch}_i$ should be included in $\mathsf{LDC}(B_i^*)$. Based on our assumption, $\mathsf{LDC}(B_i^*)$ starts when there are $r+2s$ blocks left to the end of the epoch. To include the epoch end, the length of $\mathsf{LDC}(B_i^*)$ should be equal to or greater than $r+2s$. Note that $\mathsf{LDC}(B_i^*)$ result in orphaning s honest blocks of $\texttt{epoch}_i$. Therefore, to complete $r+2s$ remaining blocks, the number of adversarial blocks within $\mathsf{LDC}(B_i^*)$ belonging to $\texttt{epoch}_{i+1}$ should be equal or greater than s. As a result, to have a successful orphan exclusion attack starting at B_i^*, the condition $k \geq s$ should be satisfied.

If $k \geq s$, the lower bound of the orphan exclusion attack length is equal to $L^{\textsf{OEA-min}} = r+s+k$. Otherwise, $L^{\textsf{OEA-min}} = 0$. The probability that the reversed mining path starting at point $(0,0)$ in the lower left quadrant reaches the line $y = x+r$ for the last time at point $(-r-s,-s)$ without never passing the line $y = x+r$ is equal to $P^1(r,s)$ presented in Lemma 3. The probability that the mining path starting at point $(0,0)$ in the upper right quadrant passes the line $y = x+r$ for the last time at point $(k,k+r)$ is equal to $P^3(r,k)$ presented in

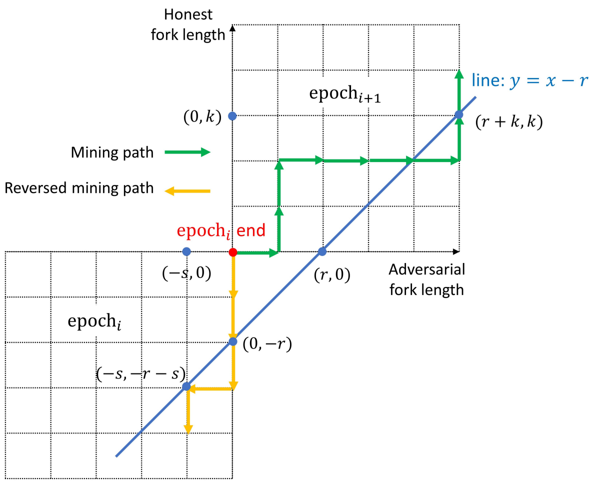

Fig. 11. Mining path representation

Lemma 5. Therefore, in the case where h_i^* is non-negative, the expected lower bound for the length of the orphan exclusion attack can be calculated as follows:

$$\mathbb{E}(L_1^{\text{OEA-min}}) = \sum_{r=1}^{\infty}\sum_{s=0}^{\infty}\sum_{k=s}^{\infty}(r+s+k)P^1(r,s)P^3(r,k) \ . \tag{65}$$

Assume the scenario in which $h_i^* = -r-1$, where $r \geq 0$. In this case, h_i^* is negative. The illustration of this scenario is depicted in Figure 11. Assume further that the promising block is located where the mining path moves from point $(-s, -r-s)$ to point $(-s+1, -r-s)$. The event that the promising block is located where the mining path moves from point $(-s, -r-s)$ to point $(-s+1, -r-s)$ is equivalent to the event that the reversed mining path starting at $(0, 0)$ reaches the point $(0, -r)$ and afterward reaches the line $y = x - r$ for the last time at point $(-s, -r-s)$ without never passing the line $y = x - r$, the probability of which is equal to $P^2(r, s)$ presented in Lemma 4. $\mathsf{LDC}(B_i^*)$ ends at an adversarial block that is the last adversarial block located below the line $y = x - r$. To ensure that $\mathsf{LDC}(B_i^*)$ leads to a successful orphan exclusion attack, the mining path in the upper right quadrant needs to get below the line $y = x-r$; and if assuming that the mining path passes the line $y = x - r$ for the last time at point $(r + k, k)$, the inequality $k \geq s$ must hold. $k \geq s$ guarantees that $\mathsf{LDC}(B_i^*)$ ends at the subsequent epoch. In this scenario, if $k \geq s$, the lower bound of the orphan exclusion attack length is equal to $L^{\text{OEA-min}} = r + s + k$. Otherwise, $L^{\text{OEA-min}} = 0$. The probability that the mining path starting at point $(0, 0)$ in the upper right quadrant passes the line $y = x - r$ for the last time at point $(k + r, k)$ is equal to $P^4(r, k)$ presented in Lemma 6. Therefore, in the case where h_i^* is negative, the expected lower bound for the length of the orphan exclusion attack can be calculated as follows:

$$\mathbb{E}(L_2^{\text{OEA-min}}) = \sum_{r=0}^{\infty}\sum_{s=1}^{\infty}\sum_{k=s}^{\infty}(r+s+k)P^2(r,s)P^4(r,k) \ . \tag{66}$$

Finally, the lower bound of the orphan exclusion attack can be obtained as follows:

$$\mathbb{E}(L^{\text{OEA-min}}) = \mathbb{E}(L_1^{\text{OEA-min}}) + \mathbb{E}(L_2^{\text{OEA-min}}) \ , \tag{67}$$

where $\mathbb{E}(L_1^{\text{OEA-min}})$ and $\mathbb{E}(L_2^{\text{OEA-min}})$ are calculated in equations 65 and 66, respectively.

2) An upper bound for the length of the orphan exclusion attack under $\mathsf{DAM}_1^{modified}$: Assume $\mathsf{LDC}(B_i^*)$ contains $N_{i+1}^{\mathcal{A}}$ adversarial blocks belonging to $\texttt{epoch}_{i+1}$ and orphans $N_i^{\mathcal{H}}$ honest blocks belonging to $\texttt{epoch}_i$. Let $L^{\text{OEA-max}}$ denote an upper bound for the length of the orphan exclusion attack. We consider as follows:

- If $N_{i+1}^{\mathcal{A}} \leq N_i^{\mathcal{H}}$, $L^{\text{OEA-max}} = L^{\mathsf{LDC}}(B_i^*)$.
- If $N_{i+1}^{\mathcal{A}} > N_i^{\mathcal{H}}$, $L^{\text{OEA-max}} = L^{\mathsf{LDC}}(B_i^*) + (N_{i+1}^{\mathcal{A}} - N_i^{\mathcal{H}} - 1)$.

We first explain why $L^{\text{OEA-max}}$ defined above is an upper bound for the length of the orphan exclusion attack. Let $h_i^* = r - 1$, where $\mathsf{LDC}(B_i^*)$ starts at point $(-r-s, -s)$ and ends at point $(k, k+r)$. This indicates that $N_{i+1}^{\mathcal{A}} = k$ and $N_i^{\mathcal{H}} = s$.

