Selfish Mining and Dyck Words in Bitcoin and Ethereum Networks

Cyril Grunspan, Ricardo Pérez-Marco

Selfish Mining and Dyck Words in Bitcoin and Ethereum Networks

Cyril Grunspan
Léonard de Vinci, Research Center, Paris - La Défense, France
cyril.grunspan@devinci.fr

Ricardo Pérez-Marco
CNRS, IMJ-PRG, Univ. Paris 7, Paris, France
ricardo.perez-marco@imj-prg.fr

Abstract

The main goal of this article is to present a direct approach for the formula giving the long-term apparent hashrates of Selfish Mining strategies using only elementary probabilities and combinatorics, more precisely, Dyck words. We can avoid computing stationary probabilities on Markov chain, nor stopping times for Poisson processes as in previous analysis. We do apply these techniques to other bockwithholding strategies in Bitcoin, and then, we consider also selfish mining in Ethereum.

2012 ACM Subject Classification Mathematics of Computing

Keywords and phrases Bitcoin, Blockchain, Ethereum, Proof-of-Work, Selfish Mining, Stubborn Mining, Apparent Hashrate, Revenue Ratio, Catalan Distributions, Dyck Words, Random Walk.

Digital Object Identifier 10.4230/OASIcs.Tokenomics.2019.9

1 Introduction

Background

Selfish mining (in short SM) is a non-stop blockwithholding attack described in [1] which exploits a flaw in the Bitcoin protocol in the difficulty adjustment formula [2]. The strategy is made of attack cycles. During each attack cycle, the attacker adds blocks to a secret fork and broadcasts them to peers with an appropriate timing. This is a deviant strategy from the Bitcoin protocol since an honest miner never withholds validated blocks and always mines on top of the last block of the official blockchain [7].

A rigorous profitability analysis that incorporates time considerations was done in [2]. The objective function based on sound economics principles that allows the comparison of profitabilities of different mining strategies with repetition is the *Revenue Ratio* $\frac{\mathbb{E}[R]}{\mathbb{E}[T]}$ where R and T are respectively the revenue and the duration time per attack cycle. A blockcwithholding attack slows down the production of blocks, hurting the profitability per unit time of all miners, including the attacker. Only after a difficulty adjustment, the attack can become profitable. The mean duration time of block production becomes equal to $\mathbb{E}[L] \cdot \tau_B$ where L is the number of blocks added to the official blockchain by the network per attack cycle and $\tau_B = 600$ sec. is the mean validation time of a block in Bitcoin network [4]. For Ethereum τ_E is around 12 sec. (in what follows, we use subscript B or E depending on which network we consider).

The Revenue Ratio becomes proportional to the long-term *apparent hashrate* of the strategy $\tilde{q} = \frac{\mathbb{E}[Z]}{\mathbb{E}[L]}$ where Z is the number of blocks added by the attacker to the official blockchain per attack cycle. This apparent hashrate becomes a proxy for the Revenue Ratio and can be used as a benchmark for profitability, but only after a difficulty adjustment. Several methods have been devised to compute $\tilde{q}$. The original approach from [1] uses a Markov model. Then the stationary probability is computed and used to compute the long term apparent hashrate. In [2] we use Martingale techniques and consider Poisson processes

and associated stopping times in order to compute the Revenue Ratio, and also the expected number of blocks $\mathbb{E}[Z]$ added by the attacker to the blockchain per attack cycle. The Revenue Ratio is computed at once using Doob's Stopping Time Theorem for Martingales. This last method has the advantage to compute the correct profitability analysis directly, not by means of the proxy of the long term apparent hashrate. For example, we can compute how long it takes for the attacker to have profit, something that is impossible to compute with the old Markov chain model. Moreover, with the Martingale techniques we clearly identify the difficulty adjustment formula as the origin of the vulnerability of the protocol. A Bitcoin Improvement Proposal (BIP) was proposed in [2] to prevent blockwithholding attacks. It consists in incorporating orphan blocks in the computation of the apparent hashrate of the network, and this is done by signaling orphan blocks. Something similar is done in Ethereum where rewards are given to some orphan blocks ("uncle" blocks). The goal was to favor mining decentralization.