If $k \leq s$, we claim there is no adversarial block before B_i^* whose longest dominant chain can result in a successful orphan exclusion attack. Let B denote a block before B_i^*. Since B_i^* is the promising block of the epoch, the number of honest blocks belonging to $\texttt{epoch}_i$ that get orphaned by $\mathsf{LDC}(B)$ is greater than s, and the number of adversarial blocks belonging to $\texttt{epoch}_{i+1}$ that are included in $\mathsf{LDC}(B)$ is less than or equal to k. As we have $k \leq s$, $\mathsf{LDC}(B)$ cannot result in a successful orphan exclusion attack. The longest dominant chains of adversarial blocks after B_i^* may result in a successful attack; however, their length is shorter than $L^{\mathsf{LDC}}(B_i^*)$. Therefore, in case where $k \leq s$, $L^{\mathsf{LDC}}(B_i^*)$ is an upper bound for the length of orphan exclusion attack.

If $k > s$, there may exist plenty of adversarial blocks before B_i^* whose longest dominant chain is longer than $\mathsf{LDC}(B_i^*)$. Let B denote a block before B_i^*. The number of adversarial blocks belonging to $\texttt{epoch}_{i+1}$ that are included in $\mathsf{LDC}(B)$ is less than or equal to k. Therefore, if the number of honest blocks belonging to $\texttt{epoch}_i$ that are included in $\mathsf{LDC}(B)$ exceeds k, $\mathsf{LDC}(B)$ cannot result in a successful orphan exclusion attack. There exist s honest blocks between B_i^* and the $\texttt{epoch}_i$'s end. Therefore, the number of honest blocks between B and B_i^* should be less than or equal to $k - s$. Knowing that there exist at most $k-s$ honest blocks between B and B_i^*, we want to find the maximum number of adversarial blocks that can exist between blocks B and B_i^*, including block B itself. Since B_i^* is the promising block of $\texttt{epoch}_i$, the reversed mining path starting at point $(-r-s, -s)$ within the lower left quadrant never reaches the line $y = x + r$ again. As a result, while the reversed mining path moves $k-s$ steps downward, it can move at most $k - s - 1$ steps to the left, showing that block B is at most $k - s - 1$ adversarial blocks away from block B_i^*.

Therefore, the upper bound for the length of the orphan

TABLE II
THE LENGTH OF THE ORPHAN EXCLUSION ATTACK UNDER $\mathsf{DAM}_1^{\text{MODIFIED}}$

mining power share	0.25	0.3	0.35	0.4	0.45
$\mathbb{E}[L^{\text{OEA-min}}]$	1.2859	2.5963	5.7596	15.6328	63.6105
$\mathbb{E}[L^{\text{OEA}}]$ (simulation result)	1.4594	2.9917	6.8328	18.3697	87.0262
$\mathbb{E}[L^{\text{OEA-max}}]$	2.2870	4.8142	10.9720	30.2100	123.0720

exclusion attack can be obtained as follows:

$$\begin{aligned}\mathbb{E}[L^{\text{OEA-max}}] &= \sum_{r=1}^{\infty}\sum_{s=0}^{\infty}\sum_{k=0}^{s}(r+s+k)P^1(r,s)P^3(r,k)\\ &+\sum_{r=1}^{\infty}\sum_{s=0}^{\infty}\sum_{k=s+1}^{\infty}(r+2k-1)P^1(r,s)P^3(r,k)\\ &+\sum_{r=0}^{\infty}\sum_{s=1}^{\infty}\sum_{k=0}^{s}(r+s+k)P^2(r,s)P^4(r,k)\\ &+\sum_{r=0}^{\infty}\sum_{s=1}^{\infty}\sum_{k=s+1}^{\infty}(r+2k-1)P^2(r,s)P^4(r,k)\ .\end{aligned} \tag{68}$$

In Table II, we provide a comparison among the lower bound calculated in equation 67, the upper bound calculated in equation 68, and the average length of the orphan exclusion attack under $\mathsf{DAM}_1^{\text{modified}}$ obtained from the simulation.

APPENDIX F
THE ORPHAN EXCLUSION ATTACK LENGTH UNDER $\mathsf{DAM}_2^{\text{MODIFIED}}$

In this section, we first discuss why the orphan exclusion attack under $\mathsf{DAM}_2^{\text{modified}}$ can be more severe than that under $\mathsf{DAM}_1^{\text{modified}}$. Then, we calculate an upper bound for the length of the orphan exclusion attack under $\mathsf{DAM}_2^{\text{modified}}$.

When there is no selfish mining and orphan exclusion attack, after every L_{epoch} blocks in the mining sequence S, an epoch ends and the DAM is applied. Under difficulty adjustment mechanism $\mathsf{DAM}_2^{\text{modified}}$, performing selfish mining during the epoch can shift the epoch end in mining sequence S and, consequently, affect the length of the orphan exclusion attack at the end of the epoch. Let L^S_{epoch} represent the number of consecutive blocks consumed from the sequence S to generate one epoch. According to the epoch definition in $\mathsf{DAM}_2^{\text{modified}}$, the epoch ends when the number of main-chain blocks gets equal to L_{epoch}. If there is no selfish mining and no orphan exclusion attack, L_{epoch} is equal to L^S_{epoch}. However, if the attacker performs selfish mining during the epoch and tries to orphan some of the honest blocks, $L_{\text{epoch}} \leq \mathbb{E}(L^S_{\text{epoch}}) \leq \frac{L_{\text{epoch}}}{1-\alpha_{\mathcal{A}}}$. The lower bound occurs when there is no selfish mining, and the upper bound occurs when $\mathcal{A}$ orphans one honest block for each of his blocks. This shows that there is a level of freedom for $\mathcal{A}$ to decide when to end the epoch. By orphaning the honest blocks during the epoch, $\mathcal{A}$ can adjust the end of the epoch in a way that increases the length of the orphan exclusion attack. For instance, consider the mining sequence depicted in Figure 12. If $\mathcal{A}$ does not perform selfish mining during the epoch, he cannot impose a successful orphan exclusion attack at the end of the epoch. However, if $\mathcal{A}$ decides to orphan 7 honest blocks during the epoch, he can shift the epoch end 7 blocks ahead and impose a successful attack. This shows that adjusting the epoch end can help $\mathcal{A}$ to impose a longer orphan exclusion attack. Note that, under difficulty adjustment mechanism $\mathsf{DAM}_1^{\text{modified}}$, performing selfish mining during the epoch cannot shift the epoch end in mining sequence S. This is due to the fact that in $\mathsf{DAM}_1^{\text{modified}}$, the epoch ends when the total number of main-chain and orphan blocks gets equal to L_{epoch}.

High-level proof overview We first present a road map for calculating an upper bound for the average length of the orphan exclusion attack under $\mathsf{DAM}_2^{\text{modified}}$:

- In the first step, we calculate the probability that the length of the longest dominant chain for an adversarial block is less than or equal to a specific amount in Theorem 2.
- In the second step, we assume there exist several independent adversarial blocks that are sampled from separate mining sequences. Each of these adversarial blocks has its own longest dominant chain. Using the probability calculated in Theorem 2, we calculate the average length of the longest chain among all the available longest dominant chains in Theorem 3.
- In the next step, we assume there exists a set of consecutive adversarial blocks sampled from the same mining sequence. Each of these adversarial blocks has its own longest dominant chain. However, these longest dominant chains are dependent on each other. Using the result of Theorem 3, we calculate an upper bound for the average length of the longest chain among all the dependent longest dominant chains in Theorem 4.
- In the last step, using Theorem 4, we find an upper bound for the average length of the orphan exclusion attack under $\mathsf{DAM}_2^{\text{modified}}$ in Theorem 5.

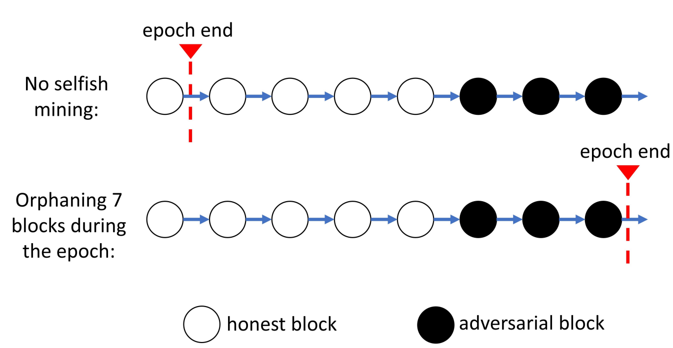

Fig. 12. The effect of selfish mining during the epoch on the end of the epoch

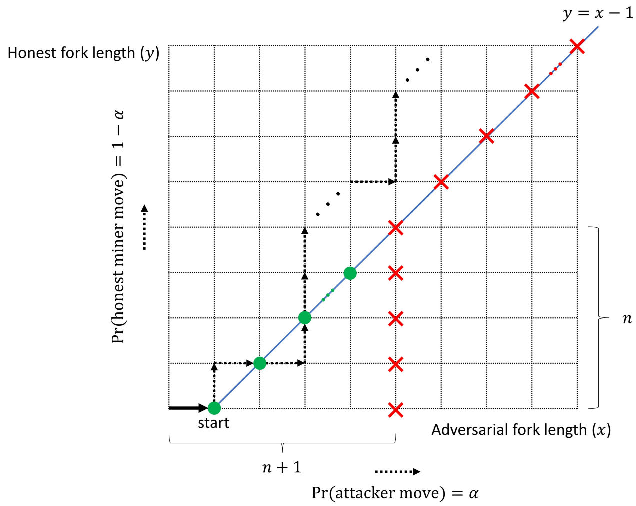

Fig. 13. Chain race representation

A. The length of the longest dominant chain

Theorem 2. *The probability of the event that the length of the longest dominant chain for an adversarial block, e.g., $B_i^{\mathcal{A}}$, is less than or equal to n can be calculated as follows:*

$$Pr(L^{\mathsf{LDC}}(B_i^{\mathcal{A}}) \le n) = 1 - 2\mathcal{I}_{\alpha}(n,n) \ , \tag{69}$$

Where $\mathcal{I}$ is the regularized incomplete beta function.

Proof of Theorem 2. The chain race starting at $B_i^{\mathcal{A}}$ is depicted in Figure 13. Since the chain race starts with an adversarial block, i.e., $B_i^{\mathcal{A}}$, the first move is always to the right. In order to have $L^{\mathsf{LDC}}(B_i^{\mathcal{A}}) \le n$, the mining path should never reach the cross marks depicted in Figure 13. Otherwise, there exists an adversarial dominant chain whose length is greater than n. All the acceptable paths pass through at least one of the green dots. Assume $L^{\mathsf{LDC}}(B_i^{\mathcal{A}}) = l$, where $l \le n$. This means the last time that the mining path visits the line $y = x - 1$ happens at the l^{th} green dot in point $(l, l-1)$. The probability that $\mathcal{A}$ can win a chain race whose length is equal to l is denoted by P_{WCR}^{l}. Winning a chain race with length l is equivalent to the event that the mining path reaches the l^{th} green dot. P_{WCR}^{l} can be calculated using the equation below:

$$P_{\mathsf{WCR}}^{l} = \binom{2(l-1)}{l-1}(\alpha(1-\alpha))^{l-1} \ . \tag{70}$$

The probability of the event that after the mining path reaches the l^{th} green dot in the diagonal line $y = x - 1$, it never reaches the line again is equal to P_{NR}, calculated in Lemma 1. Note that P_{NR} is independent of l. Having P_{WCR}^{l} and P_{NR}, we can calculate the probability that the length of the longest dominant chain is equal to l:

$$\begin{aligned}\Pr(L^{\mathsf{LDC}}(B_i^{\mathcal{A}}) = l) &= P_{\mathsf{WCR}}^{l} \cdot P_{\mathsf{NR}} \\ &= \binom{2(l-1)}{l-1}(\alpha(1-\alpha))^{l-1}(1-2\alpha) \ .\end{aligned} \tag{71}$$

Finally, the probability that the length of the longest dominant chain at $B_i^{\mathcal{A}}$ is less than or equal to n can be calculated as below:

$$\begin{aligned}\Pr(L^{\mathsf{LDC}}(B_i^{\mathcal{A}}) \le n) &= \sum_{l=1}^{n}\binom{2(l-1)}{l-1}(\alpha(1-\alpha))^{l-1}(1-2\alpha) \\ &= \sum_{i=0}^{n-1}\binom{2i}{i}(\alpha(1-\alpha))^{i}(1-2\alpha) = 1 - 2\mathcal{I}_{\alpha}(n,n) \ .\end{aligned} \tag{72}$$

In this part, we try to simplify equation 72. We define a function $F(\cdot)$ as $F(x) = \sum_{i=0}^{n-1}\binom{2i}{i}x^{i}$. Taking the derivative of $F(x)$ results in:

$$\begin{aligned}F'(x) &= \sum_{i=1}^{n-1} i\binom{2i}{i}x^{i-1} = \sum_{i=0}^{n-2}2(2i+1)\binom{2i}{i}x^{i} \\ &= -n\binom{2n}{n}x^{n-1} + \sum_{i=0}^{n-1}2(2i+1)\binom{2i}{i}x^{i} \ .\end{aligned} \tag{73}$$

Therefore, we obtain the following differential equation:

$$F'(x) = -n\binom{2n}{n}x^{n-1} + 2F(x) + 4xF'(x) \ . \tag{74}$$

By solving equation 74, we obtain:

$$F(x) = \frac{1}{\sqrt{1-4x}}\Big(1 - n\binom{2n}{n}\int_{0}^{x}\frac{u^{n-1}}{\sqrt{1-4u}}\,du\Big) \ . \tag{75}$$