Main goal

In this article we present a direct combinatorial approach for the direct computation of the apparent hashrate for different blockwithholding strategies in Bitcoin and Ethereum. These formulas are sometimes complicated, so it is remarkable that such a direct approach is possible. We don't need to use Markov chain, nor Martingale theory, and only elementary combinatorics using Dyck words. This analysis does not provide the full strength of the Martingale theory approach, but provides the basic formulas to estimate the long term apparent hashrates, and hence the profitabilities of the different strategies. The situation in Ethereum is combinatorially more complex due to the reward of "uncle" blocks and their signaling, which gives a larger spectrum of possible strategies. Our combinatorial approach also gives closed-form formulas for the apparent hashrate of one of the most effective strategy.

Notation

As usual, the relative hashrate of the honest miners (resp. attacker) is denoted by p (resp. q) and γ is its *connectivity* to the network. We have $p+q=1$, $q<\frac{1}{2}$ and $0\leq\gamma\leq 1$. When a competition occurs between two blocks or two forks, γ is the fraction of the honest miners who mine on top of a block validated by the attacker.

We will make use of Catalan numbers and Dyck words. Catalan numbers can be defined by

$$C_n=\frac{1}{2n+1}\binom{2n}{n}=\frac{(2n)!}{n!(n+1)!}$$

and their generating series is

$$C(x)=\sum_{n=0}^{+\infty}C_n x^n=\frac{1-\sqrt{1-4x}}{2x}$$

A Dyck word is a string (word) composed by two letters X and Y such that no initial segment of the string contains more Y's than X's. The relation with Catalan numbers is that the n-th Catalan number is the number of Dyck words of length $2n$. We refer to [6] for more properties and background material.

2 Selfish mining

An attack cycle for the SM strategy (see [2]) can be described as a sequence $X_0 \dots X_n$ with $X_i \in \{S, H\}$. The index i indicates the i-th block validated since the beginning of the cycle and the letter S, resp. H, indicates that the selfish, resp. honest, miner has discovered this block. From this labelling we will get the relation with Dyck words.

▶ **Example 1.** The sequence SSSHSHH means that the selfish miner has been first to validate three blocks in a row, then the honest miners have mined one, then the selfish miner has validated a new one and finally the honest miners have mined two blocks. At this point, the advantage of the selfish miner is only of one block. So according to the SM strategy, he decides to publish his whole fork and ends his attack cycle. In that case, we have $L = Z = 4$.

We are interested in the distribution of the random variable L.

▶ **Theorem 2.** *We have* $\mathbb{P}[L = 1] = p, \mathbb{P}[L = 2] = pq + pq^2$ *and for* $n \geqslant 3$, $\mathbb{P}[L = n] = pq^2(pq)^{n-2}C_{n-2}$ *where* C_n *is the* n*-th Catalan number.*

Proof. For $n \geqslant 3$, we note that $\{L = n\}$ is a collection of sequences of the form $w = \mathrm{SSX}_1 \cdots X_{2(n-2)}H$ with $X_i \in \{S, H\}$ for all i, such that if S and H are respectively replaced by the brackets "(" and ")" then, $X_1 \cdots X_{2(n-2)}$ is a Dyck word (i.e. balanced parenthesis) with length $2(n-2)$ (see [6]). The number of letters "S" (resp. "H") in w is n (resp. $n-1$). So, we get $\mathbb{P}[L = n] = p^{n-1}q^nC_{n-2}$ (see [6]). Finally, from the observation that $\{L = 1\} = \{H\}, \{L = 2\} = \{\mathrm{SSH}, \mathrm{SHS}, \mathrm{SHH}\}$, the result follows. ◀

▶ **Corollary 3.** *We have* $\mathbb{E}[L] = 1 + \frac{p^2q}{p-q}$

Proof. This formula results from the well know relations from [3]

$$\Sigma_{n\geqslant 0} p(pq)^n C_n = 1 \tag{1}$$

$$\Sigma_{n\geqslant 0} np(pq)^n C_n = \frac{q}{p-q} \tag{2}$$

◀

We can now compute the apparent hashrate.