Using the variable substitution $u = z(1-z)$, we can modify the integral above as below:

$$\begin{aligned}F(x) &= \frac{1 - n\binom{2n}{n}\int_{0}^{\frac{1-\sqrt{1-4x}}{2}} z^{n-1}(1-z)^{n-1}\,dz}{\sqrt{1-4x}} \\ &= \frac{1}{\sqrt{1-4x}}\Big(1 - n\binom{2n}{n}\mathcal{B}\big(\frac{1-\sqrt{1-4x}}{2}; n, n\big)\Big) \ .\end{aligned} \tag{76}$$

In the equation above, $\mathcal{B}(\cdot)$ is the incomplete beta function which is defined as [30]:

$$\mathcal{B}(x; a, b) = \int_{0}^{x} t^{a-1}(1-t)^{b-1}\,dt \ . \tag{77}$$

The incomplete beta function is a generalization of the complete beta function which is defined as $\mathcal{B}(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}\,dt$. $\mathcal{I}(\cdot)$ is the regularized incomplete beta function and is defined as below:

$$\mathcal{I}_{x}(a,b) = \frac{\mathcal{B}(x;a,b)}{\mathcal{B}(a,b)} \ . \tag{78}$$

Since we have $n\binom{2n}{n}\mathcal{B}(x; n, n) = 2\mathcal{I}_x(n,n)$, equation 76 can be simplified as follows:

$$F(x) = \frac{1}{\sqrt{1-4x}}\Big(1 - 2\mathcal{I}_{\frac{1-\sqrt{1-4x}}{2}}(n,n)\Big) \ . \tag{79}$$

Since $\Pr(L_i \le n) = (1-2\alpha)F\big(\alpha(1-\alpha)\big)$, we finally obtain the following result:

$$\Pr(L^{\mathsf{LDC}}(B_i^{\mathcal{A}}) \le n) = 1 - 2\mathcal{I}_{\alpha}(n,n) \ . \tag{80}$$

$\square$

B. The longest chain among independent LDCs

Definition 9 (Longest LDC)**.** *Assume there exist N adversarial blocks denoted by $B_i^{\mathcal{A}}$, $1 \leq i \leq N$. These adversarial blocks can be sampled from the same or separate mining sequences. The longest LDC of the adversarial block set $BS = \{B_1^{\mathcal{A}}, \cdots, B_N^{\mathcal{A}}\}$, which is denoted by LLDC(BS), is the longest chain(s) among the set $\{\textsf{LDC}(B_i^{\mathcal{A}}) | B_i^{\mathcal{A}} \in BS\}$.*

$L_{BS}^{\textsf{LLDC}}$ denotes the length of the longest LDC chain of the adversarial block set BS, i.e., $L_{BS}^{\textsf{LLDC}} = |\textsf{LLDC}(BS)|$.

Theorem 3. *Assume there exist N independent mining sequences, where each of them starts with an adversarial block denoted by $B_i^{\mathcal{A}}$, $1 \leq i \leq N$. Let $BS = \{B_1^{\mathcal{A}}, \cdots, B_N^{\mathcal{A}}\}$ and LDC$(B_i^{\mathcal{A}})$ denote the longest dominant chain of adversarial block $B_i^{\mathcal{A}}$. The average length of the longest LDC among the set $\{\textsf{LDC}(B_i^{\mathcal{A}}) | B_i^{\mathcal{A}} \in BS\}$ can be calculated as follows:*

$$\mathbb{E}\Big[L_{BS}^{\textsf{LLDC}}\Big] = \sum_{n=1}^{\infty} n \textsf{Pr}\Big(L_{BS}^{\textsf{LLDC}} = n\Big) \ , \tag{81}$$

where

$$\textsf{Pr}\Big(L_{BS}^{\textsf{LLDC}} = n\Big) = \begin{cases} (1-2\alpha)^N, & n = 1 \\ \big(1 - 2\mathcal{I}_\alpha(n,n)\big)^N - \big(1 - 2\mathcal{I}_\alpha(n-1,n-1)\big)^N, & n > 1 \end{cases} . \tag{82}$$

Proof of Theorem 3. $L_{BS}^{\textsf{LLDC}} = 1$ means that the length of $\textsf{LDC}(B_i^{\mathcal{A}}) = 1$ for all $B_i^{\mathcal{A}} \in BS$. The event that the length of the longest dominant chain is 1 happens when the chain race starting at $(1,0)$ never reaches the line $y = x-1$ again. Therefore, using Lemma 1, we have $\textsf{Pr}(L^{\textsf{LDC}}(B_i^{\mathcal{A}}) = 1) = 1 - 2\alpha$, and since there exist N adversarial blocks in BS, we have $\textsf{Pr}\big(L_{BS}^{\textsf{LLDC}} = 1\big) = (1-2\alpha)^N$.

In order to have $L_{BS}^{\textsf{LLDC}} = n$, where $n > 1$, there should exist one or more adversarial blocks whose longest dominant chain length is equal to n, and for the other remaining adversarial blocks, the longest dominant chain length should be less than n. Assume P_n and $P_{\leq n}$ respectively represent the probability that the length of the longest dominant chain is exactly equal to n and the probability that the length of the longest dominant chain is less than or equal to n. For $n > 1$, we have:

$$\begin{aligned} \textsf{Pr}\Big(L_{BS}^{\textsf{LLDC}} = n\Big) &= \sum_{i=1}^{N} \binom{N}{i} P_n^i P_{\leq n-1}^{N-i} \\ &= (P_n + P_{\leq n-1})^N - P_{\leq n-1}^N = P_{\leq n}^N - P_{\leq n-1}^N \\ &= \big(1 - 2\mathcal{I}_\alpha(n,n)\big)^N - \big(1 - 2\mathcal{I}_\alpha(n-1,n-1)\big)^N \quad , \quad n > 1 \ . \end{aligned} \tag{83}$$

The second equality holds based on the Binomial theorem, and the last equality is obtained from Theorem 2. Having $\textsf{Pr}\big(L_{BS}^{\textsf{LLDC}} = n\big)$ for all values of n, the expected length can be calculated as $\sum_{n=1}^{\infty} n \textsf{Pr}\big(L_{BS}^{\textsf{LLDC}} = n\big)$. □

C. The longest chain among dependent LDCs:

One can use Theorem 3 to calculate $\mathbb{E}\big[L_{BS}^{\textsf{LLDC}}\big]$, provided that the mining sequences of all the adversarial blocks in BS are independent. However, if we sample a set of N consecutive adversarial blocks from the mining sequence S to generate the adversarial block set BS, the longest dominant chains of those N adversarial blocks in BS are dependent on each other. Due to the dependency, the average length of the longest LDC will significantly decrease. In this part, we aim to calculate an upper bound for $\mathbb{E}\big[L_{BS}^{\textsf{LLDC}}\big]$, where BS is a set of consecutive adversarial blocks sampled from the same mining sequence.