▶ **Theorem 4.** *The long-term apparent hashrate of the selfish miner in Bitcoin is*

$$\tilde{q}_B = \frac{[(p-q)(1+pq)+pq]q - (p-q)p^2q(1-\gamma)}{pq^2+p-q}$$

Proof. When $L \geqslant 3$ we are in the situation where all blocks validated by the selfish miner end-up in the official blockchain. So, we have $Z = L$. If $L = 1$, then we have $Z = 0$. Moreover, we have $Z(\mathrm{SSH}) = Z(\mathrm{SHS}) = 2$ and $Z(\mathrm{SHH}) = 0$ (resp. 1) with probability $1-\gamma$ (resp. γ). So, we have

$$\begin{aligned}\mathbb{E}[Z] &= \mathbb{E}[L] - p - p^2q\gamma - 2p^2q(1-\gamma)\\ &= \mathbb{E}[L] - (p + p^2q + p^2q(1-\gamma))\end{aligned}$$

Using Corollary 3 we get,

$$\begin{aligned}\frac{\mathbb{E}[Z]}{\mathbb{E}[L]} &= \frac{p^2q + p - q - (p-q)(p + p^2q + p^2q(1-\gamma))}{pq^2+p-q}\\ &= \frac{[(p-q)(1+pq)+pq]q - (p-q)p^2q(1-\gamma)}{pq^2+p-q}\end{aligned}$$

This is Proposition 4.9 from [2] which is another form of Formula (8) from [1]. ◀

3 Stubborn Mining

We consider now two other block withholding strategies described in [8]. .

3.1 Equal Fork Stubborn Mining

In this strategy, the attacker never tries to override the official blockchain but, when possible, he broadcasts the part of his secret fork sharing the same height as the official blockchain as soon as the honest miners publish a new block. The attack cycle ends when the attacker has been caught-up and overtaken by the honest miners by one block [3, 8]. We show that the distribution of $L-1$ is what we defined as a (p,q)-Catalan distribution of first type in [3].

▶ **Theorem 5.** *For $n \geqslant 0$ we have $\mathbb{P}[L = n+1] = p(pq)^n C_n$.*

Proof. For $n \geqslant 0$, $\{L = n+1\}$ corresponds to sequences of the form $w = X_1 \cdots X_{2n} H$ with $X_i \in \{S, H\}$ for all i, such that if S and H are respectively replaced by the brackets "(" and ")" then, $X_1 \cdots X_{2n}$ is a Dyck word with length $2n$. ◀

▶ **Corollary 6.** *We have $\mathbb{E}[L] = \frac{p}{p-q}$*

Proof. Follows from (1) and (2). ◀

▶ **Theorem 7.** *The long-term apparent hashrate of a miner following the Equal-Fork Stubborn Mining strategy is given by*

$$\tilde{q} = \frac{q}{p} - \frac{(1-\gamma)(p-q)}{\gamma p}(1 - pC((1-\gamma)pq))$$

Proof. In an attack cycle, all the honest blocks except the last one have a probability γ to be replaced by the attacker. So, we have $\mathbb{E}[Z|L = n+1] = n + 1 - \frac{1-(1-\gamma)^{n+1}}{\gamma}$ (see Lemma B.1 in [3]). Conditioning by $\{L = n+1\}$ for $n \in \mathbb{N}$ and using Theorem 5, we get

$$\mathbb{E}[Z] = \frac{q}{p-q} - \frac{1-\gamma}{\gamma}(1 - pC((1-\gamma)pq))$$

and the result follows. ◀

3.2 Lead Stubborn Mining

This strategy is similar to the selfish mining strategy but this time the attacker takes the risk of being caught-up by the honest miners. When this happens, there is a final competition between two forks sharing the same height. when the competition is resolved, a new attack cycles starts. In this case, the distribution of $L-1$ turns out to be a (p,q)-Catalan distribution of second type as defined in [3].

▶ **Theorem 8.** *We have $\mathbb{P}[L = 1] = p$ and for $n \geqslant 1$, $\mathbb{P}[L = n+1] = (pq)^n C_{n-1}$.*

Proof. We have $\{L = 1\} = \{H\}$ and for $n \geqslant 0$, the condition $\{L = n+1\}$ corresponds to sequences of the form $w = \mathrm{S}X_1 \cdots X_{2(n-1)}\mathrm{HY}$ with $X_1, \ldots, X_{2(n-1)}, Y \in \{S, H\}$ and such that if S and H are respectively replaced by the brackets "(" and ")" then, $X_1 \cdots X_{2(n-1)}$ is a Dyck word with length $2(n-1)$. ◀

▶ **Corollary 9.** *We have $\mathbb{E}[L] = \frac{p-q+pq}{p-q}$*

Proof. Follows from (1) and (2). ◀

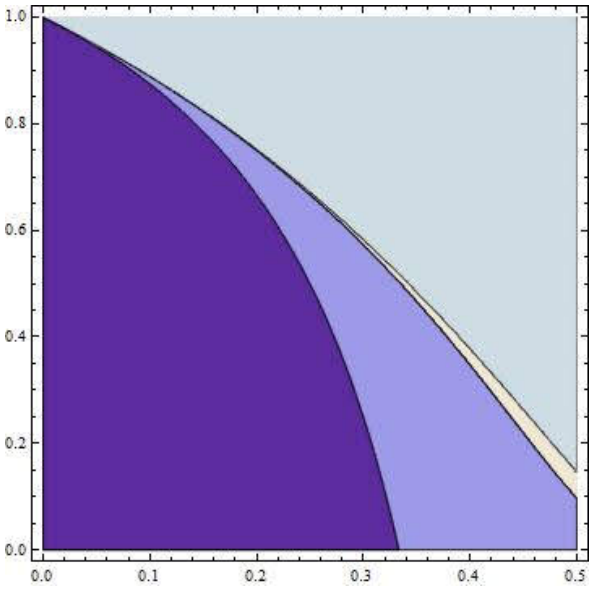

Figure 1 From left to right: HM, SM, LSM, EFSM.

By repeating the same argument as in the proof of Theorem 7 for the computation of $\mathbb{E}[Z]$, we obtain the following theorem [3].

▶ **Theorem 10.** *The long-term apparent hashrate of a miner following the Lead Stubborn Mining strategy is given by*

$$\tilde{q} = \frac{q(p+pq-q^2)}{p+pq-q} - \frac{pq(p-q)(1-\gamma)}{\gamma} \cdot \frac{1-p(1-\gamma)C((1-\gamma)pq)}{p+pq-q}$$

We plot regions in the parameter space $(q,\gamma) \in [0,0.5] \times [0,1]$ according to which strategy is more profitable. We get Figure 1 [3] (HM honest mining, SM selfish mining, LSM Lead Stubborn mining, EFSM Equal Fork Stubborn mining).

4 Selfish mining in Ethereum

Ethereum is a cryptocurrency based on a variation of the GHOST protocol [11]. The reward system is different than in Bitcoin, and this introduces a supplementary complexity in the analysis of block withholding strategies. Contrary to Bitcoin, mined orphan blocks can be rewarded like regular blocks, with a reward smaller than regular blocks. The condition for an orphan block to get a reward is to be an "uncle" referred by a "nephew" which is "not too far". By definition, an "uncle" is a stale block whose parent belongs to the main chain and a "nephew" is a regular block which refers to this "uncle". "Not too far" means that the distance d between the uncle and the nephew is less than some parameter value n_1. The distance is the number of blocks which separates the nephew to the uncle's parent in the main chain. When this situation occurs, the nephew gets an additional reward of πb and the uncle gets a reward $K_u(d)b$ where b denotes the coinbase in Ethereum. Today's parameter values are $n_1 = 6$, $K_u(d) = \frac{8-d}{8} \cdot \mathbf{1}_{1 \leqslant d \leqslant 6}$, $\pi = \frac{1}{32}$ and $b = 2$ ETH [9].

There is little published research on selfish mining in Ethereum except for [9] and [10] based on numerical simulations. In [9], through a Markov chain approach, a non-closed infinite double sum is given for the apparent hash rate of the attacker.

The general study of selfish mining in Ethereum is complex because equivalent selfish mining strategies in Bitcoin are no longer equivalent for Ethereum. The attacker can choose to refer or not uncle blocks. Referring uncle blocks provides an extra revenue but hurts the main goal of selfish mining of lowering the difficulty. Also, he can choose to create artificially more uncles by broadcasting the part of his secret fork sharing the same height as the public blockchain of the honest miners. All this different strategies are analyzed in [5]. In the present article we restrict to a couple of strategies.

In the strategy studied in [9], the attacker creates as many uncles as possible and tries to refer all of them. In [5], we prove that this strategy is not optimal and is less profitable than the strategy we study in this article, for which we obtain a closed form formula for the apparent hashrate of the attacker using only elementary combinatorics.