Theorem 4. *Assume that $L_{BS'}^{\textsf{LLDC}}(N; ind.)$ represents $|\textsf{LLDC}(BS')|$, where BS' consists of N adversarial blocks whose longest dominant chains are independent of each other, and $L_{BS}^{\textsf{LLDC}}(N; dep., S)$ represents $|\textsf{LLDC}(BS)|$, where BS consists of N consecutive adversarial blocks sampled from the mining sequence S whose longest dominant chains are dependent on each other. We have:*

$$\mathbb{E}\Big[L_{BS}^{\textsf{LLDC}}(N; dep., S)\Big] \leq \mathbb{E}\Bigg[L_{BS'}^{\textsf{LLDC}}\Big(N/\big(\frac{1-\alpha}{1-2\alpha}\big)^2; ind.\Big)\Bigg] \ , \tag{84}$$

where $\mathbb{E}\Bigg[L_{BS'}^{\textsf{LLDC}}\Big(N/\big(\frac{1-\alpha}{1-2\alpha}\big)^2; ind.\Big)\Bigg]$ can be calculated using Theorem 3.

First, we review some definitions and lemmas.

Definition 10 (Chain race advantage)**.** *For a chain race starting at the adversarial block $B_i^{\mathcal{A}}$ and ending at $B_j^{\mathcal{A}}$, where $C_{\mathcal{A}}^{B_i^{\mathcal{A}}, B_j^{\mathcal{A}}} = \mathcal{C}^{\lceil B_i^{\mathcal{A}}} || \{B_i^{\mathcal{A}}, \cdots, B_j^{\mathcal{A}}\}$ and $C_{\mathcal{H}}^{B_i^{\mathcal{A}}, B_j^{\mathcal{A}}} = \mathcal{C}^{\lceil B_i^{\mathcal{A}}} || \{B_{i'}^{H}, \cdots, B_{j'}^{H}\}$ are respectively $\mathcal{A}$'s private chain and the public chain, the chain race advantage is denoted by $\delta^{B_i^{\mathcal{A}}, B_j^{\mathcal{A}}}$ and defined as follows:*

$$\delta^{B_i^{\mathcal{A}}, B_j^{\mathcal{A}}} = |C_{\mathcal{A}}^{B_i^{\mathcal{A}}, B_j^{\mathcal{A}}}| - |C_{\mathcal{H}}^{B_i^{\mathcal{A}}, B_j^{\mathcal{A}}}| \ . \tag{85}$$

Note that $B_{i'}^{H}$ as well as $B_{j'}^{H}$ are respectively the first honest block after $B_i^{\mathcal{A}}$ and the last honest block before $B_j^{\mathcal{A}}$ in S.

Lemma 7. *If $\delta^{B_i^{\mathcal{A}}, B_j^{\mathcal{A}}} > 0$, then $\textsf{LLDC}\big(\{B_i^{\mathcal{A}}, B_j^{\mathcal{A}}\}\big) = \textsf{LDC}\big(B_i^{\mathcal{A}}\big)$.*

Proof. If $\delta^{B_i^{\mathcal{A}}, B_j^{\mathcal{A}}} > 0$, we have $\textsf{LDC}(B_j^{\mathcal{A}}) \subset \textsf{LDC}(B_i^{\mathcal{A}})$. Therefore, $\textsf{LLDC}\big(\{B_i^{\mathcal{A}}, B_j^{\mathcal{A}}\}\big)$ is always equal to the longest dominant chain starting at $B_i^{\mathcal{A}}$. □

Lemma 8. *Let S be a mining sequence starting with $B_i^{\mathcal{A}}$. The average number of adversarial blocks, e.g., $B_j^{\mathcal{A}}$, in S that satisfy $\delta^{B_i^{\mathcal{A}}, B_j^{\mathcal{A}}} > 0$ is equal to $\Big(\frac{1-\alpha}{1-2\alpha}\Big)^2$.*

Proof. Consider the chain race starting at $B_i^{\mathcal{A}}$ and ending at $B_j^{\mathcal{A}}$ whose length is equal to n. In order to have $\delta^{B_i^{\mathcal{A}}, B_j^{\mathcal{A}}} > 0$, the first time that the mining path reaches the line $x = n$ should happen in a point with $y < n$. Note that the chain race always passes through the point (1,0). For $n \geq 2$, the probability that a mining path starts at $(1,0)$ and reaches the line $x = n$ for the first time in point (n,i) is equal to $\binom{i+n-2}{n-2}\alpha^{n-1}(1-\alpha)^i$. Therefore, the probability that the advantage of a chain race with length n, which is denoted by δ^n, is greater than 0 is calculated as follows:

$$\mathsf{Pr}(\delta^n > 0) = \sum_{i=0}^{n-1} \binom{i+n-2}{n-2} \alpha^{n-1} (1-\alpha)^i \ . \tag{86}$$

The average number of adversarial blocks that satisfy $\delta^{B_i^{\mathcal{A}}, B_j^{\mathcal{A}}} > 0$ is equal to

$$\sum_{n=1}^{\infty} \mathsf{Pr}(\delta^n > 0) = 1 + \sum_{n=2}^{\infty} \sum_{i=0}^{n-1} \binom{i+n-2}{n-2} \alpha^{n-1} (1-\alpha)^i \ . \tag{87}$$

We use σ to represent the double sum above. We have:

$$\begin{aligned} \sigma &= \sum_{n=0}^{\infty} \sum_{i=0}^{n+1} \binom{i+n}{n} \alpha^{n+1} (1-\alpha)^i \\ &= \sum_{n=0}^{\infty} \alpha^{n+1} \sum_{i=0}^{n+1} \binom{i+n}{n} (1-\alpha)^i \ . \end{aligned} \tag{88}$$

We define a function $G(\cdot)$ as $G(x) = \sum_{i=0}^{n+1} \binom{i+n}{n} x^i$. Taking the derivative, we obtain:

$$\begin{aligned} G'(x) &= \sum_{i=1}^{n+1} i \binom{i+n}{n} x^{i-1} = \sum_{i=0}^{n} (n+1+i) \binom{i+n}{n} x^i \\ &= -(2n+2) \binom{2n+1}{n+1} x^{n+1} + \sum_{i=0}^{n+1} (n+1+i) \binom{i+n}{n} x^i \ . \end{aligned} \tag{89}$$

Therefore, we obtain the following differential equation:

$$G'(x) = -(2n+2) \binom{2n+1}{n+1} x^{n+1} + (n+1) F(x) + x F'(x) \ . \tag{90}$$

Solving equation 90 results in:

$$\begin{aligned} G(x) &= \frac{\left(1 - (2n+2)\binom{2n+1}{n+1} \int_0^x u^{n+1} (1-u)^n \, du\right)}{(1-x)^{n+1}} \\ &= \frac{1 - (2n+2)\binom{2n+1}{n+1} \mathcal{B}(x; n+2, n+1)}{(1-x)^{n+1}} \ . \end{aligned} \tag{91}$$