In the strategy we consider, the attacker never broadcasts his fork, which remains secret until he is on the edge of being caught-up by the honest miners or is actually caught up (this last case can only occur when the attack cycle starts with SH). In addition, the attacker always refers to all possible uncle blocks.

We denote by R the revenue by cycle of the selfish miner following this strategy. We have $R = R_s + R_u + R_n$ where R_s is the revenue coming from "static" blocks in the main chain i.e., $R_s = Zb$, R_u is the revenue coming from uncles and R_n is the additional revenue coming from nephews.

▶ Remark 11. We always have $R_u = 0$ except when the attack cycle is SHH and the last block mined by the honest miners has been mined on top of an honest block. In that case, the first block mined by the selfish miner is referred by the second block of the honest miners.

It follows from this remark that

$$\mathbb{E}\left[\frac{R_u}{b}\right] = p^2q(1-\gamma)K_u(1) \tag{3}$$

It remains to compute $\mathbb{E}[R_n]$.

▶ **Definition 12.** *If ω is an attack cycle, we denote by $U(\omega)$ (resp. $U_s(\omega)$, $U_h(\omega)$) the random variable counting the number of uncles created during the cycle ω which are referred by nephew blocks (resp. nephew blocks mined by the selfish miner, nephew blocks mined by the honest miners) in the cycle ω or in a later attack cycle.*

We denote by $V(\omega)$ the random variable counting the number of uncles created during the cycle ω and are referred by nephew blocks (honest or not) in an attack cycle strictly after ω.

▶ **Proposition 13.** *We have $\mathbb{E}[U] = q - q^{n_1+1}$.*

Proof. We have $U = 0$ if and only if the attack cycle is H or if it starts with $n_1 + 1$ blocks of type S. Otherwise, we have $U = 1$. So, $\mathbb{E}[U] = \mathbb{P}[U > 0] = 1 - (p + q^{n_1+1}) = q - q^{n_1+1}$ ◀

We compute now $\mathbb{E}[V]$

▶ **Proposition 14.** *We have $\mathbb{E}[V] = pq^2 \cdot \frac{1-(pq)^{n_1-1}}{1-pq}$.*

Proof. We have $V = 1$ if and only if the attack cycle ω is SS..SH..H with $2 \leq k \leq n_1$ S. In that case, the first block H is an uncle that will be referred by the first future official block in the attack cycle after ω. Otherwise, $V = 0$. So, $\mathbb{E}[V] = pq^2 + \ldots + p^{n_1-1}q^{n_1}$, and we get the result. ◀

▶ **Proposition 15.** *We have $\mathbb{E}[U_h] = p^2q + \big(p + (1-\gamma)p^2q\big)\, pq^2 \cdot \frac{1-(pq)^{n_1-1}}{1-pq}$.*

Proof. Let ω be an attack cycle and let ω' be the attack cycle after ω. If $U_h^{(1)}(\omega)$ (resp. $U_h^{(2)}(\omega)$) counts the number of uncles referred by honest nephews only present in ω (resp. in ω'), then we have $U_h = U_h^{(1)} + U_h^{(2)}$. Moreover, $U_h^{(1)}(\omega) = \mathbf{1}_{\omega=\text{SHH}}$ and $U_h^{(2)}(\omega) = \mathbf{1}_{\omega'\in E}\cdot V(\omega)$ where E is the event that ω' is either H or SHH with a second honest block mined on top of the first honest block. Hence we get the result by taking expectations since ω and ω' are independent. ◀

▶ **Corollary 16.** *We have*

$$\mathbb{E}\left[\frac{R_n}{\pi}\right] = q^2(1+p) - q^{n_1+1} - \left(p + (1-\gamma)p^2q\right)pq^2 \cdot \frac{1-(pq)^{n_1-1}}{1-pq} \tag{4}$$

Proof. We have $\mathbb{E}[U_s] = \mathbb{E}[U] - \mathbb{E}[U_h]$ and we use Proposition 13 and Proposition 15. ◀

We can now compute the apparent hashrate of the selfish miner in Ethereum. We have two cases to consider: The old difficulty adjustment formula (similar to the one in Bitcoin), and the current difficulty adjustment formula that takesinto account referred uncles.