Since for positive integers w and z, we have $B(z,w) = \frac{z+w}{zw\binom{z+w}{z}}$, we can write $G(x)$ as follows:

$$\begin{aligned} G(x) &= \frac{\left((2n+2)\binom{2n+1}{n+1} \int_{1-x}^1 u^{n+1} (1-u)^n \, du\right)}{(1-x)^{n+1}} \\ &= \frac{1}{(1-x)^{n+1}} \left(2(2n+1) \binom{2n}{n} \int_{1-x}^1 u^{n+1} (1-u)^n \, du \right) \ . \end{aligned} \tag{92}$$

Therefore, we can calculate σ as follows:

$$\begin{aligned} \sigma &= \sum_{n=0}^{\infty} \alpha^{n+1} F(1-\alpha) \\ &= \sum_{n=0}^{\infty} 2(2n+1) \binom{2n}{n} \int_{1-\alpha}^1 u^{n+1} (1-u)^n \, du \\ &= \int_{1-\alpha}^1 2u \sum_{n=0}^{\infty} \binom{2n}{n} \big(u(1-u)\big)^n \, du \\ &+ \int_{1-\alpha}^1 4u \sum_{n=0}^{\infty} n \binom{2n}{n} \big(u(1-u)\big)^n \, du \\ &= \int_{1-\alpha}^1 \Big(\frac{2u}{-(1-2u)} + \frac{8u^2(1-u)}{-(1-2u)^3} \Big) \, du \\ &= \int_{1-\alpha}^1 -\frac{2u}{(1-2u)^3} \, du = \frac{\alpha(1-\alpha)}{(1-2\alpha)^2} + \frac{\alpha}{1-2\alpha} \ . \end{aligned} \tag{93}$$

Finally,

$$\sum_{n=1}^{\infty} \mathsf{Pr}(\delta^n > 0) = \Big(\frac{1-\alpha}{1-2\alpha} \Big)^2 \ . \tag{94}$$

□

To better understand the effect of LDC dependencies on $\mathbb{E}\big[L_{BS}^{\mathsf{LLDC}}\big]$, consider the following example. Let BS_1 be a set of consecutive adversarial blocks sampled from the mining sequence S, which starts with $B_i^{\mathcal{A}}$, and $BS_2 = \{B_j^{\mathcal{A}} \in S | \delta^{B_i^{\mathcal{A}}, B_j^{\mathcal{A}}} > 0\} \setminus \{B_i^{\mathcal{A}}\}$. Assume BS_1 is sufficiently long to have $BS_2 \subset BS_1$. In this case, according to Lemma 7, we have $\mathsf{LLDC}(BS_1) = \mathsf{LLDC}(BS_1 \setminus BS_2)$. Therefore, according to Lemma 8, if we have already considered the longest dominant chain of $B_i^{\mathcal{A}}$ in calculation of $|\mathsf{LLDC}(BS_1)|$, in average, there exist $\left(\frac{1-\alpha}{1-2\alpha}\right)^2 - 1$ other adversarial blocks in BS_1, i.e., members of BS_2, whose longest dominant chain has no chance to increase $|\mathsf{LLDC}(BS_1)|$.

Definition 11 (Future advantage)**.** *For an adversarial block $B_i^{\mathcal{A}}$, the future advantage is denoted by $\Delta^{B_i^{\mathcal{A}}\rceil}$ and defined as follows:*

$$\Delta^{B_i^{\mathcal{A}}\rceil} = \max_{i \le k} \delta^{B_i^{\mathcal{A}}, B_k^{\mathcal{A}}} \ . \tag{95}$$

Lemma 9. *If assuming the number of paths in a mining grid from start point $(0,0)$ to the point $(s, r+s)$ without passing through the line $y = x + r$ is denoted by C_s^r, then we have:*

$$\begin{aligned} \sum_{s=0}^{\infty} C_s^r \big(\alpha(1-\alpha)\big)^s &= \frac{1}{(1-\alpha)^{r+1}} \ , \\ \sum_{s=0}^{\infty} s C_s^r \big(\alpha(1-\alpha)\big)^s &= \frac{(r+1)\alpha}{(1-2\alpha)(1-\alpha)^{r+1}} \ . \end{aligned} \tag{96}$$

Proof. The set $PS_r = \big\{(s, r+s) | s \in \mathbb{W}\big\}$ consists of all the points on the line $y = x + r$. Since we have $1 - \alpha > \alpha$, all the mining paths will finally pass through the line $y = x + r$ in one of the points in the set PS_r. Therefore, the probabilities that a mining path passes the line $y = x + r$ for the first time in $(s, r+s)$ for all $s \in \mathbb{W}$ should sum up to 1. The probability that the mining path starting in $(0,0)$ passes through the line

$y = x + r$ for the first time in $(s, r+s) \in PS_r$ is equal to $(1-\alpha)^{r+1} C^r_s \big(\alpha(1-\alpha)\big)^s$. Therefore, we have:

$$(1-\alpha)^{r+1} \sum_{s=0}^{\infty} C^r_s \big(\alpha(1-\alpha)\big)^s = 1 \Rightarrow$$
$$\sum_{s=0}^{\infty} C^r_s \big(\alpha(1-\alpha)\big)^s = \frac{1}{(1-\alpha)^{r+1}} \ . \tag{97}$$

To prove the second equality in Lemma 9, we use the variable substitution $\alpha(1-\alpha) = x$ in the equality above. We have:

$$\sum_{s=0}^{\infty} C^r_s x^s = \frac{1}{\left(\frac{1+\sqrt{1-4x}}{2}\right)^{r+1}} \ . \tag{98}$$

By taking the derivative from both sides, we obtain:

$$\sum_{s=0}^{\infty} s C^r_s x^{s-1} = \frac{r+1}{\left(\frac{1+\sqrt{1-4x}}{2}\right)^{r+2} \sqrt{1-4x}} \ . \tag{99}$$

By multiplying both sides to x and substituting $x = \alpha(1-\alpha)$, we obtain:

$$\sum_{s=0}^{\infty} s C^r_s \big(\alpha(1-\alpha)\big)^s = \frac{(r+1)\alpha}{(1-2\alpha)(1-\alpha)^{r+1}} \ . \tag{100}$$

□

Lemma 10. *The probability of the event that $\Delta^{B_i^{\mathcal{A}}\rceil} = r$ is equal to $(1-2\alpha)\frac{\alpha^{r-1}}{(1-\alpha)^r}$.*

Proof. $\Delta^{B_i^{\mathcal{A}}\rceil} = r$ means that the chain race starting at $B_i^{\mathcal{A}}$ reaches the line $y = x - r$ but never passes it. Note that a chain race always starts at point $(1, 0)$. The number of paths from point $(1, 0)$ to point $(r+s, s)$ without passing through the line $y = x - r$ is the same as C^{r-1}_s. Therefore, the probability of reaching the point $(r+s, s)$ from point $(1, 0)$ without passing the line $y = x - r$ is equal to $\alpha^{r-1} C^{r-1}_s \big(\alpha(1-\alpha)\big)^s$. The probability that once the mining path reaches the point $(r+s, s)$ on the line $y = x - r$, it never reaches the line again is $1 - 2\alpha$. Thus, the probability of reaching the line $y = x - r$ for the last time in the point $(r+s, s)$ is equal to $\alpha^{r-1} C^{r-1}_s \big(\alpha(1-\alpha)\big)^s (1-2\alpha)$. Finally, the probability that a chain race reaches the line $y = x - r$ but never passes it can be calculated as follows using Lemma 9:

$$\alpha^{r-1} \sum_{s=0}^{\infty} C^{r-1}_s \big(\alpha(1-\alpha)\big)^s (1-2\alpha) = (1-2\alpha)\frac{\alpha^{r-1}}{(1-\alpha)^r} \ . \tag{101}$$