▶ **Theorem 17.** *The long term apparent hashrate $\tilde{q}_{E,0}$ of the selfish miner in Ethereum with its old difficulty adjustment formula is given by $\tilde{q}_{E,0} = \tilde{q}_B + \tilde{q}_u K_u(1) + \tilde{q}_n \pi$ with*

$$\tilde{q}_u = \frac{p^2q(1-\gamma)(p-q)}{p-q+p^2q}$$
$$\tilde{q}_n = \frac{(p-q)\left(q^2(1+p) - q^{n_1+1} - \left(p+(1-\gamma)p^2q\right)pq^2\cdot\frac{1-(pq)^{n_1-1}}{1-pq}\right)}{p-q+p^2q}$$

The long term apparent hashrate $\tilde{q}_E$ of the selfish miner in Ethereum with its current difficulty adjustment formula is

$$\tilde{q}_E = \tilde{q}_{E,0}\cdot\xi$$

where

$$\xi = \frac{p-q+p^2q}{p^2q + (p-q)\left(1+q-q^{n_1+1}\right)}$$

Proof. We have $\tilde{q}_{E,0} = \frac{\mathbb{E}[R]}{\mathbb{E}[L]}$ and $\tilde{q}_E = \frac{\mathbb{E}[R]}{\mathbb{E}[L]+\mathbb{E}[U]}$ We then use Proposition 13, (3), (4) and the formula for $\tilde{q}_B$ in Theorem 4. ◀

We can now compare this strategy to selfish mining in Bitcoin. Observe that $\tilde{q}_{E,0} > \tilde{q}_B$, where $\tilde{q}_B$ is the long term apparent hashrate of the Bitcoin selfish miner. Therefore, the minimal threshold $q_{\min}$ such that the inequality $\tilde{q} > q$ for $q > q_{\min}$ is always lower in Ethereum with its old adjustment formula than in Bitcoin. This is due to the particular reward system that indeed favors selfish mining as we have proved. Notice also that when $q > q_{\min}$, the attack is profitable faster in Ethereum than in Bitcoin because of another difference in the protocols: In Ethereum the difficulty is updated at each block and in Bitcoin only after 2016 blocks.

Figure 2 plots the regions in parameter space $(q,\gamma) \in [0,0.5]\times[0,1]$ where each strategy HM or SM is more profitable. We find $q_{\min} \approx 9.5\%$ when $\gamma = 0$.

Now, Ethereum with its new difficulty adjustment formula is more resilient to selfish mining. Figure 3 plots the region in parameter space $(q,\gamma) \in [0,0.5]\times[0,1]$ where each strategy HM or SM is more profitable.

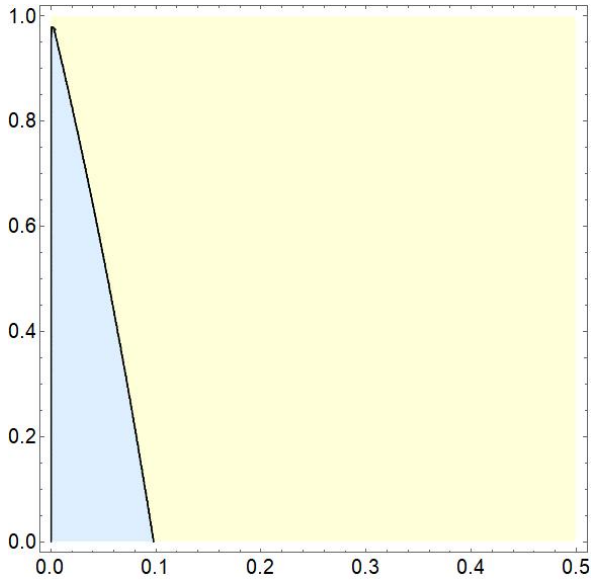

Figure 2 HM vs. SM Ethereum old difficulty adjustment.