□

Lemma 11. *The average future advantage of an adversarial block is calculated as follows:*

$$\mathbb{E}\Big[\Delta^{B_i^{\mathcal{A}}\rceil}\Big] = 1 + \frac{\alpha}{1-2\alpha} \ . \tag{102}$$

Proof. We just need to calculate the expected value of $\Delta^{B_i^{\mathcal{A}}\rceil}$ over all the values of r. Using Lemma 10, we have:

$$\mathbb{E}(\Delta^{B_i^{\mathcal{A}}\rceil}) = \sum_{r=1}^{\infty} r.\mathsf{Pr}(\Delta^{B_i^{\mathcal{A}}\rceil} = r)$$
$$= \sum_{r=1}^{\infty} r.(1-2\alpha)\frac{\alpha^{r-1}}{(1-\alpha)^r} = 1 + \frac{\alpha}{1-2\alpha} \ . \tag{103}$$

□

Lemma 12. *Let $B_i^{\mathcal{A}}$ and $B_j^{\mathcal{A}}$ be two adversarial blocks sampled from the same mining sequence, where $j < i$. We have:*

$$\delta^{B_j^{\mathcal{A}}, B_i^{\mathcal{A}}} + \Delta^{B_i^{\mathcal{A}}\rceil} - 1 > 0 \iff \mathsf{LDC}(B_j^{\mathcal{A}}) \cap \mathsf{LDC}(B_i^{\mathcal{A}}) \neq \varnothing. \tag{104}$$

Proof. $\delta^{B_j^{\mathcal{A}}, B_i^{\mathcal{A}}} + \Delta^{B_i^{\mathcal{A}}\rceil} - 1 > 0$ means that there exist a block, e.g, $B_k^{\mathcal{A}}$ with $i \leq k$, that satisfies both $\delta^{B_j^{\mathcal{A}}, B_k^{\mathcal{A}}} > 0$ and $\delta^{B_i^{\mathcal{A}}, B_k^{\mathcal{A}}} = \Delta^{B_i^{\mathcal{A}}\rceil} > 0$. Therefore, the block set $\{B_i^{\mathcal{A}}, \cdots, B_k^{\mathcal{A}}\}$ is a common subset of both $\mathsf{LDC}(B_j^{\mathcal{A}})$ and $\mathsf{LDC}(B_i^{\mathcal{A}})$. The reverse direction can be proved in an analogous way. □

Proof of Theorem 4. It is obvious that:

$$\mathbb{E}\Big[L_{BS}^{\mathsf{LLDC}}(1; \text{dep.}, S)\Big] = \mathbb{E}\Big[L_{BS'}^{\mathsf{LLDC}}(1; \text{ind.})\Big] \ . \tag{105}$$

We first prove:

$$\mathbb{E}\Big[L_{BS}^{\mathsf{LLDC}}\Big(\big(\frac{1-\alpha}{1-2\alpha}\big)^2; \text{dep.}, S\Big)\Big] \leq \mathbb{E}\Big[L_{BS'}^{\mathsf{LLDC}}(2; \text{ind.})\Big] \ . \tag{106}$$

Let $BS_3 = \{B_i^{\mathcal{A}}, B_{i-1}^{\mathcal{A}}, \cdots\}$ be a set of consecutive adversarial blocks sampled from the mining sequence S, whose indexes are ordered in a descending way. Let $B_{i-\ell}^{\mathcal{A}}$ represent the first block in BS_3 that satisfies $\delta^{B_j^{\mathcal{A}}, B_i^{\mathcal{A}}} + \Delta^{B_i^{\mathcal{A}}\rceil} - 1 \leq 0$. Let $BS_4 = \{B_{i-\ell+1}^{\mathcal{A}}, \cdots, B_{i-1}^{\mathcal{A}}, B_i^{\mathcal{A}}\}$. In this case:

$$\begin{aligned}&\forall\, B_j^{\mathcal{A}} \in BS_4 : \ \delta^{B_j^{\mathcal{A}}, B_i^{\mathcal{A}}} + \Delta^{B_i^{\mathcal{A}}\rceil} - 1 > 0 \xrightarrow{\text{Lemma 12}} \\ &\forall\, B_j^{\mathcal{A}} \in BS_4,\ \mathsf{LDC}(B_j^{\mathcal{A}}) \cap \mathsf{LDC}(B_i^{\mathcal{A}}) \neq \varnothing \ .\end{aligned} \tag{107}$$

Thus, the longest dominant chain of each of the adversarial blocks in BS_4 has an intersection with the longest dominant chain of block $B_i^{\mathcal{A}}$. This means that for all $B_j^{\mathcal{A}} \in BS_4$, the longest dominant chains of $B_j^{\mathcal{A}}$ and $B_i^{\mathcal{A}}$ are dependent on each other. Let $B_{i'}^{\mathcal{A}}$ be an adversarial block that has sampled from a separate mining sequence S', where the mining sequences S and S' are independent, and $BS' = \{B_{i'}^{\mathcal{A}}, B_i^{\mathcal{A}}\}$. Therefore, we have:

$$\begin{aligned}&\forall\, B_j^{\mathcal{A}} \in BS_4 : \\ &\mathbb{E}\Big[\big|\mathsf{LLDC}\big(\{B_j^{\mathcal{A}}, B_i^{\mathcal{A}}\}\big)\big|\Big] \leq \mathbb{E}\Big[\big|\mathsf{LLDC}\big(\{B_{i'}^{\mathcal{A}}, B_i^{\mathcal{A}}\}\big)\big|\Big] \\ &\Rightarrow \mathbb{E}\Big[\big|\mathsf{LLDC}\big(BS_4\big)\big|\Big] \leq \mathbb{E}\Big[L_{BS'}^{\mathsf{LLDC}}(2; \text{ind.})\Big] \\ &\Rightarrow \mathbb{E}\Big[L_{BS_4}^{\mathsf{LLDC}}(|BS_4|; \text{dep.}, S)\Big] \leq \mathbb{E}\Big[L_{BS'}^{\mathsf{LLDC}}(2; \text{ind.})\Big] \ .\end{aligned} \tag{108}$$

Note that $|BS_4| = \ell$, Therefore, we just need to show $\mathbb{E}[\ell] = \big(\frac{1-\alpha}{1-2\alpha}\big)^2$.

Consider the chain race starting at $B_{i-\ell}^{\mathcal{A}}$. Assume $\Delta^{B_i^{\mathcal{A}}\rceil} = r$. To have $B_{i-\ell}^{\mathcal{A}}$ be the first block in $BS_3 = \{B_i^{\mathcal{A}}, B_{i-1}^{\mathcal{A}}, \cdots\}$ that satisfies $\delta^{B_j^{\mathcal{A}}, B_i^{\mathcal{A}}} + r - 1 \leq 0$, block $B_i^{\mathcal{A}}$ should be the first adversarial block that appears after that the mining path reaches the line $y = x + r$ for the first time. Note that a chain race always starts from $(1, 0)$. The number of paths from point $(1, 0)$ to the point $(s, s+r-1)$ for $s \in \mathbb{N}$ without passing through the line $y = x + r - 1$ is equal to C^r_{s-1}. Therefore, starting from point $(1, 0)$, the number of paths to reach the line $y = x + r$ for the first time at the point $(s, s+r)$ is equal

TABLE III
COMPARISON BETWEEN SIMULATION AND THEORETICAL RESULTS

Mining power share (α)	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45
Simulation result: $\mathbb{E}\left[L_{BS}^{\mathsf{LLDC}}\left(\lfloor(\frac{1-\alpha}{1-2\alpha})^2\rfloor;\text{dep.},S\right)\right]$	1.22	1.51	1.93	2.55	3.57	5.42	9.41	20.78	82.16
Theoretical result: $\mathbb{E}\left[L_{BS}^{\mathsf{LLDC}}(2;\text{ind.})\right]$	1.31	1.67	2.11	2.72	3.99	6.00	9.74	21.60	82.83

to C_{s-1}^r. We aim to find the average distance to the y axis of the point where the mining path reaches the line $y = x + r$ for the first time, i.e., $\mathbb{E}[\ell]\Big|_{\Delta^{B_i^{\mathcal{A}}\rceil}=r}$:

$$\mathbb{E}[\ell]\Big|_{\Delta^{B_i^{\mathcal{A}}\rceil}=r} = \sum_{s=1}^{\infty} sC_{s-1}^r\alpha^{s-1}(1-\alpha)^{s+r}$$
$$= \sum_{s=0}^{\infty}(s+1)C_s^r\alpha^s(1-\alpha)^{s+r+1}$$
$$= (1-\alpha)^{r+1}\left(\sum_{s=0}^{\infty}C_s^r\alpha(1-\alpha)^s + \sum_{s=0}^{\infty}sC_s^r\alpha(1-\alpha)^s\right) . \tag{109}$$

Using Lemma 9, we have:

$$\mathbb{E}[\ell]\Big|_{\Delta^{B_i^{\mathcal{A}}\rceil}=r} = 1 + \frac{(r+1)\alpha}{(1-2\alpha)} . \tag{110}$$

By taking expected value over variable r and using Lemma 10, we can find $\mathbb{E}(\ell)$ as follows:

$$\mathbb{E}[\ell] = \sum_{r=1}^{\infty}\mathbb{E}[\ell]\Big|_{\Delta^{B_i^{\mathcal{A}}\rceil}=r}.\mathsf{Pr}(\Delta^{B_i^{\mathcal{A}}\rceil}=r)$$
$$= \sum_{r=1}^{\infty}\left(1+\frac{(r+1)\alpha}{(1-2\alpha)}\right)\left((1-2\alpha)\frac{\alpha^{r-1}}{(1-\alpha)^r}\right) \tag{111}$$
$$= 1 + \frac{\alpha}{1-2\alpha} + \frac{\alpha(1-2\alpha)}{(1-2\alpha)^2} = \left(\frac{1-\alpha}{1-2\alpha}\right)^2 .$$

Now assume M is the greatest number that satisfies:

$$\mathbb{E}\left[L_{BS}^{\mathsf{LLDC}}\Big(M;\text{dep.},S\Big)\right] \le \mathbb{E}\left[L_{BS'}^{\mathsf{LLDC}}(N;\text{ind.})\right] . \tag{112}$$

Using the same approach, we can show:

$$\mathbb{E}\left[L_{BS}^{\mathsf{LLDC}}\Big(M+(\frac{1-\alpha}{1-2\alpha})^2-1;\text{dep.},S\Big)\right] \le \mathbb{E}\left[L_{BS'}^{\mathsf{LLDC}}(N+1;\text{ind.})\right] . \tag{113}$$

Therefore, we obviously obtain equation 84. □

A comparison between the simulation and theoretical results of the LLDC length is presented in Table III.

D. An upper bound for the length of the orphan exclusion attack

Theorem 5. *Under the difficulty adjustment mechanism $\mathsf{DAM}_2^{\mathtt{modified}}$, the average length of the orphan exclusion attack, i.e., $\mathbb{E}[L^{\mathsf{OEA}}]$, performed by an attacker who owns predictive capability is upper bounded as follows:*

$$\mathbb{E}[L^{\mathsf{OEA}}] \le \mathbb{E}\left[L_{BS}^{\mathsf{LLDC}}\left(\left\lceil\frac{\alpha L_{epoch}}{(\frac{1-\alpha}{1-2\alpha})^2}\right\rceil;ind.\right)\right] , \tag{114}$$

where L_{epoch} represents the standard epoch length, which is equal to 2016 in Bitcoin.

Proof of Theorem 5. As already discussed, the attacker has a level of freedom to decide when to end the epoch under $\mathsf{DAM}_2^{\mathtt{modified}}$. In favor of the attacker, we assume all the longest dominant chains of adversarial blocks within $\mathtt{epoch}_i$ can lead to a successful orphan exclusion attack. In other words, we assume that the attacker can adjust the epoch end to guarantee that it gets included in the longest LDC of the epoch. The average number of adversarial blocks in each epoch is equal to αL_{epoch}. Therefore, the length of the orphan exclusion attack is equal to the length of the longest LDC among the LDCs that start in one of these αL_{epoch} adversarial blocks. Note that these αL_{epoch} blocks form a set of consecutive adversarial blocks all sampled from the same mining sequence. Therefore, $\mathbb{E}[L^{\mathsf{OEA}}] = \mathbb{E}\left[L_{BS}^{\mathsf{LLDC}}(\alpha L_{\text{epoch}};\text{dep.},S)\right]$. Using Theorem 4, we obtain the upper bound in equation 114. □

In Table IV, we provide a comparison between the upper bound calculated in equation 114 and the average length of the orphan exclusion attack under $\mathsf{DAM}_2^{\mathtt{modified}}$ obtained from the simulation.

TABLE IV
THE LENGTH OF THE ORPHAN EXCLUSION ATTACK UNDER $\mathsf{DAM}_2^{\mathtt{MODIFIED}}$

mining power share	0.25	0.3	0.35	0.4	0.45
$\mathbb{E}[L^{\mathsf{OEA}}]$ (simulation)	15.48	23.87	40.02	78.67	225.79
$\mathbb{E}[L^{\mathsf{OEA\text{-}max}}]$	16.67	26.39	45.58	93.07	282.09