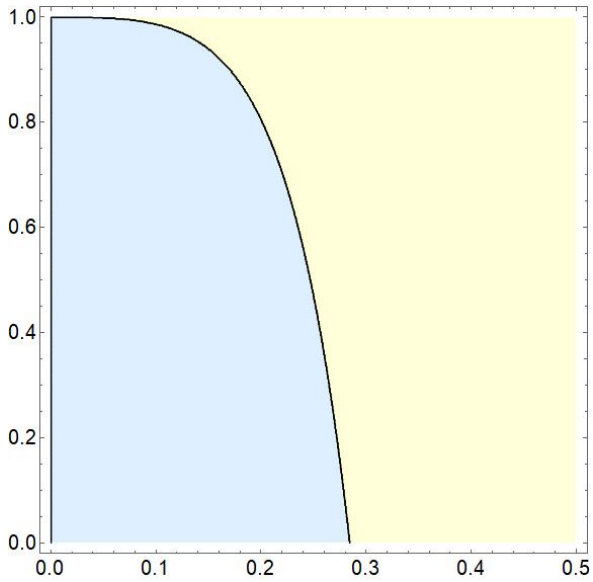

Figure 3 HM vs. SM Ethereum new difficulty adjustment.

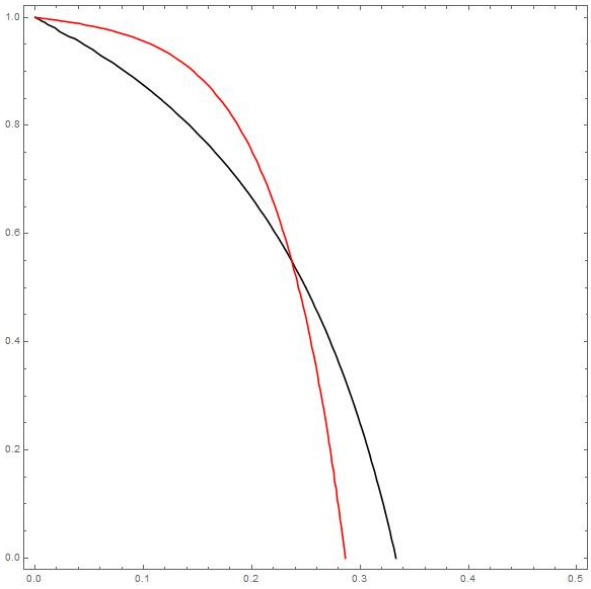

Figure 4 HM vs. SM in Bitcoin and Ethereum.

We note that Bitcoin is more resilient to selfish mining when the relative hashrate of the attacker is high, but we have the opposite for smaller relative hashrates. This means that when the relative hashrate of the attacker is small (resp. high) then, the connectivity of the attacker should be higher (resp. lower) in Ethereum than in Bitcoin for the attack to be profitable. Figure 4 compares the thresholds curves between HM and SM in Bitcoin and Ethereum.

5 Conclusions.

We have computed closed-form formulas for the long term apparent hashrate of different blockwithholding strategies for Bitcoin and Ethereum using only elementary combinatorics, Dyck words, Catalan numbers, and their properties. Although this approach does not provide a complete analysis of the profitability of the strategies, as for example the time it takes to the strategy to become profitable, this minimalist approach is sufficient to compare profitabilities in the long run. In the strategies studied we have show the impact of the different reward system. For these strategies, depending on given parameters (q, γ), relative hashrate and connectivity of the attacker, we have determined which network is more resilient to selfish mining attacks.

References

1 I. Eyal and E. G. Sirer. Majority is not enough: bitcoin mining is vulnerable. *Commun. ACM*, 61(7):95–102, 2018.
2 C. Grunspan and R. Pérez-Marco. On profitability of selfish mining. *ArXiv:1805.08281*, 2018.
3 C. Grunspan and R. Pérez-Marco. On profitability of stubborn mining. *ArXiv:1808.01041*, 2018.
4 C. Grunspan and R. Pérez-Marco. On profitability of trailing mining. *ArXiv:1811.09322*, 2018.
5 C. Grunspan and R. Pérez-Marco. Selfish mining in ethereum. *In preparation*, 2019.
6 T. Koshy. *Catalan numbers with applications*. Oxford University Press, Oxford, 2009.
7 S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

8 K. Nayak, S. Kumar, A. Miller, and E. Shi. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, pages 305–320, 2016.

9 J. Niu and C. Feng. Selfish mining in ethereum. *arXiv:1901.04620*, 2019.

10 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&P Workshops 2018, London, United Kingdom, April 23-27, 2018*, pages 50–57, 2018.

11 Y. Sompolinsky and A. Zohar. Secure high-rate transaction processing in bitcoin. In *Financial Cryptography and Data Security - 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers*, pages 507–527, 2015.